

Argumentation, paradox and kernels in directed graphs

Sjur Dyrkolbotn

Acknowledgements

I would like to thank my first supervisor, Michał Walicki, for great support and encouragement, as well as collaboration. Without him, this thesis could not have been made. I would also like to thank my second supervisor, Marc Bezem, for being supportive, understanding, and always at hand with a thoughtful comment. A great thanks is also due to the administration at the department. The people working there are so good at their job that I never really had to talk to them. Excellent! Except for the fact that they seem like really nice people too; people that I now wish I had gotten to know a little better. I know enough to give a special thanks to Ida Holen, however. Genius leader of geeks! Sadly, no longer at the department, but tales of wonder will be told about her for a long time to come, I am sure. Finally, I must thank the logic gang, including, but not necessarily limited to, the following brilliant people: Truls André Pedersen, Erik Parmann, Pål Grønås Drange, Piotr Kazmierczak, Paul Simon Svanberg and Ragnhild Hogstad Jordahl. Come to think of it, I must also thank Soda Fountain Rag, for providing an awesome soundtrack!

Contents

I	Overview	7
1	Introduction	9
1.1	Motivation and background	10
1.1.1	Kernels in digraphs	10
1.1.2	Paradox	14
1.1.3	Argumentation	17
2	Presentation of main results	23
2.1	Connections between different areas of research	23
2.2	Algorithmic results	25
2.3	Structural results	27
2.3.1	Sufficient conditions for existence of kernels	27
2.3.2	Relations that preserve structural properties	30
2.4	Reasoning about paradox and admissibility	32
2.5	Conclusion and future work	33
II	Papers	39
3	Paper A: Finding kernels or solving SAT	41
4	Paper B: Kernels in digraphs that are not kernel perfect	67
5	Paper C: Propositional Discourse Logic	85
6	Paper D: Equivalence Relations for Abstract Argumentation	129

Part I

Overview

Chapter 1

Introduction

In this thesis, we investigate propositional theories in *graph normal form*, as introduced in [3]. They are theories consisting of equivalences $x \leftrightarrow \bigwedge_{y \in X} \neg y$ for $\{x\} \cup X$ a set of propositions, such that each x appears at most once to the left of an equivalence. Intuitively, we think of the variable on the left as the *name* of the formula on the right, and we think of theories in graph normal form as giving a simple formalization of the *propositional discourse*, collections of statements that are allowed to refer to each other in an arbitrary – possibly circular, sometimes even paradoxical – manner. We tend not to think of them as theories in classical logic, however, but consider them instead from a combinatorial point of view, as directed graphs.

It was observed in [3], that the notion of a *kernel*, studied in digraph theory [6], provides means to give an equivalent definition of classical satisfiability of such theories, and this observation forms the basis for our work. We will also consider theories in graph normal form from the point of view of artificial intelligence, however, looking at them as argumentation frameworks in the sense of Dung [21]. This is natural since in fact, the notion of a kernel is essentially the same as the notion of a stable set from argumentation. Both of these notions, in particular, are equivalent to classical consistency. Our overreaching goal has been to investigate the following questions:

- (1) When are theories in graph normal form consistent?
- (2) How can we extract information from them when they are not?
- (3) What *kind* of information can be extracted?

Question (1) asks for structural conditions that ensure consistency, and for algorithmic techniques that allow us to decide consistency as efficiently as possible. Question (2) asks for alternative, non-classical, semantic notions that we can apply when classical consistency fails. Question (3) asks us how we should think of the information that we extract using such notions.

The thesis should be seen as a mainly theoretical contribution; we have studied mathematical problems that arise in the context of the three questions

presented above. The focus has not been on applications, and we have not devoted much time to discussing the exact *nature* of the phenomena we study. We think, however, that all of Questions (1)–(3), and Question (3) in particular, can also be addressed from the point of view of applications, and from a philosophical angle, as asking us to develop an *understanding* of what inconsistency means when it arises in theories in graph normal form. With regards to possible applications, we believe that the large body of work devoted to argumentation theory is sufficient to motivate theoretical investigations such as our own; it warrants the assertion that studying consistency of theories in graph normal is useful. With respect to the philosophical issues that arise, we believe that our own work, although it does not focus on this, suggests some novel perspectives. Actually, despite the fact that we maintain a focus on technical problems, some might consider our work more interesting in a conceptual regard. As we will see, it serves to connect different areas of research, and to offer a fresh point of view on established notions, both technical, philosophical and related to applications in AI.

In the remainder of this chapter we provide some further motivation, as well as a short background on the research areas we address. Then in Chapter 2 we present our main results, beginning with a brief discussion on the connection between the areas addressed in this introduction. Part II contains our papers.

1.1 Motivation and background

1.1.1 Kernels in digraphs

The notion of a kernel in a directed graph (digraph) was first introduced by Von Neumann and Morgenstern to provide an abstract solution concept in cooperative game theory [48]. It has since been studied quite extensively by graph theorists, however, from a purely theoretical point of view. We point to [6] for a recent overview of the field. In this section, we only present the basic definitions, and enough background to suggest how our own research represents a novel approach compared to earlier work in kernel theory.

We recall that a directed graph is a pair $\mathbf{G} = \langle G, N \rangle$ such that $N \subseteq G \times G$. When $(x, y) \in N$, we write $y \in N(x)$ and $x \in N^-(y)$. The notation extends pointwise to sets, e.g., such that if $X \subseteq G$, then $N(X) = \bigcup_{x \in X} N(x)$. A digraph $\mathbf{G}' = \langle G', N' \rangle$ is said to be a *subdigraph* of \mathbf{G} if $G' \subseteq G$ and $N' \subseteq N$, while it is the *subdigraph induced by G'* if $G' \subseteq G$ and $N' = \{(x, y) \in N \mid x, y \in G'\}$. For $X \subseteq G$, we typically write $\mathbf{G} \setminus X$ to denote the subdigraph induced by $G \setminus X$. A digraph is finite if G is finite and is said to be *finitary* if $N(x)$ is finite for all $x \in G$. In this thesis, unless we state otherwise, we assume all digraphs to be finite. A *walk* in \mathbf{G} is a sequence of vertices $x_1 x_2 \dots x_n$ such that for all $1 \leq i < n$ we have $x_{i+1} \in N(x_i)$. In case no vertex is repeated, the walk is referred to as a *path*. A *cycle* is a walk $x_1 x_2 \dots x_n$ such that $x_1 x_2 \dots x_{n-1}$ is a path and $x_1 = x_n$.

A *kernel* in a directed graph is a set $K \subseteq G$ such that:

$$N^-(K) = G \setminus K \quad (1.1)$$

It is useful to think of this equality in terms of two inclusions and to adopt the terminology that kernels in digraphs are exactly those sets of vertices that are both *independent* and *absorbing*:

- $N^-(K) \subseteq G \setminus K$ (K is independent)
- $N^-(K) \supseteq G \setminus K$ (K is absorbing)

We will write $Kr(G)$ for the set of all kernels in G . Not all digraphs have kernels; admitting a kernel is a non-trivial property of digraphs, and, as we will see later, it also provides a combinatorial way to look at consistency of theories in graph normal form. The basic example of a digraph that does not admit a kernel is the single loop

$$G: x \curvearrowright \quad (1.2)$$

Consulting Equation (1.1), we see that \emptyset cannot be a kernel in G since $N^-(\emptyset) = \emptyset \not\supseteq \{x\} \setminus \emptyset$, while $\{x\}$ cannot be a kernel since $N^-(x) = \{x\} \not\subseteq \{x\} \setminus \{x\}$. In particular, \emptyset is not absorbing, while $\{x\}$ is not independent.

Historically, work in kernel theory has been motivated by the close connection to the notion of *perfectness* in undirected graphs. In fact, one approach to the perfect graph conjecture, now theorem [12], addressed it from the point of view of kernels and directed graphs, see [6]. What is important for us is to note that the relevant directed notion in this regard was something stronger than existence of kernels, namely that of *kernel perfectness*, requiring existence of kernels also in all induced subdigraphs. And while work was being done concerning the connection to perfectness in undirected graphs, quite some work was also being devoted to the search for structural conditions that ensure kernel perfectness, see e.g., [20, 28, 19].

Since kernel perfectness requires existence of kernels to be preserved in all induced subgraphs, establishing existence of kernels becomes easier when this stronger notion is considered. When you aim to prove kernel perfectness you can assume, in an inductive argument, that *any* proper induced subdigraph has a kernel. In proofs, this typically means that you will do a local choice of vertices around some vertex, and then show how it gives rise to a kernel for the whole digraph by taking the union with some kernel in an appropriate induced subdigraph. Such a kernel is usually ensured by induction hypothesis, and working with kernel perfectness typically means that the conditions you consider must themselves be preserved by taking induced subdigraphs. This, as we will see later, is not the case for some conditions that it seems natural to consider.

The first non-trivial result from kernel theory is that a finitary digraph with no odd cycles is kernel perfect. The result was first obtained by Richardson [46]. The proof given there is rather complicated, but can be greatly simplified

by using the notion of a *semikernel*, first introduced by Victor Neumann Lara in [40]. A semikernel in a digraph G is a set $S \subseteq G$ such that

$$N(S) \subseteq N^-(S) \subseteq G \setminus S \quad (1.3)$$

In other words, S is a semikernel if it is independent, and also satisfies a weaker form of absorption, $N(S) \subseteq N^-(S)$, stating that all vertices pointed at by S must point back into S . We call this *local* absorption. Given a digraph G , we use $Lk(G)$ to denote the set of all semikernels in G . Notice that $\emptyset \in Lk(G)$ for any G , and that the loop does not have any non-empty semikernel. A digraph can have a non-empty semikernel without having a kernel, however, as illustrated by the following digraph:



Since they are not independent sets, neither $\{x\}$ nor $\{y\}$ can be a subset of any kernel, meaning that (global) absorption is impossible and that G has no kernel. Still, we have $Lk(G) \neq \emptyset$, with $Lk(G) = \{\{z\}\}$, the set $\{z\}$ being independent and also locally absorbing since y , the only vertex pointed to by z , is itself pointing back to z .

While non-empty semikernels can exist in digraphs that admit no kernels, the two notions coincide when we consider kernel perfectness.

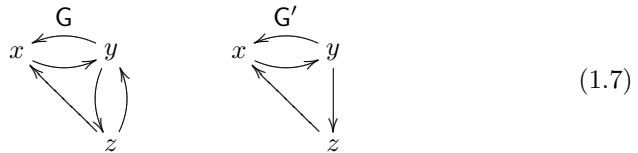
Theorem 1.5 [40] *A digraph $G = \langle G, N \rangle$ is kernel perfect iff every non-empty induced subdigraph of G admits a non-empty semikernel*

In light of this result, we can establish conditions that ensure kernel perfectness by showing that they ensure existence of non-empty semikernels for every non-empty induced subdigraph. Since semikernels are formulated locally, without demanding global absorption, this is usually much easier, and it is the approach followed in most work in kernel theory. The following theorem summarizes the most significant results. Recall that a *chord* on a cycle is an edge connecting two non-consecutive vertices.

Theorem 1.6 *For all digraphs G , we have that $Kr(G) \neq \emptyset$ if every odd cycle in G has one of the following*

- (1) *at least two symmetric edges* [19],
- (2) *at least two crossing consecutive chords* [20] or
- (3) *at least two chords with consecutive targets* [28].

As an example, consider the digraph G depicted below. It has a kernel, and this is ensured by all points of Theorem 1.6.



In fact, G has two kernels: $\{x\}$ and $\{y\}$. We notice that one of these, $\{x\}$, is also a kernel in G' . This, however, is not captured by any of the results from Theorem 1.6, suggesting the possibility of obtaining stronger results. Actually, the reader might observe that for any digraph that consists of a single odd cycle, if you add to it one symmetric edge, you obtain a kernel. This is not hard to see: simply take the target of this new edge, then skip two vertices, and from then on take every other vertex as you move along the cycle. You end up with every other vertex except for two consecutive vertices that you did not choose. But this is no problem, since one of them has the symmetric edge going back to its predecessor on the cycle – the first vertex chosen. This simple way of resolving an odd cycle does *not* generalize, however. This is illustrated by the following digraph, where each odd cycle has a symmetric edge. We leave it to the reader to verify that no kernel can be found in this digraph.

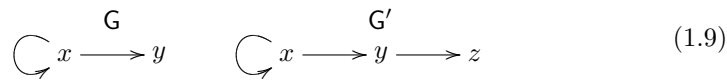


The problem is that the odd cycles *interact* in ways that make it impossible to solve them all simultaneously. This problem of *compatibility* is the essence of what makes the search for sufficient conditions both interesting and difficult, and, as we will see more clearly later, it is also one of the things that makes the connection to argumentation and paradox so natural and exciting.

It is easy to see that many digraphs exist that have kernels without being kernel perfect. Any digraph that has a kernel but also contains a loop, for instance, fall into this category. It might seem, however, that if a digraph is not kernel perfect, it will be difficult to establish any simple, local conditions that imply existence of kernels. Indeed, apart from our own work, we are not aware of any results from the literature that ensures existence of kernels in digraphs that are not kernel perfect.

When such digraphs are considered, it becomes unclear how an argument demonstrating existence of kernels should proceed. The conditions we work with no longer hold for induced subdigraphs, so if we wish to use an inductive argument, the question of when and how we can apply an induction hypothesis becomes particularly tricky.

We notice, however, going back to Richardson's Theorem, that *odd cycles* are the only structures that can prevent existence of kernels. We also see that all conditions in Theorem 1.6 concern conditions that ensure that odd cycles are somehow *resolved* by the existence of chords ensuring the existence of appropriate even subcycles. We also readily notice that as well as being resolved from "within" by chords – as in the conditions from Theorem 1.6 – an odd cycle can be resolved from the "outside", by pointing at some vertex that can be used to break it, as in the digraph G , depicted in (1.9) below.



For this digraph, we have $Kr(G) = \{\{y\}\}$; the loop at x poses no problem since it points to a vertex we can happily include in the kernel. This suggests a possible way to attack the problem of finding new structural conditions: we search for conditions under which odd cycles are harmless because they point to suitable subdigraphs that break them.

The digraph G' , also depicted in (1.9), illustrates that such an approach presents us with a challenge; G' , in particular, cannot be resolved since now, due to the presence of z , y cannot be used to break the loop at x . This leads to the following general question: *what* sort of subdigraphs are such that if the odd cycles in a digraph point to it, then existence of kernels can be ensured?

We address this in Paper **B**, where we provide some general tools that are useful to such an investigation and give several conditions that ensure resolution of odd cycles and in this way establish the existence of kernels in digraphs that need not be kernel perfect.

1.1.2 Paradox

A paradox is a surprising contradiction, a contradiction which we arrive at from premises that we think are uncontroversial.¹ Semantic paradoxes, in particular, have this property, and the relevant premises that are being challenged are the rules of classical logic, as well as a very basic intuition about *truth*, namely that a statement is true if and only if what it says is true. The liar sentence, for instance – “this sentence is false” – is inconsistent with these assumptions since they permit us to deduce that the liar is true if and only if what it says is false. This presents us with a surprising problem; either something is wrong with the rules of classical logic, or else something is wrong with our intuitive understanding of truth. The latter is often formally described by Tarski’s Convention T [47]:

$$\bullet T(\ulcorner \phi \urcorner) \leftrightarrow \phi$$

Here, T is a *truth-predicate*, the extension of which is supposed to contain all true formulas from some language which includes ϕ . $\ulcorner \phi \urcorner$, on the other hand, is the *name* of the formula ϕ , a term in the language, the purpose of which is to allow us to use the logical language to talk about its own formulas. This formalization is the standard approach in the literature, and it is rare to distinguish between the formal study of truth and the formal study of paradox; the two are almost always considered together. It seems to us, however, that a representation where truth is maintained explicitly as a predicate is only needed when we wish to challenge Convention T, i.e., when we wish to consider the possibility that some statements are *not* such that they are true iff what they say is true. Clearly, this is a possible way to approach the liar and other semantic paradoxes, but it is not the only one, and, we would argue, not even the most natural one. We will not dwell on philosophical issues, but simply

¹This only applies to what Quine calls the falsical paradoxes [44], but one might argue that the other kind he proposes – the veridical ones – are not really paradoxes at all, but merely surprising *facts*

follow Kripke in thinking that truth behaves exactly as we expect, and that the paradoxes only show that it is *partially defined* [38]. Also, our aim is not to solve the paradoxes, but to analyze them. Given our intuitive notion of truth, they do arise, and since we make the choice to stick with this notion, the interesting question becomes *when* and *why* they arise.

For these questions, it seems that an explicit representation of truth as a predicate is, at least in the first instance, redundant. Truth, when it satisfies Convention T, conflates to identity, and it seems that a paradox is simply a statement p for which $p \leftrightarrow \neg p$ can be deduced. For a further illustration of this, consider the liar sentence formulated in predicate logic: $\psi \leftrightarrow \neg T(\ulcorner \psi \urcorner)$. It is usually required that this formula must be shown to be *true* in order for the system to contain a liar. This can be done simply – by endowing the formal system with means to perform direct self-reference, as explained, for instance, in [34, Chapter 2, Section 2B] – or it can be done the hard way, as in Tarski’s original paper [47], itself an application of the diagonalization technique introduced by Gödel in his first incompleteness proof [29]. Either way, there is not yet a paradox, just a funny looking formula with some possibly non-trivial internal structure. The paradox arises upon assuming Convention T, since then we can deduce $T(\ulcorner \psi \urcorner) \leftrightarrow \neg T(\ulcorner \psi \urcorner)$. But then we have again found our p such that $p \leftrightarrow \neg p$, so why the detour?

Of course, for Tarski, and even more so for Gödel, there was true genius at work in showing that systems specifically designed to avoid this type of inconsistency would nevertheless give rise to it under what was then seen as weak assumptions. But for subsequent work, attempting to study paradox, the exact nature of the p in question seems unimportant. For the ontological concern about whether or not it exists, it seems adequate to simply point to the original liar, as it arises in natural language. Of course, one might want to ensure that the existence of a paradoxical p becomes an artifact of the formal system itself, that the truth of $p \leftrightarrow \neg p$ is indeed a possibility. We do not necessarily think this is appropriate, however, since one might as well think of $p \leftrightarrow \neg p$ as *defining* p . Still, we remark that in three-valued Lukasiewicz logic [39], the equivalence does indeed hold just in case p does not obtain a Boolean value, i.e., just in case truth is not in fact defined for it. We believe this is a nice, simple description of paradox by formal means, and we discuss it further in Paper C, see also Section 2.4. The interesting challenge, however, is to classical propositional logic, which carries implicitly the assumption that there can be no such paradoxical p . The fact that $p \leftrightarrow \neg p$ is regarded as *inconsistent*, in particular, precludes the p from becoming instantiated semantically.

In logic, contradictions are usually quite boring. In fact, it seems wrong to use the plural form to speak of them; semantically, there is typically only one contradiction, namely \perp , falsehood. What is interesting is how to locate it in the logical language. Here, it can take many forms, the challenge being to determine what formulas count as contradictions. But for any two such formulas, any difference in syntactic form is simply conflated, they are all considered equivalent. This might not always be the right way to conceive of contradictions, however. For one, they *do* arise in practice – they *exist* in the real world, so to

speak (even if they do not, perhaps, in the world of ideas) – and some taxonomy allowing to detail their properties and draw non-trivial inferences from them is a much studied challenge in many fields, especially in computer science [2].

In this thesis, however, we will not commit ourselves to any form of dialetheism. Given the formula $p \wedge \neg p$, we will not argue that there is any p for which such a formula is true, and should such a p exist, we are confident that our work does not address it in any way. Rather, we argue that the two contradictions $p \wedge \neg p$ and $q \leftrightarrow \neg q$ are *not* really equivalent. They might be both inconsistent in classical logic, but by virtue of their *form*, they should not be considered in the same light. In the case of $p \wedge \neg p$, a commitment to dialetheism seems immediate upon assuming that such p exists, but for $q \leftrightarrow \neg q$ it is not, since, as we have already argued, the liar provides a plausible witness for the existence of q such that the formula holds.² Now, while it is possible to use Łukasiewicz logic or some other means to obtain a formal reflection of this fact, doing so requires the introduction of non-classical notions already in order to *represent* the paradox. This, we feel, is hasty. Rather, we believe that classical logic is the best starting point we have, and that it should be adhered to for as long as possible. Moreover, since the challenge we wish to take on is intuitively understood as a challenge to the semantics of classical logic, we think it is questionable to abandon classical logic already in order to formalize the problem. Ideally, we want to use formal tools to analyze a question, not evaluate a possible answer. Also, the search for an answer to the paradoxes is already very widespread, and with no consensus looking likely to form any time soon, it is our belief that a more careful consideration as to the exact nature of the question is also in order.

In this thesis, we approach this by introducing the idea that the semantic paradoxes are demarcated from other contradictions by their *form*. We take them to be inconsistencies in classical logic, and we designate them as surprising for purely syntactic reasons. This leads naturally to the notion of a propositional discourse – represented formally by theories in graph normal form. Such a collection of inter-referring, named statements seems to be at the heart of all the semantic paradoxes and, moreover, it seems to us that a syntactic approach to the question of paradox is in line with some basic intuitions. Forming a statement negating itself, for instance, is indeed a possibility in natural language, and this seems to have more to do with linguistics than with any semantic notions pertaining to the possible *meaning* of such a statement. In Paper **C**, we develop this view further, providing a more comprehensive argument in favor of studying propositional discourses. We argue, in particular, that the semantic paradoxes are exactly those theories in graph normal form that are inconsistent. In fact, this becomes our *definition* of paradox.

While our perspective seems novel, it relies heavily on work done by Roy Cook [14], who introduced what was essentially a formulation of the graph normal form relying on the use of a falsity predicate instead of negation. He did not observe that it was a normal form for propositional logic, however, and his

²Of course, dialetheism is one possible response to semantic paradox, see e.g., Priest [43]. We do not find it particularly attractive, however, and since it is not really related to our own technical contributions, we do not discuss it further here.

focus was mostly on infinite paradoxes, such as Yablo’s paradox [50]. Crucially, however, he noted the connection to kernels in directed graphs.

It should be mentioned that both Cook’s work and our own bear close resemblance to work done by Haim Gaifman on so-called *pointer structures*. These are similar to theories in graph normal form, and Gaifman also relies on the use of directed graphs to analyze them, see [27, 26]. Still, Gaifman’s work is different in that his focus is on arriving at non-classical rules saying how to assign to particular statements the label “paradox” – an explicit third semantic value. When to designate something as paradox, then, becomes the main issue, and this pushes into the background the search for those combinatorial structures that are responsible for letting paradox arise in the first place. We should also mention Thomas Bolander’s PhD thesis [4], where he employed graph-theoretic techniques to study paradox, albeit in the context of a traditional formalization, relying on the use of a first order language, focusing on paradoxes that arises in arithmetic theories due to (variants of) Convention T and unrestricted universal quantification.

We believe that our approach have three particular advantages compared to earlier suggestions in a similar vein. First, it is *simple*, making do with only propositional means. Second, it is *general*, making the study of paradox the same as the study of classical consistency of theories in a normal form. Thirdly, and most importantly, the equivalent representation in terms of digraphs allows for a very fine-grained analysis of paradoxical structures in terms of kernel theory. In Paper C, we study this approach to the paradoxes in detail, consider a series of examples, survey some results, and give a logic for reasoning about discourses that are inconsistent but admit well-defined, consistent sub-discourses.

1.1.3 Argumentation

The study of argumentation has a long philosophical tradition behind it, dating back at least to Socrates and the ancient Greeks, probably further. The desire to arrive at some general notions of what counts as a logically correct argument, in particular, seems to arise naturally in all human societies. If there is interaction, there is argument, and some preliminary agreement on what is required for an argument to count as *successful* is of great importance, if nothing else, then for very pragmatic reasons.

Following Frege and the formal turn in logic, however, the study of argumentation was mostly seen as a separate issue, belonging, at best, to the informal branch. An account of argumentation, in particular, must typically provide some answers also in the context of vagueness and uncertainty – even contradiction – and this was increasingly something that was perceived to be outside the cold realm of pure logic. The search for logical perfection would famously flounder over results on incompleteness and undecidability, however, and since then, the trend has been turning. Especially following the increasing popularity of non-classical logics – designed specifically to model warmer notions – the distinction between argumentation and logic is becoming increasingly blurred.

This development took a particularly interesting turn with the seminal work

of Dung [21], who established a particularly nice formal connection between argumentation on the one hand and non-monotonic reasoning and logic programming on the other. Since then, abstract argumentation has become very popular in the AI-community. The theory proposed by Dung centers around the notion of an *argumentation framework*, which is simply a directed graph, $F = \langle \mathcal{A}, \mathcal{R} \rangle$, where vertices, \mathcal{A} , are interpreted as arguments and the edge-relation, \mathcal{R} , is interpreted as a relation of *attack*. Given two arguments, a and b , an edge (a, b) is thought of as representing an attack made by the argument a against the argument b . Following the custom in argumentation theory, we write $\mathcal{R}^+(x) = \{y \mid (x, y) \in \mathcal{R}\}$ and $\mathcal{R}^-(x) = \{y \mid (y, x) \in \mathcal{R}\}$. Walks, paths and cycles are defined as in the case of digraphs, c.f., Section 1.1.1.

Arguments are assumed to not have any internal structure, so argumentation frameworks provide an abstract point of view that allows us to investigate notions of successful argumentation without having to make any commitments with regards to the underlying logic, much less the subject matter. This means that any insights provided by such an investigation will be widely applicable. It might also raise the worry that argumentation frameworks are *too* abstract to elucidate the nature of argumentation, but the vast body of work devoted to them in recent years, and the great proliferation of different ideas, suggest that they do indeed capture a non-trivial essence worthy of theoretical consideration. For an overview of the field, including a more comprehensive historical background, and an exposition detailing its importance to research in AI, we point to [11].

In much of the work done on argumentation, the goal is to identify the successful sets of arguments in a framework $F = \langle \mathcal{A}, \mathcal{R} \rangle$. A *semantics* for argumentation, in particular, is an operator s , which provides, for any framework F , a set $s(F) \subseteq 2^{\mathcal{A}}$, containing all such sets of arguments. Now, in order to regard $A \subseteq \mathcal{A}$ as successful, it seems intuitively reasonable to require that all arguments from A are mutually compatible, and also that they are in some sense able to defend themselves against other arguments. The first intuition is made formal by the requirement that A must be *conflict free*: no two arguments from A can attack one another. The second intuition is typically made formal by considering a function $\mathcal{D} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ defined by $\mathcal{D}(A) = \{x \in \mathcal{A} \mid \mathcal{R}^-(x) \subseteq \mathcal{R}^+(A)\}$ for all $A \subseteq \mathcal{A}$. Intuitively, the function $\mathcal{D}(A)$ gives the set of arguments that A defends; the arguments x such that every argument which attacks x is in turn attacked by A . The requirement that A must defend itself against attack is then expressible as $A \subseteq \mathcal{D}(A)$.

In argumentation, a conflict free set of arguments that defends itself is called *admissible*. The term was coined by Dung in his original paper [21], and unless we state otherwise, all the other semantic notions that we consider also originate from this paper. For a more detailed account of basic semantic notions in argumentation theory, including also some that we do not discuss here, we point to [45, Chapter 2], a newly published book that presents core notions and gives a nice presentation of the wide range of different research challenges in

this field.³

While they capture an intuitive notion of success, it seems that the admissible sets are in some sense *too* permissive. The empty set, for instance, is always admissible; it has nothing to defend and is not in conflict with anything. But taking this set to be successful seems unnatural; after all, nothing was argued for! It seems, in particular, that some notion of *maximality* must be imposed on admissible sets before we can regard them as truly successful. For while it might be that a good argument can be constructed by restricting attention to only part of the framework, this will often appear inappropriate; you succeed because you choose to ignore contentious issues. From this worry arises other semantics for argumentation which are built upon the notion of admissibility. In an intuitive sense, they aim to limit the room for opportunism in putting together arguments, and they achieve this by requiring that your admissible sets should say something about as many arguments from the framework as possible (by either including or attacking them).

The strongest concept, providing a notion of success that is particularly conclusive, is that of a *stable set*, defined by saying that a set $A \subset \mathcal{A}$ is stable iff $\mathcal{R}^+(A) = \mathcal{A} \setminus A$. So A is a stable set iff it is conflict free and also such that it attacks every argument not in A . Obviously, this implies that A defends itself, so stable sets are also admissible. While stable sets provide a notion of success that seems indisputable, it cannot be adopted in general for the simple reason that such sets sometimes fail to exist. The single argument attacking itself is the basic example.

So, in general, admissibility requires too little, while stability requires too much. As a result, a range of intermediate notions have been introduced. The *complete* sets, for instance, are those admissible sets for which we have $\mathcal{D}(A) = A$, i.e., those for which A not only defends itself, but also contains everything that it defends. We note that taking an admissible set and completing it is always a possibility – you simply add all defended vertices and iterate the process until a fixed point is reached. It is not hard to show that you never violate conflict freeness in this way, so every admissible set can be extended to a complete set that contains it. An important special case is the completion of the empty set, which is called *grounded*. In fact, one popular semantics for argumentation, especially in applications, takes this as the *only* acceptable set of arguments. Then we are left with what is typically called a *unique status* semantics, the name given to all those semantics, s , such that we have, for every F , $|s(F)| = 1$.

While removing all uncertainty regarding the correct outcome of an argument might be useful in some cases, it really only shifts uncertainty away from

³We also remark that Chapter 5 gives a survey of complexity results for various decision problems that arise in argumentation. Apart from many problems related to the so-called grounded semantics, these are mostly non-tractable. We omit discussing complexity issues further here, since our own work does not address this aspect. In Paper **A**, however, we develop algorithmic techniques and give *exact* algorithms for finding kernels in digraphs, and this is relevant to argumentation. As far as we are aware, exact algorithms for problems in argumentation is still a mostly unexplored field of research.

the model, where it is an artifact of the system, and into the semantics, as a meta-question about which notion to use. To illustrate this, it is enough to note that the grounded semantics is not the only unique status semantics that has been proposed based on admissible sets. There are at least two others: the *ideal* semantics [22], and the *eager* semantics [9]. Rather than going into detail about their properties, we simply note that since no unique status semantics is likely to prove conclusive, it seems to us that in the context of a theoretical inquiry, it makes more sense to consider general notions, such as admissibility and stability, than to focus on particular notions that simply extract from the general ones a particular point of view. In fact, it seems that the choice between different notions can itself be made subject to formal inquiry, and we think, in particular, that unique status semantics should be characterized at the *logical* level, in terms of how they arise from more basic concepts. This is a challenge for future work, and we note that it has already received attention from the point of view of modal logic [31, 30, 10].

In addition to completeness, some other notions of maximality have also been considered. The *preferred* semantics, for instance, which requires A to be a *maximal* admissible set with respect to set inclusion, i.e., such that there is no admissible set A' with $A \subset A'$. Another notion that belongs to the same category is the notion of a *semi-stable* set, introduced by Caminada [8]. These are admissible sets A such that $A \cup \mathcal{R}^+(A)$ is maximal, i.e., such that you do not maximize the number of accepted arguments as with preferred sets, but the number of arguments that obtain a definite status of being either accepted or attacked.

Given some semantics, s , we can also use it to bestow a semantic status upon individual arguments. If an argument $a \in \mathcal{A}$ is such that there is some $S \in s(\mathbf{F})$ with $a \in S$, then we say that a is *credulously accepted* in \mathbf{F} with respect to s . An argument $a \in \mathcal{A}$ is said to be *skeptically accepted* if we have $a \in S$ for all $S \in s(\mathbf{F})$. If an argument is neither credulously nor skeptically accepted with respect to a semantics, it is *rejected*. Arguments that are credulously but not skeptically accepted are often called *defensible*. Intuitively, an argument that is credulously accepted is involved in some successful line of argument; it is potentially useful, and should be considered further. An argument that is skeptically accepted, on the other hand, is involved in all successful lines of arguments; it is beyond reproach, and arguing against it should be considered useless.

Notice that credulous acceptance with respect to admissible, complete and preferred semantics is the same notion. Now, rather than going further in exploring the particularities of argumentation, we remark instead that stable and admissible sets seem to form two ends of a spectrum; all stable sets are semi-stable, all semi-stable sets are preferred, all preferred sets are complete and all complete sets are admissible. Also, it is not hard to see that the grounded set is contained in every complete set of arguments, so that skeptical acceptance with respect to complete semantics is exactly the same as membership in the grounded set. While semantics have also been suggested that do not rely on admissibility (most notably the $\mathcal{CF}2$ semantics [1]), it seems that gaining a better

understanding of admissibility and stability is a key question in argumentation theory. This is what we study in this thesis, using combinatorial techniques.

In fact, the reader might have already noticed that admissible sets and stable sets correspond more or less exactly to semikernels and kernels from digraph theory. We discuss the connection further in Section 2.1, but note already that this observation seems potentially significant. It means that work done in kernel theory has immediate relevance to argumentation theory, and it might also suggest new directions of theoretical research in digraph theory based on notions coming from argumentation.

In the literature on abstract argumentation, despite the philosophical roots of the field, it has become common to adopt a pragmatic point of view, where new developments are motivated mostly by applications in AI. When some shortcoming has been identified concerning some semantic, in some context, then a new one is proposed to address these shortcomings, often, it seems, on a case by case basis. Typically, some structural and algorithmic results are provided along with new notions, but theoretical attention then subsides and the focus is again shifted towards applications. Soon, the inevitable shortcomings of new notions lead to further proliferation of proposals.

It seems to us, however, that since much of the appeal of argumentation frameworks lie in their abstract nature, more theoretical work – mathematically and logically oriented – should be devoted to the study of core semantic notions. The close connection to applications in artificial intelligence notwithstanding, a framework is nothing but a directed graph, and in many cases we wonder if it might not be best to view it as such, also in order to gain a better understanding of argumentation.

Indeed, some recent work seem to suggest that a more theoretical trend is emerging. We have seen, in particular, an increasing number of papers devoted to giving a logical account, see e.g., [31, 33, 32, 30, 10]. This work relies on the use of modal logic. In our work, we follow the same conceptual map, but we take a different route, looking at argumentation frameworks as theories in graph normal form. This allows us to connect the stable semantics directly to classical logic. Also, it turns out that the notion of a complete set corresponds to the notion of consistency in Łukasiewicz logic L3, see [24] and Paper C. In Paper C, we also develop a logic based on admissible sets, and give a sound and complete sequent calculus reasoning system for this logic. We believe that our point of view provides further motivation for a purely theoretical study of the two notions of stability and admissibility.

Studying *when* stability is unattainable, in particular, seems like an important research challenge. Moreover, studying how the notion of admissibility behaves in non-stable argumentation frameworks leads to what we think is a highly interesting perspective, suggesting the possibility of arriving at a taxonomy of different cases and different forms of inconsistency. Combinatorial tools and techniques, developed in graph theory, both provide new insights and aid in investigations that explore the structure of successful argumentation.

The combinatorial point of view also seems likely to prove itself useful with respect to the issue of how to address shortcomings of proposed semantics.

While shortcomings themselves are often seen only when applications are considered, a theoretical investigation of *why* and *when* they arise – in structural terms, and in terms of already established notions – might be the right way to proceed, perhaps a more adequate response than to keep making new proposals.

This summarizes the point of view on argumentation that we adopt in this thesis, and we hope that our work suggests the appropriateness of devoting attention to theoretical, mathematical investigations in this field. We also note, however, that while basic notions from argumentation are found in kernel theory, many of the fine-tuned distinctions made in argumentation – motivated by applications – deserve theoretical attention, and might suggest new directions for mathematical research. We hope more cross-fertilization of ideas will be possible between these two fields in the future. Argumentation seems likely to become even more important to AI, with research into logics for multi-agent societies and agreement technologies increasingly turning to abstract argumentation for ideas and formal tools [42, 5]. It seems clear to us that a formal approach, using logic and digraph theory, will prove even more valuable to this field in the future.

Chapter 2

Presentation of main results

2.1 Connections between different areas of research

In Chapter 1, we have briefly presented three topics in graph theory, philosophy and artificial intelligence, and in this section we discuss the link between them. This is not technically challenging, but we think that the connection should be exploited in future research in all of these areas, and can facilitate interesting discussions and exchange of ideas. All the connections we discuss here have been observed before. Roy Cook observed the connection between kernels and paradoxes in [14], the fact that digraphs provide a normal form for propositional logic was observed in [3], and the connection between kernels and argumentation was noted in [15]. Still, we believe this thesis is the first time that all these fields of research have been considered together. Also, unlike previous work, we do not primarily address any one of them in particular, but consider technical questions that seem common to all.

The basic observation connecting argumentation to kernel theory is quite trivial. We recall from Section 1.1.1 that a kernel $K \in Kr(\mathbf{G})$ in a digraph \mathbf{G} is a set $K \subseteq G$ such that $N^-(K) = G \setminus K$. The connection to the semantics of argumentation should be clear. If we let $\overleftarrow{\mathbf{G}}$ denote the digraph obtained by reversing all edges in \mathbf{G} , then it is not hard to verify that a kernel in \mathbf{G} is a stable set in $\overleftarrow{\mathbf{G}}$ and vice versa. Also, recall that local kernels are sets $L \subseteq G$ such that $N(L) \subseteq N^-(L) \subseteq G \setminus L$. Then it is easy to verify that a local kernel in \mathbf{G} is an admissible set in $\overleftarrow{\mathbf{G}}$ and vice versa.

It seems that techniques and results from kernel theory have yet to be fully explored from the point of view of argumentation; as far as we are aware, the connection has only been briefly mentioned. In [15], for instance, the authors study irreflexive, symmetric argumentation frameworks (frameworks for which the edge-relation is symmetric), and show that every such framework admits a stable set and that the preferred sets agree with the stable ones. They also

mention briefly that stable sets are the same as kernels in directed graphs. Still, they do not explore this aspect, and do not seem to be aware of the results we summarized in Theorem 1.6, which guarantees kernel perfectness under much weaker assumptions.¹

When discussing paradox in Section 1.1.2, we mentioned that we think of theories in graph normal form as representing propositional discourses, and we argued that studying its properties is interesting because it is in these structures that semantic paradoxes arise. We also mentioned that we wished to conduct this study using combinatorial means. Now we briefly demonstrate how this is possible by showing how kernels provide the necessary link between digraphs and logic. This observation is due to Cook [14]. We start from a digraph G , and we form the corresponding theory $T_G = \{x \leftrightarrow \bigwedge_{y \in N(x)} \neg y \mid x \in G\}$. Then, assuming that K is a kernel in G , we define $\delta_K : G \rightarrow \{\mathbf{0}, \mathbf{1}\}$ such that

$$\delta(x) = \begin{cases} \mathbf{1} & \text{if } x \in K \\ \mathbf{0} & \text{if } x \in G \setminus K \end{cases} \quad (1.1)$$

It is easy to verify that δ is a satisfying assignment to T_G , i.e., that it makes all the equivalences true under Boolean evaluation. Going the other way is also trivial; if $\delta : G \rightarrow \{\mathbf{0}, \mathbf{1}\}$ is satisfying for T_G , then it is clear that $K_\delta = \{x \in G \mid \delta(x) = \mathbf{1}\}$ will be a kernel in G .

Going from theories in graph normal form to digraphs is not much more difficult. Let $I \in \{\mathbb{N}, \{\{1, \dots, n\} \mid n \in \mathbb{N}\}\}$ and assume that $T = \{x_i \leftrightarrow \bigwedge_{x \in X_i} \neg x \mid i \in I\}$ is some (countable) theory in graph normal form. Then, we let F be the set of variables from T that does not occur on the left of any equivalence, and we form the corresponding digraph $G_T = \langle G_T, N_T \rangle$ where

$$\begin{aligned} G_T &= \{x_i \mid i \in I\} \cup \{x, \bar{x} \mid x \in F\} \\ N_T &= \bigcup_{i \in I} \{(x_i, x) \mid x \in X_i\} \cup \{(x, \bar{x}), (\bar{x}, x) \mid x \in F\} \end{aligned} \quad (1.2)$$

We introduce a fresh vertex \bar{x} for every variable x that does not occur to the left of any equivalence. The reason is that we do not want to force x to be in the kernel of the digraph corresponding to the theory. Rather, x should be open for both acceptance and rejection, depending on the rest of the theory. This is achieved by adding the symmetric edge $\{(x, \bar{x}), (\bar{x}, x)\}$.

Let $B = \{\bar{x} \mid x \in F\}$ denote all the vertices from G_T that do not correspond to propositional letters in T . Given a function $\alpha : X \rightarrow Y$, we let $\alpha|_Z$ denote its restriction to domain $Z \subseteq X$. Also, given $\delta : X \rightarrow \{\mathbf{0}, \mathbf{1}\}$, we let $\bar{\delta}$ denote the Boolean evaluation of formulas over X induced by δ according to classical logic. Now, given a kernel $K \subseteq G_T$, we define $\delta_K : G_T \rightarrow \{\mathbf{0}, \mathbf{1}\}$ as in Equation (1.1).

Then it is easily verified that $\delta_K|_{G_T \setminus B}$ is a satisfying assignment for T in classical logic. Similarly, if we are given a satisfying assignment $\delta : G_T \setminus B \rightarrow \{\mathbf{0}, \mathbf{1}\}$, we obtain the kernel $K_\delta = \{x \in G_T \mid \delta(x) = \mathbf{1}\} \cup \{\bar{x} \in G_T \mid \delta(x) = \mathbf{0}\}$.

¹In fact, remember Theorem 1.5, stating that a digraph is kernel perfect iff every non-empty induced subdigraph has a non-empty semikernel. In light of this, it is not hard to see that for a kernel perfect digraph, every semikernel can be extended to a kernel. So in kernel perfect digraphs, any preferred set is stable, covering also this aspect of their result

In [3], it is shown that the graph normal form is indeed a normal form for propositional logic; every propositional theory has an equisatisfiable one containing only formulas of this form.² This is not very difficult to demonstrate, but we omit the details in the construction, and refer the reader to the presentation in [3]. What is important is its consequence, namely that directed graphs and the notion of kernel suffice to give an equivalent definition of classical propositional logic. For *any* theory in propositional logic, there is a corresponding digraph that has a kernel iff this theory is consistent.

This allows us to study consistency in classical logic as a graph-theoretical problem. The importance of this result depends on *how* you translate theories, however. The structural information lost by the transformation to graph normal form must not exceed the gain from doing so. In fact, the simple transformation presented in [3] does not seem to fare well in this regard, so a search for more interesting ways of transforming theories into graph normal form is an interesting direction for future research. For theories already in graph normal form, however, the tools and results from digraph theory have immediate relevance, and the fact that they correspond to notions that have been independently introduced in argumentation only adds further weight to the claim that the combinatorial study of inconsistency in propositional discourse is interesting and worthwhile.

2.2 Algorithmic results

The problem of determining if a given digraph has a kernel, KER, is NP-complete [13]. While NP-complete problems are considered hard, the search for fast, exact algorithms that solve hard problems is an active field of research in computer science [49, 25]. As far as we are aware, however, the challenge of designing such algorithms for KER has not been much addressed from the point of view of either kernels in digraphs nor stable sets in argumentation.³ In Paper **A**, we make a contribution in this regard, giving an algorithm which shows that while KER is hard, it is among those hard problems for which a reasonably fast exponential algorithm can be found. We give, in particular, an algorithm for which we are able to show – using a non-trivial analysis – that the complexity is $\mathcal{O}^*(1.427^{|G|})$ for the general case and $\mathcal{O}^*(1.286^{|G|})$ for digraphs that do not contain any symmetric edges.⁴

Since both KER and SAT, the problem of determining satisfiability of propositional theories, are NP-complete, they are equivalent in terms computational complexity. However, an equivalence on the level of complexity classes does

²Equisatisfiable means that for every satisfying assignment to one there is a satisfying assignment to the other, i.e., the assignments are not necessarily the same (new propositional letters might need to be introduced)

³A notable exception is [18], which addresses algorithmic questions from both a theoretical and practical angle, but without focusing on giving tight upper bounds. We should also mention that much work has been devoted to studying where various computational problems from argumentation belong in the hierarchy of complexity classes, see e.g., [23].

⁴The notation $\mathcal{O}^*(\cdot)$ suppresses polynomial factors from exponential functions.

not necessarily say much about the relationship between the actual techniques that an optimal algorithm should employ, and how fast such an algorithm can be expected to run. Interestingly, the algorithm we propose for KER is basically just an adaptation of the standard DPLL SAT-algorithm [17, 16]. But the running time, while hypothesized to approximate 2^n for worst case SAT (where n is the number of propositional letters), is much better.⁵ Since digraphs encode the graph normal form of propositional theories, this means that the graph normal form is a normal form for which SAT can be decided relatively quickly. Moreover, translating arbitrary propositional theories to graph normal form takes only linear time. It typically requires the introduction of many additional variables, however, affecting thus a parameter on which the running time of the algorithm depends exponentially. Therefore, there are no immediate implications for solving SAT in general. Our results do suggest, however, that when small equivalent theories in graph normal form can be found, there is a computational benefit in solving SAT on these rather than the original theory.

In Paper **A**, we also address fixed parameter tractability of KER, attempting to come up with algorithms that are exponential in some other parameter than the number of vertices in the digraph. We show, in particular, how to solve KER by transforming G into a directed acyclic graph (DAG). The construction starts from an arbitrary feedback vertex set – a set of vertices such that removing them leaves an acyclic digraph – and the complexity becomes $\mathcal{O}(2^{|F|})$ where F is the set used to construct the DAG. KER, in particular, is fixed parameter tractable in the size of any feedback vertex set in the digraph, e.g., in the smallest such set. This result was also obtained in [18], using a similar, but, we feel, slightly less elegant technique. While the result is nice when small feedback vertex sets can be found easily, it is of limited significance in the general case, since small feedback vertex sets need not exist.⁶

We show, however, that KER is also fixed parameter tractable in any set of vertices that breaks only the even cycles in G . The trivial algorithm runs in time $\mathcal{O}(2^{|E|})$ in this case, where E is some set of vertices, the removal of which leaves a digraph with no even cycles. This result, while not very difficult, is structurally interesting, but as for the case of using feedback vertex sets, it can only be expected to prove useful in special cases.

We believe that our work on algorithmic techniques illustrates that KER is an interesting problem to study from the point of view of finding fast exact algorithms. It is useful to think of the problem as involving a search through the maximal independent sets in a digraph, testing whether or not they are also absorbing. It seems likely that more clever tricks can be found that can be used to rule out some such sets in advance or during computation. We have hopes, in particular, that faster algorithms can be designed. It might also be possible to analyze the running time more optimally, using the measure and conquer approach described in [25]. More generally, we think it is interesting to

⁵This assumption about the hardness of SAT is called the strong exponential time hypothesis [35] [7]

⁶Also, since the problem of finding the smallest feedback vertex set is itself NP-complete, in fact, among the original problems considered in Karp's seminal paper [36]

observe how the problem of KER is closely connected to SAT. The algorithmic techniques we adopt, while formulated on digraphs, correspond closely to techniques used to solve SAT, and we believe that future research should address further the relationship between these two problems, attempting to explore and exploit the close link between them.

2.3 Structural results

2.3.1 Sufficient conditions for existence of kernels

We mentioned in Section 1.1.1 that kernel theory has devoted quite some attention to the problem of finding structural conditions that ensure kernel perfectness. The typical approach is to look inside induced odd cycles for the presence of suitable internal structures that ensures that they can all be mutually resolved. In our own work, however, we have focused on what the odd cycles point to on the outside, trying to identify conditions under which they can be mutually resolved because they point to suitable external structures.

As usual in kernel theory, the notion of a semikernel is crucial to the analysis, and our arguments proceed by induction on the size of digraphs. We cannot assume existence of kernels in arbitrary subdigraphs, so we consider instead appropriate *sequences* of semikernels, the idea being that we can compose these, remove those vertices that obtain a definite status, and eventually reach a subdigraph for which our structural conditions hold. Then we can apply an induction hypothesis in the standard manner, combining our sequence of semikernels with a kernel provided by hypothesis. Formalizing this idea leads to the notion of a *solver*, introduced in Paper B.

Definition 3.1 *A solver for a digraph G is a sequence of induced subdigraphs and semikernels $\langle G_i, S_i \rangle_{1 \leq i \leq n}$ such that:*

- (1) $G_1 = G$
- (2) S_i is a semikernel in G_i for all $1 \leq i \leq n - 1$
- (3) $G_{i+1} = G_i \setminus (S_i \cup N^-(S_i))$ for all $1 \leq i \leq n - 1$
- (4) S_n is a kernel of G_n .

In Paper B, we also establish the following easy result, which is what makes solvers useful for establishing existence of kernels.

Theorem 3.2 *A digraph has a kernel iff it has a solver.*

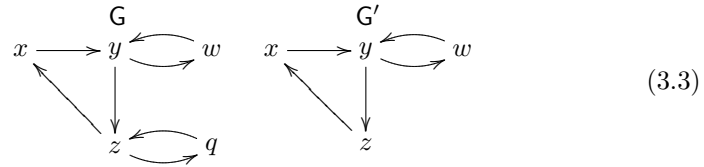
Using solvers, we are able to show, by fairly complicated arguments, that a range of various conditions is sufficient to ensure the existence of kernels in digraphs that are not kernel perfect. Furthermore, we demonstrate that conditions concerning the parity of cycles in the *underlying undirected graph*, \underline{G} , obtained from G by forgetting the direction of all edges, can be very useful in

establishing such conditions. We show, in particular, that whether or not an external resolution of odd cycles is available can depend on the nature of the interplay between the odd directed cycles of \mathbf{G} , and the odd, undirected cycles of $\underline{\mathbf{G}}$.

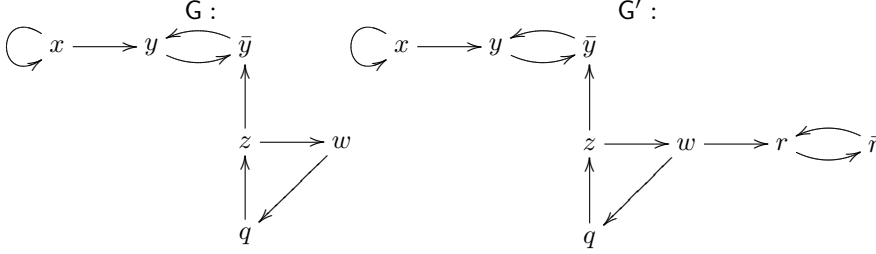
We introduce the notion of *freeness*, saying that a vertex $x \in G$ is free if it does not lie on an odd cycle from $\underline{\mathbf{G}}$. This extends to sets of vertices $F \subseteq G$ such that if every vertex in F is free, then F itself is said to be free. We then consider various ways in which a free set can be used to break odd cycles. This typically makes it necessary to place additional restrictions on F , like requiring the existence of semikernels containing its members, or restricting the allowed parity of paths between the vertices it contains. The reason why this is needed has to do with the general challenge we mentioned in Section 1.1.1, namely to break all odd cycles *simultaneously*. We must ensure, in particular, that the partial solutions we come up with are all compatible.

A basic requirement that keeps reappearing, and that seems critical in order for this to be possible, is that each odd cycle must point to at least *two* substructures of the appropriate kind. One such structure, in general, is not enough. Interestingly, this parallels the structure of the results in Theorem 1.6, where two chords of the appropriate kind (and with the appropriate interaction) is typically required, even if this is stronger than what seems to be needed when one considers simple cases.

The digraph, \mathbf{G} , for instance, depicted below, is ensured to have a kernel by Theorem 2.6 from Paper **B**



Actually, also \mathbf{G}' has a kernel ensured by Theorem 2.6, but this is little more than a sneaky trick: you can always choose *one* odd cycle to solve in the naive way, without taking compatibility into account. For all the others, however, compatibility is a subtle issue. Essentially, the reason why we can make do with two structures of the appropriate kind in our results is that the other conditions we stipulate are such that they limit the possible interaction between these two structures. Then, even if we "spend" one of them to solve a particular odd cycle, we still have at least one structure left for each among all remaining odd cycles. Moreover, since interaction is limited, we know that this structure is unaffected by our partial solution. As an example, consider the following two digraphs:



In both G and G' , there are two problematic, self-defeating sequences: (x, x) and (z, w, q, z) , and in both digraphs, it is tempting to look at vertices y, \bar{y} for a possible resolution. There is a problem, however, namely that they can only solve one of the sequences in question. In both G and G' , we have semikernels $\{y\}$ and $\{\bar{y}, w\}$, corresponding to whether we use them to solve (x, x) , or use them to solve (z, w, q, z) . In G , this is where the story ends – it is not possible to resolve both, and we conclude that $Kr(G) = \emptyset$. In G' , on the other hand, it is possible to solve both, but only if you solve (x, x) first by choosing y . This, in particular, no longer precludes solving (z, w, q, z) , since it is possible to choose r and obtain the kernel $\{y, r, z\}$, as predicted also by Theorem 2.6 from Paper **B**.

Examples such as these seem particularly interesting from the point of view of argumentation. In this context, our results can be interpreted as describing circumstances under which credulous acceptance of specific arguments leads to resolution of other, possibly problematic, self-defeating sequences of arguments. It is tempting, in particular, to think of it as *necessary* to accept arguments such as y and r . It seems necessary to accept them, not because they cannot be refuted, but because accepting them is needed in order to resolve problems with self-defeat affecting other parts of the network. A basic intuition in argumentation theory is that it is an overreaching goal to minimize the number of arguments that are not assigned any semantic status – a shared responsibility, one might say, among participants in the argumentation scenario. It is interesting, therefore, to investigate further what the results from Paper **B** imply for argumentation, especially with respect to the notions of maximality that is used to formulate non-classical semantic notions (like the preferred and semi-stable semantics).

Considering the link with propositional discourse and paradox makes this line of thought even more exciting. Could it be that the necessary truth of particular statements in natural discourse follows not from what they say about the world or other statements, but instead from what other statements say *about them*? Could it be that the paradoxes – when seen as a holistic phenomenon arising from the totality of a discourse – suggests that truth itself must be accounted for in an holistic, ungrounded fashion? Perhaps truth – like paradox – is not really a property of statements or propositions, but of *discourses*. The truth of particular statements, in particular, is perhaps only the end product of the correct semantic account of the discourse of which they form part. Such

an account, it is tempting to think, can not be aggregated from looking at the truth of individual constituents in the discourse. Rather, it seems that the discourse itself is the primitive object, and that individual statements are more like vantage points – providing only a particular perspective on something *indivisible*.

2.3.2 Relations that preserve structural properties

Given two argumentation frameworks, it is natural to ask if there are maps between their vertices that preserve and reflect stable sets, admissible sets, or sets prescribed by any of the other semantics considered in argumentation. This, in particular, leads to notions of *equivalence* between frameworks, where frameworks F and F_2 are said to be equivalent modulo some semantics s , if there is some relation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ such that

- For all $S \in s(F)$, we have $\beta(S) \in s(F_2)$ (preservation)
- For all $S' \in s(F_2)$, we have $\beta^-(S') \in s(F)$ (reflection)

Given such a notion of equivalence, the question becomes if we can find conditions such that if they are satisfied by a relation, then the relation is an equivalence relation. This has not been much explored in kernel theory, and to our knowledge, it has not been much studied in argumentation theory either. In the context of argumentation, this might be due to the abstract nature of such a notion of equivalence. In fact, it is more usual to consider two frameworks as being equivalent when they admit extensions that are *syntactically* the same, as in [41]. Still, we believe that the more general notion of equivalence presented above might prove particularly useful for argumentation, especially with regards to the overarching goal of formalizing its notions using logic.

We study equivalence in Paper D, with respect to admissible, complete, preferred, semi-stable and stable semantics. First, we attempt to argue in more detail for the point of view that this notion of equivalence is interesting. This might be a bit of a contentious issue, especially since our general notion of equivalence leads to collapses with respect to certain semantics for argumentation. With respect to unique status semantics, in particular, the general notion of equivalence separates all frameworks into two classes, those that have a non-empty extension, and those that do not. This might disconcert those coming at this from the point of view of applications in AI, but for us, coming from logic, it makes perfect sense; a unique status semantics picks a set of tautologies, arguments that you cannot dispute. The fact that any two non-empty sets of tautologies are the same does not worry us, but seems reasonable. Indeed, for us, the collapse is nothing more than a reflection of the fact that unique status semantics completely flattens the semantic structure that we are trying to explore. What interests us are semantic notions that give rise to *logical consequence*, and unique status semantics are completely uninteresting in this regard. There, the collapse has happened already at the semantic level; all arguments belong to one of three categories, accepted, rejected, or rejected *and* defeated.

Consequently, there are no interesting dependencies between them.⁷ But assume, on the other hand, that in some framework F , you are working with a semantics that requires you to accept arguments b and c whenever you accept argument a , but also allows you to defeat all of them. Or even more interesting: consider the case when there is some admissible set allowing you to choose a , but only at the cost of making it impossible to extend the set to a stable or semi-stable one. Clearly, when this happens, there are some structures in F which makes it happen, and our aim is to characterize these structures *more abstractly* than by simply pointing at them whenever we think they arise in some concrete framework. As a part of this project, we *want* to group frameworks together, and preferably, although unlikely due to the hardness of the problem, we want few classes. Collapse, in particular, is a good thing, and it is exactly what a general notion of equivalence should provide whenever possible.

In Paper **D**, we focus our technical work on bisimulations. For us, only the conditions pertaining to connectivity in the digraphs are relevant, so we take a bisimulation to be a relation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ such that for all $x \in \mathcal{A}, x' \in \mathcal{A}_2$ with $x\beta x'$, we have

- **Forth:** If $z \in \mathcal{R}^-(x)$, then there is $z' \in \mathcal{R}_2^-(x')$ such that $z\beta z'$
- **Back:** If $z' \in \mathcal{R}_2^-(x')$, then there is $z \in \mathcal{R}^-(x)$ such that $z\beta z'$

It is not hard to see that bisimulations are neither sufficient nor necessary to ensure equivalence with respect to any of the semantics for argumentation that we consider. Still, as we show in Paper **D**, bisimulations have some nice features with respect to agreement between various semantics for argumentation. We show, in particular, that if an equivalence relation with respect to admissibility is also a bisimulation, then it is also an equivalence relation with respect to the preferred, stable and semi-stable semantics. Not very surprising, perhaps, but not entirely obvious either. Interestingly, however, this result does not hold for the complete semantics. There are bisimulations that witness to equivalence under admissible semantics even when the frameworks in question are *not* equivalent with respect to complete semantics (an example is depicted in Paper **D**, Figure 2).

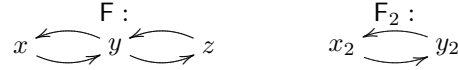
While bisimulations do not in general ensure equivalence, we show that they do so as long as they are also *finitely collapsing*, a new notion introduced in Paper **D**. Its formulation relies on the notion of an *infinite backwards walk*, a sequence of arguments $x_1x_2\dots$ such that we have $x_i \in \mathcal{R}^+(x_{i+1})$ for all $i \in \mathbb{N}$. Notice that in finite frameworks, a backwards infinite walk always involves one or more cycles. Now, we take to be finitely collapsing those bisimulations β such that

- **Global forth:** For all infinite backwards walks $\gamma = x_1x_2\dots$ in F , there is some $i \in \mathbb{N}$ such that $|\beta(x_i)| = 1$

⁷At least not until you consider questions other than those that have to do with semantic status, for instance how to *compute* the unique set of accepted arguments in the framework

- **Global back:** For all infinite backwards walks $\gamma = x_1 x_2 \dots$ in F_2 , there is some $i \in \mathbb{N}$ such that $|\beta^-(x_i)| = 1$

In Paper **D**, it is shown that finitely collapsing bisimulations are equivalence relations with respect to all the semantics that we consider. For an example of two digraphs such that their equivalence is witnessed by a finitely collapsing bisimulation, consider the following two digraphs



A bisimulation between F and F_2 is $\beta = \{(x, x_2), (y, y_2), (z, x_2)\}$, and it is easy to see that this is also finitely collapsing. Admittedly, this example is not terribly interesting, and we consider it a challenge for future research to investigate further what types of frameworks admit a finitely collapsing bisimulation between them. This is interesting in its own right, and also because it touches upon the question of how the notion of a finitely collapsing bisimulation can play a role in the search for new sufficient conditions for the existence of kernels and non-empty admissible sets. We can ask: what can the conditions we already know be transformed to under a finitely collapsing bisimulation? If some structures known to ensure kernels can be mapped to some other structures under a finitely collapsing bisimulation, then it seems likely that we will be able to demonstrate that kernels exist also in the presence of these other structures.

Even more interesting is the search for a tighter characterization of equivalence relations themselves. The ultimate goal is to arrive at some non-trivial structural conditions that are satisfied iff the relation is an equivalence with respect to some semantics. This goal seems difficult to reach, but partial results, such as those we present in Paper **D**, seeking to cover an increasing number of interesting cases, seems like the way forward.

2.4 Reasoning about paradox and admissibility

As we have stressed throughout our exposition so far, we think of our research as addressing properties and peculiarities of the graph normal form. One conceptual insight that we believe to have resulted from our work is that the graph normal form provides a particularly interesting view on classical consistency. As we argued also in Section 1.1.2, we believe that inconsistency of theories in graph normal form describes accurately the semantic paradoxes. Also, we have seen that consistency corresponds to the notion of stability in argumentation. Lastly, we have seen that when analyzing such theories, we can use tools and import results from kernel theory. A second conceptual insight seems implicit, namely that as an alternative to classical consistency, the notion of a semikernel/admissible set offers a non-classical semantics that provides, for theories in graph normal form, a interesting *local* view on classical consistency.

An admissible set requires all arguments to be defended against their attackers, i.e., that the corresponding equivalences evaluates to true under classical

rules. But it does not require a total assignment of Boolean values to all involved vertices. So have we just rediscovered strong Kleene logic [37]? In fact, the answer is no, since giving a partial assignment of Boolean values to variables in a theory in graph normal form will not – under strong Kleene logic – always be possible in such a way as to make all the equivalences in this theory true. What we have rediscovered, rather, is Łukasiewicz logic [39]. But only a special fragment of it, namely the fragment that obtains when we restrict our language exclusively to formulas of the graph normal form. This is *not*, of course, a normal form for Łukasiewicz logic, but it suggests an application of this logic to the study of paradox and admissibility. It captures, in logical terms, precisely the local view of classical inconsistency that we are after. Both \wedge and \neg are evaluated according to strong Kleene logic, and because evaluation of equivalence in Łukasiewicz logic requires *identical* values to both sides of the equivalence, this means that an essentially classical equivalence obtains whenever a boolean value appears on either side.⁸

Turning to Łukasiewicz logic, however, we are left somewhat unsatisfied with the point of view it offers. The focus, in particular, is the ordinary one, in which logical consequence is considered as taking us from a theory to the formulas that necessarily follow from it, i.e., those that come out as true under all assignments of semantic values that satisfies the theory. This, however, is a somewhat boring notion of logical consequence for theories in graph normal form. In Paper C, we show that at the level of individual arguments, it produces as logical consequences only those arguments that are in the grounded extension. The more interesting notion arises from asking what is *possible* given a theory in graph normal form – to ask what *could* be the case, given the restrictions imposed by the theory. This is where local consistency lives, together with admissible sets, semikernels, and various semantic paradoxes. In Paper C, we make a conceptual argument to the effect that this notion is important and should be studied, and we also provide a sequent calculus for reasoning about possibility in propositional discourse.

This then, in light of the connections made earlier, amounts to a reasoning system for the study of membership in semikernels, credulous acceptance with respect to admissible sets, local consistency of theories in graph normal form, and consistency for a special fragment of Łukasiewicz logic.

2.5 Conclusion and future work

In this thesis, we have studied directed graphs. We have not thought about them as purely combinatorial objects, however, but looked at them from the point of view of logic, taking them to be a particularly clear and simple representation of the propositional discourse – theories in graph normal form. We noted connections between kernel theory, logic, and the theory of abstract argumentation, and we suggested that the study of *consistency* is what unites these

⁸For what we think is a convincing argument in support of the claim that strong Kleene evaluation is essentially classical, we refer the reader to the famous footnote 18 in Kripke [38].

fields. Moreover, we suggested that inconsistency, when it arises in propositional discourse, should be thought of as *paradoxical* – that the statements leading to inconsistency do exist, but demonstrate that classical logic falls short when it is combined with a basic intuition about truth. Giving this intuition priority, while also trying to stay as close to classical logic as possible, we explored a *local view* on classical consistency – provided equivalently by semikernels from kernel theory and admissible sets from argumentation. We used combinatorial tools, and we argued that these should be exploited more actively in future work. We suggested, in particular, that kernel theory and argumentation would benefit from an increased mutual exchange of ideas and research questions.

Towards the end of each section in this chapter, we have mentioned briefly what we think are interesting directions for future research into the questions that we have addressed. In this final section, we would like to point out two possible directions for future research that we have not considered in this thesis, but which we think of as interesting challenges for the future.

First, we would like to mention *agency*. It seems, in particular, that this notion is implicit both in our account of propositional discourse, and in the account provided by argumentation. Both a discourse and an argument typically has more than one participant, in particular, and it seems that at some stage, these should be introduced into the formalism, and accounted for logically, along with the other core notions involved. Multi-agent argumentation has already received quite some attention in the literature, see for instance [45, Part III], but not much, so far, from a logical point of view. We think that the design of nice formal logics for argumentation in the context of epistemic agency is a natural and exciting direction for future research.

Secondly, we are interested in studying the *infinitary* case, when we allow infinite branching in our digraphs and infinite conjunction in our theories in graph normal form. It was shown by Cook [14] that for all finite, cyclic digraphs, there is an equivalent infinite digraph without any directed cycles, so, in fact, for the infinitary case, considering only acyclic digraphs is fully general. While it seems very difficult to arrive at elegant structural conditions ensuring kernels in this case, the goal should be to identify some structures – like the odd cycles in finitary digraphs – that are always present when inconsistency arises.⁹

We mention that the search for a characterization of such structures is closely related to the search for some nice notions of *compactness* for infinitary logic, as well as to the axiom of choice. For a presentation of the main challenges involved, and some relevant technical connections, we refer the reader to [3].

⁹One natural hypothesis, which we are currently exploring, is that some variant of Yablo’s paradox [50] is always present in infinitary acyclic digraphs that do not admit kernels

Bibliography

- [1] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1–2):162–210, 2005.
- [2] Leopoldo Bertossi, Anthony Hunter, and Torsten Schaub, editors. *Inconsistency Tolerance*, volume 3300 of *Lecture notes in computer science*. Springer, 2005.
- [3] Marc Bezem, Clemens Grabmayer, and Michal Walicki. Expressive power of digraph solvability. *Ann. Pure Appl. Logic*, 163(3):200–213, 2012.
- [4] Thomas Bolander. *Logical theories for agent introspection*. PhD thesis, Informatics and mathematical modelling, Technical University of Denmark, 2003.
- [5] Piero Bonatti, Eugenio Oliveira, Jordi Sabater-Mir, Carles Sierra, and Francesca Toni. Some open questions for the integration of trust with negotiation, argumentation and semantics. In *Cyprus panel*. 2010.
- [6] Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.
- [7] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In Jianer Chen and Fedor Fomin, editors, *Parameterized and Exact Computation*, volume 5917 of *LNCS*, pages 75–85. Springer, 2009.
- [8] Martin Caminada. Semi-stable semantics. In *Proceedings of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006*, pages 121–130, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.
- [9] Martin Caminada. Comparing two unique extension semantics for formal argumentation: Ideal and eager. In *BNAIC 2007*, pages 81–87, 2007.
- [10] Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

- [11] T. J. M. Bench Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.
- [12] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006.
- [13] Vašek Chvátal. On the computational complexity of finding a kernel. Technical Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal, 1973. <http://users.ensc.concordia.ca/~chvatal>.
- [14] Roy Cook. Patterns of paradox. *The Journal of Symbolic Logic*, 69(3):767–774, 2004.
- [15] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *Proceedings of the 8th European conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU’05*, pages 317–328, 2005.
- [16] Martin Davis, Georgs Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [17] Martin Davis and Hillary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.
- [18] Yannis Dimopoulos, Vangelis Magirou, and Christos H. Papadimitriou. On kernels, defaults and even graphs. *Annals of Mathematics and Artificial Intelligence*, 20:1–12, 1997.
- [19] Pierre Duchet. Graphes noyau-parfaits, II. *Annals of Discrete Mathematics*, 9:93–101, 1980.
- [20] Pierre Duchet and Henry Meyniel. Une généralisation du théorème de Richardson sur l’existence de noyaux dans les graphes orientés. *Discrete Mathematics*, 43(1):21–27, 1983.
- [21] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
- [22] P.M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(1015):642 – 674, 2007.
- [23] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.
- [24] Sjur Dyrkolbotn. Doing argumentation using theories in graph normal form. In Rasmus K. Rendsvig, editor, *ESSLLI 2012 Student Session Proceedings*. 2012.

- [25] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.
- [26] Haim Gaifman. Pointers to truth. *The Journal of Philosophy*, 89(5):223–261, 1992.
- [27] Haim Gaifman. Pointers to propositions. In Andre Chapuis and Anil Gupta, editors, *Circularity, Definition and Truth*, pages 79–121. Indian Council of Philosophical Research, 2000.
- [28] Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.
- [29] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatshefte für Mathematik und Physik*, 37:173–198, 1931.
- [30] Davide Grossi. Argumentation in the view of modal logic. In Peter McBurney, Iyad Rahwan, and Simon Parsons, editors, *ArgMAS*, volume 6614 of *Lecture Notes in Computer Science*, pages 190–208. Springer, 2010.
- [31] Davide Grossi. On the logic of argumentation theory. In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *AAMAS*, pages 409–416. IFAAMAS, 2010.
- [32] Davide Grossi. An application of model checking games to abstract argumentation. In Hans P. van Ditmarsch, Jérôme Lang, and Shier Ju, editors, *LORI*, volume 6953 of *Lecture Notes in Computer Science*, pages 74–86. Springer, 2011.
- [33] Davide Grossi. Fixpoints and iterated updates in abstract argumentation. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *KR*. AAAI Press, 2012.
- [34] Anil Gupta and Nuel D. Belnap. *The Revision Theory of Truth*. MIT Press, 1993.
- [35] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer System Science*, 62(2):367–375, 2001.
- [36] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
- [37] Stephen Cole Kleene. *On notiations for ordinal numbers*, volume 3. 1938.
- [38] Saul Kripke. Outline of a theory of truth. *The Journal of Philosophy*, 72(19):690–716, 1975.

- [39] Jan Łukasiewicz. On three-valued logic. In L.Borkowski, editor, *Selected works by Jan Łukasiewicz*, pages 87–88. North Holland, Amsterdam, 1970.
- [40] Victor Neumann-Lara. Seminúcleos de una digráfica. Technical report, Anales del Instituto de Matemáticas II, Universidad Nacional Autónoma México, 1971.
- [41] Emilia Oikarinen and Stefan Woltran. Characterizing strong equivalence for argumentation frameworks. *Artificial Intelligence*, 175(14–15):1985–2009, 2011.
- [42] Sascha Ossowski. Coordination and agreement in multi-agent systems. In Matthias Klusch, Michal Pechoucek, and Axel Polleres, editors, *Cooperative Information Agents XII*, volume 5180 of *Lecture Notes in Computer Science*, pages 16–23. Springer Berlin / Heidelberg, 2008.
- [43] Graham Priest. The logic of paradox. *Journal of Philosophical Logic*, 8:219–241, 1979.
- [44] W. V. Quine. *The ways of paradox and other essays*. Random House, New York, 1966.
- [45] Iyad Rahwan, editor. *Argumentation in artificial intelligence*. Springer, 2009.
- [46] Moses Richardson. Solutions of irreflexive relations. *The Annals of Mathematics, Second Series*, 58(3):573–590, 1953.
- [47] Alfred Tarski. The concept of truth in formalised languages. In John Corcoran, editor, *Logic, Semantics, Metamathematics, papers from 1923 to 1938*. Hackett Publishing Company, 1983. [translation of the Polish original from 1933].
- [48] John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944 (1947).
- [49] Gerhard J. Woeginger. Exact algorithms for np-hard problems: a survey. In Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi, editors, *Combinatorial optimization - Eureka, you shrink!*, pages 185–207. Springer-Verlag New York, Inc., New York, NY, USA, 2003.
- [50] Stephen Yablo. Paradox without self-reference. *Analysis*, 53(4):251–252, 1993.

Part II

Papers

Chapter 3

Paper A: Finding kernels or solving SAT

This paper was published in *Journal of Discrete Algorithms*, vol.10, pp.146-164, January 2012.

Finding kernels or solving SAT

Michał Walicki and Sjur Dyrkolbotn
Department of Informatics, University of Bergen

michal@ii.uib.no

Abstract

We begin by offering a new, direct proof of the equivalence between the problem of the existence of kernels in digraphs, KER, and satisfiability of propositional theories, SAT, giving linear reductions in both directions. Having introduced some linear reductions of the input graph, we present new algorithms for KER, with variations utilizing solvers of boolean equations. In the worst case, the algorithms try all assignments to either a feedback vertex set, F , or a set of nodes E touching only all even cycles. Hence KER is fixed parameter tractable not only in the size of F , as observed earlier, but also in the size of E . A slight modification of these algorithms leads to a branch and bound algorithm for KER which is virtually identical to the DPLL algorithm for SAT. This suggests deeper analogies between the two problems and the probable scenario of KER research facing the challenges known from the work on SAT. The algorithm gives also the upper bound $\mathcal{O}^*(1.427^{|G|})$ on the time complexity of general KER and $\mathcal{O}^*(1.286^{|G|})$ of KER for oriented graphs, where $|G|$ is the number of vertices.

1 Introduction

The concept of a kernel of a digraph (an independent set reachable from every outside node by an edge) was introduced in [33] as a generalization of a solution of a cooperative game and has since then found applications in both positional and cooperative game theory as well as in logic. Determining the existence of a kernel has become a problem of independent interest in graph theory, starting with the classical results of Richardson, [29, 30], and followed in the last decades by several publications, e.g., [26, 14, 15, 1, 18, 12], with a recent overview [4].

The problem of the existence of kernels in digraphs, KER, is NP-complete, [6], so in a trivial sense it is equivalent to the satisfiability of propositional theories, SAT. The equivalence has been applied, e.g., in [24] for representing finitely branching dags as consistent propositional theories, in [11, 12] for studying default logic, in [13] for correlating models of logic programs and kernels of appropriate digraphs and in [34] for analysing circularity in logical paradoxes. But it has not received a separate treatment, independent from particular applications. From an algorithmic perspective, it is natural to ask for a more fine-grained analysis of the exact relationship between SAT and KER. An answer should provide an indication both as to whether kernel theory can contribute to SAT-solving, and as to how techniques developed for SAT-solvers can be employed to increase the efficiency of deciding KER. Equivalence of the two problems with respect to some complexity class does not suffice to answer such questions because, in order for a reduction to be useful in practice, even constant factors may matter, requiring a more detailed analysis of the actual choices and possible heuristics.

In this article we focus on KER, showing that the reducibility of KER to SAT has a practical, algorithmic content. This is found not so much in the direct application of SAT-solvers, although this too is a viable approach for some cases, but rather in the similarities between the problems encountered while trying to solve KER (directly) and those faced by SAT-solvers. We present a series of novel algorithms for KER, utilizing new observations of graph-theoretical nature but also the possibility of solving SAT at appropriate places. These can be very efficient for some classes of graphs, but are hardly optimal in general. We then present our final algorithm for KER, which is

very similar to the central SAT-algorithm DPLL, [10, 9]. We review several issues which, arising from earlier experiences with SAT, are likely to affect future work on KER.

The question of how kernel theory can be used to solve SAT more effectively is left for future work, but we hope that the connection we demonstrate here indicates strongly that SAT-solvers might indeed have something to gain from utilizing the graphical nature of KER.

Section 2 introduces the basic definitions and establishes the equivalence of KER and SAT, giving new linear reductions in both directions, simpler than previously available. The problem of finding a kernel is formulated in terms of assigning boolean values to the nodes of the graph, an assignment is a *solution* when it determines a kernel and a graph is *solvable* if it has a solution. Section 3 presents some linear (or low polynomial) graph reductions which preserve and reflect solvability and are later used by the discussed algorithms. Section 4 presents several results relating solvability to various conditions on feedback vertex sets. In Subsection 4.1 we also show how to solve KER by constructing a dag from a digraph. This is essentially the technique used in the algorithms from [11, 12]. In our case, however, a single dag suffices for either finding a kernel or concluding its non-existence. In the worst case, we try all assignments to a feedback vertex set, and thus the complexity of the trivial brute-force $\mathcal{O}^*(2^{|G|})$ is reduced to $\mathcal{O}^*(2^{|F|})$, where $F \subseteq G$ is a feedback vertex set.¹ Following that, we show that one can reduce this factor even further to the number of even cycles only. Subsection 4.2 gives an algorithm which, for each assignment to a subset of nodes E touching all even cycles, determines in linear time if the resulting, induced assignment is a solution, thus giving the complexity $\mathcal{O}^*(2^{|E|})$. Both these algorithms show that the problem is fixed parameter tractable, FPT, taking the size of F , respectively E , as the parameter. We also discuss a variation which, instead of inducing the values along the obtained dag, decides solvability of the appropriate system of $|E|$ boolean equations over $|E|$ variables. Section 5 introduces the main, recursive algorithm, based on the simplifications introduced in Section 3. It subsumes the algorithm from [12] as a special case and allows to show the complexity bound $\mathcal{O}^*(1.427^{|G|})$ for the general case and $\mathcal{O}^*(1.286^{|G|})$ for oriented graphs (with no 2-cycles). It turns out that, except for the fact that it works on digraphs and not on CNFs, it is exactly the DPLL algorithm – the basis of most modern SAT-solvers. This brings a new aspect of the relationship to SAT, and we conclude listing a series of conjectures and hypotheses on the expected issues and choices in the further development of the algorithms for KER, originating from the experiences with SAT-solvers.

2 Background

A *digraph* (*directed graph*) is a pair $\mathbf{G} = \langle G, N \rangle$, where G is a finite set of nodes and $N \subseteq G \times G$ is a binary relation that describes the directed edges of \mathbf{G} .²

For a vertex $x \in G$, we denote by $N^+(x) = \{y \in G \mid N(x, y)\}$ the set of *out-neighbours* of x , and by $N^-(x) = \{y \in G \mid N(y, x)\}$ the set of *in-neighbours* of x with respect to the directed-edge relation of \mathbf{G} . *Neighbours* of x is the union of its out-neighbours and in-neighbours, $N^+(x) \cup N^-(x)$. The degree of $x \in G$, $d(x)$, is its number of neighbors. Letting $(N^+)^*$ denote the transitive closure of N^+ , we use $[x] = \{y \mid y \in (N^+)^*(x)\}$ to denote the set of vertices reachable from x and $(x) = \{y \mid x \in [y]\}$ to denote the set of vertices from which x is reachable. These notational conventions are extended to subsets of vertices, for example, for all $X \subseteq G$, we let $N^-(X) = \bigcup_{x \in X} N^-(x)$. For an $X \subseteq G$, we also write $\mathbf{G} \setminus X$ to denote the subgraph of \mathbf{G} induced by the subset $G \setminus X$.

A *walk* p is a sequence of vertices $\langle x_0, x_1, x_2, \dots, x_n \rangle$ such that $\forall 0 \leq i < n : x_{i+1} \in N^+(x_i)$ and such that all edges traversed are distinct, i.e. whenever $x_i = x_j$ for $0 \leq i \neq j < n$, we have $x_{i+1} \neq x_{j+1}$. The length of a walk is the number of edges it uses, $l(p) = n$. A walk is a *path* if it is also a sequence of distinct vertices. A *cycle* is a walk $\langle x_0, \dots, x_{n-1}, x_n \rangle$ such that $\langle x_0, \dots, x_{n-1} \rangle$

¹The notation $\mathcal{O}^*(\cdot)$ suppresses polynomial factors from exponential functions.

²Some results presented below apply to the infinite digraphs and infinitary propositional logic. However, in the present context of algorithm design, we assume all involved sets to be finite. Also, unless stated otherwise, by a graph we always mean a digraph.

is a path and $x_n = x_0 \in N^+(x_{n-1})$.

A *sink* in \mathbf{G} is a vertex $x \in G$ without out-neighbours and $\text{sinks}(\mathbf{G}) = \{x \in G \mid N^+(x) = \emptyset\}$ denotes the set of sinks of \mathbf{G} . A vertex which is not a sink is *internal*, $\text{int}(\mathbf{G}) = G \setminus \text{sinks}(\mathbf{G})$. A *root* of \mathbf{G} is a vertex $x \in G$ such that every other vertex is reachable by a path from x .

A subset of vertices $S \subseteq G$ is *strongly connected* if $(*)$: $\forall x, y \in S : x \in [y] \wedge y \in [x]$. Such an S is a strongly connected *component* if there is no set $S' \supset S$ such that $(*)$ holds. A strongly connected component S is *final* whenever $N^+(S) = S$. Since this will be of relevance for some algorithms, we remind the reader that it is possible, for instance by using Tarjan's algorithm [32], to decompose a graph into its strongly connected components in linear time.

For a digraph \mathbf{G} , $\underline{\mathbf{G}}$ denotes the undirected graph obtained by turning every directed edge $\langle x, y \rangle$ into an undirected one $\{x, y\}$. An *oriented* graph is a digraph \mathbf{G} obtained from $\underline{\mathbf{G}}$ by giving every undirected edge some direction. Such a graph does not contain any cycles of length 2.

A *kernel* of a digraph $\mathbf{G} = \langle G, N \rangle$ is a subset of vertices $K \subseteq G$ such that:

- (i) $G \setminus K \supseteq N^-(K)$ (K is an independent set in \mathbf{G}) and
- (ii) $G \setminus K \subseteq N^-(K)$ (from every non-kernel vertex there is at least one edge to a kernel vertex).

Any kernel of \mathbf{G} is an independent and dominating set in $\underline{\mathbf{G}}$. These two properties are equivalent to K being a maximal independent subset of $\underline{\mathbf{G}}$. Conversely, given a maximal independent subset K , we can determine if it is a kernel of \mathbf{G} by verifying that every vertex $x \in G \setminus K$ has a directed edge into K (a $\underline{\mathbf{G}}$ -edge in \mathbf{G} might be only to x).

Consequently, a possible (if not most efficient) algorithm for finding the kernels would unorient the input digraph \mathbf{G} , find $\underline{\mathbf{G}}$'s maximal independent subsets, and for each such check if every node outside it has a directed edge to the subset. The number of maximal independent subsets of any \mathbf{G} is limited by Moon and Moser's $3^{\frac{|G|}{3}}$ bound, [25], and such subsets can be produced with polynomial delay, [22]. It follows that there is an algorithm that finds all kernels in a graph, by just checking each such subset, in time $\mathcal{O}^*(3^{\frac{|G|}{3}})$. This running time is in fact tight for the problem of finding all kernels, as can be seen considering \mathbf{G} that is a collection of disjoint symmetric cycles of length 3, i.e. the reversal of every edge is also present. For such a graph every maximal independent subset of $\underline{\mathbf{G}}$ is a kernel and there are $3^{\frac{|G|}{3}}$ of them. Even though for most digraphs only a proper subset of the maximal independent sets will be kernels, finding all kernels is not a computationally feasible problem. We consider only the problem of determining the existence of a kernel which, when one exists, amounts usually to producing it.

The problem is addressed using the equivalence between the existence of kernels and the satisfiability of propositional theories, arising from an equivalent definition of kernels. For a digraph $\mathbf{G} = \langle G, N \rangle$, an assignment $\alpha \in \{\mathbf{0}, \mathbf{1}\}^G$ (of truth-values to the vertices of \mathbf{G}) is *correct* at a vertex $x \in G$ if $\alpha(x) = \mathbf{1} \Leftrightarrow \alpha(N^+(x)) \subseteq \{\mathbf{0}\}$ or equivalently, if:

$$(\alpha(x) = \mathbf{1} \wedge \alpha(N^+(x)) \subseteq \{\mathbf{0}\}) \vee (\alpha(x) = \mathbf{0} \wedge \mathbf{1} \in \alpha(N^+(x))) \quad (2.1)$$

An $\alpha \in \{\mathbf{0}, \mathbf{1}\}^G$ is a *solution* for \mathbf{G} , $\alpha \in \text{sol}(\mathbf{G})$, if α is correct at every vertex of \mathbf{G} , and if such an α exists \mathbf{G} is *solvable*. For any $\alpha \subseteq G \times \{\mathbf{0}, \mathbf{1}\}$, we denote $\alpha^{\mathbf{1}} = \{x \in G \mid \langle x, \mathbf{1} \rangle \in \alpha\}$ and $\alpha^{\mathbf{0}} = \{x \in G \mid \langle x, \mathbf{0} \rangle \in \alpha\}$. For all graphs \mathbf{G} and all assignments $\alpha \in \{\mathbf{0}, \mathbf{1}\}^G$ it holds that α is a solution iff $\alpha^{\mathbf{1}}$ is a kernel:

$$\alpha \in \text{sol}(\mathbf{G}) \iff \alpha^{\mathbf{1}} = G \setminus N^-(\alpha^{\mathbf{1}}) \iff \alpha^{\mathbf{1}} \text{ is a kernel of } \mathbf{G} \quad (2.2)$$

A possible algorithm for finding kernels is then based on the fact that every digraph \mathbf{G} induces a propositional theory $\mathcal{T}(\mathbf{G})$ by taking, for each $x \in G$, the formula

$$x \leftrightarrow \bigwedge_{y \in N^+(x)} \neg y, \quad (2.3)$$

with the convention that $\mathbf{1} = \bigwedge_{y \in \emptyset} y$.³ Then, letting $\text{mod}(\mathbf{T})$ denote all models of a theory \mathbf{T} , the following equality holds:

$$\text{sol}(\mathbf{G}) = \text{mod}(\mathcal{T}(\mathbf{G})). \quad (2.4)$$

Since determining kernels is a special case of determining the models of propositional theories, we can feed the equations (2.3) together with $z = \mathbf{1}$ for all $z \in \text{sinks}(\mathbf{G})$ to a solver of systems of boolean equations, to determine if \mathbf{G} has a kernel. Alternatively, we can feed the problem to a clausal SAT-solver. First, each equation (2.3) is equivalent to

$$\mathbf{1} = \left(x \vee \bigvee_{y \in N^+(x)} y \right) \wedge \bigwedge_{y \in N^+(x)} (\neg y \vee \neg x). \quad (2.5)$$

Collecting now the right-hand-sides of these new equations and adding the requirement for all $z \in \text{sinks}(\mathbf{G})$, yields the formula in CNF:

$$\text{CNF}(\mathbf{G}) = \bigwedge_{x \in \text{int}(\mathbf{G})} \left(\left(x \vee \bigvee_{y \in N^+(x)} y \right) \wedge \bigwedge_{y \in N^+(x)} (\neg y \vee \neg x) \right) \wedge \bigwedge_{z \in \text{sinks}(\mathbf{G})} z. \quad (2.6)$$

Satisfiability of $\text{CNF}(\mathbf{G})$ is equivalent to the solvability of the system of equations (2.5) for all internal nodes, with all sinks assigned $\mathbf{1}$ which, in turn, is equivalent to the existence of a kernel in \mathbf{G} , by (2.4).⁴

The above reduction and the resulting $\text{CNF}(\mathbf{G})$ are essentially the same as in [7]. The linear reduction in the opposite direction used there 3-Colorability, so we give a direct reduction from SAT: every propositional theory \mathbf{T} can be transformed in linear time into a digraph $\mathcal{G}(\mathbf{T})$ such that $\text{mod}(\mathbf{T}) = \text{sol}(\mathcal{G}(\mathbf{T}))$. Many different graphs can satisfy these requirements, so we give only one example. First, assume a theory \mathbf{T} given as a set of equivalences of the form

$$x \leftrightarrow \bigwedge_{i \in I_x} \neg y_i, \quad (2.7)$$

where all y, x_i are variables, and where every variable occurs at most once on the left of \leftrightarrow . The digraph $\mathcal{G}(\mathbf{T})$ is obtained by taking variables as vertices and, for every formula, introducing edges $x \rightarrow y_i$ for all $i \in I_x$. In addition, for every variable z *not* occurring on the left of any \leftrightarrow , we add a new vertex \bar{z} and two edges $z \rightarrow \bar{z}$ and $\bar{z} \rightarrow z$. This last addition ensures that each variable z of \mathbf{T} which would become a sink of $\mathcal{G}(\mathbf{T})$, and hence could only be assigned $\mathbf{1}$ by any solution of $\mathcal{G}(\mathbf{T})$, can be also assigned $\mathbf{0}$ (when the respective \bar{z} is assigned $\mathbf{1}$). Letting $V(\mathbf{T})$ denote all variables of \mathbf{T} , and $\text{sol}(\mathbf{X})|_Y$ the restriction of assignments in $\text{sol}(\mathbf{X})$ to the variables in Y , we have that

$$\text{mod}(\mathbf{T}) = \text{sol}(\mathcal{G}(\mathbf{T}))|_{V(\mathbf{T})} \quad (2.8)$$

Now, an arbitrary theory \mathbf{T} can be transformed into the above form. To simplify the transformation, assume \mathbf{T} to be given as a set of clauses, each clause $C = \langle C^+, C^- \rangle$ consisting of the set of positive, $C^+ = \{x_p \mid p \in P\}$, and negative, $C^- = \{\neg x_n \mid n \in N\}$, literals. First, let a_C be a new variable. The formula $C' : a_C \leftrightarrow \neg a_C \wedge \neg C$ is equisatisfiable with C , with models related by the equation $\text{mod}(C') = \text{mod}(C) \times \{\langle a_C, \mathbf{0} \rangle\}$. Substituting for $\neg C$, we obtain a more explicit form of $C' : a_C \leftrightarrow \neg a_C \wedge \bigwedge_{p \in P} \neg x_p \wedge \bigwedge_{n \in N} x_n$. We introduce for every variable in the initial theory $x \in V(\mathbf{T})$, a new variable \bar{x} . For every such pair of variables we introduce the formulae (i), and for every clause C the formula (ii):

$$(i) \quad x \leftrightarrow \neg \bar{x} \text{ and } \bar{x} \leftrightarrow \neg x.$$

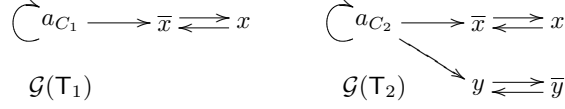
³Satisfiability of such a theory is equivalent to the existence of solutions for the corresponding system of boolean equations. This motivates the use of the name “solution”, which was also used in the early days of kernel theory, e.g., in [33], p.588, or [30].

⁴Assuming the adjacency list representation of the argument $\mathbf{G} = \langle G, N \rangle$, CNF is linear in the number of vertices, $|G|$, and edges, $|N|$ (each edge $\langle x, y \rangle$ giving rise to two pieces of data: $\neg y \vee \neg x$ and the element y in the disjunction for $x : x \vee \dots \vee y \vee \dots$).

$$(ii) \ a_C \leftrightarrow \neg a_C \wedge \bigwedge_{p \in P} \neg x_p \wedge \bigwedge_{n \in N} \neg \bar{x}_n$$

The theory C'' containing formulae (i) and (ii) is equisatisfiable with C and $mod(C) = mod(C'')|_{V(C)}$. Defining $T' = \bigcup_{C \in T} C''$ and letting $\mathcal{G}(T) = \mathcal{G}(T')$, the equality (2.8) remains valid.

Example 2.9 For $T_1 = \{\neg x\}$, respectively, $T_2 = \{\neg x \vee y\}$, we obtain the digraphs:



We note that $\mathcal{G}(T)$ can be defined so that it is oriented and has no loops. In addition to a_C , add two more nodes in a 3-cycle $\langle a_C, b_C, c_C, a_C \rangle$, and for every $x \in V(T)$, introduce in (i) two more new nodes, replacing the 2-cycle by the 4-cycle: $\langle x, \bar{x}, x', x'', x \rangle$.

Both equations (2.4) and (2.8) hold for arbitrary digraphs but when they have infinite branchings, the corresponding theory is in infinitary propositional logic. In this paper, we are concerned exclusively with usual propositional logic and finite graphs, so “graph” and “arbitrary graph” mean here only a finite digraph.

3 Preprocessing

This section presents some simplifications reducing the input graph, which will be later combined with different algorithms. In Subsection 3.1, we show that we can consider only the problem for graphs without sinks, since kernels of an arbitrary graph G are determined by the kernels of its appropriate, sinkless subgraph which can be obtained from G in linear time. Subsection 3.2 presents some further simplifications of a graph which are based on local dependencies and are of linear, or low polynomial, complexity.

3.1 Forcing values

The obvious brute-force approach, simply checking the condition (2.1) for every possible assignment, can be improved by observing consequences of a given partial assignment. The following definition captures some such consequences that are recognizable locally in the graph.

Definition 3.1 A partial assignment to a graph G is an $\alpha \in \{0, 1\}^X$ for any $X \subseteq G$. Given such an α , we define inductively its extension to the nodes which obtain forced values:

$$\begin{aligned}
 \alpha_1^1 &= \alpha^1 \\
 \alpha_1^0 &= \alpha^0 \\
 \alpha_{i>1}^0 &= N^+(\alpha_{i-1}^1) \cup N^-(\alpha_{i-1}^1) \cup \alpha_{i-1}^0 \\
 \alpha_{i>1}^1 &= \text{sinks}(G \setminus \alpha_i^0) \cup \alpha_{i-1}^1 \cup \{x \in N^+(y) \mid y \in \alpha_i^0 \wedge \{x\} = N^+(y) \setminus \alpha_i^0\}
 \end{aligned}$$

Fixed-point is reached when $\alpha_k^1 = \alpha_{k-1}^1$, no later than for $k = |G|$. We then let $\bar{\alpha}^1 = \bigcup \alpha_i^1$, $\bar{\alpha}^0 = \bigcup \alpha_i^0$ and set $\bar{\alpha} = \{(n, 1) \mid n \in \bar{\alpha}^1\} \cup \{(n, 0) \mid n \in \bar{\alpha}^0\}$.

Example 3.2 Consider the following two graphs:



In C , $\alpha = \{\langle x, 1 \rangle\}$ gives $\alpha_2^0 = \{z, y\}$, $\alpha_2^1 = \{x, z\}$ and then $\alpha_3^0 = \{x, y, z\}$, $\alpha_3^1 = \{x, y, z\}$, i.e., $\bar{\alpha} = \{x, y, z\} \times \{0, 1\}$.

In \mathbf{G} , from $\alpha = \{\langle c, \mathbf{0} \rangle\}$ we obtain $\bar{\alpha} = \{\langle c, \mathbf{0} \rangle, \langle b, \mathbf{1} \rangle, \langle a, \mathbf{0} \rangle, \langle f, \mathbf{0} \rangle, \langle e, \mathbf{1} \rangle, \langle d, \mathbf{0} \rangle\}$, while $\beta = \{\langle c, \mathbf{1} \rangle\}$ leads to $\bar{\beta} = \{\langle c, \mathbf{1} \rangle, \langle d, \mathbf{0} \rangle, \langle b, \mathbf{0} \rangle, \langle a, \mathbf{1} \rangle, \langle f, \mathbf{1} \rangle, \langle e, \mathbf{0} \rangle\}$. In both cases, the resulting assignment is a solution of \mathbf{G} . Starting with $\gamma = \{\langle e, \mathbf{0} \rangle\}$ does not induce any values, i.e., $\bar{\gamma} = \gamma$.

C shows that $\bar{\alpha}$ may happen to be a (non-functional) relation, i.e., $\bar{\alpha}^1 \cap \bar{\alpha}^0 \neq \emptyset$. If this is the case, then we cannot find a correct assignment that extends α . However, if $\bar{\alpha}$ is a function, then for all $x \in \text{dom}(\bar{\alpha})$ we have the following weaker form of correctness:

$$(\bar{\alpha}(x) = \mathbf{1} \wedge \bar{\alpha}(N^+(x)) \subseteq \{\mathbf{0}\}) \vee (\bar{\alpha}(x) = \mathbf{0} \wedge \exists y \in N^+(x) : y \notin \bar{\alpha}^0) \quad (3.3)$$

We say that $\bar{\alpha}$ is *consistent* in this case. Given a consistent partial assignment $\bar{\alpha}$, it might, but need not, be possible to extend it to a solution for \mathbf{G} . This depends on the solvability of the subgraph yet to be assigned, but also on the possibility of finding a solution for the remaining graph such that each vertex assigned $\mathbf{0}$ by $\bar{\alpha}$ is eventually justified by the assignment of $\mathbf{1}$ to one of its out-neighbours. In particular, we have to meet this constraint on the *border* of α , defined as follows:

Definition 3.4 Given a partial assignment α to a graph \mathbf{G} , the *border* of α is the set $\text{bord}(\alpha) = \{x \in \text{dom}(\alpha) \mid \alpha(x) = \mathbf{0} \wedge \mathbf{1} \notin \alpha(N^+(x))\}$.

The formula (3.3) implies that a consistent partial assignment is correct everywhere with the possible exception of its border.

Remark 3.5 When a partial assignment α is correct on its whole domain, i.e., $\alpha \in \text{sol}(\text{dom}(\alpha))$, then $\alpha^1 \subseteq G$ is called a *local kernel* (sometimes *semi-kernel*) in kernel theory. Local kernels are used in inductive proofs of sufficient conditions for the existence of kernels in digraphs from certain classes, e.g. in [2, 15, 14, 18]. Deciding if a graph has a local kernel is NP-complete, [13].

Any $\beta \in \text{sol}(\mathbf{G})$ must be such that its restriction to any subset $B \subseteq G$ is consistent on the subgraph induced by B . Also, every solution respects all values induced by its own restrictions, in particular, induced from the empty assignment. Consequently, the values induced from the empty assignment are the same in all solutions (if any). These observations are gathered in the following lemma. \mathbf{G}_α° denotes the subgraph $\mathbf{G} \setminus \text{dom}(\bar{\alpha})$, \emptyset denotes the empty assignment, and we abbreviate $\mathbf{G}^\circ = \mathbf{G}_\emptyset^\circ$.

Lemma 3.6 For an arbitrary \mathbf{G} :

1. $\text{bord}(\emptyset) = \emptyset$;
2. for any partial assignment α : $\text{sinks}(\mathbf{G}_\alpha^\circ) = \emptyset$;
3. $\forall \beta \in \text{sol}(\mathbf{G}) \forall B \subseteq G : \overline{\beta|_B} = \beta|_{\text{dom}(\overline{\beta|_B})}$;
4. $\text{sol}(\mathbf{G}) = \{\beta \cup \emptyset \mid \beta \in \text{sol}(\mathbf{G}^\circ)\}$.

PROOF. 1. It follows by induction that each \emptyset_i satisfies (2.1), i.e., $N^+(\emptyset_i^1) \subseteq \emptyset_i^0 \wedge \forall x \in \emptyset_i^0 : N^+(x) \cap \emptyset_i^1 \neq \emptyset$. This holds trivially at the start with $\emptyset_1 = \emptyset$, and after first iteration when $\emptyset_2^1 = \text{sinks}(\mathbf{G})$ and $\emptyset_2^0 = \emptyset$. Assuming (2.1) as IH for \emptyset_i , then

for each new $x \in \emptyset_{i+1}^0 : x \in N^-(\emptyset_i^1)$, because $N^+(\emptyset_i^1) \subseteq \emptyset_i^0$ by IH

for each new $x \in \emptyset_{i+1}^1 : x \in \text{sinks}(\mathbf{G} \setminus \emptyset_{i+1}^0)$ – the last component of Definition 3.1 does not apply, since for any $y \in \emptyset_i^0$ there is a $z \in N^+(y) \cap \emptyset_i^1$ by IH.

2. $\bar{\alpha} = \alpha_i = \alpha_{i+1}$ for some $i \geq 0$ and assume $x \in \text{sinks}(\mathbf{G}_\alpha^\circ)$, i.e., $N^+(x) \subseteq \text{dom}(\bar{\alpha})$. If $N^+(x) \cap \bar{\alpha}^1 \neq \emptyset$, then $x \in \alpha_{i+1}^0$ and otherwise $x \in \alpha_{i+1}^1$. In either case $x \in \text{dom}(\alpha_{i+1}) = \text{dom}(\bar{\alpha})$. Contradiction.

3. By induction on the steps used in the construction of $\overline{\beta|_B}$, Definition 3.1, we show that for all $i : (\beta|_B)_i = \beta|_{\text{dom}((\beta|_B)_i)}$. The basis is trivial since $(\beta|_B)_1 = (\beta|_B) = \beta|_{\text{dom}((\beta|_B)_1)}$. For the induction step, any $x \in (\beta|_B)_{i+1}$ gives one of the following cases:

(0) $x \in (\beta|_B)_{i+1}^0$ i.e., either

- $x \in (\beta|_B)_i^0$ which, by IH, means that $x \in \beta^0$ or
- $x \in N^+((\beta|_B)_i^1) \cup N^-((\beta|_B)_i^1)$ which, by IH and correctness of β , means that $x \in N^+(\beta^1) \cup N^-(\beta^1) \subseteq \beta^0$, or

(1) $x \in (\beta|_B)_{i+1}^1$ i.e., either

- $x \in (\beta|_B)_i^1$ which, by IH, means that $x \in \beta^1$ or
- $x \in \text{sinks}(\mathbf{G} \setminus (\beta|_B)_{i+1}^0)$, i.e., $N^+(x) \subseteq (\beta|_B)_{i+1}^0 \subseteq \beta^0$ by point (0), and $x \in \beta^1$ by correctness of β , or
- $\{x\} = N^+(y) \setminus (\beta|_B)_{i+1}^0 : y \in (\beta|_B)_{i+1}^0$, i.e. by point (0) we have $y \in \beta^0$ and $N^+(y) \setminus \{x\} \subseteq \beta^0$. Then by correctness of β we must have $\{x\} = N^+(y) \setminus \beta^0$ with $x \in \beta^1$.

4. For every $x \in G^\circ : N^+(x) \cap \overline{\beta}^1 = \emptyset$ and, by 2, $N^+(x) \cap G^\circ \neq \emptyset$. Hence, every $\beta \in \text{sol}(\mathbf{G}^\circ)$ can be combined with $\overline{\beta}$ into a correct solution for \mathbf{G} . But the values on $\text{dom}(\overline{\beta})$ can not be chosen otherwise since, by 3, $\forall \alpha : \alpha \in \text{sol}(\mathbf{G}) \rightarrow \alpha|_{\text{dom}(\overline{\beta})} = \overline{\beta}$. \square

The construction from Definition 3.1, together with Lemma 3.6, will provide the basic simplification mechanism used in all our algorithms. According to point 4, we can first (in linear time) induce all values from the sinks of \mathbf{G} , removing $\text{dom}(\overline{\beta})$ from the graph. Then, trying various partial assignments σ to the remaining, sinkless subgraph \mathbf{G}° , point 3 ensures that it suffices to consider only the induced assignment $\overline{\sigma}$, thus reducing the search space.

In the following subsection, we identify some particular, structural patterns allowing local simplifications of the graph.

3.2 Simplification

The number of possible simplifications, preserving and reflecting solvability, can be unlimited. In practice, one has to choose some which can be expected to occur frequently and can be performed cheaply. Two such simplifications are given, providing also some information about the structural properties of kernels. The first one concerns a special type of path.

Definition 3.7 A path $p = \langle x_0, x_1, \dots, x_{l(p)} \rangle$ is isolated if $\forall 0 \leq i < l(p) : N^+(x_i) = \{x_{i+1}\}$.

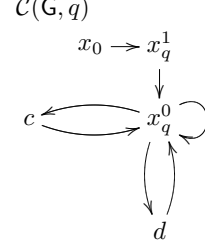
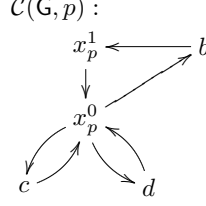
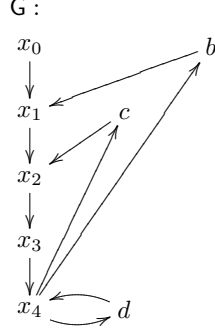
It follows from Definition 3.1 that any assignment, of **0** or **1**, to any vertex on an isolated p will induce values to every other vertex on the path. So the vertices on isolated paths do not contribute anything to the structural properties of \mathbf{G} determining its kernels. They can be removed.

Definition 3.8 For an isolated path $p = \langle x_{0,p}, \dots, x_{l(p),p} \rangle$ with $l(p) \geq 2$, let $P \subseteq G$ denote all nodes $x_{i,p}$ on p . The graph $\mathcal{C}(\mathbf{G}, p)$, the contraction of \mathbf{G} on p , is defined by a mapping $f : G \rightarrow \mathcal{C}(\mathbf{G}, p)$:

- $\mathcal{C}(G, p) = G \setminus \{x_{i,p} \mid x_{i,p} \in P\} \cup \{x_p^0, x_p^1\}$
- $f : G \rightarrow \mathcal{C}(G, p)$ is defined by $f(x) = x$ when $x \in \mathcal{C}(G, p)$, $f(x_{i,p}) = x_p^0$ when $i + l(p)$ is even and $f(x_{i,p}) = x_p^1$ otherwise
- $\mathcal{C}(N, p) = \{\langle x, y \rangle \mid \exists \langle x', y' \rangle \in (f^-(x) \times f^-(y)) \cap E : x' = x \vee x' = x_{l(p),p} \vee y' = x_{l(p),p}\}$

Example 3.9 We contract the isolated path $p = \langle x_0, x_1, x_2, x_3, x_4 \rangle$ in the digraph \mathbf{G} , obtaining the digraph $\mathcal{C}(\mathbf{G}, p)$ where f is defined on p by $f(x_0) = f(x_2) = f(x_4) = x_p^0$, $f(x_1) = f(x_3) = x_p^1$. Also shown is the digraph $\mathcal{C}(\mathbf{G}, q)$ obtained by contracting $q = \langle b, x_1, x_2, x_3, x_4 \rangle$ with $f(b) = f(x_2) =$

$$f(x_4) = x_q^0, f(x_1) = f(x_3) = x_q^1.$$



Contraction of isolated paths preserves and reflects solutions as stated in the following Fact. ($f; g$ denotes function composition in diagrammatic order: f followed by g .)

Fact 3.10 *For any isolated path p with $l(p) \geq 2$ in any \mathbf{G} :*
 $sol(\mathbf{G}) = \{\alpha \mid \exists \beta \in sol(\mathcal{C}(\mathbf{G}, p)) : \alpha = f; \beta\}.$

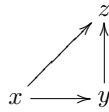
PROOF. \supseteq For a $\beta \in sol(\mathcal{C}(\mathbf{G}, p))$ define $\alpha \in \{0, 1\}^G$ by $\forall x \in G : \alpha(x) = \beta(f(x))$. To show $\alpha \in sol(\mathbf{G})$ it suffices, by definition of f , to show that α is correct on p . For $x_{i,p}$ such that $i + l(p)$ is odd or $i = l(p)$, correctness follows since by definition of f and the fact that p is isolated we have $f(N^+(x_{i,p})) = N_{\mathcal{C}(\mathbf{G}, p)}^+(f(x_{i,p}))$. All other $x_{i,p}$'s are such that $i + l(p)$ is even, and since p is isolated we have $f(N^+(N^+(x_{i,p}))) = f(x_{i,p}) = f(x_{l(p),p})$. So correctness follows from correctness of $\alpha(x_{l(p),p})$

\subseteq Assume $\alpha \in sol(\mathbf{G})$. By definition of f and the fact that p is an isolated path it follows that $\forall x : \forall y_1, y_2 \in f^-(x) : \alpha(y_1) = \alpha(y_2)$. Then we define β for every $x \in \mathcal{C}(\mathbf{G}, p)$ by choosing arbitrary $y \in f^-(x)$ and taking $\beta(x) = \alpha(y)$. Then β is correct and it satisfies $\forall x \in G : \alpha(x) = \beta(f(x))$ \square

As the second simplification we remove basic contradictions.

Definition 3.11 *An $x \in G$ is a basic contradiction if $\exists y \in N^+(x) : N^+(y) \subseteq N^+(x)$.*

Important special cases include the in-neighbours of sinks, loops, and triangles such as the following graph:

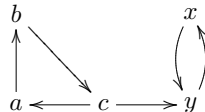


The following fact is obvious:

Fact 3.12 *If x is a basic contradiction in \mathbf{G} then $\forall \alpha \in sol(\mathbf{G}) : \alpha(x) = 0$.*

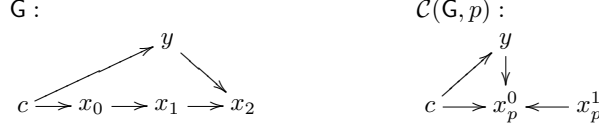
PROOF. Let $y \in N^+(x)$ be such that $N^+(y) \subseteq N^+(x)$ and $\alpha \in sol(\mathbf{G})$. If $\alpha(y) = 1$ then $\alpha(x) = 0$, while if $\alpha(y) = 0$ then, for some $z \in N^+(y) : \alpha(z) = 1$. But then also $\alpha(x) = 0$ since $z \in N^+(y) \subseteq N^+(x)$. \square

The notion of basic contradiction is motivated by the fact that a (general) contradiction, i.e. an x such that $\forall \alpha \in sol(\mathbf{G}) : \alpha(x) = 0$, may not be identifiable as such locally by inspecting its fixed neighbourhood. For instance, in the following graph, x is a contradiction since $x = 0$ is necessary (and sufficient) for the existence of a correct assignment to the rest of the graph.



The contraction of isolated paths can turn a contradiction into a basic one, as the following example illustrates.

Example 3.13 *The graph $\mathcal{C}(\mathbf{G}, p)$ results from contracting the isolated path $p = \langle x_0, x_1, x_2 \rangle$ in \mathbf{G} . After contraction, c becomes a basic contradiction, revealing that it is a contradiction in \mathbf{G} :*



The specific case of Fact 3.10, covered by the following fact, characterizes the contradictions which become basic after contraction of isolated paths. (Basic contradiction is a special case when $l(p) = 0$ and $l(q) = 1$.)

Fact 3.14 *Given isolated paths $p = \langle x_{0,p}, x_{1,p}, \dots, x_{l(p),p} \rangle$, $q = \langle x_{0,q}, x_{1,q}, \dots, x_{l(q),q} = x_{l(p),p} \rangle$ such that $l(p) + l(q)$ is odd. If $c \in G$ is such that $x_{0,p}, x_{0,q} \in N^+(c)$, then $\forall \alpha \in \text{sol}(\mathbf{G}) : \alpha(c) = \mathbf{0}$.*

PROOF. Assume arbitrarily that p has odd length and that we start by contracting p to obtain $\mathbf{H} = \mathcal{C}(\mathbf{G}, p)$. Then we have $x_p^1 \in N_{\mathbf{H}}^+(c)$, and there is an isolated path $q = \langle x_{0,q}, x_{1,q}, \dots, x_{l(q),q} = x_p^0 \rangle$ in \mathbf{H} . Contracting q to obtain $\mathbf{K} = \mathcal{C}(\mathbf{H}, q)$ we obtain a graph where $x_q^0 = x_p^0 \in N_{\mathbf{K}}^+(c)$ and $N_{\mathbf{K}}^+(x_p^1) = \{x_q^0\}$. So by facts 3.10 and 3.12 it follows that $\forall \alpha \in \text{sol}(\mathbf{G}) : \alpha(c) = \mathbf{0}$. \square

Similar facts can be proven for other situations, where contracting some collection of paths reveals a basic contradiction (for instance in the case of isolated cycles of odd length, or with two paths p, q as in Fact 3.14 but admitting also outgoing edges at nodes with even indices x_{2i} .) We do not attempt to give a complete classification, however.

Towards an algorithm for KER, we gather the two rules for isolated paths and basic contradictions into the simplification procedure $\text{simp}(\mathbf{G})$ as shown in Algorithm 3.15. The algorithm returns the error value \perp if it discovers the non-existence of solutions. Otherwise, by Facts 3.10, 3.12 and Lemma 3.6, every solution to the input graph \mathbf{G} can be obtained from a solution to the returned graph.⁵

Algorithm 3.15 $\text{simp}(\mathbf{G})$

```

if there is an isolated path  $p$  with  $l(p) \geq 2$  then
  return  $\text{simp}(\mathcal{C}(\mathbf{G}, p))$ 
else if there is a basic contradiction  $x \in G$  then
   $\alpha := \{\langle x, \mathbf{0} \rangle\}$ 
  if  $\bar{\alpha}$  is a function then
    return  $\text{simp}(\mathbf{G} \setminus \text{dom}(\bar{\alpha}))$ 
  else
    return  $\perp$ 
else
  return  $\mathbf{G}$ 

```

4 Breaking cycles

According to Richardson's theorem [30], every finitely branching (in particular, finite) graph not containing odd cycles has a kernel. Consequently, a possible approach to KER is to try *breaking*

⁵Inducing and checking the existence of isolated paths can be done in linear time. The trivial search for basic contradictions would visit, for every node x , each of its out-neighbours $y \in N^+(x)$, checking if $N^+(y) \subseteq N^+(x)$. The worst case $|G|^2$ hardly ever obtains and, in practice, even this trivial procedure is sub-quadratic.

the odd cycles. Below, we reduce the number of cycles to consider and give a general treatment of this approach utilizing the following concept.

Definition 4.1 For a graph G , we define $\mathcal{B}(G) = \{X \subseteq G \mid \forall \beta \in \text{sol}(G) : \exists \alpha \in \{0, 1\}^X : \bar{\alpha} = \beta\}$. An $X \in \mathcal{B}(G)$ is called a basis for $\text{sol}(G)$.

Thus, for any $X \in \mathcal{B}(G)$, any solution for G can be obtained by inducing from some assignment to X , reducing the complexity of the brute-force approach to $2^{|X|}$. (As inducing from a partial assignment takes linear time, the notion of a basis is almost the same as the concept of *strong backdoor* from SAT.) It remains to be proven that suitable $X \in \mathcal{B}(G)$ exists. Below we provide two types of bases, guaranteed to exist for any graph. In algorithmic terms this means that KER, when parameterized by the size of either of these bases, is FPT. It should be noted here that a more obvious choice of parameter for KER, the size of the kernel we are looking for, does not make the problem FPT for general graphs unless collapses, deemed unlikely, occur among the parameterized complexity classes.⁶ So the result in Subsection 4.2, admitting as a basis any set of vertices touching all even cycles, appears to be the best currently available regarding the parameterized complexity of KER.

4.1 Feedback Vertex Sets

A *feedback vertex set* for a graph G is a subset $F \subseteq G$ such that $G \setminus F$ is acyclic (a dag).

Proposition 4.2 For any graph G , if F is a feedback vertex set for G then $F \in \mathcal{B}(G)$.

PROOF. Let F be a feedback vertex set for G and consider arbitrary $\beta \in \text{sol}(G)$. Then by lemma 3.6 we have $\beta|_F = \beta|_{\text{dom}(\beta|_F)}$. All we need to prove is $\text{dom}(\beta|_F) = G$. So consider $G \setminus \text{dom}(\beta|_F)$. By Lemma 3.6.2, this graph has no sinks, and as F is a feedback vertex set, it has no cycles. Since G is finite, it follows that $G \setminus \text{dom}(\beta|_F) = \emptyset$, as desired. \square

This observation gives a simple algorithm for KER: find some feedback vertex set F and try all possible assignments to its nodes, verifying if the induced assignments are correct on the whole graph. More cleverly, proposition 4.2 can be used to construct a branch and bound algorithm that only branches at vertices from F . An algorithm based on this idea is presented in [12]. We will return to branch and bound algorithms in Section 5, but note here that as the success of such an approach depends on finding *small* feedback vertex sets, we can not expect it to be optimal for all graphs. It will be good enough, though, for solving KER effectively on graphs that admit small feedback vertex sets. This follows from the recent work in [5], showing that the problem of finding a minimum feedback vertex set is FPT in the size of such a set. In particular, KER is FPT in the size of a *minimum* feedback vertex set.

Feedback vertex sets are useful tools when graphs are viewed algebraically as systems of boolean equations. In this context they allow for a systematic substitution of equals for equals that both preserves and reflects solutions, allowing us to represent G more compactly than the system $\mathcal{T}(G)$ from (2.3). In the rest of this subsection we present this construction, linking substitution in systems of boolean equations with feedback vertex sets of graphs. We do this by introducing labeled dag's that are nice in their own right in that they provide a visualization of the bases originating from feedback vertex sets.⁷

F denotes such a set and given it, we represent G as a (labeled) dag $D(F) = \langle D_F, N_F \rangle$, where $F' = \{x' \mid x \in F\}$ is a set of new elements and:

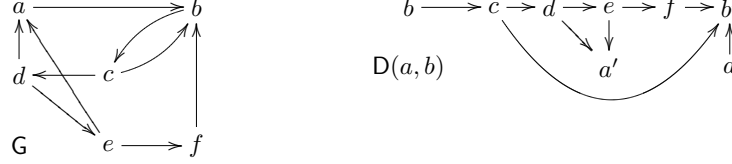
$$\begin{aligned} D_F &= G \cup F' \\ N_F &= \left(N_G \setminus \{\langle y, x \rangle \mid x \in F\} \right) \cup \{\langle y, x' \rangle \mid x' \in F' \wedge x \in N^+(y)\} \end{aligned} \quad (4.3)$$

⁶In [19] it is shown that using the size of the kernel as parameter does make the problem FPT for planar digraphs. ([27] provides an introduction to parameterized complexity.)

⁷The question whether this correspondence could be applied for solving more general systems of boolean equations seems an interesting research challenge in its own right

The new vertices are exactly the new sinks $F' = \text{sinks}(\mathbf{D}(F)) \setminus \text{sinks}(\mathbf{G})$ and $|F'| \leq$ number of cycles in \mathbf{G} . The labeling, defined by $l(x) = x$ for $x \in G$ and $l(x') = x$ for the new $x' \in F'$, serves to establish a unique correspondence between solutions of $\mathbf{D}(F)$ and of \mathbf{G} .

Example 4.4 $\mathbf{D}(a, b) \in \text{dag}(\mathbf{G})$ is obtained from the feedback vertex set $\{a, b\}$.



The double 'a' would disappear if we constructed the dag $\mathbf{D}(b)$ from the feedback set $\{b\}$. It would have an edge from a' (which became a) to b' and no extra a without incoming edges.

Let $\text{dag}(\mathbf{G})$ denote the set of so obtained dags from a given \mathbf{G} . Given a $\mathbf{D}(F) \in \text{dag}(\mathbf{G})$, we can use inductive definitions over this representation. In particular, any assignment to the new sinks, $\beta \in \{0, 1\}^{F'}$, induces an assignment $\bar{\beta}$ to the whole \mathbf{D} , in linear time. We only verify that the values assigned to the new sinks $x' \in F'$ are the same as the values induced at the respective $x \in F$. In the above $\mathbf{D}(a, b)$, trying $a' = 1 = b'$ fails inducing $a = 0$. Trying $b' = 1$ and $a' = 0$ induces the same values at b and a , allowing to conclude the existence of a kernel for \mathbf{G} .

$\text{sol}(\mathbf{G})$ becomes thus captured by a new system of equations requiring the values assigned to F' to agree with the values induced in F . The system is defined as follows. For every vertex $x \in G \setminus \text{sinks}(\mathbf{G}) = \text{int}(\mathbf{D}(F))$, divide the set of its out-neighbours $N_F^+(x)$ into two disjoint subsets: $N_L^+(x) = N_F^+(x) \cap F'$ and $N_R^+(x) = N_F^+(x) \setminus N_L^+(x)$.

Definition 4.5 For $x \in \text{sinks}(\mathbf{G})$ let $\text{FRM}_{\mathbf{D}(F)}(x) = 1$ and for $x \in \text{int}(\mathbf{D}(F))$ define:

$$\text{FRM}_{\mathbf{D}(F)}(x) = \bigwedge_{y \in l(N_L^+(x))} \neg y \wedge \bigwedge_{z \in N_R^+(x)} \neg \text{FRM}_{\mathbf{D}(F)}(z).$$

The reduced system is $\text{EQU}_{\mathbf{D}(F)}(\mathbf{G}) = \{\text{FRM}_{\mathbf{D}(F)}(x) = x \mid x \in F\}$.

Example 4.6 (4.4 continued) The reduced system $\text{EQU}_{\mathbf{D}(a, b)}(\mathbf{G})$ has two equations: $a = \neg b$ and $b = \neg(\neg b \wedge \neg(\neg a \wedge \neg(\neg a \wedge \neg b)))$.

The dag $\mathbf{D}(b) \in \text{dag}(\mathbf{G})$ would give the corresponding reduced system with only one equation (equivalent to the one obtained by substituting $a = \neg b$ in the above system), namely: $b = \neg(\neg b \wedge \neg(\neg \neg b \wedge \neg(\neg \neg b \wedge \neg \neg b)))$. Simplifying its right-hand side, we gradually obtain the trivial equation $b = \neg(\neg b \wedge \neg(\neg \neg b \wedge \neg \neg \neg b)) = \neg(\neg b \wedge \neg 0) = b$.

Each $\text{FRM}_{\mathbf{D}(F)}(x)$ contains only variables from F , so an $\alpha \in \{0, 1\}^F$ can be extended to $\alpha^* \in \{0, 1\}^G$ as follows ($\alpha[\phi]$ denotes the usual evaluation of the formula ϕ under the assignment α):

$$\alpha^*(x) = \begin{cases} \alpha(x) & \text{if } x \in F \\ \alpha[\text{FRM}_{\mathbf{D}(F)}(x)] & \text{otherwise} \end{cases} \quad (4.7)$$

This makes α^* a function consistent with $\bar{\alpha}$ induced according to Definition 3.1, i.e., $\alpha^* \subseteq \bar{\alpha}$. Every solution for \mathbf{G} is, in fact, such an α^* obtained from a solution for $\text{EQU}_{\mathbf{D}(F)}(\mathbf{G})$.

Proposition 4.8 For any $\mathbf{D}(F) \in \text{dag}(\mathbf{G})$:

$$\text{sol}(\mathbf{G}) = \{\alpha^* \mid \alpha \in \{0, 1\}^F \wedge \forall x \in F : \alpha(x) = \alpha[\text{FRM}_{\mathbf{D}(F)}(x)]\}.$$

PROOF. \supseteq) If the equality holds for F , then (4.7) makes it hold also for all other nodes. Then, for every $x \in G$, we have that (*) $\alpha^*(x) = 1 \Leftrightarrow \alpha[\text{FRM}_{\mathbf{D}(F)}(x)] = 1$, and hence

$$\begin{aligned}
\alpha^*(x) = \mathbf{1} &\Leftrightarrow \left(\bigwedge_{y \in l(N_L^+(x))} \neg \alpha^*(y) \wedge \bigwedge_{z \in N_R^+(x)} \neg \alpha[FRM_{D(F)}(z)] \right) = \mathbf{1} \quad (*) \\
&\Leftrightarrow \left(\bigwedge_{y \in l(N_L^+(x))} \neg \alpha^*(y) \wedge \bigwedge_{z \in N_R^+(x)} \neg \alpha^*(z) \right) = \mathbf{1} \quad (*) \\
&\Leftrightarrow \bigwedge_{y \in N^+(x)} \neg \alpha^*(y) = \mathbf{1} \quad N^+(x) = l(N_L^+(x)) \cup N_R^+(x) \\
&\Leftrightarrow \forall y : y \in N^+(x) \rightarrow \alpha^*(y) = \mathbf{0}
\end{aligned}$$

\subseteq) For an arbitrary $\beta \in \text{sol}(\mathbf{G})$, let $\alpha = \beta|_F$. Since $F \in \mathcal{B}(\mathbf{G})$, so $\bar{\alpha} = \beta$. But since $\bar{\alpha}$ and α^* both are functions and $\alpha^* \subseteq \bar{\alpha}$, so $\alpha^* = \bar{\alpha} = \beta$. \square

In Example 4.6, the reduced system simplified to one trivial equation $b = b$, so the graph \mathbf{G} has exactly two solutions, each induced from a solution to this equation.

Expressing this proposition in terms of the assignment $\bar{\alpha}$, induced in the dag $D(F) \in \text{dag}(\mathbf{G})$ from the assignment $\alpha \in \{\mathbf{0}, \mathbf{1}\}^{F'}$ to its new sinks F' , gives the following claim:

$$\text{sol}(\mathbf{G}) = \{\bar{\alpha}|_G \mid \alpha \in \{\mathbf{0}, \mathbf{1}\}^{F'} \wedge \forall x' \in F' : \alpha(x') = \bar{\alpha}(x)\}.$$

The above algorithms, whether utilizing the reduced system of equations $\text{EQU}_{D(F)}(\mathbf{G})$ or merely inducing values directly in $D(F)$, rely on finding an arbitrary feedback vertex set. The following subsection presents an algorithm for which it suffices to find a subset of nodes breaking only the even cycles.

4.2 Breaking Even Cycles

Dually to Richardson's theorem, we have the following fact.

Lemma 4.9 *If $\mathbf{G} \neq \emptyset$, $\text{sinks}(\mathbf{G}) = \emptyset$, and \mathbf{G} has no even cycles, then $\text{sol}(\mathbf{G}) = \emptyset$.*

PROOF. Assume towards contradiction that $\alpha \in \text{sol}(\mathbf{G})$. Clearly, $\alpha^1 \neq \emptyset$. So choose $a \in \alpha^1$ and consider a sequence of sets $V_i : \mathbb{N} \rightarrow \mathcal{P}([a])$ such that

$$\begin{aligned}
V_0 &= \{a\} \\
V_{2i+1} &= \bigcup_{x \in V_{2i}} N^+(x) \\
V_{2i+2} &= \bigcup_{x \in V_{2i+1}} \{y_x\}, \text{ where } y_x \in N^+(x) \text{ is such that } \alpha(y_x) = \mathbf{1} \text{ (if it exists).}
\end{aligned}$$

By correctness of α , such a sequence satisfies $\bigcup_i V_{2i} \subseteq \alpha^1$ and $\bigcup_i V_{2i+1} \subseteq \alpha^0$ and, as $\text{sinks}(\mathbf{G}) = \emptyset$ so $\forall i \in \mathbb{N} : V_i \neq \emptyset$. Also, it is easy to see that for every $n \in \mathbb{N}$ and every $a_n \in V_n$ there is a sequence of edges $\langle a, a_1, a_2, \dots, a_n \rangle$ such that $\forall i : a_i \in V_i \cap N^+(a_{i-1})$. So there is an infinite sequence of edges $p = \langle a, a_1, a_2, \dots \rangle$ such that $\forall i : a_i \in V_i \cap N^+(a_{i-1})$. Since \mathbf{G} is finite this is only possible if $\exists j > i : a_i = a_j$. Let i, j be a pair satisfying this condition and such that for all $i \leq k < l < j : a_k \neq a_l$. The sequence of edges $C = \langle a_i, a_{i+1}, \dots, a_j \rangle$ must be of even length since otherwise $a_i \in \alpha^1 \cap \alpha^0$. We have found an even cycle, contradicting our assumption about \mathbf{G} . \square

From an algorithmic point of view, the observation that odd cycles are the only obstacle to the existence of kernels suggests (not always efficient) algorithms based on breaking the odd cycles. The above observation suggests that we can restrict attention to even cycles, and the following proposition makes this suggestion precise. A subset of vertices $X \subseteq G$ is an even cycle transversal, if $G \setminus X$ contains no even cycles.

Proposition 4.10 *If $X \subseteq G$ is an even cycle transversal, then $X \in \mathcal{B}(\mathbf{G})$.*

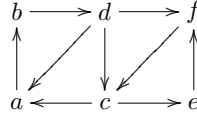
PROOF. For an arbitrary $\beta \in \text{sol}(\mathbf{G})$, Lemma 3.6 gives that $\overline{\beta|_X} = \beta|_{\text{dom}(\overline{\beta|_X})}$. Assume towards contradiction that $\mathbf{G}' = \mathbf{G} \setminus \text{dom}(\overline{\beta|_X}) \neq \emptyset$. By Lemma 3.6.2, \mathbf{G}' has no sinks, and also $\forall x \in \mathbf{G}' : \forall y \in N^+(x) \cap \text{dom}(\overline{\beta|_X}) : \beta(y) = \mathbf{0}$. This implies that $\beta = \beta|_{\text{dom}(\overline{\beta|_X})} \cup \beta'$ for some $\beta' \in \text{sol}(\mathbf{G}')$. However, as \mathbf{G}' is a graph with no sinks and no even cycles we have $\text{sol}(\mathbf{G}') = \emptyset$ by Lemma 4.9. This is our contradiction. \square

Clearly, for many graphs this represents a significant improvement over the algorithms from the previous subsection, reducing the worst case exponent from the number of cycles to the number of even cycles. Even though an implementation seeking to take advantage of this encounters the problem of finding an even cycle transversal (finding a minimum such is NP-hard, and not known to be FPT), one can argue also for the practical relevance of Proposition 4.10, besides the merely theoretical improvement. In some situations, it can happen that an even cycle transversal can be easily obtained from the input. In general, it often suffices to find a small – and not a minimum – such set and this can be done relatively efficiently.⁸

Example 4.11 (3.2, 4.4 continued) *The graph G has two even cycles: $\langle b, c, b \rangle$ and $\langle b, c, d, a, b \rangle$. Trying $b = 1$ (respectively, 0), induces the assignment $\bar{\alpha}$ (respectively, $\bar{\beta}$) as in Example 3.2. The induced assignments are functions and hence solutions by Proposition 4.10 and Definition 4.1.*

The above example might be insufficient since b is, in fact, a feedback vertex set, so the conclusion follows already by Proposition 4.2. The following example, shows the difference.

Example 4.12 *In the following graph G*



the only even cycle is $C = \langle a, b, d, c, a \rangle$. Breaking it at, say a , leads to two trials:

$a = 1$ induces $b = c = d = 0$ but then $d = 0$ gives a conflict inducing $b = 1$, and

$a = 0$ induces $b = 1, d = 0$, but no more vertices obtain any induced values.

Neither assignment induces a solution, so Definition 4.1 and Proposition 4.10 imply $\text{sol}(G) = \emptyset$.

In the graph $G \setminus \{e\}$, we have the same even cycle. Trying $a = 1$ gives a conflict as above, but from $a = 0$, we obtain $b = 1 = c$ and $d = 0 = f$, yielding a solution.

This concludes the first set of our algorithms for KER. Except for the obvious algorithm using $\text{CNF}(G)$ from (2.6), testing SAT (of boolean equations) is of use here only as a possible enhancement. The algorithms from the present section can be very efficient when applied to graphs with few (even) cycles and, particularly, when cycles or feedback vertex sets are easily read from the input. We do not think, however, that they will be optimal for all kinds of instances. Their likely shortcoming will arise from the comparison to the algorithm proposed in the following section, which also shows much tighter connections between KER and SAT.

5 KER and SAT

Algorithms in the previous section perform the initial simplification, Algorithm 3.15, extract a relevant subset X of vertices and then answer KER solving a system of equations or trying blindly assignments to X , which induce the assignments to the whole graph.

The following, recursive Algorithm 5.1 performs simplification and induction at each recursive call, returning an element of $\text{sol}(G)$, if such exists, and \perp otherwise. It takes an additional argument, the partial assignment α , and constructs its extension to a complete solution, if possible, or returns \perp if not.

The sub-routine sol_2 is used to solve more efficiently graphs that have maximum degree 2. It is given in Algorithm 5.2 and is probably best explained by simply stating Lemma 5.3.

⁸Finding a minimum feedback vertex set is shown to be FPT in [5]. Given a graph with vertices G , and such a subset $V \subseteq G$, one can try moving, one at a time, a vertex x from V back to the induced subgraph $G \setminus V$, checking if the resulting, induced subgraph $G \setminus V \cup \{x\}$ has an even cycle. This last problem is in P by the recent result from [31]. If no even cycle appears, we continue with $V \setminus \{x\}$ and the induced subgraph extended with x , while if some does, x remains in V . What remains in V , after trying all its vertices, is an even cycle transversal.

Algorithm 5.1 $sol(\mathbb{G}, \alpha)$

Input: A digraph \mathbb{G} and a partial assignment α (initially $\alpha = \emptyset$).

Output: $\beta \in sol(\mathbb{G})$ with $\alpha \subseteq \beta$ if it exists, \perp otherwise.

```
1:  $\alpha := \bar{\alpha}$  ..... // Definition 3.1 applied to  $\mathbb{G} \cup dom(\alpha)$ 
2: if  $\alpha$  is not a function then return  $\perp$ 
3:  $\mathbb{G} := \mathbb{G} \setminus dom(\alpha)$ 
4: if  $\mathbb{G} = \emptyset$  then return  $\alpha$ 
5:  $\mathbb{G} := simp(\mathbb{G})$  ..... // Algorithm 3.15
6: if  $\mathbb{G} = \perp$  then return  $\perp$ 
7: if  $\mathbb{G}$  has maximum degree 2 return  $sol_2(\mathbb{G}, \alpha)$ 
8: Choose some  $x \in \mathbb{G}$ 
9: return  $sol(\mathbb{G}, \alpha \cup \{x, 1\}) \oplus sol(\mathbb{G}, \alpha \cup \{x, 0\})$ 
```

Algorithm 5.2 $sol_2(\mathbb{G}, \alpha)$

Input: A digraph \mathbb{G} of maximum degree 2 and some partial assignment α

Output: $\beta \in sol(\mathbb{G})$ such that $\alpha \subseteq \beta$, \perp otherwise.

```
1:  $\alpha := \bar{\alpha}$  ..... // Definition 3.1
2: if  $\alpha$  is not a function then return  $\perp$ 
3:  $\mathbb{G} := \mathbb{G} \setminus dom(\alpha)$ 
4: if  $\mathbb{G}$  contains an odd cycle without any reversible edge then
5:   return  $\perp$ 
6: if  $bord(\alpha) = \emptyset$  then
7:   return  $\alpha \cup \beta$  for any  $\beta \in sol(\mathbb{G})$ 
8: Choose some connected component  $S \in \mathbb{G}$ 
9:  $B = \{\beta \in sol(S) \mid N^+(bord(\alpha)) \cap \beta^1 \neq \emptyset\}$ 
10: if  $B \neq \emptyset$  return  $\bigoplus_{\beta \in B} sol_2(\mathbb{G}, \alpha \cup \beta)$ 
11: else return  $sol_2(\mathbb{G}, \alpha \cup \beta)$  for any  $\beta \in sol(S)$ 
```

Lemma 5.3 *For any, sinkless, loopless graph G of maximum degree 2: G has a solution iff every odd cycle in G has a reversed edge*

PROOF. Let $G_S = \{S_1, S_2, \dots, S_n\}$ be the n connected components of G . Clearly, $\text{sol}(G) \neq \emptyset$ iff $\forall 1 \leq i \leq n : \text{sol}(S_i) \neq \emptyset$. Since each component S_i has maximum degree 2, so its underlying graph \underline{S}_i is either a path or a chordless cycle. So either S_i does not have an odd cycle or else it is an odd cycle, possibly with some reversed edges. Solvability of S_i follows from Richardson's theorem in the first case. For the second case of S_i being an odd cycle, if S_i has no reversed edge then, by lemma 4.9, S_i does not have a solution. If S_i has one or more reversed edges we show that it has a solution. Write S_i as $x_0x_1x_2\dots x_n$ where $n+1 = |S_i|$ is odd, and $x_{i+1} \in N^+(x_i)$ for all $i \geq 0$, with addition modulo n . Choose x_i, x_{i+1} such that also $x_i \in N^+(x_{i+1})$, and define α by $\alpha^1 = \{x_i\} \cup \bigcup_{0 \leq j < i} \{x_j \mid (i=j) \bmod 2\} \cup \bigcup_{i+1 < j \leq n} \{x_j \mid (i \neq j) \bmod 2\}$. It is easily verified that α is a solution for S_i . \square

Correctness of Algorithm 5.2 follows readily from lemma 5.3. In particular, line 4 determines if G has an odd cycle without reversed edge and the algorithm proceeds only if it does. After this the question of solvability of G is settled by Lemma 5.3. We must acquire the actual solutions for use later in the algorithm. But this is easy - the brute-force approach, for instance, would do it by simply computing all maximal independent sets in all components. In lines 7 and 11 we require only one solution (to G and a component $S \subseteq G$ respectively), and this is even easier. For the case of odd cycles with reversed edges we refer to the proof of lemma 5.3 where we construct an actual solution. For all other components, S , a solution is found by simply assigning **1** and **0** to every other vertex along \underline{S} , which works since \underline{S} is either a path or an even cycle without any sinks nor loops. Consequently, the vertices assigned **1** form an independent set, while every other vertex, on pain of contradicting sinklessness, will have some edge going into this set.

Now, the reason why the algorithm cannot simply stop upon having noted that G is solvable is that each vertex in $\text{bord}(\alpha)$ requires that one of the vertices it points to is **1**. This means we have to search through a potentially large collection of solutions for G . This happens in lines 8-10 and is considered in more detail in the proof of Proposition 5.5 below.

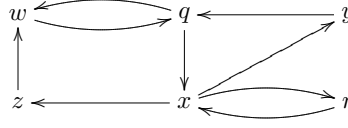
Correctness of Algorithm 5.1 is now quite obvious, although strictly speaking, it is not *an* algorithm but *a class* of algorithms. For instance, the simplification of a graph, as well as inducing of values from a given partial assignment, could be defined otherwise and replace those used here. Also, several minor issues are left for more detailed decisions. For instance, α is only a solution to the reduced graph obtained after a possible series of contractions at line 5. The solution for the actual input graph has to be reconstructed from it by a corresponding series of applications of Fact 3.10.

The polynomial (in the size of the original graph) time, spent in each recursive call on the computation of $\bar{\alpha}$, can be improved since, once it starts going, it only needs to consider border vertices from $\text{dom}(\alpha)$, Definition 3.4. A conflict (a vertex assigned two values so that $\bar{\alpha}$ is no longer a function) must occur at these vertices, and once it is detected, the algorithm can return the failure value at line 2.

Two more central decisions are left open. The operation \oplus denotes angelic choice, ignoring the possible argument \perp , i.e., $x \oplus \perp = \perp \oplus x = x$, and $x \neq \perp \wedge y \neq \perp \rightarrow (x \oplus y) \in \{x, y\}$. (It is also used in line 10 of Algorithm 5.2.) An implementation has to decide how to perform the choice of the first value tried. Finally, we have not specified how to choose an $x \in G$ for branching at line 8. A specific instance of the above algorithm was presented in [12]. It performs no simplification and branches only from maximal degree cyclic vertices. Proposition 4.2 guarantees sufficiency of branching only from cyclic vertices, and choosing maximal degree is often sound.⁹ However, it is not always the best choice, as evidenced by the following example.

Example 5.4 *Consider the graph G :*

⁹Whether degree refers to the in-degree, out-degree or their sum may depend on the implementation and, in particular, on the way of inducing. In our case, inducing happens both along and against the direction of the edges, so degree of a vertex means for us the number of all (bot in and out) neighbours.



The minimal degree is $2 = \deg(y) = \deg(z) = \deg(r)$. Branching from y gives two cases:

$y = \mathbf{1}$ induces the solution with $x = q = z = \mathbf{0}$ and $y = w = r = \mathbf{1}$

$y = \mathbf{0}$ induces the solution with $q = z = r = \mathbf{1}$ and $x = y = w = \mathbf{0}$.

Similarly, each branching from z induces a solution. So, Algorithm 5.1 branching first on some nodes with minimal degree, terminates successfully in just one recursive call. This need not happen when branching on x , the only vertex with maximal degree. Inducing from $x = \mathbf{1}$ gives $y = q = \mathbf{0}$ which yields a conflict at y , while $x = \mathbf{0}$ induces only $r = \mathbf{1}$, i.e., requires further recursive calls.

It is probably too much to ask for an algorithm that always makes the optimal choice of branching vertex. As a simple and general rule for arbitrary graphs, choosing a vertex with maximal degree may be a good heuristic. Taken in conjunction with Algorithm 5.2, it suffices to establish the following upper bound on the time spent by Algorithm 5.1.

Proposition 5.5 *Algorithm 5.1 can be made to run in time $\mathcal{O}^*(1.427^{|G|})$, for all G .*

PROOF. Sinks are removed already when inducing from the empty assignment and loops when removing the contradictions, so all graphs, G' , considered by Algorithm 5.1 after these initial steps are loopless and sinkless. Thus, all vertices have degree 1 or more.

(A) Consider first the case of repeated branching at line 8 on vertices with degree 3 or more. Methods invoked in lines 1-6 are linear (or low polynomial). Trying assignment of $\mathbf{0}$ may not induce any values, while assignment of $\mathbf{1}$ induces values to at least 3 neighbours. This gives the recurrence $T(|G'|) = T(|G'| - 1) + T(|G'| - 4)$ for arbitrary subgraph $G' \subseteq G$. So if the algorithm branches always at line 8 and terminates at 2, 4 or 6, we get the upper bound of $\mathcal{O}^*(1.381^{|G|})$.

(B) Another case occurs when recursion terminates at line 7 with calls to sol_2 . There are then two subcases:

(B.1) If $bord(\alpha) = \emptyset$, then if $G' = G \setminus dom(\alpha)$ is solvable, any solution to G' extends α and its existence is checked in polynomial time in line 4. (If needed, a solution can be found as suggested in the discussion following Lemma 5.3.) The whole case is therefore handled, lines 1-7, in polynomial time with no branching.

(B.2) The complex case is when $bord(\alpha) \neq \emptyset$. Then, if the subgraph $G' = G \setminus dom(\alpha)$ is solvable, we may risk having to generate all its solutions, in search for one matching its $\mathbf{1}$'s against all border vertices. This happens in lines 8-10 of Algorithm 5.2. The algorithm chooses some component S of G' at line 8, and branches on its every solution that leads to a reduction in the size of the border. The depth of the recursion in sol_2 is thus bounded by the minimum among the number of disjoint components in G' and the size $k = |bord(\alpha)|$ of the border. Taking all components to be of the same size l , gives $\frac{|G'|}{l}$ as their number in G' . (This simplification is justified by the cases identified below.) At each recursion level, line 9 of Algorithm 5.2 inspects all solutions of the current connected component S with max degree 2 and size l . Denoting the number of such solutions by $S(l)$, gives the following formula for the upper bound on the complexity of $sol_2(G', \alpha)$:

$$S(l)^{\min(\frac{|G'|}{l}, k)}. \quad (5.6)$$

This reflects the worst case of branching at line 10, for each component of G' , which contributes a multiplicative factor $S(l) * \dots$ to the complexity. Recursive calls at line 11 contribute only an additive factor $S(l) + \dots$, since they allow to use and propagate arbitrary solution β of the current component. Now, $S(l)$ is limited from above by the number of maximal independent sets which, for the concerned graphs with max degree 2, satisfy the recurrence $S(l) = S(l-2) + S(l-3)$, with $S(l)$ approaching 1.321^l as l grows, cf. [17]. Still, this is only the behaviour in the limit, while the initial conditions for low l 's give worse bounds, as can be seen considering the following two cases:

B.2.1 If for all $S_i \in G'$, $l_i = |S_i| \geq 4$, then the number of solutions in any such S_i is bounded from above by 1.381^l , the worst case obtaining for 5-cycles, $l = 5$, with up to $5 < 1.381^5$ solutions.

By substituting into (5.6) we get $1.381^{5 \cdot \min(\frac{|G'|}{5}, k)} \leq 1.381^{5 \cdot \frac{|G'|}{5}} = 1.381^{|G'|}$, i.e. the upper bound not exceeding that from case (A).

B.2.2 If for some $S_i \in G'$, we have $l_i \leq 3$, the worst case obtains when all components of G' are symmetric 3-cycles (i.e., with all edges reversed), since they have 3 solutions each.¹⁰ Now, let n denote the number of times we encounter the upper-bound in (5.6), i.e., the number of times Algorithm 5.1 terminates in line 7 with a call to sol_2 on such a collection of 3-cycles. Let G_1, G_2, \dots, G_n and k_1, k_2, \dots, k_n denote the subgraphs and the numbers of border vertices in each of these n instances of the problem (assuming always the greatest possible $k_i = |dom(\alpha_i)|$, i.e. $|G_i| = |G| - k_i$). Instead of giving the running time in terms of each $|G_i|$ we express it directly in terms of $|G|$, as the sum of *all* n instances of (5.6), each with $S(3) = 3$ for symmetric 3-cycles:

$$\sum_{i=1}^n 3^{\min(\frac{|G|-k_i}{3}, k_i)} \quad (5.7)$$

We maximize $3^{\min(\frac{|G|-k_i}{3}, k_i)}$ for each k_i , noting that since both functions of k_i are monotone, one decreasing and the other increasing, the maximum is obtained when both are equal, i.e., for $k_i = \frac{|G|}{4}$. Now, n and the k_i 's are mutually dependent but (5.7) reaches the maximum when *all* branches of the recursion tree have already processed $k_i = \frac{|G|}{4}$ vertices, and call sol_2 with $|G_i| = \frac{3|G|}{4}$. Then, according to (A), $n = 1.381^{\frac{|G|}{4}}$. To see that this is the worst case, first assume that any of the branches continues splitting, i.e., that Algorithm 5.1 continues branching at line 9 with subgraphs smaller than $\frac{|G|}{4}$. This amounts to increasing k_i , so $\min(\frac{|G|-k_i}{3}, k_i) = \frac{|G|-k_i}{3}$. Consequently, in (5.7) the term $t = 3^{\frac{|G|}{4}} = 3^{\frac{3|G|}{12}}$ is replaced by two terms, $t_1 = 3^{\frac{3|G|-4}{12}}$ and $t_2 = 3^{\frac{3|G|-16}{12}}$, for the subgraph reduced by 1, respectively 4 vertices, according to the recurrence from (A). But $t_1 + t_2 < t$, and the sum keeps thus decreasing if splitting and reducing the size of the subgraphs continues in this way. If, on the other hand, there are fewer than n branches since some terminate earlier for $k_i < \frac{|G|}{4}$, then (5.7) has fewer terms, some smaller than $3^{\frac{|G|}{4}}$. Thus, the maximum of (5.7) is $1.381^{\frac{|G|}{4}} * 3^{\frac{|G|}{4}} = 4.143^{\frac{|G|}{4}} \leq 1.427^{|G|}$.

B.2.2 is the overall worst case. The obtained $1.427^{|G|}$ dominates $1.381^{\frac{|G|}{4}}$, which should be added as the time spent on reaching the n calls to sol_2 terminating all the branches. We obtain thus $\mathcal{O}^*(1.427^{|G|})$ as the overall upper bound for the whole Algorithm 5.1. \square

A more detailed analysis might improve this bound but even the estimate given here shows that Algorithm 5.1 is better than checking every possible maximal independent subset of G , i.e., every potential solution. In particular situations it is possible to specify the choice of branching vertex more carefully and thereby get improved running times for certain classes of graphs. This is done below for the class of oriented graphs.

5.1 Oriented graphs

Many possibilities of improvements of Algorithm 5.1 exist in the interplay of the different choices involved and an interesting, related, question is how much the complexity can be improved when attention is restricted to special classes of graphs. We show that for the oriented graphs it is possible, by choosing $x \in G$ for branching in a certain way, to obtain a better bound than $\mathcal{O}^*(1.427^{|G|})$. For the analysis in this subsection Algorithm 5.2 will play no role, so for simplicity we assume

¹⁰This might not be immediately obvious, but the analysis performed here for 3-cycles can be performed in exactly the same way for 2-cycles, yielding an upper bound which is smaller than for 3-cycles. The result for 3-cycles below is worse than for $l_i \geq 4$ in B.2.1, so assuming only some (but not all) components to be 3-cycles gives a better case.

that Algorithm 5.1 is run to completion, i.e. without calling sol_2 in line 7. In fact, we establish an upper bound on the number of kernels in oriented graphs, not just the time it takes to decide if one exists.

Proposition 5.8 *For oriented $G : |sol(G)| \leq 1.286^{|G|}$ and Algorithm 5.1 can run in $\mathcal{O}^*(1.286^{|G|})$.*

PROOF. We run Algorithm 5.1 on some oriented graph G . Since sinks, contractible paths and basic contradictions are removed without branching we assume that G is free of these. We analyze the running time by considering the following possible cases:

1. G has a vertex x with one or more in-neighbours, $|N^-(x)| \geq 1$, and a single out-neighbour $\{y\} = N^+(x)$. Branching from any such $x \in G$, gives the two cases:
 - (a) x has a single in-neighbour: Then, since G does not contain any contractible paths, the in-neighbour of x must be y and then we have a cycle of length 2. Since G is obtained from some oriented graph this cycle must originate from the contraction of some path which led to the reduction of input size by at least 2. We are thus justified to view the actual size as $|G| - 2$, this earlier reduction being polynomial. The assignment of either **1** or **0** to x induces a value at least to y , giving the recurrence $T(|G|) = T(|G| - 2 - 2) + T(|G| - 2 - 2) = 2T(|G| - 4)$.
 - (b) x has two in-neighbours: Then there are two further subcases:
 - y is an in-neighbour of x : Then there is a 2-cycle in G and we can view its size as in the previous case, $|G| - 2$. Since x has another neighbour, distinct from y , we induce values to at least 2 other vertices when we assign **1** to x while we induce **1** to y when we assign **0** to x . So we obtain the recurrence $T(|G|) = T(|G| - 5) + T(|G| - 4)$.
 - y is not an in-neighbour of x : Then we induce values to 3 other nodes when we assign **1** to x . If we assign **0** to x we induce **1** to y . But since G is sinkless and y is not an in-neighbour of x , there is some out-neighbour of y , distinct from x , that is induced **0** in this case. It follows that we get the recurrence $T(|G|) = T(|G| - 4) + T(|G| - 3)$. This is the worst possible situation for case 1).

Every G with maximum degree 3 or less will fall under the current case, i.e., have a vertex x with a single out-neighbour. This follows since G is sinkless and therefore has at least one final strongly connected component S with $|S| \geq 2$. Clearly, every vertex in S has at least one in-neighbour and one out-neighbour. But if every vertex in S has two out-neighbours then since S is final it follows that there is a vertex in S with two or more in-neighbours, contradicting the fact that no vertex has degree more than 3. Consequently, in the following case, when no vertex with a single out-neighbour exists, the degree of a graph is at least 4.

2. Case 1) does not apply and G has maximum degree 4. Then there is some final strongly connected component $S \subseteq G$ where every vertex has more than one out-neighbour. It follows that every vertex in S has exactly two out-neighbours and exactly two in-neighbours. To see this note that since S is final the sum of out-neighbors over vertices in S must be less than or equal to the sum of in-neighbours over vertices of S . So either every vertex has two in-neighbours or else there is some vertex that has more than two in-neighbors, the latter option being ruled out by G having maximum degree 4. We analyze the situation with branching on an arbitrary vertex x from S , obtaining two cases:
 - (a) x has some out-neighbour that is also an in-neighbour. Then there is a cycle of length two that has been obtained from contracting a path. This yields $T(|G|) = T(|G| - 2)$. Also, on assignment of **1** to x we induce values to at least 2 other vertices while on assignment of **0** to x we do not necessarily induce any values. So the recurrence becomes $T(|G|) = T(|G| - 5) + T(|G| - 3)$.

- (b) All neighbours of x are distinct. Then, on assignment of **1** to x we induce values to 4 other vertices while if we assign **0** to x then we might not induce anything and so might have to solve a problem of size $T(|G| - 1)$. However, since we removed x from G to obtain the graph corresponding to this subproblem there is a vertex with only one out-neighbour in this graph. This gives us, by case 1), the worst case recurrence $T(|G| - 1) = T(|G| - 1 - 4) + T(|G| - 1 - 3) = T(|G| - 5) + T(|G| - 4)$. The recurrence is thus $T(|G|) = 2T(|G| - 5) + T(|G| - 4)$.
3. Case 1) does not apply and G has a vertex of degree 5. Then we branch on such a vertex obtaining $T(|G|) = T(|G| - 1) + T(|G| - 6)$.

The recurrence from 3 is the overall worst-case in this analysis, giving the bound $\mathcal{O}^*(1.286^{|G|})$. \square

Notice that the proofs of Propositions 5.5 and 5.8 do not specify completely how to choose a branching vertex, but only narrow the choices down to sets of vertices with some desired properties. The question about the exact choice is still highly relevant for an implementation.

5.2 DPLL

If, in Algorithm 5.1, we take G to be a CNF formula, the algorithm turns out to be exactly the pseudo-code for the DPLL algorithm for satisfiability, [10, 9], which is the basis of virtually all modern SAT-solvers (for a relatively recent overview, one can consult e.g., [23]). Inducing in the first line amounts then, typically, to the unit propagation and the condition ‘ α is not a function’ amounts to the ‘conflict’ in the SAT-solving parlance. An α satisfying all clauses in G is returned, line 4. Otherwise, the remaining problem is preprocessed for the next recursive call, line 5. Simplification may include elimination of clauses with pure literal (occurring only positively or only negatively), as well as learning and many other heuristics depending on the implementation. We suggested, similarly, a wide range of possible choices in Section 3. Choosing then wisely the branching literal x is one of the crucial aspects of successful SAT-solvers.

The coincidence of Algorithm 5.1 and DPLL goes beyond the mere fact of both instantiating the general branch and bound schema. It involves also the fact that kernels can be seen as solutions, (2.2), and that during their gradual construction, partial assignments induce values to the neighbourhoods, in analogy to unit propagation and other constraint propagation techniques in SAT. One may therefore expect the lessons from SAT-solving to be relevant for KER-solving. The crucial aspects of SAT-solvers concern the efficiency and range of inducing values from a given, partial assignment, line 1, and the choices of the branching point and its value required to get the most out of the propagation of constraints implied by the performed choices, line 8. These two elements occupy the critical position, as SAT-solvers spend around 80% of time on this phase. It is reasonable to expect a similar situation in KER-solving. The importance of this aspect has been illustrated by the complexity analysis. The bound $\mathcal{O}^*(1.427^{|G|})$ for the general case and its improvement to $\mathcal{O}^*(1.286^{|G|})$ for oriented graphs, were obtained due to the respective graphs enabling, at each recursive call, some minimal extension of the current partial assignment, thus reducing the remaining search space. This justifies also the expectation that Algorithm 5.1, propagating partial assignments recursively, will outperform the algorithms from Section 4, which do not take any advantages of the attempted partial assignments.

There seems to be no general guidance in actually performing the choice of branching vertex. High degree may often work well but, as we saw in example 5.4, is not necessarily optimal. It might be too much to ask for a strategy working best in all cases but uncertainty at this point may also reflect the lack of experience and overview of the problem instances. In SAT-solvers, the choice is performed depending on the subclass of instances for which the solver is designed. Choice of the branching literal in solvers for random-SAT uses a lookahead procedure, which determines the reduction in the search space effected by each choice. Solvers for industrial-SAT can use the results of learning from the earlier encountered conflicts.

We have thus mentioned another important aspect: a SAT-solver is designed for a specific category of instances. A solver deciding SAT quickly on instances from industrial, or other rational

and systematic contexts (using additional techniques of conflict analysis and clause learning), may perform poorly on random instances. For random instances, “local search” heuristics for merely finding a solution may be extremely efficient but remain incomplete, being unable to conclude unsatisfiability. The winner of several categories of the SAT-competition in previous years, SATzilla, is actually a collection of various algorithms, which are only chosen appropriately depending on the analysis of the actual instance. The lack of one, uniform approach and the need to adjust solutions and heuristics to appropriately limited subclasses of instances is a general lesson from SAT. One can expect KER to face the same challenge of identifying such relevant subclasses. It is likely, however, that just as the DPLL schema is at the core of virtually all efficient procedures for solving SAT, so does Algorithm 5.1 express the core structure of efficient approaches for solving KER.

An important case of subclasses are those for which the problem becomes tractable. For instance, 2-SAT is NL-complete and Horn-SAT is P-complete. Search for sufficient conditions for kernel existence is an active research field, e.g., [1, 14, 15, 18, 3], with a recent overview in [4]. Further research should, in our opinion, consider also the problem of finding classes of graphs which may not admit kernels but have complexity bounds for the KER-problem below NP-completeness.¹¹

Finally, let us mention an interesting SAT phenomenon – phase transition. When the clausal density (the ratio of number of clauses to the number of variables) is below 4, the theory is, with high probability satisfiable, while when it exceeds 4.5, the theory is almost certainly unsatisfiable. The instances with the clausal density around the transition value, 4.25, are the most difficult to solve. It is not obvious how to translate this into the graph language. Graph density (average degree) seems to be a relative of the clausal density, so one might conjecture that sparse graphs should be solvable with high probability (as are, e.g., all trees, dags and 50% of all cycles.) Very dense graphs might be expected to be relatively easy (e.g., kernels in a weakly complete digraph G (one with a complete underlying graph \underline{G}) are exactly nodes x satisfying $N^-(x) = G \setminus \{x\}$) but should be expected to be unsolvable. A naive guess might expect the most difficult problems somewhere in the middle between these two extremes. This is partially confirmed by the tests of the algorithm presented in [12]. According to them, sparse graphs and graphs with density over 50% are relatively easy, while those with density around 15-20% are most difficult. On the other hand, it has been shown in [16] that the kernel problem is NP-complete for planar digraphs of degree at most 3, so the “easy” instances of KER can certainly be difficult enough. It remains to be seen if phase transition from SAT has a counterpart in KER and, if so, under what measure of graph density.

5.2.1 Why not reducing to SAT?

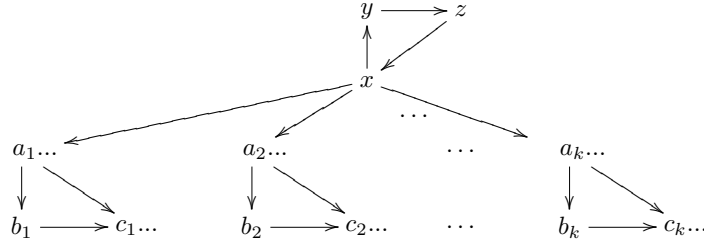
The relevance of SAT for KER should not be overestimated to the point of dismissing the latter by merely translating its instances into SAT and using SAT-solvers. Sophistication of modern SAT-solvers might suggest such a move and can even be expected to yield good results in various cases.

The main reasons, justifying our separate treatment of KER, are complexity considerations. According to the bound on the number of maximal independent subsets, $3^{\frac{|G|}{3}}$, and the possibility to produce them with polynomial delay, general KER can be solved in $\mathcal{O}^*(1.443^{|G|})$. The analysis of Algorithm 5.1 lowers this bound to $\mathcal{O}^*(1.427^{|G|})$. For general SAT, on the other hand, the best known upper bound is of order $\mathcal{O}^*(2^{n(1 - \frac{1}{\mathcal{O}(\log(\frac{m}{n}))})})$, where n is the number of variables and m the number of clauses, [8]. (Note that this converges to $\mathcal{O}^*(2^n)$ as the ratio m/n grows. For DPLL solving k -SAT, 2^n is also the lower bound, as k goes towards infinity, [28].) SAT-instances of form (2.6), representing KER, provide thus a non-trivial subclass which can be solved more efficiently than this general upper bound, irrespectively of the m/n ratio.

¹¹One non-trivial result of this kind follows from the work done on stable matchings. An algorithm presented in [21] decides existence of stable matchings for the roommates problem in polynomial time. This solves KER in polynomial time for any digraph that is an orientation of a line graph and for which every weakly complete subgraph is acyclic.

On a more practical side, it is possible that instances of KER, when translated into SAT, would be amenable to a uniform processing with the gains comparable to the difference between these two upper bounds. It is also possible that subclasses of digraphs, like the oriented ones, making KER easier, could be translated into instances making also SAT easier. But the gains in complexity, both for the general KER and for the oriented graphs, were based on specific strategies for selecting active literals, which do not seem to be reflected in those applied in SAT-solvers. To use SAT-solvers with comparable increase in efficiency will almost certainly require their adjustments and is a possible direction for further research. Similar remark applies to the simplifications from Section 3. When translated into the operations on the translated SAT instances, they correspond to techniques used in SAT-solving: basic contradictions can be discovered by means of implication graphs for binary clauses, while contraction of isolated paths amounts to detection of equivalent variables from binary clauses. Many SAT-solvers perform such simplifications only at the stage of preprocessing, while Algorithm 5.1 performs them dynamically at each recursive call. Indeed, for general instances of SAT, the cost of their repetitive use tends to exceed the gains (though some use of such “inprocessing” may be viable, [20]). But for KER instances, propagation of values along the isolated paths and checking neighbourhood of a vertex are among the simplest possible operations. As many isolated paths or basic contradictions can be expected to appear only dynamically, their repeated simplification may be worthwhile. A similar move could be easily implemented in a SAT-solver, but it is typically not included, due to low average cost/gain ratio. In special situations, the difference may be of exponential order.

Example 5.9 Consider the following graph which should be seen as arising only during computation from removal of nodes assigned 0. All nodes c_i and a_i can have more outgoing edges.



Its translation into CNF, following (2.6), yields the clauses:

- i) $a_i \vee b_i \vee c_i$, $\neg a_i \vee \neg b_i$, $\neg a_i \vee \neg c_i$ and $b_i \vee c_i$, $\neg b_i \vee \neg c_i$ – for each lower triangle, and
- ii) $x \vee y \vee \bigvee_i a_i$, $\neg x \vee \neg y$, $\neg x \vee \neg a_i$ (for all $1 \leq i \leq k$), $y \vee z$, $\neg y \vee \neg z$ and $x \vee z$, $\neg x \vee \neg z$.

Assume a SAT-solver selects, as the active literals, consecutive c_i 's. Trying

$c_1 = 1$, the unit propagation yields $a_1 = b_1 = 0$

and this operation is repeated k times, after which the clauses in ii) are reduced to:

- iii) $x \vee y$, $\neg x \vee \neg y$, $y \vee z$, $\neg y \vee \neg z$ and $x \vee z$, $\neg x \vee \neg z$.

Now a conflict results trying both values 0, 1 with any choice of the active literal. Backtracking one c_i literal at the time, and trying $c_i = 0$, gives the same result, so that after 2^{k+1} trials, this backtrack search concludes unsatisfiability.¹²

Algorithm 3.15 identifies contradictions efficiently (footnote 5) and sets first all $a_i = 0$. The rest of the graph is processed in linear time. Algorithm 5.1 may assign various c_i 's and/or b_i 's before going to x, y, z , where two attempts with 0 and 1 at any of these nodes unveil unsatisfiability.

In short, solving KER directly gives gains in complexity and efficiency and corresponding gains could be achieved by fine-tuning a SAT-solver to the specific form of SAT-instances arising from KER. However, in order for this fine-tuning to give comparable effects, it would have to follow the results of the direct analysis of KER, as presented in this paper.

¹²Such claims must, of course, be taken with serious reservations. A particular solver, using a particular strategy and heuristic, might actually happen to avoid the problem. Although the example seems also to depend on the strategy for selecting the active literals, one can adjust it to many different strategies, since all c_i might possess other outgoing edges (e.g., occur in most clauses). The main point is that simplifications of such form are, typically, performed by a SAT-solver only in preprocessing and not during the computation.

6 Conclusions

We have studied the problem, KER, of solvability of digraphs or, in the more standard language, of determining if a given digraph has a kernel. We began by observing its equivalence to the problem of satisfiability of propositional formulae, whether in usual or infinitary propositional logic. Seeing different applications of digraph kernels, in areas such as game theory and non-classical logics, it is conceptually rewarding in itself to see that kernels can be expressed – equivalently and naturally – as models of propositional theories.

We have proposed a series of graph reductions which preserve and reflect solvability and, being linear (or low polynomial), can be incorporated into the algorithms for KER-solving. In Section 4, we gave two such algorithms: one based on the extraction of a feedback vertex set, F , and another reducing the complexity even to $\mathcal{O}^*(2^{|E|})$, where E is an even cycle transversal. As a consequence, KER is FPT not only in the size of F but also of E . (As an instance of reducing KER to SAT, we gave a variant of the first algorithm where solving a reduced system of boolean equations replaced blind trials of all assignments to the sinks of a labeled dag, representing the input graph.) These algorithms can be expected to perform well on graphs with few (even) cycles and, especially, when even cycle transversal or, at least, feedback vertex set can be easily obtained from the input.

The question about a general algorithm for arbitrary instances of KER, led in Section 5 to another, new algorithm, which turns out to be virtually identical to the well-known DPLL algorithm, underlying modern SAT-solvers. From this we dare draw a series of conjectures for further development of the research on KER. It suggests that this final algorithm may outperform others on the large, practical instances of KER. This, however, will depend on more detailed decisions, because the presented sketch gives only a class of algorithms. It leaves open the possibility for further choices and improvements at points where such possibilities were realised or are still investigated in the context of SAT-solving. Experience with SAT-solving suggests that one will have to adjust choices and heuristics to specific subclasses of instances. As a particular case, we showed that, with a specific branching strategy, oriented digraphs guarantee a certain minimum of inducing during the recursive trials, allowing to reduce their worst case bound to $\mathcal{O}^*(1.286^{|G|})$. For the general case of arbitrary graphs, one can still ensure minimum inducing guaranteeing the worst case bound not exceeding $\mathcal{O}^*(1.427^{|G|})$.

We have shown that SAT-solving can be, to some extent, incorporated in KER-algorithms. More importantly and generally, however, solving KER appears to pose the same kind of choices and challenges, as met earlier in the design of SAT-algorithms. One can therefore expect that issues known from SAT, like those exemplified in Section 5.2, have graph-theoretic counterparts that will come up in the design of KER-algorithms. This itself may provide an independent motivation, and a specific direction, for the further study of KER. On the other hand, it does not seem unreasonable to expect that SAT-solvers may eventually benefit from KER-algorithms. The fact that KER can be formulated just as naturally in the language of graphs as in the language of logic or of game-theory, suggests that the problem can act as a useful point of reference for the exchange of ideas between these different fields. A better understanding of KER might very well foster a better understanding of the relationship between different problems that, apart from being computationally demanding, often appear to have little in common.

Having seen several new algorithms, the reader might expect also a report of their implementation and performance in practice. However, the analogy to SAT suggests that one should not rely here on any simple statements of the kind “algorithms perform well in practice”. More precisely, any such statement should be qualified by a careful description of the instances and actual performance measures. Experimentation with various implementations seems to be, in the case of KER as it is in the case of SAT, an independent and extensive field of work, not to be dismissed in a few sentences. We leave this important aspect for future work.

References

- [1] Martine Anciaux-Mendeleer and Pierre Hansen. On kernels in strongly connected graphs. *Networks*, 7(3):263–266, 1977.
- [2] Claude Berge and Pierre Duchet. Recent problems and results about kernels in directed graphs. *Discrete Mathematics*, 86:27–31, 1990.
- [3] Mostafa Blidia. A parity digraph has a kernel. *Combinatorica*, 6(1):23–27, 1986.
- [4] Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.
- [5] Jianer Chen, Yang Liu, Songjian Lu, Barry O’Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *Journal of ACM*, 55(5):1–19, 2008.
- [6] Vašek Chvátal. On the computational complexity of finding a kernel. Technical Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal, 1973. <http://users.encs.concordia.ca/~chvatal>.
- [7] Nadia Creignou. The class of problems that are linearly equivalent to satisfiability or a uniform method for proving NP-completeness. *Theoretical Computer Science*, 145:111–145, 1995.
- [8] Evgeny Dantsin and Edward A. Hirsch. Worst-case upper bounds. In *Handbook of Satisfiability*, pages 341–362. 2008.
- [9] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [10] Martin Davis and Hillary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.
- [11] Yannis Dimopoulos and Vangelis Magirou. A graph theoretic approach to default logic. *Information and Computation*, 112:239–256, 1994.
- [12] Yannis Dimopoulos, Vangelis Magirou, and Christos H. Papadimitriou. On kernels, defaults and even graphs. *Annals of Mathematics and Artificial Intelligence*, 20:1–12, 1997.
- [13] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170(1-2):209–244, 1996.
- [14] Pierre Duchet. Graphes noyau-parfaits, ii. *Annals of Discrete Mathematics*, 9:93–101, 1980.
- [15] Pierre Duchet and Henry Meyniel. Une généralisation du théorème de Richardson sur l’existence de noyaux dans les graphes orientés. *Discrete Mathematics*, 43(1):21–27, 1983.
- [16] Aviezri S. Fraenkel. Planar kernel and grundy with $d \leq 3$, $d_{out} \leq 2$, $d_{in} \leq 2$ are NP-complete. *Discrete Applied Mathematics*, 3(4):257–262, 1981.
- [17] Zoltán Füredi. The number of maximal independent sets in connected graphs. *Journal of Graph Theory*, 11:463–470, 1987.
- [18] Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.
- [19] Gregory Gutin, Ton Kloks, Chuan Min Lee, and Anders Yeo. Kernels in planar digraphs. *Journal of Computer and System Sciences*, 71(2):174–184, 2005.

- [20] Marijn Heule, Matti Järvisalo, and Armin Biere. Efficient CNF simplification based on binary implication graphs. In *Theory and Application of Satisfiability Testing*, volume 6695 of *LNCS*, pages 201–215, 2011.
- [21] Robert W. Irving. An efficient algorithm for the stable roommates problem. *J. Algorithms*, 6(4):577–595, 1985.
- [22] David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent subsets. *Information Processing Letters*, 27:119–123, 1988.
- [23] Inês Lynce and João P. Marques-Silva. An overview of backtrack search satisfiability algorithms. *Annals of Mathematics and Artificial Intelligence*, 37:307–326, 2003.
- [24] Eric C. Milner and Robert E. Woodrow. On directed graphs with an independent covering set. *Graphs and Combinatorics*, 5:363–369, 1989.
- [25] John W. Moon and Leo Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:2328, 1965.
- [26] Victor Neumann-Lara. Seminúcleos de una digráfica. Technical report, Anales del Instituto de Matemáticas II, Universidad Nacional Autónoma México, 1971.
- [27] Rolf Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, 2006.
- [28] Pavel Pudlák and Russell Impagliazzo. A lower bound for dll algorithms for k-sat. In *In Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA00*, 2000.
- [29] Moses Richardson. On weakly ordered systems. *Bulletin of the American Mathematical Society*, 52:113–116, 1946.
- [30] Moses Richardson. Solutions of irreflexive relations. *The Annals of Mathematics, Second Series*, 58(3):573–590, 1953.
- [31] Neil Robertson, Paul D. Seymour, and Robin Thomas. Permanents, pfaffian orientations and even directed cycles. *Annals of Mathematics (2)*, 150(3):929–975, 1999.
- [32] Robert Tarjan. Depth-first search and linear graph algorithms. In *Switching and Automata Theory, 1971., 12th Annual Symposium on*, pages 114–121, Oct. 1971.
- [33] John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944 (1947).
- [34] Michał Walicki and Sjur Dyrkolbotn. The graphical structure of paradox. 2010. <http://www.ii.uib.no/~michal/graph-paradox.pdf>.

Chapter 4

Paper B: Kernels in digraphs that are not kernel perfect

This paper was published in *Discrete Mathematics*, vol. 312, no 16, pp. 2498-2505, August 2012.

Kernels in digraphs that are not kernel-perfect

Sjur Dyrkolbotn and Michał Walicki
Department of Informatics,
University of Bergen, Norway

Abstract

An equivalent of kernel existence is formulated using semikernels. It facilitates inductive arguments, which allow to establish several sufficient conditions for existence of kernels in finite digraphs. The conditions identify classes of digraphs that have kernels without necessarily being kernel-perfect.

Kernel theory received particular attention due to the connections with perfect graphs, and much work has revolved around the kernel-perfect digraphs – ones in which every induced subdigraph has a kernel. A lot of the theoretical interest in the field undoubtedly stems from the connections between such digraphs and orientations of perfect graphs ([2] gives an overview). The perfect graph conjecture has now become theorem and the interest in kernel theory among graph theorists seems to have diminished. This is unfortunate as the kernel-problem seems interesting in its own right, also because it encodes easily many problems from game theory, argumentation theory, logic and logic programming, see e.g. [3, 4, 5, 6, 7, 8]. In such a wider context, kernel-perfectness is all too strong a property, which can hardly be expected in most structures of potential interest.

1 Basic concepts, semikernels and solvers

We consider only finite digraphs, $G = \langle G, N \rangle$, with G a set of vertices and $N \subseteq G \times G$ a set of directed edges. We use the notation $N^+(x) = \{y \mid \langle x, y \rangle \in N\}$ and $N^-(y) = \{x \mid \langle x, y \rangle \in N\}$ and write $N^+(G, x)$ when we need to identify the appropriate G . We often work with $N^-[x] = \{x\} \cup N^-(x)$, the *closed in-neighborhood* of x . Function applications are extended point-wise to sets, e.g., $N^-(X) = \bigcup_{x \in X} N^-(x)$, $N^-[X] = \bigcup_{x \in X} N^-[x]$, etc.

A subset of nodes $X \subseteq G$ may be treated as the induced subdigraph X , and $G \setminus X$ denotes the subdigraph induced by $G \setminus X$. \underline{G} denotes the underlying undirected graph, i.e. with $\underline{N} = \{\{x, y\} \subseteq G \mid \langle x, y \rangle \in N\}$. We consider only simple *paths* and *cycles*, i.e. no vertex is repeated except that on a cycle the first is the same as the last. The term *walk* is reserved for cases when vertices are internally repeated. We write $P = P_{x,y}$ to indicate that P is a path from x

to y . Then $P_{z,w}$ denotes the sub-path of P going from z to w . A path $P = P_{x,y}$ *alternates* on $F \subseteq G$ if every other vertex of P is in F . (It is not required that the first vertex of P is in F .) We allow ourselves to denote directed and undirected paths in the same way, but state explicitly that P is in \mathbf{G} or in $\underline{\mathbf{G}}$. When P is viewed as an induced subgraph we might write \underline{P} to stress that we are thinking of it as undirected. Paths and cycles are *odd* or *even*, depending on the parity of the number of edges used.

Paths $P = P_{x,y}, Q = Q_{y,z}$ can be *appended* whenever $P \cap Q = \{y\}$, giving the path PQ from x to z (with only one occurrence of y). Paths $P = P_{x,y}, Q = Q_{z,w}$, with $z \in N^+(y)$, on the other hand, can be *concatenated* when $P \cap Q = \emptyset$, written $P;Q$. A vertex is also treated as an even path (the empty path), and we write $P;z$ to denote the path $P = P_{x,y}$ extended with $z \in N^+(y)$. For a path $P = P_{x,y}$, $\text{int}(P)$ denotes the set of internal vertices of P , i.e. all vertices on P except $\{x, y\}$.

$[\mathbf{G}, x)$ denotes the *cone* of x in \mathbf{G} – the set of vertices to which x has a directed path in \mathbf{G} . We write $[\mathbf{G}, x)_e, [\mathbf{G}, x)_o$ to denote only those vertices to which x has an even or an odd directed path, respectively. The definition extends to the undirected case in the obvious way with $[\underline{\mathbf{G}}, x)$ denoting the set of vertices to which x has an undirected path in $\underline{\mathbf{G}}$. Since x is regarded as an even (empty) path, we always have $x \in [\mathbf{G}, x)_e \subseteq [\mathbf{G}, x)$.

A *kernel* of a digraph $\mathbf{G} = \langle G, N \rangle$, is a set $K \subseteq G$ such that:

$$N^-(K) = G \setminus K \quad (1.1)$$

This is the case iff K is *independent*, $N^-(K) \subseteq G \setminus K$, and *absorbing*, $N^-(K) \supseteq G \setminus K$. The empty digraph, with $G = \emptyset$, has the unique kernel $K = \emptyset$. Not every digraph has a kernel, the obvious example being an odd directed cycle. A fundamental result is Richardson's theorem [12], which states that if a (finitely branching) digraph has no odd directed cycles then it has a kernel.

The concept of a semikernel is a useful technical tool in kernel theory, introduced by Victor Neumann-Lara in [11]. A *semikernel* is an independent subset of a digraph $L \subseteq G$ that is *locally absorbing*, i.e. such that:

$$N^+(L) \subseteq N^-(L) \subseteq G \setminus L \quad (1.2)$$

A kernel is a semikernel, while a semikernel L satisfying $N^-[L] = G$ is a kernel. $Sk(\mathbf{G})$ denotes the set of semikernels in \mathbf{G} and $Kr(\mathbf{G})$ the set of its kernels.

Virtually all results in the literature about the existence of kernels address kernel-perfect digraphs, namely, ones where every induced subdigraph has a kernel ([2] gives an overview). A basic result, stating that a digraph is kernel-perfect iff each induced subdigraph has a semikernel, is typically used to obtain sufficient conditions for kernel-perfectness, e.g. in [9]. We use semikernel in a different way, based on the concept of a solver.

Definition 1.3 A solver for a digraph \mathbf{G} is a sequence of induced subdigraphs and semikernels $\langle \mathbf{G}_i, L_i \rangle_{1 \leq i \leq n}$ such that:

$$(1) \quad \mathbf{G}_1 = \mathbf{G}$$

- (2) L_i is a semikernel in G_i for all $1 \leq i \leq n-1$
- (3) $G_{i+1} = G_i \setminus N^-[L_i]$ for all $1 \leq i \leq n-1$
- (4) L_n is a kernel of G_n .

Having a solver is equivalent to having a kernel:

Theorem 1.4 *A digraph has a kernel iff it has a solver.*

PROOF. \Rightarrow) If $K \in Kr(G)$, then $\langle G, K \rangle$ is a solver for G .

\Leftarrow) Let $\langle G_i, L_i \rangle_{1 \leq i \leq n}$ be a solver for G and let $K = \bigcup_{1 \leq i \leq n} L_i$. We show that K is (i) independent and (ii) absorbing.

(i) Assume towards contradiction that there are $x, y \in K$ with $y \in N^+(x)$. Since every semikernel is independent and K is a union of semikernels, x and y belong to different ones, say $x \in L_i, y \in L_j$. There are two cases, both leading to contradiction. If $i < j$, then $y \in N^-(L_i)$ since L_i is a semikernel. Then, by Definition 1.3, $y \notin G_j$ and so $y \notin L_j$. If $j < i$, then $x \in N^-(L_j)$, so $x \notin G_i$ and $x \notin L_i$.

(ii) If there is some $x \in G \setminus N^-[K]$, then $x \notin N^-[L_i]$ for all $1 \leq i \leq n$. But then $x \in G_n \setminus N^-[L_n]$, contradicting the fact that L_n is a kernel in G_n . \square

In employing solvers to prove existence of kernels, it will be useful to consider sets of semikernels containing some given vertex $x \in G$, denoted $Sk(x)$, or $Sk(G, x)$ when we need to identify the digraph. We will be particularly interested in the *minimal* members of $Sk(x)$ (w.r.t. set-inclusion). We denote the set of these by $minSk(x)$. We will also use *completions* of semikernels, defined as follows:

Definition 1.5 *The completion \bar{L} of an $L \in Sk(G)$ is defined inductively:*

$$\begin{aligned} L_0 &= L \\ L_{i+1} &= sinks(G \setminus N^-(L_i)) \end{aligned}$$

Fixed-point, $\bar{L} = L_{i+1} = L_i$, is reached no later than at $i = |G|$.

Every L_i , in particular \bar{L} , is a semikernel. \bar{L} can be characterized equivalently as the minimal $M \in Sk(G)$ such that $L \subseteq M$ and $G \setminus N^-[M]$ is sinkless.

A particularly important case obtains starting with $L_0 = \emptyset$. Then $L_1 = sinks(G)$ and the semikernel $\bar{\emptyset}$ is a subset of every kernel (if G has any). This observation originates from the proof of Richardson's theorem, [12], was clarified in [10] and is stated generally, without redundant side conditions, in [1].

Fact 1.6 *For any $G : Kr(G) = \{K \cup \bar{\emptyset} \mid K \in Kr(G^\circ)\}$, where $G^\circ = G \setminus N^-[\bar{\emptyset}]$.*

As a consequence, possible restrictions to sinkless digraphs are inessential, since the existence of kernels in any G is determined by their existence in its sinkless residuum G° . In particular, every digraph with $G^\circ = \emptyset$ has a unique kernel,

for instance, every dag with no infinite directed path. We will not use this fact explicitly, but one main result presented in the next section is stated only for sinkless digraphs.

The minimal semikernels containing x and their completions will be useful because they ensure the existence of certain paths in G , as detailed in the following lemma.

Lemma 1.7 *For a digraph G , any $x \in G$ and any $L \in \min Sk(x)$ we have:*

- (1) *For every $y \in L$ there is a directed path $P = P_{x,y}$ of even length, alternating on L .*
- (2) *For every $y \in \bar{L} \setminus \text{sinks}(G)$ there is some $z \in L$ for which there are directed paths $P = P_{x,z}$ and $Q = Q_{y,z}$ of even length, alternating on \bar{L} .*

PROOF. (1) We consider some arbitrary $L \in \min Sk(x)$ and form the following set of vertices:

$$L' = \{y \in L \mid \exists P = P_{x,y} \text{ in } G \text{ which is even and alternating on } L\} \quad (1.8)$$

Since x is regarded as the empty (even) path, $x \in L'$. Consider arbitrary $y \in L'$ and let $P = P_{x,y}$ be even and alternating on L . Then P is also alternating on L' , since each even vertex on P , being in L , is also in L' .

We show $L' \in Sk(x)$, thereby proving the claim (since then either $L = L'$ or else we have contradicted minimality of L). Now, from $L' \subseteq L$ it follows that L' is independent. Assume towards contradiction that L' is not locally absorbing, i.e. there is some $y \in L'$ with $z \in N^+(y)$ for which $N^+(z) \cap L' = \emptyset$. Since L is a semikernel it follows that there is some $w \in N^+(z) \cap (L \setminus L')$. Consider the even directed path $P = P_{x,y}$, alternating on L . Since $w \notin L'$ we know that w is not on P (since otherwise $P_{x,w}$ would witness to $w \in L'$). If z is on P , then $P_{x,z}$ is odd and alternating on L and so $P_{x,z}; w$ is even and alternating on L , contradicting $w \notin L'$. If z is not on P , we obtain the directed path $P; z; w$ that is even and alternating on L , again contradicting $w \notin L'$.

(2) For all $y \in L$ the claim follows from (1) (remembering that we have the empty path y). Any $y \in \bar{L} \setminus L$ is in L_i for some i , by Definition 1.5, so we proceed by induction on i . The basis case $i = 0$ is already established. For the induction step we consider an arbitrary $y \in L_i \setminus L_{i-1}$. By Definition 1.5, $y \in \text{sinks}(G \setminus N^-(L_{i-1}))$ and since $y \notin \text{sinks}(G)$, we have $N^+(y) \neq \emptyset$. In particular, there is some $z \in N^+(y) \cap N^-(L_{i-1})$ which means there is some $v \in L_{i-1} \cap N^+(z)$. Then by IH there is $r \in L$ with directed paths $P = P_{x,r}$ and $Q = Q_{v,r}$, both even and alternating on \bar{L} . To prove the induction step we show that there is a directed path $R = R_{y,r}$ that is even and alternating on \bar{L} . There are three simple cases to consider. If $y \in Q$, then since $y \in \bar{L}$ and Q is even and alternating on \bar{L} we can take $R = Q_{y,r}$ and R will then be even and alternating on \bar{L} . If $y \notin Q$ but $z \in Q$, then $Q_{z,r}$ is odd and alternating on \bar{L} and so we can take $R = y; Q_{z,r}$. The only possibility left is $y \notin Q$ and $z \notin Q$. In this case, we take $R = y; z; Q$. \square

The usefulness of the lemma will become clear in the following section.

2 Some sufficient conditions for kernel existence

Our results involve combinations of the properties from the following definition.

Definition 2.1 *A vertex $x \in G$ is free if it does not lie on any undirected odd cycle in \underline{G} . A subset F of vertices from \mathbf{G} is:*

- (1) free iff all $x \in F$ are free;
- (2) even iff there is no odd directed path in \mathbf{G} between any distinct $x, y \in F$;
- (3) strongly even iff $[\mathbf{G}, x)_e \cap [\mathbf{G}, y)_o = \emptyset$ and $[\mathbf{G}, x)_o \cap [\mathbf{G}, y)_e = \emptyset$ for all distinct $x, y \in F$;
- (4) a candidate iff $Sk(\mathbf{G}, x) \neq \emptyset$ for every $x \in F$;
- (5) a perfect candidate iff it is a candidate and $Sk(\mathbf{G}', x) \neq \emptyset$ for every $x \in F$ and every induced sinkless subdigraph $\mathbf{G}' \subseteq \mathbf{G}$ that contains x .

\mathbf{G} is said to be:

- (a) separated by F iff for every directed odd cycle C in \mathbf{G} , there is an $x' \in C$ such that $N^+(x') \cap F \neq \emptyset$;
- (b) doubly separated by F iff it is separated by F and for all odd directed cycles C in \mathbf{G} , with the exception of at most one, there are distinct $x', y' \in C$ such that $N^+(x') \cap F \neq \emptyset$ and $N^+(y') \cap F \neq \emptyset$;
- (c) strongly separated by F iff for every odd undirected cycle C in \underline{G} there are distinct $x', y' \in C$ s.t $N^+(x') \cap F \neq \emptyset$ and $N^+(y') \cap F \neq \emptyset$.

The remainder of the paper shows that the following combinations are sufficient for the existence of kernels:

- (3)+(4)+(a) – Theorem 2.2;
- (1)+(5)+(b) – Theorem 2.6;
- (1)+(2)+(c) – Corollary 2.14, for sinkless digraphs.

Corollary 2.14 follows from Theorem 2.12, stated with more general properties to be introduced in due course. We also give counterexamples showing insufficiency of some weaker conditions, in particular of (1)+(c) and of (1)+(2)+(b), for sinkless digraphs.

The conditions require the existence of $F \subseteq G$ with some separation property (a)-(c), which allows to “break” every odd cycle. Conditions from (1)-(5) are placed on F to ensure that this can be done and, in particular, that it can be done simultaneously for all odd cycles. Obviously, (3) implies (2), (5) implies (4), while (c) implies (b) which implies (a), so strengthening any conditions yields trivial corollaries.

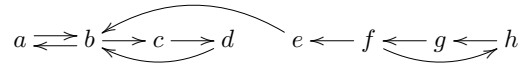
Theorem 2.2 ((3)+(4)+(a)) *A digraph G which is separated by a strongly even candidate F , has a kernel $K \supseteq F$.*

PROOF. We prove the claim by induction on the number of odd directed cycles in G . The basis case is covered by Richardson's theorem, [12], according to which a (finite) digraph with no odd directed cycle has a kernel. For the induction step we choose some arbitrary odd directed cycle C in G . We choose some $x \in F \cap N^+(C)$ and $L \in \min Sk(G, x)$, which exist because F is a candidate that separates G . We then consider $G' = G \setminus N^-[L]$ and $F' = F \setminus N^-[L]$. Obviously, $F \setminus F' \subseteq N^-[L]$, but in fact also $F \setminus F' \subseteq L$. To see this, recall that Lemma 1.7.(1) gives us even directed paths $P = P_{x,y}$ in G for all $y \in L$, i.e. $L \subseteq [G, x]_e$. On the other hand, if there is $z \in F \cap N^-(L)$ then there must be $w \in N^+(z) \cap L$, i.e. $w \in [G, z]_o$. But then $w \in [G, z]_o \cap [G, x]_e$, contradicting strong evenness of F .

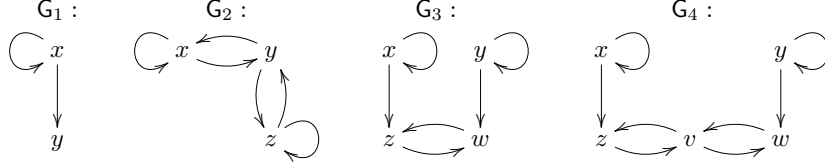
Now, if F' is a strongly even candidate that separates G' it follows by IH that G' has a kernel K' with $F' \subseteq K'$. In this case, $\langle G, L \rangle$ combined with $\langle G', K' \rangle$ gives a solver for G and $L \cup K' \in Kr(G)$ by Theorem 1.4. Since $F \setminus F' \subseteq L$ and $F' \subseteq K'$, we also get $F \subseteq L \cup K'$ so this completes the induction step. We show that F' satisfies the conditions of IH with respect to G' .

- (a) F' is obviously strongly even in G' , since F is such in G .
- (b) To show that F' separates G' , assume towards contradiction that it does not. Then there is an odd directed cycle $C' \subseteq G'$, for which there is a free vertex $y \in F \cap N^+(C')$ in G but not in G' , i.e., $y \in N^-[L]$. If $y \in L$ then C' could not be a directed cycle in G' , since $N^-(y) \subseteq N^-[L]$. So $y \in N^-(L)$ and this means that there is an odd directed path (single edge) from y to some $z \in N^+(y) \cap L$, i.e. $z \in [G, y]_o$. Since $z \in L \in \min Sk(G, x)$ we have, by Lemma 1.7.(1), that there is an even directed path $P = P_{x,z}$ in G , i.e. $z \in [G, x]_e$. But then $z \in [G, x]_e \cap [G, y]_o$, contradicting strong evenness of F .
- (c) To show that F' is a candidate, assume towards contradiction that it is not, i.e., for some $y \in F' : Sk(G', y) = \emptyset$. Choose some $M \in \min Sk(G, y)$. Since $M' = M \cap G' = M \setminus N^-[L]$ is independent and non-empty but is not a semikernel in G' , there exists an $r \in N^+(M') \cap G'$ with $N^+(r) \cap M' = \emptyset$. Since M is a semikernel G this means that there is some $z \in N^+(r) \cap M \cap N^-[L]$. If $z \in L$ then $r \in N^-(z) \subseteq N^-[L]$ and $r \notin G'$, so it must be the case that $z \in M \cap N^-(L)$. Let $w \in N^+(z) \cap L$. Then by Lemma 1.7.(1) there is a directed path $P = P_{x,w}$ in G of even length that is alternating on L . In particular, $w \in [G, x]_e$. But we have, also by Lemma 1.7.(1), a directed path $Q = Q_{y,z}$ in G that is even and alternating on M . Since $w \in N^+(z) \subseteq N^+(M)$ we have $w \notin M$ by M being independent and so either $Q_{y,w}$ (if w is on Q) or $Q;w$ is an odd directed path in G , giving us $w \in [G, y]_o$. This contradicts F being strongly even. \square

For instance, $\{a, e\}$ is a strongly even candidate separating the following G , which therefore has a kernel:



Example 2.3 Consider the following digraphs:



Both digraphs G_1 and G_2 have kernels by Theorem 2.2, witnessed by the strongly even candidates $\{y\}$. G_3 has no kernel and Theorem 2.2 fails for the candidate $\{z, w\}$, which is not strongly even. G_4 has a kernel since here the candidate $\{z, w\}$ is strongly even due to the presence of v .

In the following, we will consider conditions on a free subset F , Definition 2.1.(1), and some additional conventions will simplify the presentation. A vertex $x \in F$ is *free* for an odd (directed or undirected) cycle C , if $x \in N^+(C)$, i.e., if there is an $x' \in C$ with $x \in N^+(x')$ (since x is free, it is not on C). We denote it $Fr(G, x, x', C, F)$. Such an $x' \in C$ is *safe* (for C), denoted $Sa(G, x', C, F)$. A free vertex $x \in F$ is *critical* for x' on C , $Cr(G, x, x', C, F)$, if x' is the only vertex on C which has an out-neighbour in F . It is always intended that C (or a variant like C', C_i etc.) denotes an odd cycle. We often drop arguments if they are clear from the context or irrelevant, e.g. $Fr(x, x', C)$ is written when G and F are clear, and often it suffices with merely $Fr(x, C)$, $Sa(y', C)$, $Cr(x, C)$ (implicitly, only the primed arguments are on odd cycles).

For an $F \subseteq G$, an (*in*-)neighborhood function $f : F \rightarrow 2^G$ is one with $f(x) \subseteq N^-(x)$ for all $x \in F$. When $F \subseteq G$ is free, the *associated* neighborhood function returns the set of vertices on all directed and undirected odd cycles for which $x \in F$ is free:

$$f(x) = \{x' \mid \exists C, x' \in C : Fr(x, x', C, F)\} \text{ for all } x \in F. \quad (2.4)$$

The following lemma will prove quite useful; point (2) ensures that if x is free for an odd cycle C , the associated $f(x)$ contains a unique node from C .

Lemma 2.5 For any digraph G and any free set $F \subseteq G$ that separates G , let f be associated according to (2.4). Then, for all odd undirected cycles C in \underline{G} :

- (1) If $Fr(x, C)$, $y \in C$ and there is a $P = P_{x,y}$ in \underline{G} , then P meets $f(x) \cap C$.
- (2) If $Fr(x, C)$ then there is one and only one $x' \in C$ such that $x' \in f(x)$.
- (3) If distinct x', y' are safe for C and this is witnessed by $x \in N^+(x') \cap F$ and $y \in N^+(y') \cap F$, then every undirected path $P = P_{x,y}$ in \underline{G} meets either x' or y' .

PROOF. (1) Assume towards contradiction that $Fr(x, x', C)$, $y \in C$ and there is an undirected path $P = P_{x,y}$ in \underline{G} that does not meet $f(x) \cap C$. Let w be the first vertex on P that meets C and consider $P' = P_{x,w}$. Then P' does not meet C on any internal vertex and $w \notin f(x)$ by assumption. We have $x \in N^+(x')$

and $w \neq x'$, so there are undirected paths $A = A_{w,x'}$ and $B = B_{w,x'}$ in \underline{C} , with different parity (they are obtained from traversing C from w to x' along and against the direction of edges). Then $P'A;x$ and $P'B;x$ are undirected cycles from \underline{G} and one of them is odd, contradicting freeness of $x \in F$.

(2) Existence of x' is direct from definition of $Fr(x, C)$. For uniqueness, assume towards contradiction that there are two distinct $x', x'' \in C$ such that $x \in N^+(x') \cap N^+(x'')$. Since x' and x'' are distinct, there are undirected paths $A = A_{x',x''}$ and $B = B_{x',x''}$ in \underline{C} with different parities. So either $x; A; x$ or $x; B; x$ is an odd undirected cycle in \underline{G} , contradicting freeness of x .

(3) Let $Sa(x', C)$ and $Sa(y', C)$ hold, witnessed by $x \in N^+(x') \cap F$ and $y \in N^+(y') \cap F$ and assume towards contradiction that there is an undirected path $P = P_{x,y}$ in \underline{G} that does not meet x' or y' . By points (1) and (2) it follows that P does not meet C . Since $x' \neq y'$, there are undirected paths $A = A_{y',x'}$, $B = B_{y',x'}$ in \underline{C} that have different parity. So $P; A; x$ or $P; B; x$ is an odd undirected cycle from \underline{G} , contradicting freeness of x . \square

Theorem 2.6 ((1)+(5)+(b)) *A digraph G that is doubly separated by a free, perfect candidate $F \subseteq G$, has a kernel.*

PROOF. We proceed by induction on the number of odd directed cycles in G , with the basis case given by Richardson's theorem. For the induction step, we construct sequences $\langle G_i \rangle_{1 \leq i \leq n}$ and $\langle \bar{L}_i, x_i \rangle_{1 \leq i \leq n-1}$ with $G_1 = G$, such that:

$$\begin{aligned} & Fr(G_1, x_1, C_1) \text{ for some } C_1 \subseteq G_1 \text{ \& } L_1 \in \min Sk(G_1, x_1) \\ & G_{i+1} = G_i \setminus N^-[\bar{L}_i] \text{ for } 1 \leq i < n, \text{ where :} \\ & \forall 2 \leq i < n : Cr(G_i, x_i, C_i) \text{ for some } C_i \subseteq G_i \text{ \& } L_i \in \min Sk(G_i, x_i) \\ & \forall x \in F \cap G_n : \neg Cr(G_n, x, C) \text{ for all } C \subseteq G_n \end{aligned} \quad (2.7)$$

If there is an odd directed cycle C with only one free vertex x , as allowed by Definition 2.1.(b), then $C_1 = C$ and $x_1 = x$. All odd cycles C_i are directed, i.e. from G .

Properties 2.7 express an attempt to construct a solver for G . For the resulting G_n , we have three possibilities: either (i) it is empty, or (ii) it satisfies the assumption of IH or (iii) it does not satisfy the assumption of IH. In case (i) G_n has the kernel $K = \emptyset$ and in (ii) it has a kernel by IH, having fewer odd directed cycles than G . Then a solver for G is obtained by appending $\langle G_n, K \rangle$ to the sequence $\langle G_i, L_i \rangle_{1 \leq i \leq n-1}$ – and G has a kernel by Theorem 1.4. The rest of the proof shows that case (iii) can not happen.

G_n is sinkless by the observation following Definition 1.5. $F \cap G_n$ is free in G_n , since F is free in G , and it is a perfect candidate, because G_n is sinkless and because every sinkless induced subdigraph G' of G_n is also a sinkless induced subdigraph of G . To show that $F \cap G_n$ doubly separates G_n , we will use the following claim. For $1 \leq i \leq n$ we let $G_i^- = G \setminus G_i = \bigcup_{k=1}^{i-1} N^-[\bar{L}_k]$, so for $1 \leq i \leq j \leq n : G_i^- \subseteq G_j^-$.

Claim (A): For all $2 \leq i \leq n$ and $x \in G_i^-$, there is an undirected path $X = X_{x,x_1}$ in \underline{G} such that:

- (1) For all $q \in X \cap G_i$ there is an odd undirected cycle C_q in \underline{G} containing q and the vertex immediately preceding it on X .
- (2) If $x \in F$ then $N^-(x) \cap X \cap G_i = \emptyset$.

We prove it by induction on i . For the basis case, $x \in G_2^- = N^-[L_1]$, Lemma 1.7 gives a directed and hence undirected path $X = X_{x,x_1}$ in \underline{G}_2^- . So $X \cap G_2 = \emptyset$ and points (1) and (2) hold trivially.

For the induction step consider any $x \in G_{i+1}^- \setminus G_i^- = N^-[L_i]$. By Lemma 1.7 there is an undirected path $P = P_{x,x_i}$ that is in $N^-[L_i] \subseteq \underline{G}_{i+1}^-$. By (2.7) we have $Cr(G_i, x_i, x'_i, C_i, F)$ for some odd directed cycle $C_i \subseteq G_i$. Remember that if there is an odd directed cycle C in G with only one free vertex, as allowed by Definition 2.1.(b), then $C = C_1$. Since $i \geq 2$ this means that $C \not\subseteq G_i$. Since $C_i \subseteq G_i$, $C \neq C_i$ and the fact that G is doubly separated by F ensures that there is $y' \in C_i$ such that $y' \neq x'_i$ and y' is safe in G but not in G_i . Since y' is safe in G , there is some $y \in F \cap N^+(y')$ such that $Fr(G, y, y', C_i)$, and since y' is not safe in G_i , we have $y \notin G_i$, that is $y \in G_i^-$. By IH, there is an undirected path $R = R_{y,x_1}$ in \underline{G} satisfying conditions (1) and (2).

To fill the gap between the paths P and R , we consider an undirected path $Q = Q_{x_i,y}$ in \underline{G} , passing through x'_i and y' (in that order) with every internal vertex of Q lying on C_i . Let z be the first vertex on P that is also on Q (possibly, $z = x_i$ or $z = y$), and set $P' = P_{x,z}$, $Q' = Q_{z,y}$ and $S = P'Q'$.

The undirected path S (or its prefix) will start the desired path X . We argue that it satisfies condition (1). It might meet G_{i+1} , say at w , but since $y \in G_i^-$, $P' \subseteq P \subseteq G_{i+1}^-$ and $x'_i \in G_{i+1}^-$, we have $w \in Q' \setminus \{y, x_i, x'_i\}$, i.e., w is on C_i . Let v be the vertex immediately preceding w on S . Since P' is in \underline{G}_{i+1}^- and also $x'_i \in G_{i+1}^-$ it follows that S meets C_i for the first time at some vertex from \underline{G}_{i+1}^- . In particular, w is not the first vertex from S that is on C_i . Since the internal part of Q is in \underline{C}_i , it further follows that v is on C_i . So C_i is the desired odd cycle proving condition (1) for S .

Let p be the first vertex from S (starting from x) that is on R and let $R' = R_{p,x_1}$. Now, assume towards contradiction that $p \in G_{i+1}$. Then, since P is in \underline{G}_{i+1}^- and Q is in \underline{C}_i except for $\{x_i, y\} \subseteq G_{i+1}^-$, we have $p \in C_i$. So, in particular, R meets C_i . Let p' be the first vertex from R (starting from y) that is on C_i . Since $y' \in G_i$ we know from IH, condition (2), that $p' \neq y'$. It follows that there are undirected paths $A = A_{p',y'}$, $B = B_{p',y'}$ of different parity and so either $R_{y,p'}A_{p',y'}; y$ or $R_{y,p'}B_{p',y'}; y$ is an odd undirected cycle, contradicting freeness of $y \in F$. (Since we pick p' to be the first vertex from R that is on C_i we know that $R_{y,p'}$ and A, B does not meet internally.) It follows that $p \notin G_{i+1}$.

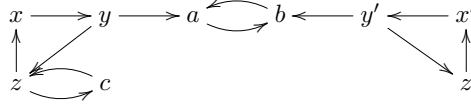
Consequently, for all $q \in R' \cap G_{i+1}$ there is some $r \in R'$ preceding it on R' . Then, by IH, there is an odd undirected cycle C_q in \underline{G} containing both q and r (remember that $G_{i+1} \subseteq G_i$). It follows that claim (1) holds for $X = S_{x,p}R'$.

To show point (2), assume towards contradiction that there is $x' \in N^-(x) \cap G_{i+1} \cap X$. Let x'' be the vertex preceding x' on X . Then by (1) we have an odd undirected cycle $C_{x'}$ in \underline{G} containing both x'' and x' . Now, let t be the first vertex on X that is on $C_{x'}$ (possibly, $t = x''$). Since x'' is on $C_{x'}$, $t \neq x'$. So there are undirected paths $A = A_{t,x'}$, $B = B_{t,x'}$ of different parity, both in $C_{x'}$. So one of $X_{x,t}A; x, X_{x,t}B; x$ is an odd undirected cycle in \underline{G} , contradicting freeness of $x \in F$. This completes the proof of Claim (A).

Assume now that G_n is not doubly separated by $F \cap G_n$. We have $\forall x \in F \cap G_n : \neg Cr(G_n, x, C)$ for all $C \subseteq G_n$ by construction (2.7), so from the assumption that G_n does not satisfy IH it follows that here must be at least one C_n in G_n with two distinct vertices $x', y' \in C_n$ that are safe in G but not in G_n . This means that there are free vertices $x \in N^+(x') \cap G_n^-$ and $y \in N^+(y') \cap G_n^-$. $x' \neq y'$, so $y' \notin f(x)$ and $x' \notin f(y)$, by Lemma 2.5.(2). By Claim (A), there are undirected paths $X = X_{x,x_1}$ and $Y = Y_{y,y_1}$ in \underline{G} that do not meet $N^-(x) \cap G_n$ and $N^-(y) \cap G_n$, respectively. So X does not meet x' and Y does not meet y' , and by Lemma 2.5.(1), neither X meets y' nor Y meets x' . It follows that there is an undirected path $P = P_{x,y}$ in \underline{G} that does not meet $\{x', y'\}$. But this contradicts Lemma 2.5.(3). This contradiction shows that G_n satisfies the assumptions of IH, i.e., case (iii) can not happen. This completes the proof. \square

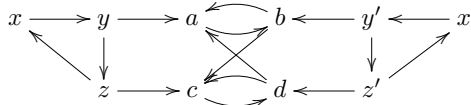
Obviously, if a digraph consists of disconnected components, Theorem 2.6 can be applied to each component separately (allowing one odd directed cycle with only one safe vertex in each component.)

Example 2.8 (a) *Theorem 2.2 does not apply to the following G , since no separating set – neither $\{a, b, c\}$, $\{a, b\}$ nor $\{c, b\}$ – is strongly even:*



It has a kernel by Theorem 2.6 since $\{a, b, c\}$ is a free, perfect candidate, doubly separating G . Although there is no local kernel that includes both a and b , things work out thanks to the additional free vertex c . Removing it, would leave a digraph without any kernel, so this illustrates also that, in general, at most one odd directed cycle can be allowed to have only one free vertex, as in Definition 2.1.(b).

(b) *Freeness of a doubly separating, perfect candidate F in Theorem 2.6, can not be weakened by requiring merely that no vertex in F lies on an odd directed cycle. In the following digraph, $\{a, b, c, d\}$ satisfies such a requirement:*



Still, there is no kernel, since it is not possible for both one of $\{a, c\}$ and one of $\{b, d\}$ to be in the same semikernel (which is needed to “break” both odd directed cycles).

Freeness and its consequences in Lemma 2.5 seem crucial for Theorem 2.6. Quite generally, the essential conditions for kernel existence seem to concern parity, as exemplified already by Richardson's theorem. Although imposing such conditions on the *undirected* paths in the underlying graph may appear unnecessarily restrictive, the above theorem shows that it may be useful and, perhaps, even necessary.

This is further illustrated by our last theorem. It does not make any assumption about the existence of semikernel, replacing it by a structural condition: in a sinkless \mathbf{G} , strongly separated by a free set F , at least one local kernel exists. Additional parity condition (evenness of F) ensures that one can keep constructing a solver for \mathbf{G} . The proof involves a new auxiliary concept of reachability in $\underline{\mathbf{G}}$, relatively to a given neighborhood function. For a digraph \mathbf{G} , $F \subseteq G$ and any neighborhood function $f : F \rightarrow 2^G$, we denote by

$[\underline{\mathbf{G}}, x)^{-f}$ the set of all vertices $y \in G$ such that $\underline{\mathbf{G}}$ contains an undirected path $P = P_{x,y}$ that does not meet $f(x)$ internally.

The requirement that $\text{int}(P) \cap f(x) = \emptyset$ means, in particular, that if $y \in f(x)$ and there is an undirected path $P = P_{x,y}$ that does not meet $f(x)$ before reaching y , then $y \in [\underline{\mathbf{G}}, x)^{-f}$. What makes this notion of reachability useful is that it allows us to define for any $F \subseteq G$ and any neighborhood function $f : F \rightarrow 2^G$, the following strict partial order on F :

$$x <_f y \text{ iff } (y \in [\underline{\mathbf{G}}, x)^{-f} \wedge x \notin [\underline{\mathbf{G}}, y)^{-f}),$$

Fact 2.9 *For any digraph \mathbf{G} and neighborhood function f , the relation $<_f$ is a strict partial order.*

PROOF. The relation is clearly irreflexive. To prove transitivity, assume $x <_f y <_f z$ and let $P = P_{x,y}$, $Q = Q_{y,z}$ be the undirected paths witnessing to this fact (so they are in $\underline{\mathbf{G}}$). First we prove that $z \in [\underline{\mathbf{G}}, x)^{-f}$. Assume it is not. Letting p be the first vertex on P that is on Q we obtain the undirected path $R = P_{x,p}Q_{p,z}$. Then R must intersect $f(x)$ internally and as P witnesses to $x <_f y$, it follows that Q must intersect $f(x)$ say, at q . We have $q \in \text{int}(R) \cap Q$, $\text{int}(Q) \cap f(y) = \emptyset$ and $y \notin f(y)$, so $Q_{y,q} \cap f(y) = \emptyset$ meaning that $Q_{y,q}; x$ witnesses to $x \in [\underline{\mathbf{G}}, y)^{-f}$, contradicting $x <_f y$.

To prove $x \notin [\underline{\mathbf{G}}, z)^{-f}$, assume towards contradiction that $x \in [\underline{\mathbf{G}}, z)^{-f}$ is witnessed by $Z = Z_{z,x}$. Let q be the first vertex on Q that is on Z . Then $S = Q_{y,q}Z_{q,x}$ is an undirected path from y to x and, since $x \notin [\underline{\mathbf{G}}, y)^{-f}$, there must be some $r \in f(y)$ on $\text{int}(Z)$ (since none such exists on Q , which witnesses to $z \in [\underline{\mathbf{G}}, y)^{-f}$). But then $Z_{z,r}; y$ gives $y \in [\underline{\mathbf{G}}, z)^{-f}$, contradicting $y <_f z$. \square

Consequently, for any neighborhood function f on F and any $<_f$ -maximal $x \in F$, we have:

$$\forall y \in F : y \in [\underline{\mathbf{G}}, x)^{-f} \rightarrow x \in [\underline{\mathbf{G}}, y)^{-f}. \quad (2.10)$$

This helps to prove the following lemma.

Lemma 2.11 *Given any \mathbb{G} separated by a free $F \subseteq G$, let f be given by (2.4). If $[\underline{\mathbb{G}}, y]^{-f}$ is strongly separated by F for all $y \in F$ then for every $<_f$ -maximal $x \in F$:*

- (1) $[\underline{\mathbb{G}}, x]^{-f}$ induces a bipartite subdigraph.
- (2) For every $y \in F$ and every odd undirected cycle C in $\underline{\mathbb{G}}$: if $Fr(y, C)$ and $y \in [\underline{\mathbb{G}}, x]^{-f}$, then $Fr(x, C)$.

PROOF. (1) We show that any $<_f$ -maximal $x \in F$ satisfies the claim. Assume towards contradiction that there is an odd undirected cycle C in the underlying subgraph $[\underline{\mathbb{G}}, x]^{-f}$. From the fact that F strongly separates $[\underline{\mathbb{G}}, x]^{-f}$ it follows that this subdigraph is loopless, i.e. all odd cycles contain at least 3 vertices. Then $C \cap f(x) = \emptyset$, for otherwise x would be on an odd undirected cycle in $\underline{\mathbb{G}}$, contradicting its freeness. To see this, note that if $x' \in C \cap f(x)$ and z is a first node from C on some undirected path $P = P_{x, z'}$ (in $\underline{\mathbb{G}}$) witnessing to $z' \in [\underline{\mathbb{G}}, x]^{-f}$ for some $z' \neq x'$ on C , then $z \neq x'$. So there are two undirected paths $A_{z, x'}$ and $B_{z, x'}$, both in $\underline{\mathbb{G}}$, of different parities, giving an odd undirected cycle $P_{x, z} A_{z, x'}; x$ or $P_{x, z} B_{z, x'}; x$. Now, since F strongly separates $[\underline{\mathbb{G}}, x]^{-f}$ there are distinct $y', z' \in C$ such that $Sa([\underline{\mathbb{G}}, x]^{-f}, y', C)$ and $Sa([\underline{\mathbb{G}}, x]^{-f}, z', C)$, witnessed by $y, z \in F$ with $y \in N^+(y')$ and $z \in N^+(z')$. Since C is in $[\underline{\mathbb{G}}, x]^{-f}$, there are undirected paths $P = P_{x, y'}$, $Q = Q_{x, z'}$ in $[\underline{\mathbb{G}}, x]^{-f}$. Also, since $C \cap f(x) = \emptyset$, there are undirected paths $P' = P'_{x, y}$ and $Q' = Q'_{x, z}$, also in $[\underline{\mathbb{G}}, x]^{-f}$. P' can be taken as $P; y$ or, if that is a walk (due to y occurring on P), as the path $P_{x, y}$ (similarly for Q'). Since x is $<_f$ -maximal, (2.10) implies that there are undirected paths $U = U_{y, x}$ and $V = V_{z, x}$, contained in $[\underline{\mathbb{G}}, y]^{-f}$ and $[\underline{\mathbb{G}}, z]^{-f}$, respectively. It follows from Lemma 2.5.(1) that neither U nor V meets C on any internal vertex. Also, x is certainly not on C since it is free. It follows that there is an undirected path $W = W_{y, z}$ in $\underline{\mathbb{G}}$ that meets neither y' nor z' (obtained from U and V). This contradicts Lemma 2.5.(3).

(2) Assume contrapositively that there is an odd undirected cycle C in $\underline{\mathbb{G}}$ and some $y \in F$ with $Fr(y, C)$, $y \in [\underline{\mathbb{G}}, x]^{-f}$ and $\neg Fr(x, C)$. Since $y \in F$, it does not lie on any odd undirected cycle, so $y \notin f(x)$. Hence there are undirected paths from x to every vertex on C that do not meet $f(x)$ internally (obtained by extending such a path going to y). Then C is in $[\underline{\mathbb{G}}, x]^{-f}$, but this contradicts point (1). \square

Theorem 2.12 *For a sinkless \mathbb{G} separated by a free, even $F \subseteq G$, let f be given by (2.4). If F strongly separates $[\underline{\mathbb{G}}, x]^{-f}$ for every $x \in F$, then \mathbb{G} has a kernel.*

PROOF. We proceed by induction on the number of odd directed cycles in \mathbb{G} . The basis case is Richardson's theorem. For the induction step we choose some $x \in F$ that is maximal with respect to $<_f$. Let $\mathbb{G}' = [\underline{\mathbb{G}}, x]^{-f}$. We prove first:

Claim (B): \mathbb{G}' is sinkless.

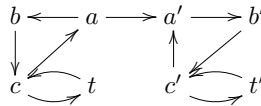
To show it, assume contrapositively that there is some $y \in \mathbb{G}'$ such that $N^+(y) \cap$

$G' = \emptyset$. Now, since $y \in [\underline{G}, x)^{-f}$, there is some undirected path $P = P_{x,y}$ in \underline{G} that does not meet $f(x)$ internally. Since \underline{G} is sinkless we know that $N^+(y) \neq \emptyset$. So then there must be some $z \in N^+(y)$ such that $P; z$ does meet $f(x)$ internally. It follows that $y \in f(x)$. But then $x \in N^+(y) \cap G'$ so y is not a sink in G' after all. The contradiction establishes Claim (B).

By Lemma 2.11.(1), \underline{G} is bipartite, so let B_1, B_2 be a bipartition. Assuming w.l.o.g. $x \in B_1$, we show that $B_1 \in Sk(\underline{G}, x)$. Indeed, B_1 is independent in \underline{G} , so assume towards contradiction that it is not locally absorbing, i.e., for some $y \in N^+(B_1) : N^+(y) \cap B_1 = \emptyset$. Now, since $f(x) \subseteq N^-(x)$ it follows that $f(x) \subseteq B_2$, so $N^+(B_1) \subseteq [\underline{G}, x)^{-f}$. In particular, $y \in B_2 \subseteq G'$. But we have $N^+(y) \cap G' \cap B_1 = \emptyset$ and, obviously, $N^+(y) \cap G' \cap B_2 = \emptyset$, so y is a sink in G' , contradicting Claim (B).

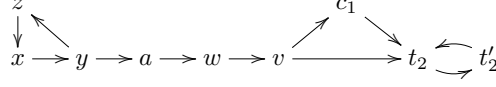
Let $L \in minSk(x)$ be such that $L \subseteq B_1$ and consider its completion \bar{L} . We obtain a solver for \underline{G} if we can establish that the IH applies to $\underline{G}_2 = \underline{G} \setminus N^-(\bar{L})$. $F' = F \cap \underline{G}_2$ is a free, even set in \underline{G}_2 , since F is such in \underline{G} , so the IH fails to apply to \underline{G}_2 only if F' does not separate \underline{G}_2 or does not strongly separate $[\underline{G}_2, y)^{-f}$ for some $y \in F'$. This can be the case only if there is an odd undirected cycle C in \underline{G}_2 with an $y' \in C$ such that there is $y \in (F \cap N^+(y')) \setminus \underline{G}_2$. Since C is in \underline{G}_2 it follows that $y \in N^-(\bar{L})$ (rather than \bar{L} , which would mean that $N^-(\bar{L}) \cap C \neq \emptyset$ and thereby contradict C being in \underline{G}_2). It follows that there is $z \in N^+(y) \cap \bar{L}$ and by Lemma 1.7.(2) there is an even directed path $P = P_{z,v}$, alternating on \bar{L} , where $v \in L \subseteq B_1$. Since $y \in F$, $Fr(y, C)$ and $\neg Fr(x, C)$ (since C is in \underline{G}_2), we know from Lemma 2.11.(2) that $y \notin G'$. Now, from the directed path $P_{z,v}$ we obtain an alternating directed path $P' = P'_{y,v}$ such that every internal vertex from P' is on P . (P' can be taken as $y; P$ or, if that is not a path, as $P_{y,v}$.) We have $v \in G'$ and $y \notin G'$ so $int(P')$ must meet $f(x)$ at some vertex x' . If it does not, we obtain $y \in [\underline{G}, x)^{-f} = G'$ from the existence of an undirected path from x to y not passing through $f(x)$ (obtained from some such path $U = U_{x,v}$ and P' traversed from v towards y). Now, $f(x) \subseteq N^-(\bar{L})$ so from the fact that P' is alternating on \bar{L} and $y \in N^-(\bar{L})$ it follows that $P'_{y,x'}$ has even length. So we obtain the directed odd path $P'_{y,x'}; x$ in \underline{G} , contradicting the fact that F is an even set. \square

Example 2.13 (a) *In the following digraph \underline{G} , the existence of a kernel is witnessed by the free, even set $\{t, t'\}$, with $[\underline{G}, t)^{-f} = \emptyset = [\underline{G}, t')^{-f}$. (Neither Theorem 2.2 nor 2.6 is applicable.)*

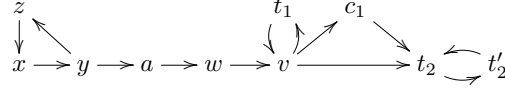


(b) *Strong separation (of odd undirected cycles) can't be weakened to double separation (of odd directed cycles). In the following \underline{G}' , $[\underline{G}', a)^{-f}$ is doubly separated (trivially, since it has no directed odd cycle) and the free, even set $\{a\}$ separates*

G' , but G' has no kernel.

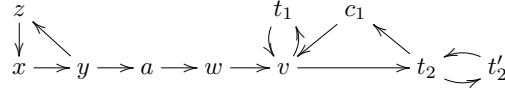


(c) Neither can we drop the evenness condition, as shown by the following G'' :



$F = \{a, t_1, t'_2\}$ is free, separating and it strongly separates the cone $[\underline{G}'', a)^{-f}$. It is not even, however, having an odd directed path, e.g., $a - t'_1$, and G'' has no kernel, as can be seen by checking that $Sk(a) = \emptyset$.

(d) The following digraph, with a directed odd cycle instead of the undirected one in G'' , has a kernel – not by Theorem 2.12, but by 2.6. It has no even, strongly separating set, but $\{a, t_1, t'_2\}$ is a free, doubly separating, perfect candidate.



Theorem 2.12 gives trivially the following corollary.

Corollary 2.14 ((1)+(2)+(c)) *A sinkless digraph G , that is strongly separated by a free, even $F \subseteq G$, has a kernel.*

References

- [1] Marc Bezem, Clemens Grabmayer, and Michał Walicki. Expressive power of digraph solvability. *Annals of Pure and Applied Logic*, 163(2):200–212, 2012.
- [2] Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.
- [3] Nadia Creignou. The class of problems that are linearly equivalent to satisfiability or a uniform method for proving np-completeness. *Theoretical Computer Science*, 145:111–145, 1995.
- [4] Yannis Dimopoulos and Vangelis Magirou. A graph theoretic approach to default logic. *Information and Computation*, 112:239–256, 1994.
- [5] Yannis Dimopoulos, Vangelis Magirou, and Christos H. Papadimitriou. On kernels, defaults and even graphs. *Annals of Mathematics and Artificial Intelligence*, 20:1–12, 1997.

- [6] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170(1-2):209–244, 1996.
- [7] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
- [8] Aviezri S. Fraenkel. Combinatorial game theory foundations applied to digraph kernels. *Electronic Journal of Combinatorics*, 4(2), 1997.
- [9] Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.
- [10] John R. Isbell. On a theorem of Richardson. *Proceedings of the AMS*, 8(5):928–929, 1957.
- [11] Victor Neumann-Lara. Seminúcleos de una digráfica. Technical report, Anales del Instituto de Matemáticas II, Universidad Nacional Autónoma México, 1971.
- [12] Moses Richardson. Solutions of irreflexive relations. *The Annals of Mathematics, Second Series*, 58(3):573–590, 1953.

Chapter 5

Paper C: Propositional Discourse Logic

This paper has been submitted to Synthese

Propositional Discourse Logic

Sjur Dyrkolbotn and Michał Walicki
Institute of Informatics
University of Bergen, Norway

Abstract

A novel normal form for propositional theories underlies the logic PDL, which captures some essential features of natural discourse, independent from any particular subject matter and related only to its referential structure. In particular, PDL allows to distinguish vicious circularity from the innocent one, and to reason in the presence of inconsistency using a minimal number of extraneous assumptions, beyond the classical ones. Several, formally equivalent decision problems are identified as potential applications: non-paradoxical character of discourses, admissibility of arguments in argumentation networks, propositional satisfiability, and the existence of kernels of directed graphs. Directed graphs provide the basis for the semantics of PDL and the paper concludes by an overview of relevant graph-theoretical results and their applications in diagnosing paradoxical character of natural discourses.

1 Introduction

The natural discourse, in idealized form, can be seen as a network of cross-references, statements that assert or deny each other. Some statements assert external facts. What should count as fact, however, might be a very contentious issue, in which case it seems safest to regard a fact as nothing more than the statement expressing it. The idealization amounts to abstracting from the specificity of facts, which depend on the actual subject matter, and concentrating on the referential structure, the mutual dependencies between the involved statements. Facts can be then taken as statements which are considered true, independently from any other statements.¹

A statement is considered true if what it claims is accepted to hold and, typically, all statements of a discourse obtain (at least possible) truth-values corresponding to the status of their claims. Occasionally, however, a discourse malfunctions, resulting in the impossibility of assigning any truth-value to some of its statements. The liar and other standard paradoxes provide obvious examples, but typical situations tend to be more complex. If, for instance, Frank

¹Of course, we are *not* saying that this is what facts *are*, only that they can be treated in this way, without impairing correctness of the formal model. Our model works unchanged also when no such facts are available.

asserts the opposite of John, John asserts the same as Paul while Paul expresses agreement with Frank then, at first, the situation may be unclear. But a moment of reflection shows that *regardless* of the actual subject matter, no one can be right and no one can be wrong. If Frank is right, then John is wrong, but then Paul is wrong so Frank must be wrong too. In a similar way all possibilities end up undermining themselves and from this we are forced to conclude, by logic alone, that the discourse has malfunctioned.

We are concerned here with diagnosing the disease, the inability of a classical, propositional discourse to express something that can be consistently evaluated as true or false. The *mere diagnosis* of such cases is of independent interest and importance, because it is only from a proper diagnosis that one can begin to analyze *why* things went awry in a particular case. For example, a lawyer cross-examining a witness looks first for inner contradictions, and if he finds some, these provide the cue as to which claims deserve particular scrutiny. While the answer to the question of why anomalies appear may be specific to the domain and circumstances of the discourse, the question of coherence does not hinge on any such extraneous elements. Certainly, agreement with facts (undisputed statements) is mandatory but, in general, does not suffice for ensuring coherence of the discourse. This problem deserves a separate treatment.

The referential structure of discourses will be represented as directed graphs, which capture many essential properties, circularity in particular, in a simple and intuitively appealing way. The associated logic PDL allows to localize malfunctioning (sub)discourses and gives precise insight into the structural causes of the anomalies, in particular, the vicious circularity. The reported results can be summarized as follows:

Section 2: Deciding if a propositional discourse hides any anomalies or else can be consistently evaluated, has several, formally equivalent decision problems: stable extensions in argumentation networks, the existence of kernels in digraphs and satisfiability of propositional theories. Collecting these equivalences (some known earlier only separately) unifies various fields, simplifying also many proofs.

Section 3: Local kernels (generalizing kernels) of digraphs provide semantics for arbitrary, also inconsistent discourses, and the logic is not explosive, allowing to establish valid consequences also when the discourse is inconsistent, for instance, identifying consistent subdiscourses. Local kernels of a graph G can be captured logically using a simple axiomatization of G in Łukasiewicz's logic $L3$. In particular, we show that kernels correspond to consistent assignments in classical logic while local kernels correspond to consistent assignments in $L3$. We note, however, some shortcomings of $L3$, the central one relating to the difficulties with treating the third value (paradox) in the same way as the two classical ones. Paradox seems to be admitted into a discourse only when it is not possible to find any classical truth assignment. In this sense it has a necessary character, as opposed to the classical values, which must be only possible, for the discourse to be meaningful. In $L3$, as is typical in non-modal logics, questions of possibility (consistency) can only be addressed by some indirect

means. We therefore introduce *propositional discourse logic*, PDL. It allows to decide possible truth-values of complex formulae over arbitrary discourse graphs. When graphs correspond to syntax trees, the logic becomes classical propositional logic. By the equivalences from Section 2, it gives the means for deciding:

- paradoxical character of discourses;
- satisfiability of propositional theories;
- acceptability and admissibility of arguments in argumentation networks;
- the existence of (and membership in) local kernels in digraphs.

Meaningful information will be deduced also from inconsistent discourses, but we use only two truth-values with connectives evaluated by the standard rules. Avoiding any extraneous assumptions, which can affect what counts as an anomaly, should make the resulting diagnosis no more dubious than the classical intuitions on which it is based. In light of this, we dare call our logic *essentially* classical.

Section 4: PDL is based on the concept of local kernel and kernel-theory provides valuable results for the analysis of discourses. We cite a series of such results and show their applications in diagnosing problematic cases. At the same time, these results make precise many intuitions, in particular, concerning (vicious) circular reference.

More involved proofs, not included in the text, can be found in the appendix.

Before we embark on the technical parts we present some of the intuitions about natural discourse that we seek to capture. They may enhance understanding of and motivate the technical parts, but are not necessary and the reader interested only in the latter, can go directly to Section 2.

Elements of natural discourse

Natural discourse is open-ended, it has no discernible beginning nor end, there is no period, only “...” preceding and following every statement. Yet, every now and then we have to stop and consider some part of it, some *relative* totality.

Consider the series of consecutive statements, with some longer suspensions, marked by the horizontal lines, at which the possible truth-values of the statements made so far are evaluated, giving the results in the respective column:

a	...The next statement is false...	?	⊥	?	0	0	⊥
b	The next statement is false...	?	⊥	?	1	1	⊥
c	The first statement (a) is false ...		⊥				
c'	... and, by the way, so is the next one...	?		?	0	0	⊥
d	The next statement is false...	?		?	1	1	0
e	The previous statement is false...				0	0	1
f	The previous statement is false...				1		
f'	... and so is this one...					⊥	0

(1.1)

At point *b*, the truth-values of the two statements are unclear. If the person suspends the voice after *c*, we may think that he has said the last word, making no

sense. But if he continues as indicated, the discourse becomes again potentially meaningful at c' , while at e and f all statements can even be assigned classical truth-values. At f' , however, it dissolves again. At this point it exemplifies all phenomena we will address, so there is no need to extend it.

Open-endedness means, in particular, that many statements are *undetermined* at the moment they are made; their truth-value is not exactly known, either because one can not provide their ultimate justification in terms of “hard facts” or other statements, or else because they address future contingents, depending on events or statements which have not yet been clarified. “Snow is white” may be unproblematic, but is a rather special case, representative at most of a special class. The pair $d - e$ illustrates well this modal element. In the absence of any additional information, there seems to be no reason to choose between d and e , and keeping both possibilities is the most natural, not to say ethical, way. Under special circumstances, such sets of possibilities can be narrowed to unique truth-values, resolving indeterminacy into certainty. Yet even with the simplest, empirical claims, one does not have the capacity to verify them all against their eventual justification basis. Most statements are therefore accepted on the basis of other statements, in many situations on the basis of faith, in others on the basis of some defaults or coherence.

Importantly, even when empirical evidence is insufficient, truth-values of many statements can be intuitively ascertained. This does not imply any idealism nor any reduction of truth to coherence, only that it might be difficult to argue with one who does not agree that if e is wrong then d is right. It seems hard to deny that internal coherence is an indispensable feature of meaningful discourse and equally hard to deny that our *approximations* of truth often falter on exactly this point. Most significantly, accepting some statements on the basis of (empirical) facts does not in any way exclude accepting others on the basis of internal coherence. We will show how to draw a line separating these two kinds in any propositional discourse.

The inherent indeterminacy and possible reliance on “mere” coherence reflect the holistic character of such cross-referential networks. Due to mutual dependencies, the discourse can not always be evaluated assigning step by step correct values to single statements. If c' is right, then d must be wrong, but this may, in turn, depend on e, c , etc.² Consequently, an anomaly is an accident of *the whole* discourse, not of any particular among its statements. Just like no particular statement among $a - b - c$ is wrong, there is no single culprit among all statements $a - f'$ – they malfunction only *together*. Certainly, easiest to identify are single paradoxical statements, like the liar, but they represent only special cases of discourses, namely, those limited to a single statement. There is no need to distinguish such special cases from more complex ones, like $a - b - c$ or $a - f'$, once we accept the anomaly as an holistic phenomenon of the totality of a discourse. The meaning of this, possibly controversial claim, should be

²When unfolded in time as a sequence of consecutive statements, such a holistic network of mutual dependencies gives rise to anaphoric and cataphoric references, yielding the non-monotonic character of the discourse. But since non-monotonicity appears thus only as a special, temporal view of mutual dependencies, we will not devote it separate treatment.

transparent in view of the just mentioned examples. It is also in line with more recent developments. In the infinitary Yablo’s paradox [32], for instance, no single statement is paradoxical, taken on its own. Saying, on the other hand, that every one of them is, requires to consider them in conjunction with all others.³

Classical logic expresses an extreme desire for coherence – in the presence of any inconsistencies, discourses simply “explode”. The process of reasoning *towards* consistency is certainly very natural and classical logic captures many of its essential features. What is not natural, however, is the exploding, and as we shall see, not even classical logic needs it. The classical way of reasoning makes perfect sense for discourses that are, when viewed as a whole, inconsistent. If this isn’t immediately obvious, consider how one concludes paradoxicality of, say, $a - b - c$ in (1.1). Trying $a = \mathbf{1}$ leads to a contradiction and so does $a = \mathbf{0}$ – in both cases, using classical means alone. PDL will allow such classical reasoning in the presence of inconsistency, without any deductive explosion.

This is made possible because, as in natural discourse, contradictions are regarded in the first instance as only *locally* significant. They may render the possible truth or falsity of some statements unclear, but do not pollute all statements in the discourse. A person contradicting himself at some point, may say valid things in the next moment.

For the open-ended natural discourse, a local analysis is inevitable since every totality is only relative. Moreover, what is taken as the *actual* totality bears a crucial influence on the truth-values of the involved statements. If, at point f' , we view only the last three statements $d - e - f'$, it is consistent. But if we go all the way back to a , then there is no way of assigning, in a consistent way, truth-values to all statements – the discourse is paradoxical. Likewise, the discourse which is inconsistent after $a - b - c$, can acquire a promise of potential meaning at c' , and even become fully meaningful, allowing classical distribution of truth-values among all its statements at e or f .

The inconsistency of a discourse D does not prevent us from deducing useful information about its particular statement x . For instance, there can be *subdiscourses* that are consistent, and if x is true or false in some of these, then we might want to know – after all, D itself is just a snapshot of some larger totality. Its choice seems, at least in part, guided by the desire to avoid inconsistency. So if we can do better by looking at smaller or larger discourses, why not? A counter-argument might be that it is unclear where to draw the line. Admitting *any* subdiscourse containing x might be too permissive. In discourse (1.1), for instance, stopping after d is hazardous. It refers to the later statement e , so a judgment about d commits us also to a specific judgment about e and such necessary dependencies should be taken into account.

The logic PDL allows us to do this, capturing a natural condition that sepa-

³“Holism” does not refer here to any universal totality of everything, settling all particular issues in a final way. We do not deny its possibility but neither know where to find it nor attempt to do it. Locally meaningful, relative totalities, on the other hand, appear every time we conduct a conversation and our holism relies only on such relative totalities of actual interest. Its essential aspect is possible lack of compositionality: a series of meaningful and consistent statements may yield a paradoxical totality.

rates the coherent, acceptable subdiscourses from the others. Loosely expressed, the condition says that *a subdiscourse is an acceptable totality only if one can make a consistent assertion about the truth of its statements that cannot be disproven by extending it*. Such a subdiscourse admits a complete evaluation of its statements – a proof, if you like, of its admissibility. The condition expresses its robustness – whatever happens to the rest of the discourse does not affect the truth-values within the subdiscourse. “Hard facts”, statements accepted as true independently from the rest of the discourse, provide a basic example. Much more involved examples, involving circular and even ungrounded subdiscourses, will be given once we have defined precisely the necessary notions.

2 Formalization

A discourse, over a set of propositional variables Σ , is a finite propositional theory consisting of a series of equivalences

$$x \leftrightarrow \bigwedge_{y \in I_x} \neg y \quad (2.1)$$

where each $I_x \subseteq \Sigma$ is finite and each $x \in \Sigma$ occurs exactly once on the left of such an equivalence.⁴ We use the convention that the right-hand side is **1** when $I_x = \emptyset$. Rendered in this pattern, discourse (1.1) becomes:

$$\begin{array}{ll} a & \leftrightarrow \neg b \\ b & \leftrightarrow \neg c' \\ c' & \leftrightarrow \neg a \wedge \neg d \end{array} \quad \begin{array}{ll} d & \leftrightarrow \neg e \\ e & \leftrightarrow \neg d \\ f' & \leftrightarrow \neg e \wedge \neg f' \end{array} \quad (2.2)$$

The variable on the left of each equivalence acts as the unique identifier of the actually pronounced statement, occurring on its right. The intuitive incoherence of a discourse, the impossibility of assigning truth-values to all its statements (variables on the left), corresponds exactly to the inconsistency of such a theory. Variants of this format were implicit in [7, 20], and elaborated in [30]. As we will see in Corollary 3.22, it does not limit the expressive power, providing a normal form for propositional theories.⁵

The consistency of discourses turns out to be equivalent to two other problems: the existence of stable extensions in argumentation networks and the existence of kernels in digraphs.

⁴Many results that will be presented hold also for infinite discourses (theories) and infinitary logic (allowing infinite I_x 's), but we are addressing primarily the finite and finitary case.

⁵One can think of a propositional letter appearing on the left of an equivalence as naming the complex formula that appears on the right. The equivalences become then instances of Tarski's T-schema, formulated in propositional logic

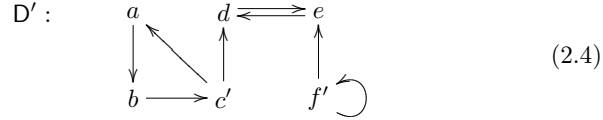
2.1 Argumentation networks

Consider the discourse (1.1) with all statements claiming falsity or truth of others replaced by arguments contesting validity of other arguments.

a	...The next argument is wrong...	?	\perp	?	0	0	\perp
b	The next argument is wrong...	?	\perp	?	1	1	\perp
c	The first argument (a) is wrong ...		\perp				
c'	... and, by the way, so is the next one...	?			0	0	\perp
d	The next argument is wrong...	?			1	1	0
e	The previous argument is wrong...				0	0	1
f	The previous argument is wrong...				1		
f'	... and so is this one...					\perp	0

(2.3)

One seldom encounters arguments like f' in practice, but $a - b - c$ is quite possible.⁶ Argumentation theory, at least in its AI version arising from [16], addresses coherence of such argumentation networks. The analogy between the two examples is obvious and it illustrates the general equivalence between the argumentative and discursive networks. Claims of falsity of other statements act exactly as the arguments attacking other arguments: if a claims falsity of b , and b turns out to be true, then a is false; while if an argument a attacks b and b turns out to be valid/accepted then a becomes defeated/invalidated. We can therefore conflate these two scenarios and represent the fact that a claims falsity of (respectively, attacks) b by an edge in a directed graph $a \rightarrow b$. Discourses (1.1) and (2.3) become thus the graph D' :⁷



The discourse immediately after c , and before c' , is the triangle $a - b - c$ without other nodes nor the edge $\langle c', d \rangle$, while after f , but just before f' , is D' without the loop at f' , which we will denote D .

Formally, an argumentation network is a directed graph, $G = \langle G, E \rangle$, with $E : G \rightarrow \mathcal{P}(G)$ determining the out-neighbours of each node. (All functional notation is extended pointwise to sets, e.g., for $X \subseteq G : E(X) = \bigcup_{x \in X} E(x)$). We consider only directed graphs, so “graph” means digraph unless explicitly

⁶In [25], p.238, the authors note disappointingly little attention paid to the self-defeating arguments in the argumentation literature. Although psychologically very different from incoherent totalities of arguments, like $a - b - c$, their formal role and effects are entirely analogous.

⁷Direction of the edges may be reversed, provided that it is done consistently throughout the whole development. Argumentation networks, or various derivative concepts, may be formulated in the literature with edges going in the opposite direction.

A particular consequence of the representation (2.1) and this graphical counterpart is that statements, the actual carriers of truth-values, correspond to the sentence tokens and not types. Saying the same sentence (type) at two different points may turn it into different statements. A token, or a statement, is in this context just a point in a network of cross-references, a node of the discourse graph.

stated otherwise.) A *solution* is an assignment $\alpha \in \{0, 1\}^G$ of boolean values **1** (true, accepted) or **0** (false, defeated), respecting the following rules:

$$\begin{aligned} (1) \quad & \forall x \in \text{dom}(\alpha) : \alpha(x) = \mathbf{1} \Leftrightarrow \forall y \in E(x) : \alpha(y) = \mathbf{0} \\ (2) \quad & \forall x \in \text{dom}(\alpha) : \alpha(x) = \mathbf{0} \Leftrightarrow \exists y \in E(x) : \alpha(y) = \mathbf{1} \end{aligned} \quad (2.5)$$

The rules are equivalent in the context of consistent, classical theories, but we record them both for further use. The set of solutions for a graph is denoted $\text{sol}(\mathbf{G})$. In argumentation theory, a solution α corresponds to a *stable extension*, given by the set $\alpha^1 = \{x \in G \mid \alpha(x) = \mathbf{1}\}$ ([16], Definition 13, Lemma 14). In terms of discourses, it gives a consistent assignment of truth-values to all statements, i.e., a model of the respective theory (2.1). Non-existence of a solution indicates an anomaly, an incoherent set of arguments or a paradoxical element in the discourse. A trivial example is the liar, an argument defeating itself – the graph $x \curvearrowright$ has no solution. A more elaborate example is \mathbf{D}' in (2.4). Its lack of coherence can be seen trying, for instance, first to make $d = \mathbf{1}$. This forces $e = \mathbf{0}$ and leaves the liar node f' with its loop without any possible assignment. Trying instead $d = \mathbf{0}$, leaves the triangle $a - b - c$, which can not be assigned any value. \mathbf{D} , on the other hand, is not problematic, since assigning $d = b = f = \mathbf{1}$ and $a = c' = e = \mathbf{0}$, respects (2.5). Informal verification of the dialogues confirms the intuitive correctness of the conclusion that $\text{sol}(\mathbf{D}') = \emptyset$.

Any discourse \mathbf{T} , in the form (2.1), gives a graph $\mathcal{G}(\mathbf{T})$, by taking all propositional variables of \mathbf{T} as the nodes, and defining the out-neighbours $E(x) = I_x$ for every variable x . In particular, variables which occur on the left-hand side of the equivalences $x \leftrightarrow \mathbf{1}$ become the sinks of $\mathcal{G}(\mathbf{T})$. Conversely, a digraph \mathbf{G} gives a discourse $\mathcal{D}(\mathbf{G})$ by taking its nodes G as variables and forming the equivalence $x \leftrightarrow \bigwedge_{y \in E(x)} \neg y$ for each $x \in G$. For the graph \mathbf{D}' in (2.4), $\mathcal{D}(\mathbf{D}')$ gives the discourse (2.2), while $\mathcal{G}(\mathcal{D}(\mathbf{D}')) = \mathbf{D}'$. These transformations yield easily the following fact. It only specializes a particular case of Theorem 3.21, but gives here the first indication of the equivalence of the logical and graphical formulations of the problem.

Fact 2.6 *For every graph \mathbf{G} and discourse \mathbf{T} ($\text{mod}(\mathbf{T})$ denotes all models of \mathbf{T}):*
 $\text{sol}(\mathbf{G}) = \text{mod}(\mathcal{D}(\mathbf{G}))$, and
 $\text{mod}(\mathbf{T}) = \text{sol}(\mathcal{G}(\mathbf{T}))$.

Argumentation theory was worth mentioning both because it is a field of wide interest ([25] gives a good overview, [18] shows newer developments), and because the plain equivalence to the problem of paradox makes the transfer of our results straightforward. But we neither assume familiarity with its details nor intend to present them. We will only parenthetically mention relations to the concepts from argumentation theory. The connections to graphs, on the other hand, are of central importance, as suggested by the above fact and explained further below.

2.2 Kernels of digraphs

A *kernel* of a digraph $\mathbf{G} = \langle G, E \rangle$ is a subset $K \subseteq G$ which is independent (no edges between nodes in K) and absorbing (every node outside K has an edge

to some node in K):

$$\begin{array}{lcl} G \setminus K \supseteq E^\sim(K) & \text{(independent)} & \\ \text{and } G \setminus K \subseteq E^\sim(K) & \text{(absorbing)} & \\ \hline \text{i.e., } G \setminus K = E^\sim(K), & & (2.7) \end{array}$$

where E^\sim denotes the converse of E , i.e., $E^\sim(y) = \{x \in G \mid y \in E(x)\}$. One checks easily that K is a kernel iff the assignment $\alpha_K = (K \times \mathbf{1}) \cup ((G \setminus K) \times \mathbf{0})$ is a solution, i.e., satisfies conditions (2.5). For instance, D from (2.4) has a unique kernel, containing nodes assigned $\mathbf{1}$ at point f in (1.1)-(2.3); while D' , i.e., D with the additional loop at f' , has no kernel, representing paradoxical discourse, the whole $a - f'$.

The main semantic notion associated with our graphical representation, generalizing the notion of a kernel, is a *local kernel*, [24]. It is an independent subset L which absorbs its out-neighbours, i.e., an $L \subseteq G$ satisfying:

$$E(L) \subseteq E^\sim(L) \subseteq G \setminus L. \quad (2.8)$$

One verifies easily that a kernel is a local kernel, while a local kernel need not be a kernel. $Lk(G)$ denotes the set of local kernels in G . In argumentation theory, a local kernel is called an *admissible* extension, and an argument is *acceptable* if it can be added to it, resulting in a new admissible extension. Iterating such a process leads to a *complete* extension \bar{L} – the unique, maximal extension that extends the admissible set L . In terms of graphs, for any local kernel $L \in Lk(G)$ one obtains inductively its *completion*, \bar{L} , defined as follows:

Definition 2.9 *The completion \bar{L} of an $L \in Lk(G)$ is defined inductively:*

$$\begin{aligned} L_0 &= L \\ L_{i+1} &= \text{sinks}(G \setminus E^\sim(L_i)) \end{aligned}$$

Fixed-point, $\bar{L} = L_{i+1} = L_i$, is reached no later than at $i = |G|$.

For all $i : L_i \in Lk(G)$ and $G \setminus (\bar{L} \cup E^\sim(\bar{L}))$ has no sinks. Of special interest will be the completion of the empty local kernel, \emptyset , representing the values necessarily induced from the “facts”, sinks of the graph. We then let G° be the subgraph of G induced by $G^\circ = G \setminus (\emptyset \cup E^\sim(\emptyset))$. It represents the sinkless residuum of G , remaining after removal of all nodes with values induced from the sinks. Since for any $L \in Lk(G) : \text{sinks}(G) \subseteq \text{sinks}(G \setminus E^\sim(L))$, \emptyset is obviously contained in the completion of every local kernel:

$$\text{For every } L \in Lk(G) : \emptyset \subseteq \bar{L}. \quad (2.10)$$

Now, for any local kernel $L \in Lk(G)$, the assignment

$$\alpha_L = (L \times \mathbf{1}) \cup (E^\sim(L) \times \mathbf{0}) \quad (2.11)$$

is, so to speak, “justified”: each node assigned $\mathbf{0}$ has an out-neighbour assigned $\mathbf{1}$, while all out-neighbours of a node assigned $\mathbf{1}$ are assigned $\mathbf{0}$. Interestingly, this is equivalent to satisfaction of (2.5), as ensured by the following fact (recall that for an $\alpha \in \{\mathbf{0}, \mathbf{1}\}^G$, we denote $\alpha^1 = \{x \in \text{dom}(\alpha) \mid \alpha(x) = \mathbf{1}\}$).

Fact 2.12 For any graph G , subset $H \subseteq G$ and $\alpha \in \{0, 1\}^H$:

α satisfies both conditions (2.5) iff α^1 is a local kernel of G and $\alpha = \alpha_{\alpha^1}$.

PROOF. The condition (1) implies that α^1 must be independent, so $E^\sim(\alpha^1) \subseteq G \setminus \alpha^1$ and, moreover, that $E(\alpha^1)$ be assigned 0 . But then (2) requires for any $x \in E(\alpha^1)$ to have an edge back to α^1 , i.e., $E(\alpha^1) \subseteq E^\sim(\alpha^1)$. The equality $\alpha = \alpha_{\alpha^1}$ is then obvious.

Conversely, making $\alpha_L(L) = 1$ for a local kernel L ensures (1) when also $\alpha_L(E^\sim(L)) = 0$. The latter ensures then trivially (2), since for each $x \in E^\sim(L)$: $E(x) \cap L \neq \emptyset$. \square

In particular, for a total $\alpha \in \{0, 1\}^G$, α^1 is a kernel of G iff α is its solution which, by Fact 2.6, is equivalent to α being a model of the discourse $\mathcal{D}(G)$.⁸

Example 2.13 (1) *Sinks of a graph, $\text{sinks}(G) = \{x \in G \mid E(x) = \emptyset\}$, can be seen as “external facts”, accepted as true. A statement directly negating such a fact is a node pointing at it. Every collection $L \subseteq \text{sinks}(G)$ is a local kernel (since $E(L) = \emptyset$), inducing the assignment of 0 to all nodes in $E^\sim(L)$.*

(2) *Consider the subdiscourse F of D' from (2.4) induced by $d - e - f'$:⁹*

d : The next statement is false.

e : The previous statement is false.

f' : The previous statement is false, and so is this one.

F : $d \rightleftarrows e \longleftarrow f' \curvearrowright$

This subdiscourse arises from the local kernel $E = \{e\}$, as $\text{dom}(\alpha_E)$ according to (2.11). The local kernel $\{d\}$ induces even smaller subdiscourse $d \rightleftharpoons e$ of F . In either case, the induced assignment respects (2.5) independently from the values (or their lack) assigned to the rest of D' .

In terms of discourses, a local kernel gives a consistent – possibly partial – evaluation, which can be seen as internally justified: all its statements can be made simultaneously true, while all statements they claim to be false, are made false. A local kernel L gives thus a general concept of a “coherent subdiscourse”, in the sense of a subset of statements, namely $\text{dom}(\alpha_L)$, which can be consistently assigned truth-values, obeying the rules (2.5), irrespectively of the assignment to all other statements. For instance, the graph D' from (2.4) has no kernel, but $\{d, b\}$ is its local kernel, and so is $\{e\}$ (the latter inducing the subdiscourse F from Example 2.13.(2).) The lack of any kernel suggests some anomaly, as we can see considering the triangle $a - b - c$ or the whole graph D' . But an anomaly does not mean meaninglessness – the discourse may still possess a lot of information, which can be recovered from its local kernels. These provide the semantic basis for the logic PDL which is introduced in the following section.

⁸The equivalence of kernels and non-paradoxical discourses was first noted in [7], while of kernels and stable extensions of argumentation networks in [13].

⁹We speak about a subdiscourse of $G = \langle G, E \rangle$ induced by a set of statements $H \subseteq G$, in the sense of the induced subgraph, i.e., $H = \langle H, E \cap (H \times H) \rangle$, and likewise about a subdiscourse induced by a local kernel $L \subseteq G$, namely, the subgraph induced by $L \cup E^\sim(L)$. The meaning should be clear from the context.

3 The Propositional Discourse Logic

The logic PDL allows to establish facts about possible truth or falsity of statements in any finite propositional discourse. Semantics of a discourse is determined by the assignments induced, according to (2.11), from the local kernels of the network of its cross-references, represented by the digraph, as exemplified by (2.4).

Given a graph $G = \langle G, E \rangle$, we let wff_G be the set of all propositional formulae over the alphabet G , formed using the set of connectives $\{\neg, \wedge\}$.¹⁰ $\Gamma \subseteq \text{wff}_G$ denotes any finite set of such formulae. The *basic* formulae of the language \mathcal{L} have the format $[\Gamma : G]$ and are understood as saying that all formulae in Γ can be made simultaneously true over the graph G by some (possibly partial) assignment respecting (2.5), i.e., induced by a local kernel according to (2.11). Just as any actual totality of a natural discourse limits the range of possible distributions of truth-values between its statements, so the graph acts as a restriction on the relevant assignments to its nodes, the tokens of the discourse.

A basic formula $[\Gamma : G]$ is *atomic* if Γ contains only literals (positive, Γ^+ , or negated variables, Γ^-). Atomic formulae appear in the first point of the following definition.

Definition 3.1 *The relation $\models \subseteq Lk(G) \times \text{basic}(\mathcal{L})$ is defined inductively:*

- $L \models [\Gamma : G]$ iff $\Gamma = \{a \mid a \in \Gamma^+ \subseteq G\} \cup \{\neg b \mid b \in \Gamma^- \subseteq G\}$ and $\Gamma^+ \subseteq L$ and $\Gamma^- \subseteq E^\sim(L)$
- $L \models [\Gamma, A \wedge B : G]$ iff $L \models [\Gamma, A : G]$
- $L \models [\Gamma, \neg(A \wedge B) : G]$ iff $L \models [\Gamma, \neg A : G]$ or $L \models [\Gamma, \neg B : G]$
- $L \models [\Gamma, \neg\neg A : G]$ iff $L \models [\Gamma, A : G]$

It is easy to see that \models is monotone with respect to the first argument – if $L, M \in Lk(G)$, $L \subseteq M$ and $L \models [F : G]$ then $M \models [F : G]$ for any $F \in \text{wff}_G$.

The full language \mathcal{L} is given by *composite formulae*, namely, propositional combinations of the basic formulae, again, using only \neg and \wedge . Their finite sets Θ, Φ form sequents, $\Theta \vdash \Phi$, using notational conventions of sequent calculi.

Definition 3.2 *The true formulae of \mathcal{L} , $\models \subseteq \mathcal{L}$, are defined as follows:*

- $\models [\Gamma : G]$ iff there is some $L \in Lk(G)$ such that $L \models [\Gamma : G]$
- $\models \neg\phi$ iff $\not\models \phi$
- $\models \phi \wedge \theta$ iff $\models \phi$ and $\models \theta$

The logical consequence is defined in the standard way, for $\Theta, \Phi \subset \mathcal{L}$:

¹⁰All other connectives can be defined from $\{\neg, \wedge\}$ in the classical manner. This choice does not in any way limit the expressivity of the language, and is made only for establishing an easy connection to graphs.

- $\Theta \models \Phi$ iff there is $\theta \in \Theta$ such that $\not\models \theta$ or there is $\phi \in \Phi$ such that $\models \phi$.

Two points should be noticed. The first is that truth is defined relatively to a discourse, the $[\dots : G]$. Furthermore, the definition requires only the *existence* of local kernels, without requiring the relation $L \models _$ to hold for *all* local kernels L . A basic formula $[\Gamma : G]$ is true if it is satisfied by *some* local kernel $L \in Lk(G)$. Hence, $\models [\Gamma : G]$ can be read as “possibly Γ in G ”. (This generalizes the notion of *admissibility* of arguments in argumentation theory, which considers only Γ consisting of a single propositional variable.) Truth of a basic negation, $\models \neg[\Gamma : G]$, denotes thus non-existence of any local kernel satisfying Γ , which can be read as “impossibly Γ in G ”.

Example 3.3 *The following lists some examples of (in)valid statements in the discourse F from Example 2.13.(2). The only two local kernels are $D = \{d\}$ and $E = \{e\}$, where the latter is also a kernel of F .*

1. $\models [d : F]$ since $D \models [d : F]$ $F : \quad d \rightleftarrows e \longleftarrow f' \curvearrowright$
2. $\models [\neg d : F]$ since $E \models [\neg d : F]$
3. $\models [d : F] \wedge [\neg d : F]$ since $\models [d : F]$ and $\models [\neg d : F]$
4. $\not\models [d \wedge \neg d : F]$ since for any $L \in Lk(F)$ if $d \in L$ then $d \notin E^\sim(L)$
5. $\models [\neg e : F]$ since $D \models [\neg e : F]$ (since $e \in E^\sim(\{d\})$)
6. $\models [e : F]$ since $E \models [e : F]$
7. $\not\models [d \wedge e : F]$ since there is no $L \in Lk(F)$ such that $\{d, e\} \subseteq L$
8. $\not\models [f' : F]$ since there is no $L \in Lk(F)$ such that $f' \in L$
9. $\models [e, \neg f', \neg d : F]$ since $E \models [e, \neg f', \neg d : F]$
10. $\not\models [d \wedge \neg(\neg f' \wedge f') : F]$ since for each $L \in Lk(F) : d \notin L$ or $f' \notin L \cup E^\sim(L)$

In 9, Γ contains a literal for each variable from F , so this validity means that $\{e\}$ is actually a kernel of the graph F . Validity of 3 and invalidity of 4 corresponds to the non-distributivity of the existential quantifier (or diamond) over conjunction. The former says that there is a local kernel making $d = \mathbf{1}$ and there is one making $d = \mathbf{0}$. The latter claims the existence of a local kernel making both simultaneously. Its justification shows that a contradiction, like $d \wedge \neg d$, is not satisfied in any discourse. But as suggested by 10, also its negation may fail. Such a failure, amounting to the impossibility of assigning either $\mathbf{0}$ or $\mathbf{1}$ to a node, means that the statement does not appear in any acceptable, coherent (sub)discourse. Such statements deserve special attention.

Definition 3.4 *In a graph G , $x \in G$ is a paradox iff $\models \neg[\neg(\neg x \wedge x) : G]$*

The definition provides means to move from the meta-level, where paradox is a property – inconsistency – of discourses, to the object-level, where we would like to identify particular statements as paradoxical. Familiar examples turn out as expected. The liar, for instance, must be a paradox, $\models \neg[\neg(x \wedge \neg x) : x \curvearrowright]$, for the simple reason that the graph has no local kernels at all. In the more complex discourse $H : x_1 \xleftrightarrow{\quad} x_2 \xrightarrow{\quad} x_3 \rightarrow \cdot \rightarrow s, \{s\}$ is (the only) local kernel, and all x_i are paradoxical: $\models \neg[\neg(x_i \wedge \neg x_i) : H]$.

Note that $[x \wedge \neg x : G]$ does not hold in any graph so, in particular, a paradox does not come out here as any dialetheia. According to the above definition, it is a statement which can not possibly witness to the negation of such a contradiction.

The definition captures only statements which are *necessarily* paradoxical, failing to function in *all* acceptable subdiscourses.¹¹ *Contingent* paradoxes are statements $x \in G$ which are paradoxical only under specific circumstances, expressed by some formula $F \in \text{wff}_G$, i.e., such that:

$$\models \neg[F \wedge \neg(x \wedge \neg x) : G]. \quad (3.5)$$

This validity means the impossibility of satisfying both conjuncts simultaneously in G : whenever F is satisfied, then x *necessarily* becomes paradoxical. In Example 3.3, for instance, 10 confirms the intuition that whenever $d = \mathbf{1}$ in F , then f' becomes paradoxical. To capture the real possibility of x being a paradox, however, the above does not suffice. (3.5) is satisfied, for instance, for any contradiction F . One should, in addition, verify that F indeed can be true, i.e., extend (3.5) with the conjunct expressing the factual possibility of F :

$$\models [F : G]. \quad (3.6)$$

For 3.3.10, for instance, the additional verification of $\models [d : F]$ in 3.3.1, shows that in fact there is an acceptable subdiscourse making f' paradoxical.

Paradox, being a necessary consequence of a discourse, has thus a different status than merely possible truth or falsehood. Trying to bring all three on equal footing would lead to a three-valued logic and involve replacing our existential truth by the universal one (i.e., the existential quantifier in the first point of Definition 3.2 by the universal one.) Some consequences of such a move can be obtained from the following fact.

Fact 3.7 *For any $\Gamma \subset \text{wff}_G$ and $L \in Lk(G)$, if $\overline{\varnothing} \models [\Gamma : G]$ then $\overline{L} \models [\Gamma : G]$.*

PROOF. By structural induction on Γ . Since $\overline{\varnothing} \subseteq \overline{L}$ for all $L \in Lk(G)$, so the basis for atomic formula (Γ containing only literals) follows from the monotonicity of \models . The inductive steps for \neg, \wedge and $\neg(\dots \wedge \dots)$ are all trivial. \square

Since \varnothing is a local kernel in which no atom is true, defining a statement as true only if it is satisfied in every local kernel would render all atoms neither true

¹¹The corresponding idea in Kripke's theory of truth from [23] would be to take as paradox only those sentences which are neither true nor false in any fixed-point. We do not claim that this is appropriate for a general theory of truth, which is not our object.

nor false. One might therefore try taking as true the formulae holding in the completion of every local kernel. But then, by the above fact, what is true is simply all that is true in $\overline{\mathcal{O}}$. Before presenting a complete and adequate reasoning system for PDL, we note first that this notion of truth – as validity in a discourse – is captured exactly by Lukasiewicz’s logic L3.

3.1 Łukasiewicz’s logic L3

We first show that just like the classical models of $\mathcal{D}(\mathbf{G})$ determine kernels of \mathbf{G} , so the models of $\mathcal{D}(\mathbf{G})$, viewed now as a theory in L3, determine the local kernels of \mathbf{G} . Recall the L3-tables for the relevant connectives:

$$\begin{array}{c|c} \neg & \\ \hline \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ \perp & \perp \end{array} \quad \begin{array}{c|c|c|c} \wedge & \mathbf{1} & \perp & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{1} & \perp & \mathbf{0} \\ \perp & \perp & \perp & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \quad \begin{array}{c|c|c|c} \leftrightarrow & \mathbf{1} & \perp & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{1} & \perp & \mathbf{0} \\ \perp & \perp & \mathbf{1} & \perp \\ \mathbf{0} & \mathbf{0} & \perp & \mathbf{1} \end{array} \quad (3.8)$$

Because \leftrightarrow occurs only as the main connective forming the equivalences (2.1), their right hand sides are evaluated as in strong Kleene logic, which shares the tables for \neg and \wedge with L3. One can therefore introduce there other connectives, disjunction and implication in particular, using classical definitions as in strong Kleene logic. Our restricted use of Łukasiewicz’s biconditional captures the difference between treating paradox as a third logical value, where all three appear with the same “necessary character” – and treating it only as a limiting case, the impossibility of classically meeting the intuitive meta-requirement that statements have the same semantic value as the content of what they say.

The semantics of \leftrightarrow in L3 captures this identity, leading to the following characterization of local kernels. \models_L denotes satisfaction defined by tables (3.8), with $\mathbf{1}$ as the only designated value. In this context, an assignment α_L induced from a local kernel according to (2.11), is treated as a total 3-valued assignment with $\alpha_L(x) = \perp$ for all $x \notin L \cup E^\sim(L)$.

Proposition 3.9 *For a graph \mathbf{G} , we have:*

- a) *if $L \in Lk(\mathbf{G})$ then $\alpha_L \models_L \mathcal{D}(\mathbf{G})$, and*
- b) *for any $\alpha \in \{\mathbf{1}, \mathbf{0}, \perp\}^G$, if $\alpha \models_L \mathcal{D}(\mathbf{G})$ then $\overline{\mathcal{O}} \subseteq \alpha^1 \in Lk(\mathbf{G})$.*

This allows us to replace local kernels by \models_L in the formulation of the semantics of PDL.

Theorem 3.10 *For $\Gamma \subset \text{wff}_G$:*

$$\models [\Gamma : \mathbf{G}] \text{ iff there is some } \alpha \in \{\mathbf{1}, \mathbf{0}, \perp\}^G \text{ such that } \alpha \models_L \mathcal{D}(\mathbf{G}) \text{ and } \alpha \models_L \Gamma.$$

Consequently, any reasoning system for L3 can be used to establish validity or contradiction of a formula F in a discourse given by a graph \mathbf{G} . The logical consequence $\mathcal{D}(\mathbf{G}) \models_L F$ means that F is true in the completion of *every* local kernel of \mathbf{G} . In particular, $\mathcal{D}(\mathbf{G}) \models_L x \leftrightarrow \neg x$ (with L3 biconditional) iff $x = \perp$ in every model of $\mathcal{D}(\mathbf{G})$. This is certainly an elegant, logical characterization of necessarily paradoxical statements. Definition 3.4 may be less appealing but, in

this respect, PDL coincides with L3. One verifies easily that $\models \neg[\neg(x \wedge \neg x) : G]$ iff $\mathcal{D}(G) \models_L x \leftrightarrow \neg x$. The former states the non-existence of any local kernel of G assigning a truth-value to x , while the latter that every local kernel induces the assignment $x = \perp$.

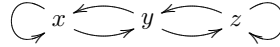
Our initial intuitions suggested the importance of the undetermined character of typical statements, whose truth is a mere possibility. Unlike L3 and most other non-modal logics, PDL captures the possible truth/falsehood as the natural dual to the impossible truth and falsehood of paradoxes. To achieve this in L3 one must take an indirect meta-route, for instance, using Theorem 3.10 or the following corollary. The possibility (satisfiability) of a formula F is equivalent to the non-validity of its “negation”: it is satisfiable iff $\neg F \vee (F \leftrightarrow \neg F)$ is not valid (with L3 biconditional, and $x \vee y = \neg(\neg x \wedge \neg y)$.) Let $\neg\Gamma \vee (\Gamma \leftrightarrow \neg\Gamma)$ denote the disjunction of all respective formulae $\bigvee \{\neg F \vee (F \leftrightarrow \neg F) \mid F \in \Gamma\}$.

Corollary 3.11 *For $\Gamma \subset \text{wff}_G : \models [\Gamma : G]$ iff $\mathcal{D}(G) \not\models_L \neg\Gamma \vee (\Gamma \leftrightarrow \neg\Gamma)$.*

PROOF. \Rightarrow) If $\models [\Gamma : G]$ then Theorem 3.10 gives an $\alpha \in \{\mathbf{1}, \mathbf{0}, \perp\}^G$ such that $\alpha \models_L \mathcal{D}(G)$ and $\alpha \models_L \Gamma$. In particular, for every $F \in \Gamma : \alpha \not\models_L \neg F \vee (F \leftrightarrow \neg F)$. \Leftarrow) Since $\mathcal{D}(G) \not\models_L \neg\Gamma \vee (\Gamma \leftrightarrow \neg\Gamma)$, so there is some $\alpha \in \{\mathbf{0}, \mathbf{1}, \perp\}^G$ such that $\alpha \models_L \mathcal{D}(G)$ and $\alpha \not\models_L \neg\Gamma \vee (\Gamma \leftrightarrow \neg\Gamma)$. That is, for every $F \in \Gamma : \alpha(\neg F) \in \{\mathbf{0}, \perp\}$, i.e., $\alpha(F) \in \{\mathbf{1}, \perp\}$ and $\alpha(F \leftrightarrow \neg F) \in \{\mathbf{0}, \perp\}$. But if $\alpha(F) = \perp$ then $\alpha(F \leftrightarrow \neg F) = \mathbf{1}$, hence $\alpha(F) = \mathbf{1}$ for all $F \in \Gamma$. So $\alpha \models_L \Gamma$, yielding $\models [\Gamma : G]$ by Theorem 3.10, as desired. \square

For instance, for any cycle X , L3 does not establish anything about the truth of any node $x_i \in X$: $\mathcal{D}(X) \not\models_L x_i \vee \neg x_i$, since in the absence of sinks, the empty local kernel provides a model of $\mathcal{D}(X)$ with \perp at all nodes. For an odd cycle, e.g., a 3-cycle $A = a_1 - a_2 - a_3$, L3 establishes the paradoxicality of all three nodes, $\mathcal{D}(A) \models_L a_i \leftrightarrow \neg a_i$. An even cycle, e.g., a 2-cycle $B = b_1 \rightleftharpoons b_2$, does not satisfy such a formula, $\mathcal{D}(B) \not\models_L b_i \leftrightarrow \neg b_i$, but there is no L3-consequence of $\mathcal{D}(B)$ which would witness to the absence of paradox. The possibility of any b_i being true follows only by indirect analysis, e.g., using the above corollary $\mathcal{D}(B) \not\models_L \neg b_1 \vee (b_1 \leftrightarrow \neg b_1) \vee \neg b_2 \vee (b_2 \leftrightarrow \neg b_2)$.

Relevance of such merely possible truth/falsehood has been noted, for instance, in Example 3.3.10. Here, consider the following discourse G :



In L3, we have the entailment $\mathcal{D}(G) \models_L \neg y \rightarrow (x \leftrightarrow \neg x)$ (L3 arrows), expressing the intuition that if y is false then x is paradoxical. Although correct, this is not very informative since y cannot possibly be false, which is captured in PDL by $\models \neg[\neg y : G]$. In addition, PDL also gives the possibility of y being true: $\models [y : G]$, i.e., y can be true and can not be false. L3 gives only conditional dependencies, like the one above. In particular, it does not establish the truth of y , since \perp at all nodes is a possible L3-model.

This shortcoming results from the fact that L3 addresses only the semantic information that is already present in the completion $\overline{\mathcal{D}}$ – the truths and falsehoods induced from sinks, the undisputed facts. Although every local kernel

provides a model of the discourse, the L3-consequences of a discourse can be determined by looking only at the truths in $\overline{\mathcal{O}}$.

Theorem 3.12 *For $F \in \text{wff}_G$, we have $\mathcal{D}(G) \models_L F$ iff $\overline{\mathcal{O}} \models [F : G]$.*

PROOF. \Rightarrow) For any graph G , \mathcal{O} is a local kernel, so $\alpha_{\overline{\mathcal{O}}} \models_L \mathcal{D}(G)$ by Proposition 3.9.a) and $\alpha_{\overline{\mathcal{O}}} \models_L F$ by assumption. Since F is formed using $\{\neg, \wedge\}$ it is not hard to see, consulting Definition 3.1 and tables (3.8), that $\overline{\mathcal{O}} \models [F : G]$.

\Leftarrow) For any α with $\alpha \models_L \mathcal{D}(G)$, $\overline{\mathcal{O}} \subseteq \alpha^1$ by Lemma 3.9.b), so $\alpha^1 \models [F : G]$ by the monotonicity of \models , and hence $\alpha \models_L F$ (since F is formed using $\{\neg, \wedge\}$). \square

As the mere consequences of undisputed facts, these truths from $\overline{\mathcal{O}}$ seem unproblematic. The problematic and more interesting things happen in the sinkless residuum, $G^\circ = G \setminus (\overline{\mathcal{O}} \cup E^-(\overline{\mathcal{O}}))$, as will be further illustrated in Section 4.2. If needed, one can therefore use L3 for analyzing statements which are true in every acceptable subdiscourse. This novel application of L3, although potentially useful, seems however to rest on all too strong a concept, as witnessed by Theorem 3.12 (so called sceptical semantics in argumentation theory). The inquiry into the alternatives actually present in the acceptable subdiscourses, on the other hand, brings us outside this usual scope of universal truth, away from L3 and towards PDL.

3.2 Reasoning in PDL

Consider the 3-cycle, as $a - b - c$ in the introductory example (2.4). Informally, one analyses it by assuming, say, $a = 1$, which requires $b = 0$ and, in turn, $c = 1$. But c can not be true when a is true, so this possibility is excluded. Alternatively, trying $a = 0$ makes $b = 1$ which, in turn, requires $c = 0$. But this, again, gives a conflict since falsity of c means that a is true. In short, and quite generally, we follow the chain of cross-references (to truth of other statements) and assign values, observing the rules (2.5). At the same time, we also decompose the discourse, in the sense that starting with $a = 1$, we never revise this trial but only check “at the end” if the resulting assignment conforms to (2.5). Paradox amounts to the impossibility of assigning either value to some nodes. Informal analysis of the whole discourse G' from (2.4) will be more complex, but along the same lines. Its paradoxical character can be confirmed by observing that the only two possible assignments to $d - e$, make it impossible to assign any values to either the cycle $a - b - c$ or the loop at f' .

The reasoning system PDL, given below in Figure 1, reflects this informal procedure. The composite and basic formula are handled by the standard sequent rules. The non-standard elements are axioms and the first four rules, which address only literals in Γ 's. Unlike the standard rules, these decompose both the considered formulae *and* the discourse in which it is evaluated. In this way, and just as the informal analysis sketched above, it is trivially finite and decidable, avoiding any non-terminating revisions of attempted assignments.

A closer look at the rule $(\vdash a)$ should explain the connections to the intuitive procedure above. To establish a possibility of a (being true) in G , $\vdash [a : G]$,

the rule's premise requires establishing the possibility of all a 's out-neighbours being simultaneously false. This is just (2.5).(1). But the premise is verified in the reduced graph $G \setminus out(a)$, where $out(a) = \{\langle a, b \rangle \mid b \in E(a)\}$ denotes all edges going out of a . In so reduced graph, a becomes a sink. This is exactly the informal move of assuming a true and checking what happens “at the end”, as this value is propagated through the discourse. If such a check “returns to” a without making any of its out-neighbours true, one concludes the possibility of a . This is what happens at the axioms, which require that all things assumed true, end up among the sinks of the resulting, reduced graph.

<u>ATOMIC FORMULAE</u> (literals $a, \neg a$ in Γ):	
Axioms: $[\Gamma, \neg a : G], \Theta \vdash \Phi$ if $a \in sinks(G)$ $\Theta \vdash \Phi, [\Gamma : G]$ if $\Gamma \subseteq sinks(G)$	
$(a \vdash)$	$\frac{[\Gamma \cup \{\neg a_i \mid a_i \in E(a)\} : G \setminus out(a)], \Theta \vdash \Phi}{[\Gamma, a : G], \Theta \vdash \Phi} \quad \text{if } E(a) \neq \emptyset$
$(\vdash a)$	$\frac{\Theta \vdash \Phi, [\Gamma \cup \{\neg a_i \mid a_i \in E(a)\} : G \setminus out(a)]}{\Theta \vdash \Phi, [\Gamma, a : G]}$
$(\neg \vdash)$	$\frac{[\Gamma, a_1 : G], \Theta \vdash \Phi ; \dots ; [\Gamma, a_n : G], \Theta \vdash \Phi}{[\Gamma, \neg a : G], \Theta \vdash \Phi} \quad \text{if } \{a_1, \dots, a_n\} = E(a) \neq \emptyset$
$(\vdash \neg)$	$\frac{\Theta \vdash \Phi, [\Gamma, a_1 : G], \dots, [\Gamma, a_n : G]}{\Theta \vdash \Phi, [\Gamma, \neg a : G]}$
The side conditions on $E(a)$ apply to both rules between which they appear.	
<u>BASIC FORMULAE</u> (one-sided sequent rules):	
$(\wedge \vdash)$	$\frac{...[\Gamma, A, B : G] \vdash ...}{...[\Gamma, A \wedge B : G] \vdash ...}$
$(\vdash \wedge)$	$\frac{... \vdash [\Gamma, A, B : G], ...}{... \vdash [\Gamma, A \wedge B : G], ...}$
$(\neg \neg \vdash)$	$\frac{...[\Gamma, A : G] \vdash ...}{...[\Gamma, \neg \neg A : G] \vdash ...}$
$(\vdash \neg \neg)$	$\frac{... \vdash [\Gamma, A : G], ...}{... \vdash [\Gamma, \neg \neg A : G], ...}$
$(\neg \wedge \vdash)$	$\frac{...[\Gamma, \neg A : G] \vdash ... ; ...[\Gamma, \neg B : G] \vdash ...}{...[\Gamma, \neg(A \wedge B) : G] \vdash ...}$
$(\vdash \neg \wedge)$	$\frac{... \vdash [\Gamma, \neg A : G], [\Gamma, \neg B : G], ...}{... \vdash [\Gamma, \neg(A \wedge B) : G], ...}$
<u>COMPOSITE FORMULAE</u> (two-sided sequent rules):	
$((\neg \vdash))$	$\frac{\Theta \vdash \Phi, \phi}{\neg \phi, \Theta \vdash \Phi}$
$((\vdash \neg))$	$\frac{\phi, \Theta \vdash \Phi}{\Theta \vdash \Phi, \neg \phi}$
$((\wedge \vdash))$	$\frac{\phi, \psi, \Theta \vdash \Phi}{\phi \wedge \psi, \Theta \vdash \Phi}$
$((\vdash \wedge))$	$\frac{\Theta \vdash \Phi, \phi ; \Theta \vdash \Phi, \psi}{\Theta \vdash \Phi, \phi \wedge \psi}$

Figure 1: The reasoning system PDL.

Example 3.13 In a 4-cycle, $a - b - c - d$, a (or any other node) can be true:

$$\begin{array}{c}
\frac{a \in \text{sinks}(b \rightarrow c \quad d \rightarrow a)}{\vdash [a : b \rightarrow c \quad d \rightarrow a]} \\
\frac{\vdash [a : b \rightarrow c \quad d \rightarrow a]}{\vdash [\neg d : b \rightarrow c \quad d \rightarrow a]} (\vdash \neg) \\
\frac{\vdash [\neg d : b \rightarrow c \quad d \rightarrow a]}{\vdash [c : b \rightarrow c \rightarrow d \rightarrow a]} (\vdash c) \\
\frac{\vdash [c : b \rightarrow c \rightarrow d \rightarrow a]}{\vdash [\neg b : b \rightarrow c \rightarrow d \rightarrow a]} (\vdash \neg) \\
\frac{\vdash [\neg b : b \rightarrow c \rightarrow d \rightarrow a]}{\vdash [a : b \rightarrow c \rightarrow d \rightarrow a]} (\vdash a)
\end{array}$$

Proofs of paradoxicality use the same graph reductions, but involve two sub-proofs, showing the impossibility of being true and of being false.

Example 3.14 x with the liar loop is paradoxical – it can be neither true (the left branch) nor false (the right branch):¹²

$$\begin{array}{c}
\frac{x \in \text{sinks}(x)}{(x \vdash) \frac{[\neg x : x] \vdash}{[x : \langle x, x \rangle] \vdash}} \quad \frac{x \in \text{sinks}(x)}{(x \vdash) \frac{[\neg x : x] \vdash}{[x : \langle x, x \rangle] \vdash}} \\
(\neg \vdash) \frac{(\neg \vdash) \frac{[\neg \neg x : \langle x, x \rangle] \vdash}{[\neg \neg x : \langle x, x \rangle] \vdash}}{[\neg(\neg x \wedge x) : \langle x, x \rangle] \vdash} \quad \frac{(\neg \vdash) \frac{[\neg \neg x : \langle x, x \rangle] \vdash}{[\neg(\neg x \wedge x) : \langle x, x \rangle] \vdash}}{[\neg(\neg x \wedge x) : \langle x, x \rangle] \vdash} \\
\frac{[\neg(\neg x \wedge x) : \langle x, x \rangle] \vdash}{\vdash \neg[\neg(\neg x \wedge x) : \langle x, x \rangle]} ((\vdash \neg))
\end{array}$$

The following proof shows that a , in the triangle $a-b-c$, is impossibly true (left branch) and impossibly false (right branch; the same holds also for b and c):

$$\begin{array}{c}
\frac{a \in \text{sinks}(a, c, \langle b, c \rangle)}{(c \vdash) \frac{[\neg a : a, c, \langle b, c \rangle] \vdash}{[c : a, \langle b, c \rangle, \langle c, a \rangle] \vdash}} \quad \frac{b \in \text{sinks}(b, a, \langle c, a \rangle)}{(a \vdash) \frac{[\neg b : b, a, \langle c, a \rangle] \vdash}{[a : b, \langle a, b \rangle, \langle c, a \rangle] \vdash}} \\
(\neg \vdash) \frac{(\neg \vdash) \frac{[\neg b : a, \langle b, c \rangle, \langle c, a \rangle] \vdash}{[a : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle] \vdash}}{[\neg a : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle] \vdash} \quad \frac{(\neg \vdash) \frac{[\neg c : b, \langle a, b \rangle, \langle c, a \rangle] \vdash}{[b : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle] \vdash}}{[\neg a : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle] \vdash} \\
((\vdash \neg)) \frac{((\vdash \neg)) \frac{[\neg a : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle] \vdash}{\vdash \neg[a : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle]}}{\vdash \neg[a : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle] \wedge \neg[\neg a : \langle a, b \rangle, \langle b, c \rangle, \langle c, a \rangle]} ((\vdash \wedge))
\end{array}$$

The second proof suggests that there may be other characterizations of a paradox, besides Definition 3.4. Indeed, inspecting the rules, we see the equivalence of the provability of the following formulae:

$$\vdash \neg[\neg(x \wedge \neg x) : \mathbf{G}] \Leftrightarrow \vdash \neg[\neg x : \mathbf{G}] \wedge \neg[x : \mathbf{G}]. \quad (3.15)$$

Hence, the second proof in Example 3.14 does show that a is paradoxical according to Definition 3.4 (assuming soundness of PDL, which is proven in Appendix).

¹²For displaying proofs, it is convenient to write a graph as a list of sinks and edges, e.g., $\langle x, x \rangle$ is the liar graph, while x the same graph with the loop removed. Some redundancy in notation may ease readability, e.g., $a, \langle b, a \rangle$ and $\langle b, a \rangle$ denote the same graph $b \rightarrow a$.

This is a special case of the general equivalence, corresponding to the distributivity of universal quantifier (or box) over conjunction:

$$\vdash \neg[\neg(A \wedge B) : G] \Leftrightarrow \vdash \neg[\neg A : G] \wedge \neg[\neg B : G].$$

As noted before, we have thus a logic of *possible truth and falsehood*, in the context where *tertium datur*, namely, the paradox. However, a paradoxical statement is not a functional consequence of the contingent values assigned to its substatements. It is not a mere possibility but occurs only as the impossibility of truth and of falsehood. It is always necessary and appears only as an unavoidable consequence of the discourse.

An interesting phenomenon is that when paradox, as a property of statements, is understood in this way, then a discourse can be paradoxical without having any paradoxical statements. It may be namely undetermined which part of the discourse is paradoxical. In the discourse (2.4), depending on the choice of the local kernel for $d - e$, either $a - b - c'$ becomes paradoxical, or else only f' . In this way, a network of contingent paradoxes, captured using pattern (3.5), might be, as a whole, a necessary paradox. Holism strikes back, as it where, and rightly so, since our judgment about the paradoxicality of particular statements is *derived* from the discourse. Traditionally simple examples, like the liar, do not contradict this in any way. They provide only examples of very simple discourses, but not any argument for restricting paradox to single statements.

Provability of the following conditional paradoxes in D' from (2.4) is left as a simple exercise to the interested reader:

$\vdash \neg[d \wedge \neg(f' \wedge \neg f') : F']$, i.e., when $d = \mathbf{1}$ then f' is paradoxical, and

$\vdash \neg[\neg d \wedge \neg(c' \wedge \neg c') : F']$, i.e., when $d = \mathbf{0}$ then c' is paradoxical.

PDL is sound and complete with respect to the semantics from Definition 3.2. (The proof is in the appendix. Inspecting the rules, in particular, for literals in Γ , one verifies easily decidability of PDL.)

Theorem 3.16 *For all finite sets $\Theta, \Phi \subset \mathcal{L} : \Theta \models \Phi \Leftrightarrow \Theta \vdash \Phi$.*

3.3 Classical logic

Semantically, PDL generalizes classical logic by considering not only assignments induced, according to (2.11), from kernels of a graph G , but also those induced from *local* kernels. By Fact 2.12, the former correspond exactly to the classical models of the respective theory $\mathcal{D}(G)$, while the latter correspond to partial L3-models, Theorem 3.10, which exist also when the theory is classically inconsistent. Partiality might seem a significant departure from the classical logic – after all, the latter does not admit any truth-gaps. But since the failure to evaluate classically is the phenomenon under investigation, there is only the question of approaching it in a more or less classical way. Concerning its effects at the level of discourse, strong Kleene logic is generally viewed as a minimal departure, preserving the classical intuitions as far as possible by respecting (2.5). This is the logic used to represent statements of discourses on the right

of the equivalences. (The only used element of L3, different from Kleene, is the main equivalence, which yields smooth transitions between logic and graphs, like Proposition 3.9 and Theorem 3.10.)

Most significantly, even if classical strictures were directed against these partial assignments and evaluations of the analyzed discourses, PDL itself, which is the meta-logic *of* such assignments and evaluations, is classical. On the reasoning side, note first that the standard axioms of sequent calculus (*) $\Theta, \phi \vdash \phi, \Phi$ are not needed. For any basic formula $\phi = [\Gamma : G]$, we either have $\models \phi$ or $\models \neg\phi$. Consequently, every instance of the standard axiom (*) becomes provable, since either $\vdash \phi$ or $\phi \vdash$. This bivalence is a genuinely classical feature of PDL. The classical character is further confirmed by the rules for composite expressions in Γ and the rules ((.)) being the classical sequent rules (for one-sided and two-sided sequent calculus, respectively), which reflect the classical semantic definitions of the connectives in Definitions 3.1 and 3.2.)

The only difference is that discourses in PDL – the basic formulae – can be evaluated over general graphs and not only, as in classical logic, over well-founded syntax graphs. At this level, classical logic can be obtained, for instance, by ignoring the graph and leaving only, for any given formula A , the collection of disjoint 2-cycles $a \rightleftharpoons \bar{a}$, one for each variable a occurring in A . If \bar{A} denotes such a graph, then a proof $\vdash [A : \bar{A}]$ establishes the consistency of A , while a proof $\vdash \neg[A : \bar{A}]$ shows that A is a tautology (a proof $\vdash \neg[A : \bar{A}]$ shows that A is a contradiction.)

An equivalent representation results from taking as the graph for a given formula A , essentially its syntax tree, where edges of the tree represent negation. All leaves with the same label a are identified, and a fresh node \bar{a} is added, with a 2-cycle, $a \rightleftharpoons \bar{a}$. If a occurs positively in some conjunction, it is first replaced by $\neg\neg a$. We call the resulting graph *syntax dag* of A (disregarding the 2-cycles at its leafs), denote it by $\text{sd}(A)$ and its root by r_A . More precisely, $\text{sd}(A)$ is defined recursively:

- $\text{sd}(\bigwedge_{i \in I} \phi_i)$ consists of a fresh node n , $\text{sd}(\neg\phi_i)$ for all $i \in I$, and edges from n to the roots of all $\text{sd}(\neg\phi_i)$,
- $\text{sd}(\neg\phi)$ consists of a fresh node r , $\text{sd}(\phi)$, and an edge from r to the root r_ϕ of $\text{sd}(\phi)$,
- $\text{sd}(a)$ for a variable a , is a two cycle $a \rightleftharpoons \bar{a}$, with a treated as the root.

For instance, for $A = \neg(\neg a \wedge a)$, the syntax dag is obtained as follows:

$$\begin{aligned}
\text{sd}(A) &= r_A \rightarrow \text{sd}(\neg a \wedge a) \\
&= r_A \longrightarrow n \begin{array}{c} \xrightarrow{\quad} \text{sd}(\neg\neg a) \\ \xrightarrow{\quad} \text{sd}(\neg a) \end{array} \\
&= r_A \longrightarrow n \begin{array}{c} \xrightarrow{\quad} r' \longrightarrow \text{sd}(\neg a) \\ \xrightarrow{\quad} r'' \longrightarrow \text{sd}(a) \end{array} \\
&= r_A \longrightarrow n \begin{array}{c} \xrightarrow{\quad} r' \longrightarrow r'' \longrightarrow a \rightleftharpoons \bar{a} \end{array}
\end{aligned}$$

Some simplifications can be made like, for instance, contracting a path of 3 edges with no outgoing edges to a single edge, or replacing $\text{sd}(\neg a)$ by $\text{sd}(\bar{a})$. After such

simplifications, we might use instead of the above graph the following one:

$$\text{sd}(A) = r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a \quad (3.17)$$

A is consistent iff there is a kernel of $\text{sd}(A)$ containing its root r_A , i.e., if there is a proof $\vdash [r_A : \text{sd}(A)]$. A proof $\vdash [\neg r_A : \text{sd}(A)]$ shows that A is a tautology. We register this fact (\models_C denotes the classical satisfaction).

Fact 3.18 *The following equivalences hold for every $A \in \text{wff}_G$:*

$$\begin{aligned} \exists V \in \{0, 1\}^G : V \models_C A &\Leftrightarrow \models [A : \bar{A}] \Leftrightarrow \models [r_A : \text{sd}(A)] \\ \forall V \in \{0, 1\}^G : V \models_C A &\Leftrightarrow \models [\neg A : \bar{A}] \Leftrightarrow \models [\neg r_A : \text{sd}(A)] \end{aligned}$$

Example 3.19 *Tautology of the non-contradiction principle, $A = \neg(a \wedge \neg a)$, is shown, to the left, using the full formula and 2-cycles \bar{A} and, to the right, using its syntax dag (3.17) and atomic labels for all subformulae:*

$$\begin{array}{c} \frac{a \in \text{sinks}(a \leftarrow \bar{a})}{[\neg \bar{a}, \neg a : a \leftarrow \bar{a}] \vdash} (\neg \vdash) \\ \frac{[a, \neg a : a \Leftarrow \bar{a}] \vdash}{[a \wedge \neg a : a \Leftarrow \bar{a}] \vdash} (\wedge \vdash) \\ \frac{[\neg \neg(a \wedge \neg a) : a \Leftarrow \bar{a}] \vdash}{\vdash [\neg \neg(a \wedge \neg a) : a \Leftarrow \bar{a}]} ((\vdash \neg)) \end{array} \quad \begin{array}{c} \frac{\bar{a} \in \text{sinks}(r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a)}{[\neg a, \neg \bar{a} : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a] \vdash} (\bar{a} \vdash) \\ \frac{[\bar{a}, \neg \bar{a} : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a] \vdash}{[\neg r', \neg \bar{a} : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow \bar{a}] \vdash} (\neg \vdash) \\ \frac{[\neg r', \neg \bar{a} : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow \bar{a}] \vdash}{[n : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a] \vdash} (n \vdash) \\ \frac{[n : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a] \vdash}{[\neg r_A : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a] \vdash} (\neg \vdash) \\ \frac{[\neg r_A : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a] \vdash}{\vdash [\neg r_A : r_A \rightarrow n \xrightarrow{r'} \bar{a} \Leftarrow a]} ((\vdash \neg)) \end{array}$$

The difference between proving $[A : \bar{A}]$ and $[r_A : \text{sd}(A)]$ is that in the former the complex element is the formula A , which is decomposed using the logical rules until only literals are left. In the latter, all intermediary subformulae appear as atomic names representing the nodes of the respective graph, which is the complex object being decomposed as the proof proceeds.

A bit loosely, one might say that classical logic is a logic without any circularity, where everything can be expressed using only well-founded syntax dags. This guarantees that every formulae obtains a unique value for every assignment to its variables, a special case of the fact that every well-founded dag has a unique kernel.¹³

Arbitrary digraphs introduce circularity in the form of loops and cycles and, as a consequence, subformulae (nodes in the graph) possibly lacking any truth-value. The non-contradiction formula A from Example 3.19, for instance, ceases to be a tautology and becomes unprovable over the liar graph $a \rightarrow a$. Even if this were damaging for the actual discourse, it is not for PDL. Such a situation, when the discourse graph G has no local kernel, can be likened to the classical inconsistency – no involved formula has any possible truth-value. No explosion

¹³For finite dags, this is the first result in kernel theory from [29]. The general statement appears, for instance, in [3], but also in the context of argumentation theory, as Theorem 30 in [16].

results, however, but on the contrary, no positive basic formula is satisfied. For any $\Gamma \subset \text{wff}_G$, we obtain $\not\models [\Gamma : G]$, i.e., $\models \neg[\Gamma : G]$. Such rare discourses are indeed ones where nothing positive can be stated about *any* subformula, each being an unavoidable paradox.

That truth-value gaps, or inconsistency of a discourse, are indeed *consequences* of circularity, and even of a special kind of circularity, follows from Richardson's theorem, [27]. We will see it by logical analysis in Subsection 3.5, and then in precise, graph-theoretical terms in Subsection 4.1.

3.4 Paradox and inconsistency

As remarked in the introduction, inconsistency may sometimes refer to propositional theories, and sometimes to the paradoxicality of discourses. Blurring the distinction is justified by their formal equivalence, whose preliminary version was given in Fact 2.6. Here, we give the general statement. The equivalence of the problem of kernel existence, KER, and propositional consistency, SAT, has been observed earlier [9]. But the graphical representation as the syntax dags gives it a new and very simple proof.

Given a digraph G , one forms the corresponding propositional theory $\mathcal{D}(G)$, obtaining the identity $\text{sol}(G) = \text{mod}(\mathcal{D}(G))$, cf. Fact 2.6. On the other hand, given a theory T (with formulae using only \neg and \wedge , but not necessarily in the discourse form (2.1)), the relevant graph $\mathcal{G}(T)$ is obtained by collecting the syntax dags of all formulae $\{\text{sd}(A) \mid A \in T\}$, adding to each dag $\text{sd}(A)$ a new node x_A with the loop $\langle x_A, x_A \rangle$ and an edge $\langle x_A, r_A \rangle$ to the root r_A of $\text{sd}(A)$ and, finally, identifying the leafs ($a \leftrightarrow \bar{a}$) with the same labels across all different subgraphs. For the formula $A = \neg(\neg a \wedge a)$ and its syntax dag from (3.17), this yields the graph

$$\mathcal{G}(A) = \begin{array}{c} \curvearrowright x_A \longrightarrow r_A \longrightarrow n \longrightarrow r' \longrightarrow \bar{a} \rightleftharpoons a \end{array} \quad (3.20)$$

Two models of A , $a = \mathbf{0}$ and $a = \mathbf{1}$, give two kernels of $\mathcal{G}(A)$, $\{\bar{a}, r_A\}$ and $\{a, r', r_A\}$. Extending the theory A with the formula $B = \neg a$, would result in the graph $\mathcal{G}(A, B)$ extending the above one with $\dots \hookrightarrow a \leftarrow r_B \leftarrow x_B \curvearrowright$.

According to Fact 3.18, a formula A is satisfiable iff $\models [r_A : \text{sd}(A)]$ holds, i.e., iff there is a kernel of $\text{sd}(A)$ including its root r_A , which induces $\mathbf{0}$ to the “loopy” nodes x_A . Filling in trivial details, this gives:

Theorem 3.21 *For every propositional theory T and digraph G :*

- a) *kernels of G and models of $\mathcal{D}(G)$ are in bijective correspondence,*
- b) *models of T and kernels of $\mathcal{G}(T)$ are in bijective correspondence.*

In particular, for every theory T , the theory $\text{GNF}(T) = \mathcal{D}(\mathcal{G}(T))$ is equisatisfiable with T : both have essentially the same models.¹⁴ Moreover, this theory has the graph normal form, as given in (2.1).

¹⁴The only additional variables in $\text{GNF}(T)$ arise from the naming of all subformulae – as the additional nodes in the syntax dag $\text{sd}(A)$, in comparison to the mere collection of 2-cycles \bar{A} . They obtain induced values whenever the theory is consistent, while a model of $\text{GNF}(T)$ gives a model of T by just forgetting such additional variables.

Corollary 3.22 *For every propositional theory T , there is an equisatisfiable theory $\text{GNF}(\mathsf{T})$ in the graph normal form (2.1).*

Now, inconsistency in propositional logic is certainly a different concept from paradoxical discourses. But intuitively, the two coincide for the discourses in this special graph form. The intuitive paradoxicality of a discourse, the impossibility of assigning truth-values to all its statements, results from the conflict between every attempted truth-values and the one which results from evaluating some statements. This is exactly inconsistency of the equivalences which determine the value of each variable on the left-hand side, as a result of the evaluation of its right-hand side. The intuitive impossibility means simply that every assignment violates some of the equivalences.

The overall consistency seems one of the meta-assumptions of the declarative discourse, giving rise to the feeling of unease when it fails and, at least sometimes, its designation as paradox. A plain contradiction does not cause similar trouble. Hearing $a \wedge \neg a$, we give it a new identifier (usually only implicitly), say n_a , and note $n_a \leftrightarrow a \wedge \neg a$. No inconsistency results, only false statement n_a . However, inconsistency of a collection of such equivalences is a different matter, meaning exactly the impossibility of all its statements n_x having a truth-value. The format (2.1) was discussed in more detail in [30]. But only in [3] one finds the observation, expressed in Corollary 3.22, that it gives a normal form for propositional theories. Consequently, not only every paradox, when represented in classical logic, gives an inconsistent theory, but also every inconsistent theory, when formulated in a graph normal form, produces a discourse which, intuitively, will be classified as paradoxical, violating the meta-principle of bivalence or contravalence. The crux of this construction, formalized by $\mathcal{D}(\mathcal{G}(-))$, is to form, given an inconsistent formula F , a contingent liar $x \leftrightarrow \neg x \wedge \neg F$. E.g., taking $F = a \wedge \neg a$ instead of A in (3.20), will give the graph without the node r_A , but with a direct edge from x_F to n .

$$\mathcal{G}(F) = \left(x_F \longrightarrow n \xrightarrow{r'} \bar{a} \rightleftharpoons a$$

Since n is a contradiction, x_F is a non-contingent paradox.

3.5 A structure of paradox

PDL provides a tool for detailed investigation of the paradoxical character of particular discourses but, as we saw in Section 3.1, also L3 could be used for this purpose. PDL's ability to handle merely possible truth was mentioned as its advantage over L3. Another advantage is PDL's sequent calculus. Analysis of the proofs of paradoxicality provides an insight into the general structure of paradox as we will now show.¹⁵ Let's keep in mind here that out-neighbours of a node x represent the statements directly negated by x .

¹⁵It is not clear whether the sequent calculus for L3 presented in [4] could be used in a similar way. This seems rather unlikely, in particular, as it has multiple rules with the same principal formula.

The equivalence (3.15) gives a simple example stating that a node necessarily violates the law of excluded middle if and only if it can not possibly be made false and can not possibly be made true. This is hardly unexpected, but correct and simple expression of basic intuitions is as reassuring as it may be non-trivial.

More interestingly, the fact that x is paradoxical, i.e., impossibly true and impossibly false, can be expressed equivalently as *both x and all its out-neighbours being impossibly true*. Indeed, impossibility of assigning $\mathbf{0}$ to a node amounts to the non-existence of any local kernel containing some of its out-neighbours, and provability in PDL satisfies the equivalence

$$\vdash \neg[x : G] \wedge \neg[\neg x : G] \iff \left((\vdash \neg[x : G]) \text{ and } (\vdash \neg[y_i : G] \text{ for all } y_i \in E(x)) \right).$$

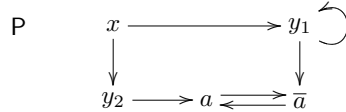
This follows immediately from the rule $(\neg\vdash)$, which is the only one yielding the impossibility of x being false, $[\neg x : G] \vdash$, and requiring for this the premises stating impossibility of y_i being true, $[y_i : G] \vdash$, for every $y_i \in E(x)$:

$$\begin{array}{c} \frac{([x : G] \vdash)}{(\vdash \neg)} \frac{[x : G] \vdash}{\vdash \neg[x : G]} \quad \frac{\frac{[y_1 : G] \vdash ; \dots ; [y_n : G] \vdash}{(\neg\vdash)} \quad \frac{[\neg x : G] \vdash}{(\vdash \neg)}}{\vdash \neg[\neg x : G]} \quad \frac{(\vdash \neg)}{(\vdash \wedge)} \frac{\vdash \neg[x : G] \quad \vdash \neg[\neg x : G]}{\vdash \neg[x : G] \wedge \neg[\neg x : G]} \end{array}$$

Another characterization of paradox can be read from further analysis of the proof of the first conjunct. It is established according to the rule $(x\vdash)$, only when some x 's out-neighbour $y_i \in E(x)$ can't be false, $[\dots\neg y_i \dots : G] \vdash$. The two subproofs together mean that if x is paradoxical (under any circumstances F , which may affect which node y_i it actually is) then it has a paradoxical out-neighbour y_i :

$$\vdash \neg[F \wedge \neg(x \wedge \neg x) : G] \implies \vdash \neg[F \wedge \neg(y_i \wedge \neg y_i) : G] \text{ for some } y_i \in E(x). \quad (3.23)$$

The implication can not be reversed, as illustrated by the following example: $\bigcirc y_1 \leftarrow x \rightarrow y_2$. Although x has a paradoxical out-neighbour y_1 , it is not itself paradoxical since the sink y_2 is a local kernel, making $x = \mathbf{0}$. A bit imprecisely, we can claim the equivalence of x being paradoxical with all its out-neighbours being impossibly true *and* at least one of them being impossibly false. In the following discourse P, x has a possibly paradoxical out-neighbour y_1 , but for this to make x paradoxical, also y_2 can not be true.



The general schema of the proof of paradoxicality of x , with $E(x) = \{y_1, \dots, y_n\}$, shows the sufficiency and necessity of the following two assumptions:

$$\begin{array}{c}
(x \vdash) \frac{[F, \neg y_1, \dots, \neg y_n : \mathbf{G} \setminus \text{out}(x)] \vdash}{(\neg \neg \vdash) \frac{[F, x : \mathbf{G}] \vdash}{[F, \neg \neg x : \mathbf{G}] \vdash} \quad \frac{[F, y_1 : \mathbf{G}] \vdash ; \dots ; [F, y_n : \mathbf{G}] \vdash}{[F, \neg x : \mathbf{G}] \vdash} (\neg \vdash)} \\
(\neg \wedge \vdash) \frac{(\neg \neg \vdash) \frac{[F, \neg(\neg x \wedge x) : y\mathbf{G}] \vdash}{\vdash \neg[F, \neg(\neg x \wedge x) : \mathbf{G}]} ((\vdash \neg))}
\end{array}$$

The premise in the right branch demands, as noted before, the impossibility of any y_i being true, while the left one the impossibility of all y_i being simultaneously false. Thus, there must exist at least one y_i which can be neither true nor false, while all the remaining ones can not be true. In \mathbf{P} , both y_1 can be false and so can y_2 . In particular, y_1 is not necessarily paradoxical. But they can not be false simultaneously. When $a = \mathbf{1}$, then x is paradoxical, but when $a = \mathbf{0}$ then the graph has a kernel, i.e.: $\vdash \neg[a, \neg(x \wedge \neg x) : \mathbf{P}]$ and $\vdash [\bar{a}, y_2 : \mathbf{P}]$. The latter says that $\{\bar{a}, y_2\}$ is a local kernel of \mathbf{P} and it can be extended to the provable claim $\vdash [\bar{a}, y_2, \neg a, \neg x, \neg y_1 : \mathbf{P}]$. Since now each node of \mathbf{P} figures in one literal of the claim, this shows that $\{\bar{a}, y_2\}$ is actually a kernel of \mathbf{P} .

Implication (3.23) shows that circularity is indispensable for obtaining a finitary paradox. A paradox must negate a paradox.¹⁶ In case of the liar, such a paradoxical out-neighbour is the liar itself while in general, it requires its own paradoxical out-neighbour, etc.. Hence, in a finite graph, a paradox requires a cycle. (The only alternative would be an infinite chain of paradoxical statements, but this requires moving to infinite and, as we will see in the next section, infinitary discourses.) Although this has always been a basic intuition about (finitary) paradoxes, we are not aware of any other, general and strictly formal expression of this idea.¹⁷

Instead of continuing this logical analysis, we switch now to a graphical one. It captures evil circularity in a more direct and more precise way, providing also a series of general results useful for the diagnosis of discursive anomalies.

¹⁶Curry's paradox may be negation-free only if $x \leftrightarrow (x \rightarrow y)$ does *not* abbreviate $x \leftrightarrow \neg(x \wedge \neg y)$. In our case, it is exactly what it does, as the arrow \rightarrow on the right is defined as in strong Kleene logic.

¹⁷This may require a qualification. On the one hand, purely logical means are inherently inadequate, since the language of classical logic is designed exactly so as to prevent any direct self-reference. One is forced to step beyond first-order logic and apply intricate Gödelizations in order to express something as simple as the liar (as a matter of fact, only something which merely reminds of the liar). On the other hand, one may take a more semantic approach. A good example is the use of non-well-founded sets, that is, eventually arbitrary graphs in [1], as the semantic basis for modeling circularity of discourses. Accepting the anti-foundation axiom is, however, a dramatic step, bringing us out of classical set theory. It may happen that a general solution to paradoxes of all kinds might require such a fundamental departure. We prefer to avoid it as long as possible, in particular, when it suffices to represent circularity in classical set theory and analyze it using essentially classical logic.

4 Some applications of kernel theory

The kernel-theoretic approach provides new means and several results for the analysis of discourses which are often easier and more intuitive than those offered by classical logic. It informs and extends accepted intuitions in a formally precise, yet intuitively appealing way. This section illustrates applicability of kernel theorems for diagnosing paradoxical character of discourses.

The development so far indicates that the truth-operator (or truth-predicate) plays no essential role for the appearance of paradoxes – object-level negation suffices. As long as we are working with essentially classical, two-valued semantics, truth-operator can be plausibly taken as the identity and represented by double negation. This may not reflect all intentions and intuitions about it, but does not affect the correctness of the diagnosis of (non)paradoxicality. In a consistent discourse, the truth-operator applied to any statement should leave its value unchanged. When applied to paradox in an inconsistent discourse, it might be asked to act differently. But this poses the questions about the nature of the truth-operator, which are different than the questions about paradox. The latter can be addressed fruitfully without settling the former.

With these reservations, all particular claims about (non)paradoxical character of specific, finitary discourses in the following examples can be proven in PDL. But the presentation should benefit from dispensing with such detailed formalities and keeping it at a more intuitive level.

Occasionally, we address also infinite cases, possibly even in infinitary logic (admitting infinite conjunctions in equivalences (2.1)). Although PDL would require extensions (to infinitary formulae and rules), the basic semantic facts hold unchanged: transformations \mathcal{D} , \mathcal{G} and Fact 2.6, definitions in Section 2.2 with Fact 2.12, as well as Fact 3.18 (with satisfaction in infinitary logic replacing \models_C) and Theorem 3.21 – all these retain their validity when passing to infinitary logic. Graphically, infinitary logic corresponds to digraphs with infinite branching, and the main difference is that such discourses, unlike the finitary ones, may be inconsistent without involving any circularity.

4.1 Circularity

Circularity seems inherently difficult to capture by logical means alone. Some cases are vicious, others are not and although it has always seemed the key to paradox, not only its nature but even its very occurrence may be disputed. The amount of implicit agreement, underlying most of its discussions, fails in the face of more complex examples or, perhaps, of more involved notions of circularity. For instance, although Yablo’s paradox appears at first sight uncontroversially non-circular, this has been challenged and disputed by a series of authors, e.g., [26, 28, 2, 8], some claiming it to possess a sort of circularity. One can construe circularity so that it applies to Yablo’s paradox, but this is then a different notion from the simple one, which does not apply to it. The graphical representation offers the standard notion of a cycle which is hardly disputable. A finite path in a graph is a sequence of nodes $x_0x_1x_2\dots x_n$, where for all $0 \leq i < n : x_{i+1} \in E(x_i)$.

A path is simple when it has no repeating vertices. A cycle is a path $x_0x_1x_2\dots x_n$, which is simple except for $x_n = x_0$. The cycle is odd/even when n is. A special case is an odd cycle of length 1, i.e., a loop xx , when $x \in E(x)$.¹⁸

4.1.1 Only cycles are vicious

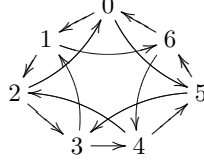
According to the theorem from [29], which appears to be the first result in kernel theory, *every finite, directed acyclic graph (dag) has a unique kernel*. By the equivalence with the propositional theories and the compactness theorem, this extends to finitely branching, infinite dags, [3]. As long as statements refer only to finitely many other statements, paradoxicality will only arise from circularity, and this holds even when one considers infinitely many statements. The infinite path of statements $x_0x_1x_2x_3\dots$, each saying: “The next 3 statements are false.” is not paradoxical. Its theory, for all $i \in \mathbb{N} : x_i \leftrightarrow \neg x_{i+1} \wedge \neg x_{i+2} \wedge \neg x_{i+3}$, is consistent, corresponding to a finitely branching dag. No matter what the statements say, as long as each claims something only about finitely many of its followers, the discourse is not paradoxical.

Thus, only cycles can become vicious in finitary discourses and examples abound. A chordless odd cycle has no kernel. This obvious fact subsumes the simplest paradoxes. The liar, $x \leftrightarrow \neg x$, is a loop, $x \overset{\curvearrowright}{\rightarrow} x$, and “I am not true” or “I am not non-false”, $x \leftrightarrow \neg\neg\neg x$, is a 3-cycle, $x \overset{\curvearrowright}{\rightarrow} y \overset{\rightarrow}{\rightarrow} z$.

A more general statement is that *a non-empty, finite, sinkless graph, which has no even cycle has no kernels*, [31]. For instance, an odd number n of persons standing in a ring, with every x_i claiming that his successor x_{i+1} is lying and so does the predecessor of his predecessor, x_{i-2} (with addition and subtraction modulo $n - 1$), forms a paradox. For $n = 3$ this is just a 3-cycle, but for larger

¹⁸This excludes any “infinite cycle” and makes Yablo’s paradox non-circular. (Infinite cycles can be introduced into infinite graphs, by topological means, using completions of infinite rays. They seem to have no relation to infinitary paradoxes, though, and Yablo remains acyclic also when such cycles are allowed.) Yablo’s circularity, suggested in [26], concerned its finitary formulation but not its actual referential structure. This is the relevant structure, captured by our graphs. On the other hand, since every person in the Yablo’s path says (*) “All my followers are lying”, Priests suggests that “one individuates the thought in such a way that all the people are thinking the same thought”. This is certainly possible, but asks us to ignore the crucial structure of the reference involved: the thought of the n -th person includes the $(n+1)$ -th person, while the thought of the $(n+1)$ -th person does not. As observed in footnote 7, one can plausibly ask also here to individuate the thought (*) – if one wants to insist on the singular form – at the level of tokens and not of its type. The isomorphism of every tail of the Yablo graph with the whole graph does not mean that they are identical.

n this involves chords, as shown for such a paradoxical discourse with $n = 7$:¹⁹



4.1.2 Vicious cycles are odd

Although circularity is necessary for finitary paradoxes, it remains innocent as long as it does not result in any self-negation. The standard example is the truth-teller, which can be formulated in different ways, all giving the same graphical representation:

- (1) “This statement is true.” or
- (2) “This statement is not false.” or
- (3) “The next statement is false.” and “The previous statement is false.”

The corresponding theory $x \leftrightarrow \neg \bar{x}$ and $\bar{x} \leftrightarrow \neg x$ gives a 2-cycle $x \rightleftharpoons \bar{x}$ with two solutions, each assigning complementary values to both statements.²⁰

The informal intuition that circularity may be vicious only when it involves some sort of self-negation, is captured precisely by the central result of kernel theory, Richardson’s theorem from the early 50-ties, [27], stating that *every finitely branching graph with no odd cycles has a kernel*. Solvability of finitely branching dags is its special case. As a more complex example, consider the infinite T with the statements, for all integers $i \in \mathbb{Z}$, of the form:

$$\begin{aligned} x_{2i} &\leftrightarrow \neg x_{2i-1} \wedge \neg x_{2i+1} \\ x_{2i+1} &\leftrightarrow \neg y_{2i+1} \\ y_{2i} &\leftrightarrow \neg x_{2i} \\ y_{2i+1} &\leftrightarrow \neg y_{2i} \wedge \neg y_{2i+2} \end{aligned}$$

Its finitely branching graph $\mathcal{G}(\mathsf{T})$ has the form

$$\begin{array}{ccccccccccc} \dots & \leftarrow & y_1 & \rightarrow & y_2 & \leftarrow & y_3 & \rightarrow & y_4 & \leftarrow & \dots \\ & & \uparrow & & \downarrow & & \uparrow & & \downarrow & & \\ \dots & \rightarrow & x_1 & \leftarrow & x_2 & \rightarrow & x_3 & \leftarrow & x_4 & \rightarrow & \dots \end{array}$$

Since $\mathcal{G}(\mathsf{T})$ has no odd cycles, T is not paradoxical.

¹⁹Incidentally, this form of discourse (a ring of size $n \geq 3$ where each x_i claims falsity of x_{i+1} and x_{i-2}) is paradoxical even when the ring is even, but this follows from a particular argument concerning the impossibility of breaking the involved odd 3-cycles. (To see this, assume a solution, pick a node x_i that is **1** and look for any **1**-successor of its **0**-successor x_{i+1} on the ring. No such can exist, since all successors of x_{i+1} are in- or out-neighbours of x_i .)

²⁰One can propose finer criteria for distinguishing statements, so that (1)-(3) come out as different, even to the point where (3) becomes a No-No paradox. But as far as their truth-conditions under the classical semantics are concerned, there is as little problem with their equivalence – and the absence of paradox – as with the fact that among two persons accusing each other of lying, only one is telling the truth, the symmetry of appearances notwithstanding. Which one it is, may vary between various tokens of truth-teller.

4.1.3 Not all odd cycles are vicious

Richardson's theorem has been generalized in various ways by giving conditions on the odd cycles ensuring the existence of a kernel. For instance, *a finite G has a kernel if each of its odd cycles $C = x_0x_1\dots x_{2k+1}$ has at least:*

- a) two reversed edges ($x_{i+1} \in E(x_i)$ is reversed if also $x_i \in E(x_{i+1})$), [14],*
- b) two crossing consecutive chords, [15], or*
- c) two chords whose targets are two consecutive nodes of the cycle, [22].*

Five persons in a ring, each accusing his right neighbour of lying, form an odd cycle and a paradoxical discourse. By a), if two persons accuse, in addition, also the person to their left, the paradox is resolved. Incidentally, for an isolated odd cycle it is sufficient for only one person to make such an additional claim, but the general result, for arbitrary finite graphs, requires two.

The conditions become more complex as one tries to cover more cases left open by the elegant theorem of Richardson ([5] lists some more results.) As in the case of finite satisfiability, the intractability of the problem of kernel existence, [6, 9], leaves little hope for any compositional criteria for deciding if odd cycles in a given discourse are vicious or not.

4.2 Ungroundedness

Ungroundedness, as introduced by Kripke in [23], subsumes circularity and relates to the issue of contingency. According to Kripke's terminology a statement x is grounded, modulo some monotone operator on partial semantic assignments, if starting from some collection of true atomic statements, the truth/falsity of x is determined by the semantic assignment that is the least fixed-point of the operator in question. (The typical operator used is obtained from the inductive step in the definition of satisfaction in strong Kleene logic.) The construction has a natural expression in terms of graphs. According to definition (2.7), sinks belong to every kernel, encoding true atomic statements, $x \leftrightarrow \mathbf{1}$. Now, as sinks belong to every kernel, their predecessors do not belong to any, so we can designate all sinks true and all predecessors of sinks false. Iterating this process leads to the assignment $\alpha_{\overline{\emptyset}}$, obtained as in (2.11), where $\overline{\emptyset}$ is the completion of the trivial local kernel \emptyset , cf. Definition 2.9. In argumentation theory, the induced assignment $\alpha_{\overline{\emptyset}}$ gives the so called sceptical (or grounded) semantics. The following fact, originating from [27] and stated generally in [3], gives substance to our claim that, for the general purposes, it is of very limited value. Paradoxical anomalies occur only after such grounded truths have been taken into account. At the center of the problem of solvability are sinkless graphs: every solution for any graph G consists of the uniquely induced (grounded) $\alpha_{\overline{\emptyset}}$ composed with a solution for the ungrounded, sinkless residuum G° (recall that G° is the subgraph induced by $G^\circ = G \setminus (\overline{\emptyset} \cup E^-(\overline{\emptyset}))$).

Fact 4.1 *For any G :*

- 1. $\text{sinks}(G^\circ) = \emptyset$, and*

2. $\text{sol}(\mathbf{G}) = \{\alpha \cup \alpha_{\overline{\mathbf{G}}} \mid \alpha \in \text{sol}(\mathbf{G}^\circ)\}$, hence also: $\text{sol}(\mathbf{G}) \neq \emptyset \Leftrightarrow \text{sol}(\mathbf{G}^\circ) \neq \emptyset$.

So, although empirical contingency may influence the (non)paradoxical character of the actual discourse, eventually, it is always the ungrounded, non-empirical residuum of the discourse which determines such a character. In case of a “fully grounded” discourse, a dag with no infinite paths, \mathbf{G}° is empty and the induced $\alpha_{\overline{\mathbf{G}}}$ is the unique solution. But \mathbf{G}° may also be empty when the graph contains cycles. In the example a) below, all statements obtain induced values as indicated; b) is paradoxical, as inducing leaves the unresolved liar, while c) is not paradoxical, having a truth-teller as the ungrounded residuum.

a) This sentence is false and the Earth isn't round.

$$\begin{array}{c} \text{ } \\ \text{ } \end{array} \mathbf{0} \rightarrow \mathbf{1}$$

b) This sentence is false and the Earth is round.

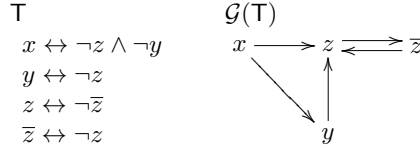
$$\begin{array}{c} \text{ } \\ \text{ } \end{array} b \rightarrow \mathbf{0} \rightarrow \mathbf{1}$$

(4.2)

c) This sentence is true and the Earth is round.

$$\overline{c} \rightleftarrows c \rightarrow \mathbf{0} \rightarrow \mathbf{1}$$

Groundedness is sometimes taken to provide *the* source of definite and unavoidable truth-values. However, we have already seen several examples of discourses where statements that are not grounded still have truth-values that can be, intuitively, ascertained. A further example may be the statement x claiming both falsity and truth of the truth-teller z :



Accepting propositional logic only, such a statement is false, even if ungrounded. Indeed, \mathbf{T} is consistent and proves classically $\neg x$. In our case, unlike in Kripke's least fixed-point, this conclusion is obtained by noting that no kernel of the solvable $\mathcal{G}(\mathbf{T})$ contains x , or by proving in PDL the possibility of $x = \mathbf{0}$, $\vdash [\neg x : \mathcal{G}(\mathbf{T})]$, and the impossibility of $x = \mathbf{1}$, $\vdash \neg[x : \mathcal{G}(\mathbf{T})]$.

Shortcomings of the least fixed-point approach have been addressed by proposing various other fixed-points as alternatives or as additions. But like every proliferation disease, this too poses the question where to stop. Possibilities suggested by Kripke were revised by the revision theory which, along with its own notion of stability, introduced a new plethora of different notions of validity and truth. This may possess some merit, and the comparison of different solutions is worthwhile, provided that it leads to a more definitive understanding and more definite theory of the phenomenon under question. In our case such problems do not arise – a discourse is paradoxical iff it corresponds to an inconsistent theory iff its graph has no kernel. All these questions are decided by PDL for the finitary discourses, and settled unambiguously by the general semantic results for the infinitary ones.

4.2.1 Non-empirical inducing

Groundedness is related to empirical contingency. As for the contingent liar a in (4.2), groundedness allows to dissolve the paradox, since Earth is round. But contingency need not be empirical. Consider F from Example 2.13.(2) of the liar f' contingent on the truth-teller e . If one stipulates that e is false then this discourse is paradoxical, if e is true it is not. This warrants the conclusion that not only the truth-teller e *could* be true but that it *must* be true in order for the discourse as a whole to function properly – T has a unique model, $e = 1$. This is a general phenomenon, illustrating again the holistic character of discourses: contingent paradoxes can give rise to definite truth-values of the ungrounded statements on which they depend. In light of the plausible meta-assumption that discourses which need not be paradoxical should *not* be designated as such, the presence of a potential paradox becomes *a fact* which, like empirical facts captured by $\overline{\mathcal{O}}$, may force the truth-values of other statements.


There is, to our knowledge, no formal account of the semantical paradoxes that provides for this kind of reasoning. Note that although it may be referred to the holistic meta-imperative of avoiding paradoxes, it is simply a sound classical inference, concluding e (and $\neg f'$) from $f' \leftrightarrow \neg f' \wedge \neg e$. If the goal is to avoid paradox whenever possible then such inferences are appropriate. The example also seems to provide a counterargument against viewing the truth-teller itself as pathological because its truth-value is arbitrary, depending only on itself. Apparently – and classically! – the truth-value of a truth-teller may depend on other statements not because it refers to them, but because they refer to the truth-teller.

Since this claim invokes a meta-principle (of global consistency), it creates a tension with the possibility of viewing the truth-teller $d - e$ as an isolated and independent subdiscourse, which has two local kernels. Limiting one's view to one's own business, without any consideration for larger issues, is a psychological possibility and it is not for logic to exclude it. The role of logic may be to describe such phenomena accurately and derive their unavoidable consequences. The provability $\vdash [\neg e : F]$ in PDL demonstrates only the possible falsity of e . But also necessary consequences of this option are provable, for instance, that it entails paradoxicality of $f' : \vdash \neg[\neg e \wedge \neg(\neg f' \wedge f') : F]$.

4.2.2 No discourse of truth-tellers is paradoxical

More can be said about the communities of truth-tellers, irrespectively of their (un)groundedness. Buridan's early solution to the liar and other paradoxes claimed that every statement, saying whatever it might be saying, says also "... and I am true." In terms of a graph, such a community X of truth-tellers involves, in addition to the actual edges between the nodes X , their copies $\overline{X} = \{\overline{x} \mid x \in X\}$, with a two cycle $x \rightleftharpoons \overline{x}$ for every $x \in X$. Every kernel of the original graph, determines also a kernel of the new one (inducing $\overline{x} = \neg x$). But one obtains also the kernel \overline{X} , which means that all $x \in X$ can be 0 , irrespectively of any connections between them. This makes any discourse almost void since,

no matter the values of various (sub)statements, every statement can always be 0. No matter what various truth-tellers say about each other, they never become paradoxical, but they never become tautological either. This applies equally to any single truth-teller involved in any discourse: it is never paradoxical, because it can always be false. An argument asserting it's own truthfulness does not add any weight but, inadvertently, gives others the possibility to consider it false, irrespectively of any other circumstances.

Truth-teller appears also in the statement “This sentence is a paradox”, with paradox understood dialetheically, i.e., x : “This sentence is true and false”. The graph contains the truth-teller x claiming also its own falsity: $\bar{x} \rightleftharpoons x$ . Its only solution makes x false. (The statement is false also when the paradox is taken as a gap, neither true nor false, though this is no longer a truth-tellers community. Its graph becomes then $\bar{x} \rightleftharpoons x \rightleftharpoons x_2 \rightarrow x_1$ and has the only solution making x false.)

4.2.3 Every discourse of liars is paradoxical.

Unlike a truth-teller graph, with a 2-cycle at each vertex, a reflexive graph has no kernel since a node with a loop, $x \in E(x)$, can not belong to any kernel. In such a liar community everybody claims, in addition to whatever he may be claiming about others, also its own falsity. Every such a liar community is paradoxical, for instance, each of the following discourses, where X is an arbitrary set with $|X| > 1$ and every $x \in X$ says:

- (i) “I am lying.” – this is just a collection of unrelated liars,
- (ii) “Everybody, including me, is lying.” or
- (iii) “Everybody else is speaking truth but I am lying.” or
- (iv) “Every person with my eye-colour is lying.” or
- (v) “My right neighbour and both his neighbours are lying” (standing in a ring or an infinite line).

4.2.4 Accusations breed guilt

A graph G is weakly complete when its underlying, undirected graph \underline{G} is complete, i.e., when for each pair of distinct nodes $x \neq y$ in G , either $x \in E(y)$ or $y \in E(x)$. As is easy to see, *any kernel of a weakly complete graph – if it exists – is a single node x satisfying the kernel equation (2.7): $E^-(x) = G \setminus \{x\}$, [3].*

For instance, any company in which, for every two persons at least one accuses the other of lying, is a paradox, unless there is a person accused of lying by everybody except himself. Exactly one such person is telling the truth.

As a special case, for any set X with $|X| > 1$, let every $x \in X$ accuse everybody else (except himself) of lying. This gives a strongly (and hence also weakly) complete graph without loops, where each node satisfies the equation (2.7), and hence gives a possible kernel. Exactly one x is speaking the truth but it can be chosen arbitrarily, as can be the value for the truth-teller (which is the special case with $|X| = 2$.)

Also Yablo’s graph, with the natural numbers \mathbb{N} as nodes and the edge relation $E(x) = \{y \in \mathbb{N} \mid y > x\}$, is weakly complete. Its unsolvability follows, since for every $x \in \mathbb{N} : E^-(x) = \{y \in \mathbb{N} \mid y < x\} \neq \mathbb{N} \setminus \{x\}$.

The same argument shows paradox in any generalization of Yablo where, instead of \mathbb{N} , one takes integers, rationals, reals, or any other total order without greatest element.

5 Related work and concluding remarks

Before concluding the paper, it seems appropriate to comment briefly relation to modal logic.

Our existentially quantified truth carries some modal element. Indeed, one can view it as a dialect of **S5**, since all assignments, determined by local kernels, are “equally accessible” from each other. However, viewing modality of PDL in this way, cripples it instead of clarifying. What is essential for the involved notion of possibility are the available assignments. These are determined by the (graphical) structure of the discourse. The fact that they all are mutually and “equally accessible” is of marginal importance. This, so to say “material” or “extensional” character of the modality, excluding also any nesting of modalities (in a different way but with similar effects as in **S5**), seem to make it a close relative of the informal, minimal modality of natural discourse.

This modal element, as the concern with mere consistency and not validity, may capture most informal intuitions but is not what logicians understand by modality. Since modal logic with Kripke semantics can be seen as a logic of graphs (or of movements along directed edges of graphs), our use of graphs might suggest turning to some existing modal logic. However, since each among typical modal logics corresponds only to *some* subclass of graphs, a new variant would be needed, allowing to model arbitrary referential structures. Such a modal foundation has been proposed for argumentation networks in [17]. The basic form of the modal formula for a graph constructed there, $\mu(G)$, bears close similarities to our graph normal form, $\text{GNF}(G)$. Its advantage arises in argumentation theory, since it allows to characterize logically various kinds of extensions.²¹ The expressive power, however, comes at a price. First, in spite of the same basic form, $\mu(G)$ is significantly more complex than $\text{GNF}(G)$. More importantly, the semantic view involves at least three values and a level of detail that eventually leads to very fine-grained distinctions, for instance, between the following two discourses:



If the goal is to capture logically the exact structure of a graph, this is certainly an advantage. But along with each discourse, its graph is given, so there seems

²¹It can also be seen as a successful continuation of the attempts to determine where to place the third value in pointer semantics for circular discourses, arising from [19].

to be little need to duplicate its representation. Logic serves rather to obtain some more abstract view of it. One can, for instance, view logic as a way of capturing the elements of the discourse which are relevant for the truth-values of its statements. In such a context, distinguishing between a discourse A , where all statements are paradoxical, and B , where all statements are paradoxical, may be more than what is actually called for.

Theorems 3.10 and 3.12 established a tight relation between the local kernel semantics of graphs, which underlies $\mu(G)$, and the well-known logic $L3$. The applicability of $L3$ to the analysis of local kernels provides a novel perspective on this logic. (As a curiosity, let us note how Łukasiewicz's third semantic value returns thus, through graphs, to the modal applications, which motivated its introduction.) More importantly, the essentially classical character of PDL and the orderliness of sequent calculus, allow to draw some conclusions about the structure of paradox from the properties of proofs of paradoxicality, as we saw in Section 3.5. It is not impossible that similar results might be obtained in the modal scenario from [17]. But since it is a strong modal logic (extending $K4$ with Löb's axiom and more), one should expect problems with forming any elegant and informative proof theory for it.

Hopefully, all other elements, besides the modal one, mentioned informally in the introduction are easily discernible in the logical system PDL. The holistic character of the discourse is reflected in that kernels can not be obtained by any straightforward, compositional rules. Consistency is not a compositional property. At the same time, local kernels represent subdiscourses where (elements of) compositionality can be regained and from which meaningful information can be extracted even when other parts, or the totality of the discourse, are inconsistent. The logic is paraconsistent in the sense that it handles meaningfully inconsistent discourses without any deductive explosion. But, at the same time, it is essentially classical, using only boolean truth-values and classical evaluation of connectives, with paradox (or lack of truth-value) appearing only as a non-functional consequence of the inconsistency of the discourse.

PDL can be seen as a formalization and completion of the project of logic of statements from [30], which asked exactly for such a propositional logic of discourses, represented as graphs in the same way as is done here.²² Having now observed also the equivalence between a series of different problems, which so far have been considered in isolation, PDL can serve for addressing instances of each such separate problem: consistency of discourses, existence of kernels in digraphs, presence of semantic paradoxes, coherence of argumentation networks and even propositional satisfiability. (Applications to non-monotonic reasoning or logic programming were not considered here, but they are possible, too, as shown initially in [16], and later, for instance, in [10, 11, 12].) On the other hand, the equivalence of different problems helps understanding each single one,

²²The only exception is the lack of the explicit truth-operator in PDL. Possibility of including such an operator remains to be investigated.

in the light of its relations to others. In this respect, kernel theory has proved particularly enlightening, clarifying and making precise many intuitions about the nature of circularity.

On a more philosophical note, we saw that in practice there is no necessary opposition between the correspondence and coherence view of truth. The two can function in unison. External facts, sinks, induce values to some statements but, typically, do not cover the whole discourse. Once such inducing has taken place, there remains the problem of potential paradoxicality of the remaining part. For this ungrounded residuum, no sufficient, external criteria of truth are available and there remain only necessary, negative criteria demanding exclusion of undesirable effects, primarily, of inconsistency. Paradoxes appear only in this inner circle. Fact 4.1 captures precisely the intuition that while empirical contingency may contribute to dissolving them, it never creates any new paradoxes – in a finitary discourse, a paradox arises only due to some self-negation, a vicious circle. Furthermore, the problem of distinguishing between evil and innocent circularity has been extensively addressed in kernel theory. Applicability of its results to the anomalies of natural discourse seems a valuable insight, providing both precise results and an enhanced general understanding of the phenomenon.

Finally, defining PDL by the essentially classical conditions (2.5), we have avoided the problematic issue of the behaviour of logical connectives in the presence of paradox. Intuitively, saying that the liar is false, seems false, just as saying that it is true, seems false (unless one turns dialetheic). Yet from (3.23) we see that every paradox is a statement negating a paradox – the obvious example being the liar. So, sometimes, negation of a paradox is a paradox and other times, it is false? It might be possible to *impose* such a distinction between the *statements* of a discourse, resulting in a new, non-classical semantics, as the one given and elaborated by Gaifman in [19, 21, 20]. Its intuitive appeal seems to rest in large part on viewing paradox as a property of individual statements. If one insists on this, then it is only reasonable to let \perp result in a functional, compositional propagation of semantic values. However, the long lasting difficulties with agreeing on a single, stable set of rules determining such a functional propagation, let alone appearance, of paradox, should be very discouraging for this approach. The difficulties and proliferation of alternatives become more understandable, if paradox does not arise from any property of individual statements, but is a holistic effect of the totality of the discourse. Then it seems more appropriate to start with a logic that treats it as such and does not try to pin it down to particular statements. Also, although positive description of a paradox might be easy to ask for, it has proven notoriously difficult to obtain. It then seems more prudent to designate paradox negatively, as the limit (of consistency), found when we reach the point where classical intuitions, despite our best efforts, fail to provide any definitive answers.

Appendix: Proofs

Proposition 3.9 *Given a graph G , we have:*

- a) If $L \in Lk(\mathbf{G})$ then $\alpha_{\bar{L}} \models_L \mathcal{D}(\mathbf{G})$ and;
b) If $\alpha \models_L \mathcal{D}(\mathbf{G})$ for $\alpha : G \rightarrow \{\mathbf{1}, \mathbf{0}, \perp\}$ then $\bar{\alpha} \subseteq \alpha^1 \in Lk(\mathbf{G})$

PROOF. a) Assume $L \in Lk(\mathbf{G})$ and consider arbitrary $x \leftrightarrow \bigwedge_{y \in E(x)} \neg y \in \mathcal{D}(\mathbf{G})$. Assume towards contradiction that $\alpha_{\bar{L}} \not\models_L x \leftrightarrow \bigwedge_{y \in E(x)} \neg y$. We let $\bar{\alpha}_{\bar{L}}$ denote the evaluation of complex formulae obtained from $\alpha_{\bar{L}}$ according to tables (3.8). Then we have $\alpha_{\bar{L}}(x) \neq \bar{\alpha}_{\bar{L}}(\bigwedge_{y \in E(x)} \neg y)$. If $\alpha_{\bar{L}}(x) = \mathbf{1}$ this inequality means that there is one $y \in E(x)$ such that $\alpha_{\bar{L}}(y) \in \{\mathbf{1}, \perp\}$, impossible by the fact that \bar{L} is a local kernel (which requires, for all $y \in E(x)$, $y \in E^\sim(\bar{L})$, i.e. $\alpha_{\bar{L}}(y) = \mathbf{0}$). If $\alpha_{\bar{L}}(x) = \mathbf{0}$ we must then have, for every $y \in E(x)$, $\alpha_{\bar{L}}(y) \in \{\mathbf{0}, \perp\}$ but this is also ruled out by the fact that \bar{L} is a local kernel (which requires existence of some $y \in E(x)$ such that $\alpha_{\bar{L}}(y) = \mathbf{1}$). The last possibility is that $\alpha_{\bar{L}}(x) = \perp$ in which case there are two possibilities. 1) We have some $y \in E(x)$ such that $\alpha_{\bar{L}}(y) = \mathbf{1}$. This contradicts $x \notin E^\sim(L)$ (required since we have $\alpha_{\bar{L}}(x) = \perp$). 2) For all $y \in E(x)$ we have $\alpha_{\bar{L}}(y) = \mathbf{0}$. This means $x \in \text{sinks}(\mathbf{G} \setminus (\bar{L} \cup E^\sim(\bar{L})))$, impossible by Definition 2.9 of \bar{L} .

b) Assume $\alpha \models_L \mathcal{D}(\mathbf{G})$. We show that α^1 is a local kernel. We show first that α^1 is independent. Assume towards contradiction that it is not. Then there are $x, y \in \alpha^1$ with $y \in E(x)$. So we have $\alpha(x) = \alpha(y) = \mathbf{1}$ and from inspecting the tables (3.8) we see that $\bar{\alpha}(\bigwedge_{y \in E(x)} \neg y) = \mathbf{0}$. In particular, we have $\alpha(x) \neq \bar{\alpha}(\bigwedge_{y \in E(x)} \neg y)$, contrary to hypothesis. Assume towards contradiction that α^1 is not locally absorbing. Then there is some $x \in \alpha^1$ with $y \in E(x)$ such that $E(y) \cap \alpha^1 = \emptyset$. Since $\alpha(z) \neq \mathbf{1}$ for all $z \in E(x)$, by α^1 being independent, this means that $\alpha(y) = \perp$ and that $\bar{\alpha}(\bigwedge_{y \in E(x)} \neg y) = \perp \neq \alpha(x) = \mathbf{1}$, contrary to hypothesis. To show that $\bar{\alpha} \subseteq \alpha^1$ is a simple proof by induction over definition 2.9. For the basis, if $x \in \text{sinks}(\mathbf{G})$, i.e. $x \in \emptyset_1$, then $x \leftrightarrow \mathbf{1} \in \mathcal{D}(\mathbf{G})$. Then it is clear that $x \in \alpha^1$. The inductive step is also trivial. \square

Theorem 3.10 $\models [\Gamma : \mathbf{G}]$ iff there is some $\alpha : G \rightarrow \{\mathbf{1}, \mathbf{0}, \perp\}$ such that $\alpha \models_L \mathcal{D}(\mathbf{G})$ and $\alpha \models_L \Gamma$.

PROOF. \Rightarrow) Assume that $\models [\Gamma : \mathbf{G}]$. Then by Definition 3.2 there is some local kernel $L \in Lk(\mathbf{G})$ such that $L \models [\Gamma : \mathbf{G}]$. We have from Proposition 3.9.a) that $\alpha_{\bar{L}} \models_L \mathcal{D}(\mathbf{G})$. We show $\alpha_{\bar{L}} \models_L \Gamma$ by induction on the complexity of Γ . We take its complexity to be the sum of the complexity of its formulae divided by $|\Gamma|$. The basis is for Γ a collection of literals. Then $\Gamma^+ \subseteq L$ and $\Gamma^- \subseteq E^\sim(L)$, so for all $x \in \Gamma^+$ we have $\alpha_{\bar{L}}(x) = \mathbf{1}$ and for all $y \in \Gamma^-$ we have $\alpha_{\bar{L}}(y) = \mathbf{0}$ (remember that $L \subseteq \bar{L}$). It follows from inspecting tables (3.8) that $\alpha_{\bar{L}} \models \Gamma$. The inductive steps are easy. For instance, if $\models [\Gamma : \mathbf{G}]$ and there is $\neg\neg A \in \Gamma$ that has maximal complexity among formulae of Γ , then we form Γ' which is like Γ except that A replaces $\neg\neg A$. Γ' has lower complexity than Γ and, obviously from Definitions 3.1 and 3.2, $\models [\Gamma' : \mathbf{G}]$. So by IH we get $\models_L \Gamma'$. Consulting tables (3.8) we see that this gives us $\models_L \Gamma$ so we are done. The cases for $\neg(A \wedge B)$ and $A \wedge B$ are equally easy.

\Leftarrow) Assume $\alpha \models_L \mathcal{D}(\mathbf{G})$ and $\alpha \models_L \Gamma$. We have $\alpha^1 \in Lk(\mathbf{G})$ from 3.9.b) and obtain $\alpha^1 \models [\Gamma : \mathbf{G}]$ by induction on the complexity of Γ , measured as in the

proof of \Rightarrow). From this $\models [\Gamma : G]$ follows by Definition 3.2. The basis is for Γ a collection of literals. Consulting tables (3.8), we see that for all $x \in \Gamma^+$, we have $\alpha(x) = \mathbf{1}$ so $x \in \alpha^1$. For all $y \in \Gamma^-$, on the other hand, we have $\alpha(y) = \mathbf{0}$. Since $\alpha \models_L \mathcal{D}(G)$, we have $\alpha \models_L y \leftrightarrow \bigwedge_{z \in E(y)} \neg z$, meaning $\alpha(y) = \bar{\alpha}(\bigwedge_{z \in E(y)} \neg z)$ (where $\bar{\alpha}$ is the evaluation of α according to tables 3.8). From tables (3.8) we see that there must then be some $z \in E(y)$ such that $\alpha(z) = \mathbf{1}$, meaning $y \in E^\sim(\alpha^1)$. It follows from Definition 3.1 that $\alpha^1 \models [\Gamma : G]$. The inductive steps are straightforward. For instance, if there is some $A \wedge B \in \Gamma$ that has maximal complexity among formulae of Γ , we form Γ' which is like Γ except that we replace $A \wedge B$ by A and B . Then $\models_L \Gamma'$ and Γ' has smaller complexity than Γ so by IH $\alpha^1 \models [\Gamma' : G]$. It follows immediately from Definition 3.1 that $\alpha^1 \models [\Gamma : G]$ and we are done. The cases of $\neg\neg A$ and $\neg(A \wedge B)$ are equally easy. \square

Soundness and completeness of PDL

Soundness and completeness follow easily from the following simple lemma giving us the compositionality we need with respect to admissibility in graphs.

Lemma 5.1 *For any graph G and $a \in G$ we have:*

- (1) $\models [\Gamma, a : G]$ iff $\models [\Gamma, \{\neg b \mid b \in E(a)\} : G \setminus out(a)]$
- (2) $\models [\Gamma, \neg a : G]$ iff for some $b \in E(a)$, $\models [\Gamma, b : G]$

PROOF. (1) \Rightarrow) Assume Γ, a is admissible in G and let $L \subseteq G$ be a local kernel witnessing to Γ and containing a . Clearly, L is a local kernel also in $G \setminus out(a)$. Now, since $a \in L$ it follows that $E(a) \subseteq E^\sim(L)$, so $\Gamma \cup \{\neg b \mid b \in E(a)\}$ is indeed admissible (in both G and $G \setminus out(a)$)

\Leftarrow) Assume $\Gamma \cup \{\neg b \mid b \in E(a)\}$ is admissible in $G \setminus out(a)$ and let $L \subseteq G$ be an arbitrary local kernel in $G \setminus out(a)$ witnessing to this fact. Then for every $b \in E(a)$ we have $E(b) \cap L \neq \emptyset$ so $L \cup \{a\}$ is a local kernel in G (as well as in $G \setminus out(a)$)

(2) \Rightarrow) Let $L \subseteq G$ be a local kernel witnessing to the admissibility of $\Gamma, \neg a$ in G . Then, for some $b \in E(a)$, we have $b \in L$. So Γ, b is admissible in G .

\Leftarrow) Assume that there is some $b \in E(a)$ such that Γ, b is admissible. Let $L \subseteq G$ be a witness. Then L also witness to the admissibility of $\Gamma, \neg a$ in G . \square

This lemma establishes soundness and invertibility of the only rules from PDL that are not essentially classical. The rest is easily verified, yielding

Theorem 5.2 *PDL is sound and all its rules are invertible.*

PROOF. The standard sequent rules for the composite formulae in $\Theta \vdash \Phi$ are trivially invertible, as are the rules for non-atomic basic $[\Gamma : G]$ (which form a one-sided sequent system for propositional logic). Lemma 5.1 established soundness and invertibility of the four rules for literals in Γ . We only have to

show that the two axiom schemata are valid:

(1) $\Theta, [\Gamma, \neg a : G] \vdash \Phi$ for some $a \in \text{sinks}(G)$.

To show $\Theta, [\Gamma, \neg a : G] \models \Phi$, it suffices to show that $\not\models [\neg a : G]$, by Definition 3.2. By Definition 3.1, this amounts to the nonexistence of a local kernel L of G containing a successor of a . But since a is a sink in G , no such L exists.

(2) $\Theta \vdash [\Gamma : G], \Phi$ for some $\Gamma \subseteq \text{sinks}(G)$.

To show $\Theta \models [\Gamma : G], \Phi$, it suffices to show $\models [\Gamma : G]$. Since Γ is a collection of atomic expressions this amounts to showing that there is a local kernel L in G such that $\Gamma \subseteq L$. But $\text{sinks}(G)$ is such a local kernel in G so the claim follows. \square

Completeness of PDL follows now by the standard line of reasoning, demonstrating invalidity of any unprovable sequent. We say that a sequent $\Theta \vdash \Phi$ is *reduced* when Θ and Φ contain only atomic formulae, i.e., every $[\Gamma : G] \in \Theta \cup \Phi$ contains only literals and, moreover, literals over sinks of G , i.e.,

$$\Gamma = \{a \mid a \in \Gamma^+ \subseteq \text{sinks}(G)\} \cup \{\neg b \mid b \in \Gamma^- \subseteq \text{sinks}(G)\}.$$

We first argue that, for any sequent, the rules suffice to create a proof-tree with all leaves reduced.

Trivially, the top level rules and rules for composite Γ suffice to create a proof-tree where all leaves have the form $\Theta \vdash \Phi$ with Θ and Φ being collections of atomic expressions $[\Gamma : G]$, i.e., each Γ being a collection of literals. Now, we employ the rules for literals, as long as there is some $a \in \Gamma$ or $\neg a \in \Gamma$ with $E(a) \neq \emptyset$, i.e. as long as the sequent is not reduced. For any finite graph G , it is clear that by employing these rules we will eventually reach a stage where all sequents have been reduced. If (i) $a \in \Gamma$ is not a sink, an application of the rule $(\vdash a)$, resp., $(a \vdash)$, makes it a sink. If (ii) $\neg a \in \Gamma$ is not a sink, then an application of the rule $(\vdash \neg)$, resp., $(\neg \vdash)$, replaces it by all its out-neighbours with positive polarity, for which case (i) applies in the next round.

Theorem 3.16 *System PDL is sound and complete: $\Theta \vdash \Phi$ iff $\Theta \models \Phi$.*

PROOF. We show that reduced, non-axiomatic $\Theta \vdash \Phi$, is invalid. We have:

(1) $\forall [\Gamma_T : G_T] \in \Theta : \Gamma_T^- = \emptyset$ and (2) $\forall [\Gamma_F : G_F] \in \Phi : \Gamma_F^- \neq \emptyset$.

(1) follows since $\Theta \vdash \Phi$ is reduced, so for all $[\Gamma_T : G_T] \in \Theta : \Gamma_T^+ \cup \Gamma_T^- \subseteq \text{sinks}(G_T)$. Since the sequent is not axiomatic, we must have $\Gamma_T^- = \emptyset$. Consequently, $\Gamma_T \subseteq \text{sinks}(G_T)$ and, since $\text{sinks}(G_T) \in Lk(G_T)$, so $\models [\Gamma_T : G_T]$.

(2) follows since, as before, $\Gamma_F^+ \cup \Gamma_F^- \subseteq \text{sinks}(G_F)$ and, since the sequent is not axiomatic, $\Gamma_F \not\subseteq \text{sinks}(G_F)$. Consequently, $\Gamma_F^- \neq \emptyset$ (since atoms from this set are negated in Γ_F). So there is some $a \in \Gamma_F^- \subseteq \text{sinks}(G_F)$, i.e. $\neg a \in \Gamma_F$ while $a \in \text{sinks}(G_F)$. It follows that $\not\models [\Gamma_F : G_F]$.

Having obtained $\models [\Gamma_T : G_T]$ for all $[\Gamma_T : G_T] \in \Theta$ and $\not\models [\Gamma_F : G_F]$ for all $[\Gamma_F : G_F] \in \Phi$, we conclude by Definition 3.2 that $\Theta \not\models \Phi$. Invertibility of all the rules ensures that if such a reduced sequent is obtained as a leaf in a proof tree from some initial sequent S , then also S is invalid. Invertibility was shown in Theorem 5.2 and here we also established soundness of the system. \square

References

- [1] Jon Barwise and Lawrence Moss. *Vicious Circles: On the Mathematics of Non-Wellfounded Phenomena*. CSLI, Stanford, 1996.
- [2] J. C. Beall. Is Yablo’s paradox non-circular? *Analysis*, 63(1):176–187, 2001.
- [3] Marc Bezem, Clemens Grabmayer, and Michał Walicki. Expressive power of digraph solvability. *Annals of Pure and Applied Logic*, 163(2):200–212, 2012.
- [4] Jean-Yves Béziau. A sequent calculus for Łukasiewicz’s three-valued logic based on suszko’s bivalent semantics. *Bulletin of the Section of Logic*, 28(2):89–97, 1998.
- [5] Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.
- [6] Vašek Chvátal. On the computational complexity of finding a kernel. Technical Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal, 1973. <http://users.encs.concordia.ca/~chvatal>.
- [7] Roy Cook. Patterns of paradox. *The Journal of Symbolic Logic*, 69(3):767–774, 2004.
- [8] Roy Cook. There are non-circular paradoxes (but Yablo’s isn’t one of them). *The Monist*, 89:118–149, 2006.
- [9] Nadia Creignou. The class of problems that are linearly equivalent to satisfiability or a uniform method for proving np-completeness. *Theoretical Computer Science*, 145:111–145, 1995.
- [10] Yannis Dimopoulos and Vangelis Magirou. A graph theoretic approach to default logic. *Information and Computation*, 112:239–256, 1994.
- [11] Yannis Dimopoulos, Vangelis Magirou, and Christos H. Papadimitriou. On kernels, defaults and even graphs. *Annals of Mathematics and Artificial Intelligence*, 20:1–12, 1997.
- [12] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170(1-2):209–244, 1996.
- [13] Sylvie Doutre. *Autour de la sémantique préférée des systèmes d’argumentation*. PhD thesis, Université Paul Sabatier, Toulouse, 2002.
- [14] Pierre Duchet. Graphes noyau-parfaits, II. *Annals of Discrete Mathematics*, 9:93–101, 1980.

- [15] Pierre Duchet and Henry Meyniel. Une généralisation du théorème de Richardson sur l’existence de noyaux dans les graphes orientés. *Discrete Mathematics*, 43(1):21–27, 1983.
- [16] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
- [17] Dov Gabbay. Modal provability foundations for argumentation networks. *Studia Logica*, 93(2-3):181–198, 2009.
- [18] Dov Gabbay and Leendert van der Torre, editors. *New Ideas in Argumentation Theory: Special Issue*, volume 93 (2-3). *Studia Logica*, 2009.
- [19] Haim Gaifman. Operational pointer semantics: solution to self-referential puzzles. In Moshe Vardi, editor, *Theoretical Aspects of Reasoning about Knowledge*, pages 43–59. Morgan Kaufman, 1988.
- [20] Haim Gaifman. Pointers to truth. *The Journal of Philosophy*, 89(5):223–261, 1992.
- [21] Haim Gaifman. Pointers to propositions. In Andre Chapuis and Anil Gupta, editors, *Circularity, Definition and Truth*, pages 79–121. Indian Council of Philosophical Research, 2000.
- [22] Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.
- [23] Saul Kripke. Outline of a theory of truth. *The Journal of Philosophy*, 72(19):690–716, 1975.
- [24] Victor Neumann-Lara. Seminúcleos de una digráfica. Technical report, Anales del Instituto de Matemáticas II, Universidad Nacional Autónoma México, 1971.
- [25] H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In Dov Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume 4, pages 219–318. Kluwer Academic Publishers, 2002.
- [26] Graham Priest. Yablo’s paradox. *Analysis*, 57:236–242, 1997.
- [27] Moses Richardson. Solutions of irreflexive relations. *The Annals of Mathematics, Second Series*, 58(3):573–590, 1953.
- [28] Roy Sorensen. Yablo’s paradox and kindred infinite liars. *Mind*, 107:137–155, 1998.
- [29] John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944 (1947).

- [30] Michał Walicki. Reference, paradoxes and truth. *Synthese*, 171:195–226, 2009.
- [31] Michał Walicki and Sjur Dyrkolbotn. Finding kernels or solving SAT. 2010.
<http://www.ii.uib.no/~michal/KernelSAT.pdf>.
- [32] Stephen Yablo. Paradox without self-reference. *Analysis*, 53(4):251–252, 1993.

Chapter 6

Paper D: Equivalence Relations for Abstract Argumentation

This paper was accepted at *BNC@ECAI 2012*, Montpellier France. We remark that the two digraphs in Figure 3 have been replaced in the version of the paper included here. The original digraphs we used did not illustrate what they were supposed to.

Equivalence Relations for Abstract Argumentation

Sjur K Dyrkolbotn
Department of Informatics
University of Bergen, Norway

Abstract

We study equivalence relations between argumentation frameworks, taking a relation to be an equivalence with respect to some semantics if it preserves and reflects the extensions of that semantics. We argue that this notion of equivalence is useful and should be considered in abstract argumentation. We go on to consider what conditions can be placed on arbitrary relations to ensure that they behave nicely with respect to equivalence. This leads us to consider bisimulations, and we show that while they do not ensure equivalence, equivalences that are also bisimulations have some nice properties with respect to semantic agreement. Then we introduce bisimulations that we call finitely collapsing. They satisfy an additional, non-local condition, and we show that they are equivalence relations with respect to all the semantics for argumentation that we consider.

1 Introduction

In abstract argumentation following Dung [8], the notion of equivalence usually adopted states that two frameworks are equivalent with respect to a semantics if they have *syntactically identical* sets of extensions under that semantics, see e.g., [13]. This is problematic for a number of reasons. First of all, it involves a peculiar attachment to the *names* of arguments - out of place, we think, in the study of abstract argumentation. This objection is typically countered by a statement to the effect that it is both well known and trivial that you can rename arguments without affecting their semantical status. While true, this is hardly satisfactory. The question immediately becomes *how* we should rename arguments so that two argumentation frameworks admit the same extensions. This, it seems, is the most interesting question, far more significant than trying to describe circumstances when the relation of identity happens to be an equivalence.

Secondly, we do not in general wish to restrict attention only to bijective functional relations that can be thought of as renamings. In fact, what seems more interesting and useful is to introduce congruences, grouping arguments

together whenever they display the same behavior with respect to some semantics. The natural way to do this, we think, is to introduce a more general notion of equivalence, saying that two frameworks are equivalent with respect to a semantics if there is a relation between their arguments that both preserves and reflects extensions of that semantics. Then we must ask: *when* is a relation an equivalence? What structural properties does it need to preserve? This is the question we address in this paper.

To motivate the general notion of equivalence we adopt, we remark that relations which preserve and reflect extensions preserve and reflect what we will call *consistency*: the ability of a semantics to provide any answers about the status of an argument as either accepted or defeated. In general, semantics for argumentation can only provide a partial answer. Some arguments have no clear status, the paradigmatic example being that of a single self-attacking argument. Such an argument is inconsistent in the sense that it cannot be accepted without being defeated, and cannot be defeated without being accepted. This, it seems to us, is the general property that arguments that can neither be accepted nor defeated always share (although in general, such a picture might arise only when we consider a chain of dependencies, e.g., an attack-cycle of odd length).

This notion of consistency, while non-standard, seems like a very natural and suggestive way to talk about arguments that do not have a clear status, and for semantics based on admissible sets, a formal connection to consistency in classical logic can also be established, c.f., the discussion in Section 2. Moreover, we hope that the general notion of equivalence presented in this paper can be used to shed light on two questions that seem to be of great importance to abstract argumentation: *why* do inconsistencies sometimes arise, and *how* do we deal with them? Apart from the case of the grounded semantics, these two questions, albeit phrased in a different manner, seem to both motivate and confound most of the usual semantics adopted for argumentation frameworks.

We think that a very interesting direction of research is to attempt at exploiting the graph-theoretical structure of argumentation frameworks in order to see if some combinatorial account of inconsistency can be given. Under the stable semantics, this question is particularly critical: an argument is inconsistent (can be neither defeated nor accepted) precisely when *all* arguments are inconsistent. This happens iff the framework does not admit a stable set, and the result that a finite framework admits a stable extension as long as it does not contain attack-cycles of odd length can therefore be seen as the first non-trivial result concerning consistency in argumentation. This result was established by Dung in his original paper [8], and by Richardson, with respect to a different (but equivalent) formalism, already in the 1950s [14]. The result is very satisfying, and we find it somewhat strange that this general direction of research has received so little attention from the community. We find it strange, in particular, that not more work has been devoted to the question of establishing structural conditions on frameworks that ensure the existence of stable sets (or, more generally, the existence of non-empty admissible sets). Hopefully, this paper can generate some renewed interest. We show, in particular, that it is possible to arrive at non-trivial structural conditions ensuring that a relation

between frameworks is an equivalence (which preserves and reflects consistency). This, we believe, suggests that the general notion of equivalence deserves attention, especially from the point of view of trying to arrive at a graph-theoretical account of the semantic behavior of argumentation frameworks, and especially with regards to questions regarding inconsistency.

2 Background

An *argumentation framework*, framework for short, is a digraph, $F = \langle \mathcal{A}, \mathcal{R} \rangle$, with \mathcal{A} a set of vertices, called *arguments*, and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ a set of directed edges, called the *attack relation*. Unless stated otherwise, we also consider argumentation frameworks that are infinite. For $(a, b) \in \mathcal{R}$ we say that the argument a *attacks* the argument b . We use the notation $\mathcal{R}^-(x) = \{y \mid (y, x) \in \mathcal{R}\}$ and $\mathcal{R}^+(x) = \{y \mid (x, y) \in \mathcal{R}\}$, extended pointwise to sets, such that, for instance, $\mathcal{R}^+(X) = \bigcup_{x \in X} \mathcal{R}^+(x)$. For general relations $\alpha \subseteq X \times Y$, we drop $+$ as a superscript and use $\alpha(x) = \{y \mid (x, y) \in \alpha\}$ and $\alpha^-(y) = \{x \mid (x, y) \in \alpha\}$. This notation also extends pointwise to sets.

A framework $F = \langle \mathcal{A}, \mathcal{R} \rangle$ is a *subframework* of a framework $F_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ iff $\mathcal{A} \subseteq \mathcal{A}_2$ and $\mathcal{R} \subseteq \mathcal{R}_2$. A subset of arguments $X \subseteq \mathcal{A}$ gives rise to the *induced subframework* $X = \langle X, \mathcal{R}_X \rangle$ with $\mathcal{R}_X = \{(x, y) \in \mathcal{R} \mid x, y \in X\}$. $F \setminus X$ denotes the subframework of F induced by $\mathcal{A} \setminus X$. A backwards infinite walk is a sequence $\lambda = x_1 x_2 x_3 \dots$ such that $x_{i+1} \in \mathcal{R}^-(x_i)$ for all $i \geq 1$. Notice that in finite argumentation frameworks, there can be backwards infinite walks, but they must involve one or more arguments twice, i.e., they involve cycles.

The most well-known semantics for argumentation, first introduced in [8] and [3] (semi-stable semantics), are given in the following definition.¹

Definition 2.1 *Given any argumentation framework $F = \langle \mathcal{A}, \mathcal{R} \rangle$ and a subset $A \subseteq \mathcal{A}$, we define $\mathcal{D}(A) = \{x \in \mathcal{A} \mid \mathcal{R}^-(x) \subseteq \mathcal{R}^+(A)\}$, the set of vertices defended by A . We say that*

- *A is conflict-free if $\mathcal{R}^+(A) \subseteq \mathcal{A} \setminus A$, i.e., if there are no two arguments in A that attack each other.*
- *A is admissible if it is conflict free and $A \subseteq \mathcal{D}(A)$. The set of all admissible sets in F is denoted $a(F)$.*
- *A is complete if it is conflict free and $A = \mathcal{D}(A)$. The set of all complete sets in F is denoted $c(F)$.*
- *A is the grounded set if it is complete and there is no complete set $B \subseteq A$ such that $B \subset A$, it is the unique member of $g(F)$.*
- *A is preferred if it is admissible and not strictly contained in any admissible set. The set of all preferred sets in F is denoted $p(F)$.*

¹The formulation used here is not always identical to the one originally given, but is easily seen to be equivalent to it

- A is stable if $\mathcal{R}^+(A) = \mathcal{A} \setminus A$. The set of all stable sets in \mathbf{F} is denoted $s(\mathbf{F})$
- A is semi-stable if it is admissible and there is no admissible set B such that $A \cup \mathcal{R}^+(A) \subset B \cup \mathcal{R}^+(B)$. The set of all semi-stable sets in \mathbf{F} is denoted by $ss(\mathbf{F})$.

For any $\mathcal{S} \in \{a, c, g, p, s, ss\}$, one also says that $A \in \mathcal{S}(\mathbf{F})$ is an *extension* (of the type prescribed by \mathcal{S}). For an argument $x \in \mathcal{A}$, one says that x is *credulously* accepted with respect to $\mathcal{S} \in \{a, c, g, p, s, ss\}$ if there is some $S \in \mathcal{S}(\mathbf{F})$ such that $x \in S$. One says that x is *sceptically* accepted with respect to $\mathcal{S} \in \{a, c, g, p, s, ss\}$ if $x \in \bigcap \mathcal{S}(\mathbf{F})$.

Before we embark on the question of equivalence, we briefly survey some links between argumentation, graph theory and logic. We start with graph theory. Given a directed graph (digraph) $\mathbf{D} = \langle D, N \rangle$ with $N \subseteq D \times D$, a set $K \subseteq D$ is said to be a *kernel* in \mathbf{D} if:

$$N^-(K) = D \setminus K$$

Kernels were introduced by Von Neumann and Morgenstern in the 1940s [15] in the context of cooperative game theory and they have later attracted a fair bit of interest from graph-theorists, see [2] for a recent overview. The connection to argumentation should be apparent. If we let $\overleftarrow{\mathbf{D}}$ denote the digraph obtained by reversing all edges in \mathbf{D} , then it is not hard to verify that a kernel in \mathbf{D} is a stable set in $\overleftarrow{\mathbf{D}}$ and vice versa.

In kernel theory, one also considers *semikernels* [12], which are sets $L \subseteq D$ such that

$$N^+(L) \subseteq N^-(L) \subseteq D \setminus L$$

It is easy to verify that a semikernel in \mathbf{D} is an admissible set in $\overleftarrow{\mathbf{D}}$ and vice versa. In the context of graph theory, several interesting results and techniques have been found, especially concerning the question of finding structural conditions that ensure the *existence* of kernels, see e.g., [11, 6, 7]. In our view, the connection to argumentation has not received the attention it deserves, although it has been mentioned, for instance in [5]

The second link we wish to present is with classical logic and classical consistency. This link is implicit already in much work done on argumentation, but as far as we are aware, it has only recently been pointed out that argumentation frameworks and the stable actually provide an equivalent formulation of classical propositional logic [10]. We would like to stress this point a little, since it shows that when we study structural conditions that ensure preservation of extensions based on admissible sets under mappings between frameworks, we are also studying - from a novel point of view - conditions that ensure preservation of classical consistency of theories.

For a formal account of the connection we have in mind, we refer to [1]. There the authors show that digraphs provide a normal form for propositional theories such that an assignment is satisfying for a theory iff it gives rise to a kernel in

the corresponding digraph [1]. They introduce, in particular, a new normal form for propositional logic, called the *graph normal form*, where a formula ϕ is said to be in graph normal form iff $\phi = x \leftrightarrow \bigwedge_{y \in X} \neg y$ for propositional letters $\{x\} \cup X$. It is shown that it is indeed a normal form for propositional logic - every propositional theory has an equisatisfiable one containing only formulas of this form.² The connection between theories in graph normal form and argumentation frameworks is quite obvious, and obtaining a theory from an argumentation framework is particularly easy; given a framework F , we simply form the following set of equivalences:

$$\text{TF} = \{x \leftrightarrow \bigwedge_{y \in \mathcal{R}^-(x)} \neg y \mid x \in \mathcal{A}\} \quad (2.2)$$

We adopt the convention that $x \leftrightarrow \bigwedge \emptyset$ is a tautology, and then it is easy to see that an assignment $\Gamma : \mathcal{A} \rightarrow \{0, 1\}$ is a satisfying assignment for TF iff $S_\Gamma = \{x \in \mathcal{A} \mid \Gamma(x) = 1\}$ is a stable set in F . Going the other way, from theories in graph normal form to argumentation frameworks, is also straightforward, but for the details we refer to [1] (the construction is presented with respect to directed graphs, so edges must be reversed for argumentation).

So we have an immediate formal expression of the conceptual link between stable sets in argumentation and classical consistency. The difference is only a matter of *perspective*, and it is our belief that both the combinatorial perspective offered by directed graphs, and the procedural, somewhat pragmatic, perspective offered by argumentation, can serve to enhance our understanding of classical intuitions. Also, while the stable semantics expresses full classical consistency, i.e., consistency of the theory corresponding to the whole framework, other semantics based on admissible sets can be seen as identifying consistent subparts of a framework/theory that satisfy certain additional properties. To see this, it is enough to note that if $A \in a(F)$ is an admissible set in F , then it is a stable set in the subframework of F induced by $A \cup \mathcal{R}^+(S)$, so it corresponds to a satisfying assignment to the theory which represents this subframework. The upshot is that *all* semantic notions expressed in Definition 2.1 are based on, and expand upon, a notion of consistency that is essentially classical. This provides a fresh point of view, and we think it is particularly interesting to ask about preservation of various forms of consistency under relations between frameworks, not only because it is relevant for abstract argumentation, but also because it addresses consistency in classical logic from a new perspective.

3 A General Notion of Equivalence

Consider two arbitrary attack-cycles of even length, say F and F_2 depicted in Figure 1. How do we reason semantically about an even length attack-cycle? Well, suppose that the argument x_1 from F has some proponent. Then this

²Equisatisfiable means that for every satisfying assignment to one there is a satisfying assignment to the other, i.e., the assignments are not necessarily the same (new propositional letter might need to be introduced)

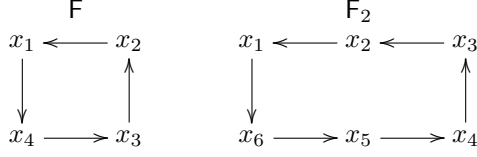


Figure 1: Two even cycles

proponent will probably recognize that his argument is attacked by the argument x_2 , and, most likely, he will then become a proponent of argument x_3 , recognizing that this argument attacks x_2 and therefore defends x_1 . In F , this is when the story stops, since the proponent notices at this point that although x_4 attacks x_3 , it is in turn attacked by x_1 . In F_2 , the story is basically the same; a proponent of x_1 realizes he should also support x_3 , but now, since the cycle is longer, he also comes to support x_5 .

The observation we want to make is that while the length of cycles F and F_2 differ, they are still similar. So similar, in fact, that it seems completely natural - at this level of abstraction - to say that they are semantically the same. More generally, it seems that whatever an even length cycle has to tell us with respect to any semantical notion from Definition 2.1 has been told already by this one: $x \rightleftarrows y$. Essentially, all even cycles behave the same way; they are different manifestations of exactly the same argumentation scenario. Unfortunately, the notion of equivalence adopted in the literature on argumentation does not allow us to conclude this; even cycles of different length do not have the same set of extensions under any reasonable semantics.

The case of even cycles seems to illustrate in a very simple way why the current notion of equivalence used in argumentation is too restrictive. It relies on a crude syntactic criterion requiring extension - semantic in nature - to be syntactically the same. In light of this, we believe that the following notion of equivalence should be investigated. It seems completely natural and is determined not by looking for syntactic identity between sets of arguments, but by looking for sets of arguments that can be grouped together upon noting that they have the same semantic status.

Definition 3.1 *Given two argumentation frameworks F and F_2 , we say that they are equivalent with respect to $\mathcal{S} \in \{a, c, g, p, s, ss\}$, and we write $F \equiv^{\mathcal{S}} F_2$, if there is a relation $\alpha \subseteq \mathcal{A} \times \mathcal{A}_2$ such that*

- *If $A \in \mathcal{S}(F)$, then $\alpha(A) \in \mathcal{S}(F_2)$ - the relation preserves extensions*
- *If $A_2 \in \mathcal{S}(F_2)$ then $\alpha^{-}(A_2) \in \mathcal{S}(F)$ - the relation reflects extensions*

If $\alpha \subseteq \mathcal{A} \times \mathcal{A}_2$ witnesses to the equivalence of F and F_2 , we say that α is an equivalence relation. For the case of even cycles, it is easy to see that this definition is adequate. It allows us to state formally what our intuition told us to be the case, namely $F \equiv^{\mathcal{S}} F_2$ for all $\mathcal{S} \in \{a, c, g, p, s, ss\}$. The relation

$\alpha = \{(x_1, x_1), (x_1, x_3), (x_2, x_2), (x_2, x_4), (x_3, x_5), (x_4, x_6)\}$, for instance, is easily seen to be an equivalence relation with respect to all $\mathcal{S} \in \{a, c, g, p, s, ss\}$. Indeed, for arbitrary even cycles $x_1 \dots x_{2i} x_1$, it is easy to see that for all $\mathcal{S} \in \{a, c, g, p, s, ss\}$ they are all equivalent to each other. In particular, they are equivalent to the even cycle $x_1 x_2 x_1$, witnessed by the equivalence relation $\alpha = \bigcup_{1 \leq i \leq n} \{(x_1, x_{2i-1}), (x_2, x_{2i})\}$.

3.1 First Observation: Skeptical and Credulous Acceptance

The first observation we would like to make regarding Definition 3.1 is that - unsurprisingly - equivalences preserve and reflect skeptical and credulous acceptance of arguments. It is clear, in particular, that if $F \equiv^{\mathcal{S}} F_2$ and $S \subseteq \mathcal{A}$ is a set of skeptically accepted arguments from F , then $\alpha(S)$ is a skeptically accepted set of arguments in F_2 (and similarly for the inverse α^-). Also, if $C \subseteq \mathcal{A}$ is a set of credulously accepted arguments, then for each $x \in C$, we have $S_x \in \mathcal{S}(F)$ such that $x \in S_x$, and since $\alpha(S_x) \in \mathcal{S}(F_2)$ by α being an equivalence, it follows that $\alpha(C)$ is a set of credulously accepted arguments in F_2 as well. More is true, however, and what our definition of equivalence ensures is that the *logical* properties of frameworks are preserved. For instance, if one of the logical properties of F is that all extension under \mathcal{S} containing $x \in \mathcal{A}$ must also contain $y \in \mathcal{A}$, the same relationship obtains between all $x_2 \in \alpha(x)$ and all $y_2 \in \alpha(y)$. We obtain, in particular, two collections of equivalent arguments in F_2 such that one logically implies the other. Then the benefit of having defined equivalence as in Definition 3.1 becomes clear; since our notion of an equivalence does not impose any restrictions on what the relation must look like, we can investigate logical properties of complex frameworks by looking for equivalences with more simple frameworks that have already been analyzed.

3.2 Second Observation: Collapse with respect to the Single-Status Semantics

The second observation we will make is almost as trivial as the first, but might make the notion of equivalence introduced in Definition 3.1 somewhat controversial to the argumentation community. Consider, in particular, two frameworks F and F_2 and a semantics $\mathcal{S} \in \{a, c, g, p, s, ss\}$ such that both F and F_2 have a *unique* extension $\{S\} = \mathcal{S}(F), \{S_2\} = \mathcal{S}(F_2)$. If we assume both S, S_2 to be non-empty, it is obvious that we can always construct a relation $\alpha \subseteq \mathcal{A} \times \mathcal{A}_2$ such that $\alpha(S) = S_2$ and $\alpha^-(S_2) = S$, allowing us to conclude that $F \equiv^{\mathcal{S}} F_2$.

With respect to the grounded extension, which always gives rise to a unique extension, this means that all frameworks fall into one of two classes; those that admit non-empty grounded extensions and those that do not. More is true, since it is well known, see e.g., [8], that for any two non-empty *finite acyclic* frameworks, all semantics from Definition 2.1 coincide and deliver a unique non-empty extension - the grounded one. This means, in particular, that with equivalence conceived of as in Definition 3.1, all finite, non-empty, acyclic frameworks are

equivalent. Also, we note that other semantics for argumentation have also been proposed that always yield a unique extension - they are called *single-status* in the literature. In light of this, the collapse of frameworks with respect to all such semantics might disconcert some, but to us it signals only that we have arrived at a notion of equivalence that is appropriate. It allows us to abstract away from superficial syntactical differences and focus instead on genuine semantic problems.

The grounded semantics for argumentation is particularly trivial; the grounded extension can always be computed in linear time (iterate $\mathcal{D}()$ from Definition 2.1, starting from the set, U , of unattacked arguments), and it contains arguments that, intuitively speaking, cannot be disputed by any rational agent. Indeed, if a semantics for argumentation was proposed that did not include the grounded extension as a subset of all extensions, it would probably be dismissed without further comment. But in some sense - and we believe it is the most relevant sense - all single-status semantics are trivial. They leave no room for dispute, no contingency, and, most critically, no interesting dependencies between arguments. Such semantics simply pick a set, and it seems clear that the interesting question, and the only possible source of non-triviality, lies in *how* the set is chosen. Clearly, if this is something more than an arbitrary choice, it must involve other notions, and it is these notions - which typically *do* involve interesting dependencies - that are truly semantic in nature and deserve attention. The point we are trying to make is beautifully illustrated by the so-called *ideal semantics* [9]. The ideal set of arguments is the maximal set of arguments that is contained in all preferred extensions. As such, the ideal semantics should, in our opinion, not be seen as a separate semantics at all, but just as a new notion of acceptance for preferred semantics, asking you to accept an argument only if it is skeptically accepted and is also in an admissible set which contains only skeptically accepted arguments (since defense is preserved under union and the set of skeptically accepted arguments is conflict-free, the set of all such arguments will obviously be the maximal admissible subset of skeptically accepted arguments). It seems to us that the relevant notion of equivalence is still the one which preserves and reflects preferred sets - there is nothing you can say about the ideal set and what it captures unless you make reference to the notion of a preferred set.³

We remark that the collapse with respect to single-status semantics has an obvious generalization, allowing us to conclude that any two frameworks with exactly $n \in \mathbb{N}$ disjoint extensions under some semantics are equivalent with respect to that semantics. Any two such frameworks are equivalent, as they should be, because there is a way to associate arguments such that a one-to-one correspondence between the extensions of these frameworks will result.

Thinking of arguments as propositional formulas (remember the discussion in Section 2), makes for a further argument in favor of the possibly controversial point of view that we adopt here. What single-status approaches provide us

³We mention that we can impose the same restriction starting from semi-stable semantics, leading to the *eager* set [4]

with is basically a set of tautologies - arguments that cannot be disputed. In a logical sense, any two collections of tautologies are equivalent, and they should be; no questions arise at all about how their semantic status is dependent on that of other formulas, the point being precisely that no such dependencies influence their status as indisputable. It seems clear, therefore, that a collection of arguments that cannot be disputed should be regarded as logically equivalent to any other such collection, in exactly the same way as a collection of tautologies of some logical language is equivalent to any other such collection. What is interesting about tautologies is how to locate them, and the general notion of equivalence is potentially useful in this regard precisely because it does not care what they look like. That way, it becomes possible to look for relations that allows simplification of the framework under consideration, potentially simplifying the search for tautologies. For the finite case and semantics based on admissible sets, this is only a relevant consideration for cyclic frameworks, however, since the search for tautologies in a finite acyclic frameworks is already completely trivial.

3.3 Third Observation: Structural Conditions Needed

We have introduced a new semantic notion of equivalence between frameworks, and argued that it is the appropriate notion that we want to work with when we consider two frameworks and ask about the relationship between them. Some might object that it is too abstract, referring to how it conflates frameworks with respect to the grounded semantics and in any unique status situation. But as we have tried to argue above, we actually believe that such a conflation is in order when we work at a high level of abstraction. For the case of the grounded extension, in particular, it seems to us that there is not much more to be said about it at the level of abstraction that we address. The grounded extension might be very useful in applications, and it might be possible to focus on more intermediate levels of abstraction where some, but not all implementation-specific aspects are studied. But from the point of view of *pure* abstract argumentation, as introduced by Dung, we are bold enough to suggest that the grounded extension is perhaps properly understood already. What is not understood, however, not even at a high level of abstraction, is the notion of an admissible set; in particular, we do not seem to have a clear understanding of *when* non-empty such sets can be found, *why* they sometimes fail to exist, and *how* we best should go about locating them. As discussed earlier, this question hinges on the notion of *consistency*, in various forms and guises. If the question is simply whether or not a framework admits a stable set, the question becomes that of deciding classical consistency, as discussed above in Section 2. But when we make the move to consider admissible sets, we are free to also reason about and locate consistent sub-parts of a system that could, as a whole, be inconsistent. However, since what - in terms of structural properties - leads to inconsistency in argumentation frameworks is not properly understood, it is also difficult to pin down where the problem lies, with repercussion also for what exactly the non-stable semantics contribute in such cases. A fundamental, overreaching re-

search goal - as we see it - should be to attempt giving an account of this by combinatorial means.

We think it is obvious that in this regard, the notion provided by Definition 3.1 is appropriate and should be considered. Still, it only states what an equivalence is, not how to find one. Unless we can establish some structural properties on relations that ensure that they are equivalences, it would be fairly useless, pointing only to an unattainable ideal that would have to be replaced by more pragmatic notions in practice. In the following section, however, we present first results on this, exploring the notion of bisimulation.

4 Bisimulation and Equivalence in Argumentation

In this section, we first work with a standard notion of bisimulation, and show that if equivalence with respect to admissible semantics is witnessed by a bisimulation, we can conclude equivalence also for some (but not all) semantics based on admissible sets. Then we add a further requirement to bisimulations - introducing finitely collapsing bisimulations - and we show that they are equivalences with respect to all the semantics we consider in this paper.

Definition 4.1 *Given argumentation frameworks F and F_2 , a relation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ is said to be a bisimulation if we have:*

forth: *For every $x \in \mathcal{A}$, $y \in \mathcal{R}^-(x)$, for all $x_2 \in \beta(x)$ there is $y_2 \in \mathcal{R}_2^-(x_2) \cap \beta(y)$*

back: *For every $x_2 \in \mathcal{A}_2$, $y_2 \in \mathcal{R}_2^-(x_2)$, for all $x \in \beta^-(x_2)$ there is $y \in \mathcal{R}^-(x) \cap \beta^-(y_2)$*

Notice that the definition asks for mutual simulation of *incoming* attacks. For $\mathcal{S} \in \{a, c, p, s, ss\}$, it is not hard to see that bisimulations are neither necessary nor sufficient for equivalence. The problem is that a bisimulation ensures only that attacks between arguments are preserved and reflected, but does not ensure that attacks are absent when they need to be in order to ensure conflict-freeness. It is easy to see, for instance, that an even cycle is bisimilar to a single self-attacking argument, and these two frameworks are only equivalent under the grounded semantics. We have the following easy fact, however, stating that bisimulation behaves nicely when it comes to defense.

Fact 4.2 *Assume we have frameworks F, F_2 and some bisimulation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$. Then we have*

- (1) *For all $A \subseteq \mathcal{A}$, $\beta(\mathcal{D}(A)) = \mathcal{D}(\beta(A))$ - β preserves defended arguments*
- (2) *For all $A_2 \subseteq \mathcal{A}_2$, $\beta^-(\mathcal{D}(A_2)) = \mathcal{D}(\beta^-(A_2))$ - β reflects defended arguments*

PROOF. (1) We consider arbitrary $A \subseteq \mathcal{A}$ and prove the claim by showing both inclusions.

(\subseteq) Consider arbitrary $y \in \mathcal{D}(A)$, $y_2 \in \beta(y)$ and $z_2 \in \mathcal{R}_2^-(y_2)$. Then by β being a bisimulation (back), it follows that there is some $z \in \beta^-(z_2)$ such that $z \in \mathcal{R}^-(y)$. Since $y \in \mathcal{D}(A)$ it follows that there is $x \in A$ such that $x \in \mathcal{R}^-(z)$. Then by β being a bisimulation (forth) it follows that there is some $x_2 \in \beta(x)$ such that $x_2 \in \mathcal{R}_2^-(z_2)$, meaning $z_2 \in \mathcal{R}_2^-(\beta(A))$. We conclude $y_2 \in \mathcal{D}(\beta(A))$ as desired.

(\supseteq) Consider arbitrary $y_2 \in \mathcal{D}(\beta(A))$, $y \in \beta^-(y_2)$ and $z \in \mathcal{R}^-(y)$. Then by β being a bisimulation (forth), it follows that there is some $z_2 \in \beta(z)$ such that $z_2 \in \mathcal{R}^-(y_2)$. Since $y_2 \in \mathcal{D}(\beta(A))$, it follows that there is some $x_2 \in \beta(A)$ such that $x_2 \in \mathcal{R}^-(z_2)$. From β being a bisimulation (back), it follows that there is some $x \in \beta^-(x_2)$ such that $x \in \mathcal{R}^-(z)$. It follows that $y \in \mathcal{D}(A)$, meaning that $y_2 \in \beta(\mathcal{D}(A))$ as desired.

(2) The argument is symmetric to that used to show (1). \square

We note that a trivial corollary of this is that bisimulations are equivalences with respect to the grounded semantics. The next result concerns the relationship between various semantics. We ask, in particular, if equivalences that are also bisimulations will automatically preserve and reflect extensions for more than one type of semantics from Definition 2.1 at once. We show, in particular, that if an equivalence with respect to admissible sets is also a bisimulation, then it is also an equivalence with respect to preferred, stable and semi-stable semantics, yet *not* with respect to the complete semantics.

Theorem 4.3 *Given frameworks F and F_2 , if $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ is a bisimulation, then if β preserves and reflects admissible sets, it also preserves and reflects preferred, semi-stable and stable sets.*

PROOF. For all semantics, we only show preservation. Reflection can be shown symmetrically.

Stable: Assume that $S \subseteq \mathcal{A}$ is stable. We know $\beta(S)$ is conflict-free and must show $\mathcal{A}_2 \setminus \beta(S) = \mathcal{R}_2^+(\beta(S))$. Consider arbitrary $x_2 \in \mathcal{A}_2 \setminus \beta(S)$. Then $\beta^-(x_2) \subseteq \mathcal{A} \setminus S$, so there is $y \in S$ such that $y \in \mathcal{R}^-(\beta^-(x_2))$. By β being a bisimulation ("forth"), we have $x_2 \in \mathcal{R}_2^+(\beta(S))$ as desired.

Preferred: Assume that $S \subseteq \mathcal{A}$ is preferred. Then $\beta(S)$ is admissible. Assume towards contradiction that there is $A_2 \supset \beta(S)$ which is admissible in F_2 . Then $\beta^-(A_2)$ is admissible in F and since $\beta(\beta^-(A_2)) \supseteq A_2 \supset \beta(S)$, we have $\beta^-(A_2) \supset S$, contradiction.

Semi-stable: Assume that $S \subseteq \mathcal{A}$ is semi-stable, i.e. that S is admissible, and that there is no admissible $A \subseteq \mathcal{A}$ such that $S \cup \mathcal{R}^+(S) \subset A \cup \mathcal{R}^+(A)$. Assume towards contradiction that $\beta(S)$ is not semi-stable. Then there is $S_2 \subseteq \mathcal{A}_2$ such that a) $S_2 \cup \mathcal{R}_2^+(S_2) \supset \beta(S) \cup \mathcal{R}_2^+(\beta(S))$. By β being a bisimulation ("forth"), we have b) $\beta(\mathcal{R}^+(S)) \subseteq \mathcal{R}_2^+(\beta(S))$ and also ("back") that c) $\beta^-(\mathcal{R}_2^+(S_2)) \subseteq \mathcal{R}^+(\beta^-(S_2))$. We will show that $\beta^-(S_2 \cup \mathcal{R}_2^+(S_2)) = \beta^-(S_2) \cup \beta^-(\mathcal{R}_2^+(S_2)) \supset$

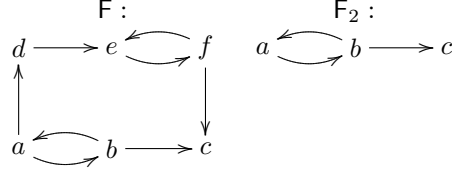


Figure 2: Frameworks F, F_2 such that we have $F \equiv^S F_2$ for $S \in \{g, a, p, s, ss\}$ but $F \not\equiv^c F_2$

$S \cup \mathcal{R}^+(S)$, which is a contradiction since it allows us to conclude, by applying c), that $\beta^-(S_2) \cup \mathcal{R}^+(\beta^-(S_2)) \supset S \cup \mathcal{R}^+(S)$. We show inclusion first.

$$\begin{aligned}
 \beta^-(S_2 \cup \mathcal{R}_2^+(S_2)) &\stackrel{a)}{\supseteq} \beta^-(\beta(S) \cup \mathcal{R}_2^+(\beta(S))) \\
 &= \beta^-(\beta(S)) \cup \beta^-(\mathcal{R}_2^+(\beta(S))) \\
 &\stackrel{b)}{\supseteq} \beta^-(\beta(S)) \cup \beta^-(\beta(\mathcal{R}^+(S))) \\
 &\supseteq S \cup \mathcal{R}^+(S)
 \end{aligned}$$

To show that the inclusion is strict, consider $x_2 \in (S_2 \cup \mathcal{R}_2^+(S_2)) \setminus (\beta(S) \cup \mathcal{R}_2^+(\beta(S)))$. For arbitrary $x \in \beta^-(x_2)$, observe first that since $x_2 \notin \beta(S)$, we have $x \notin S$. We also have $x_2 \notin \mathcal{R}_2^+(\beta(S))$ and from b) it follows that $x_2 \notin \beta(\mathcal{R}^+(S))$. Then we conclude that $x \notin \mathcal{R}^+(S)$. \square

Interestingly, a bisimulation that preserves and reflects admissible sets might not preserve complete sets, as shown by the frameworks F and F_2 in Figure 2. Here, we have the bisimulation $\beta = \{(a, a), (e, a), (b, b), (d, b), (f, b), (c, c)\}$ which is also an equivalence with respect to the admissible semantics. We notice, however, that $\{a\}$ is a complete set in F while $\beta(a) = \{a\}$ is not complete in F_2 since d is defended by $\{a\}$.

As mentioned, the intuitive reason why bisimulations do not preserve extensions is that they do not preserve conflict-freeness. Still, they fail to do so only in specific circumstances. To see how this works, assume that you have two arguments a, b in some framework F such that a and b are not in any conflict, and that you then relate them by a bisimulation β to some a_2, b_2 in F_2 with $b_2 \in \mathcal{R}^-(a_2)$. It then follows by β being a bisimulation (back), that there must be some $c \in \beta^-(b_2)$ such that $c \in \mathcal{R}^-(a)$. So an attacker of a , the argument c , was merged with a non-attacker of a , the argument b . So this type of collapse has to occur when bisimulations fail to be equivalences. It makes sense, then, to see what happens if we attempt to limit it by introducing a further requirement. In particular, we will investigate what happens when we do not allow the collapse of any two disjoint infinite backwards walks.

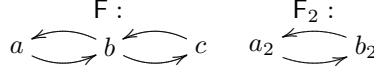


Figure 3: Two fc-bisimilar argumentation frameworks

Definition 4.4 Given two frameworks F and F_2 , a bisimulation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ is finitely collapsing if the following holds:

global forth: For every backwards infinite walk $\lambda = x_1 x_2 x_3 \dots$ in F_2 , there exists some $i \in \mathbb{N}$ such that $|\beta^-(x_i)| = 1$

global back: For every backwards infinite walk $\lambda = x_1 x_2 x_3 \dots$ in F , there exists some $i \in \mathbb{N}$ such that $|\beta(x_i)| = 1$

For short we will call bisimulations that are finitely collapsing fc-bisimulations. As an example, consider the frameworks in Figure 3. They are fc-bisimilar witnessed by $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ where $\beta(a) = a_2, \beta(b) = b_2, \beta(c) = a_2$.

The main result in this paper now follows. It shows that fc-bisimulations are equivalences with respect to all semantics in Definition 2.1. We remark that it is sufficient to show that fc-bisimulations preserve and reflect admissible and complete sets, from which it follows by Theorem 4.3 that they also preserve and reflect preferred, stable and semi-stable sets.

Theorem 4.5 Given frameworks F and F_2 , if there is an fc-bisimulation $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$, then $F \equiv^S F_2$ for all $S \in \{s, a, p, ss, c\}$

PROOF. Admissible sets: Let $\beta \subseteq \mathcal{A} \times \mathcal{A}_2$ be an arbitrary fc-bisimulation. We show that β preserves admissible sets. Then, by symmetry, β also reflects them, since the inverse of β , $\beta^- \subseteq \mathcal{A}_2 \times \mathcal{A}$ is clearly also an fc-bisimulation. Let $E \subseteq \mathcal{A}$ be an admissible set in F and consider $E_2 = \beta(E)$. If $x_2 \in \mathcal{R}_2^-(y_2)$ for $y_2 \in E_2$, then there is $y \in E$ such that $y_2 \in \beta(y)$, and by β being a bisimulation ("back"), there is some $x \in \mathcal{R}^-(y)$ such that $x_2 \in \beta(x)$. Since E defends itself, it follows that there is $z \in \mathcal{R}^-(x) \cap E$. Then, by β being a bisimulation ("forth"), it follows that there is some $z_2 \in \mathcal{R}_2^-(x_2)$ such that $z_2 \in \beta(z)$, meaning $z_2 \in E_2$. This shows that $E_2 \subseteq \mathcal{D}(E_2)$. To show that E_2 is conflict free, assume towards contradiction that there is $x_2, b' \in E_2$ with $x_2 \in \mathcal{R}_2^-(b')$. Then, by definition of E_2 , there is $x, b \in E$ with $x_2 \in \beta(x)$ and $b' \in \beta(b)$. Also, we know that $x \notin \mathcal{R}^-(b)$ since E is conflict-free. But by β being a bisimulation ("back"), there must be $z \in \mathcal{R}^-(b)$ such that $x_2 \in \beta(z)$. Since E is conflict-free, we know that $z \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$. Now we have $x_2 \in E_2 \cap \beta(x) \cap \beta(z)$ such that z attacks E , and this is the first step towards showing that there exists an infinite backwards walk $\lambda = y_1 y_2 y_3 \dots$ in \mathcal{A}_2 such that for all $i \geq 1$, we have $|\beta^-(y_i)| \geq 2$. This will contradict the assumption that β is an fc-bisimulation ("global forth"). We take $y_1 = x_2$ and let $w_1 = x, v_1 = z$. Then for all $i \geq 2$, we define y_i, w_i, v_i inductively, assuming that $y_{i-1}, w_{i-1}, v_{i-1}$ have been defined such that $w_{i-1} \in E, v_{i-1} \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$ and $y_{i-1} \in \beta(w_{i-1}) \cap \beta(v_{i-1})$. The

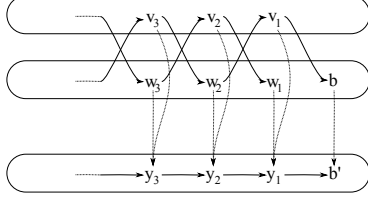


Figure 4: Illustrating the construction of $\lambda = y_1y_2y_3 \dots$

construction is visualized in Figure 4. Since E defends itself against all attacks, we can find $w_i \in E \cap \mathcal{R}^-(v_{i-1})$. Since we have $y_{i-1} \in \beta(v_{i-1})$ it follows by β being a bisimulation ("forth") that we can find $y_i \in \beta(w_i) \cap \mathcal{R}^-(y_{i-1})$. But we also have $y_{i-1} \in \beta(w_{i-1})$, so by β being a bisimulation ("back"), we find $v_i \in \beta^-(y_i) \cap \mathcal{R}^-(w_{i-1})$. Since $w_{i-1} \in E$ and E is conflict-free, it follows that $v_i \in \mathcal{R}^-(E) \subseteq \mathcal{A} \setminus E$. So y_i, w_i, v_i can be found for all $i \in \mathbb{N}$, proving existence of λ that contradicts "global forth".

Complete sets: We know that β preserves and reflects admissible sets, and now we assume that $S \subseteq \mathcal{A}$ is complete. Consider arbitrary $x_2 \in \mathcal{A}_2 \setminus (\beta(S) \cup \mathcal{R}_2^+(\beta(S)))$. By β being a bisimulation ("forth"), we get $\beta^-(x_2) \cap \mathcal{R}^+(S) = \emptyset$, which implies $\beta^-(x_2) \subseteq \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$. Then, since S is complete, there is $y \in \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$ such that $y \in \mathcal{R}^-(\beta^-(x_2))$. Then, since β is a bisimulation ("forth"), it follows that there is $y_2 \in \beta(y) \cap \mathcal{R}_2^-(x_2)$. Since $x_2 \notin \mathcal{R}_2^+(\beta(S))$ it follows that $y_2 \notin \beta(S)$. Assume towards contradiction that $y_2 \in \mathcal{R}_2^+(z_2)$ for some $z_2 \in \beta(S)$. Then there is $z \in S \cap \beta^-(z_2)$ and also, since β is a bisimulation ("back"), there is $z' \in \mathcal{R}^-(y) \cap \beta^-(z_2)$. Since $y \notin \mathcal{R}^+(S)$, $z' \notin S$. Since β is a bisimulation ("forth") and $z_2 \in \beta(S)$ and $\beta(S)$ is conflict-free, $z' \notin \mathcal{R}^+(S)$. It follows that $z' \in \mathcal{A} \setminus (S \cup \mathcal{R}^+(S))$. To contradict global forth, we prove existence of a backwards infinite walk $\lambda = x_1x_2x_3 \dots$ in F_2 such that for all $i \geq 1$ we have $|\beta^-(x_i)| \geq 2$. We take $x_1 = z_2$, $v_1 = z'$, $w_1 = z$ and for all $i \geq 2$, we assume that we have $x_{i-1}, v_{i-1}, w_{i-1}$ with $x_{i-1} \in \beta(S) \cup \mathcal{R}_2^+(\beta(S))$ and $w_{i-1} \in (S \cup \mathcal{R}^+(S)) \cap \beta^-(x_{i-1})$, $v_{i-1} \in (\mathcal{A} \setminus (S \cup \mathcal{R}^+(S))) \cap \beta^-(x_{i-1})$. There are two cases.

I) $x_{i-1} \in \beta(S)$. Then since $\beta(S)$ is admissible and $w_{i-1} \in \beta^-(x_{i-1})$, we have $w_{i-1} \notin \mathcal{R}^+(S)$ by β being a bisimulation ("forth"). Since S is complete, we find $v_i \in \mathcal{R}^-(v_{i-1}) \cap (\mathcal{A} \setminus (S \cup \mathcal{R}^+(S)))$. Since β is a bisimulation ("forth"), we find $x_i \in \mathcal{R}_2^-(x_{i-1}) \cap \beta(v_i)$, and since $\beta(S)$ is admissible, $x_i \in \mathcal{R}_2^+(\beta(S))$. Then, going back, we find $w_i \in \beta^-(x_i) \cap \mathcal{R}^-(w_{i-1})$, and since $w_{i-1} \in S$ and S is admissible, $w_i \in \mathcal{R}^+(S)$.

II) $x_{i-1} \in \mathcal{R}_2^+(\beta(S))$. Since $w_{i-1} \in \beta^-(x_{i-1}) \cap (S \cup \mathcal{R}^+(S))$ and $\beta(S)$ is admissible, we have $w_{i-1} \in \mathcal{R}^+(S)$. We choose $w_i \in S \cap \mathcal{R}^-(w_{i-1})$. By β being a bisimulation ("forth"), we find $x_i \in \beta(w_i) \cap \mathcal{R}_2^-(x_{i-1})$ and ("back") $v_i \in \beta^-(x_i) \cap \mathcal{R}^-(v_{i-1})$. Since $v_{i-1} \notin \mathcal{R}^+(S)$, $v_i \notin S$. Also, by β being a bisimulation ("forth") and $x_i \in \beta(v_i) \cap \beta(S)$ and $\beta(S)$ being conflict-free, we have $v_i \notin \mathcal{R}^+(S)$.

Having established the claim for $\mathcal{S} \in \{a, c\}$, the claim follows by Theorem 4.3

for all $\mathcal{S} \in \{a, c, p, ss, s\}$

□

5 Conclusion

We have addressed the notion of equivalence in abstract argumentation, arguing for a general notion that allows us to consider arbitrary relations between frameworks. We suggested that searching for maps between frameworks that preserve and reflect extensions is worthwhile, and we established a first result on this, introducing finitely collapsing bisimulations and proving that they are equivalences with respect to all the semantics we consider. On a more general note, we suggested that investigating equivalence should be conceived of as part of a direction of research where one attempts to provide graph-theoretical characterizations of various logical properties of argumentation frameworks. We suggested that the notion of *consistency*, in particular, is interesting to look at from a combinatorial point of view. For future work, we hope to be able to identify further structural requirements that ensure relations to be equivalences, and we hope to arrive at a more complete understanding of what structures need to be present in frameworks in order for different semantics for argumentation to actually disagree.

References

- [1] Marc Bezem, Clemens Grabmayer, and Michał Walicki. Expressive power of digraph solvability. *Annals of Pure and Applied Logic*, 163(2):200–212, 2012.
- [2] Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.
- [3] Martin Caminada. Semi-stable semantics. In *Proceedings of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006*, pages 121–130, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.
- [4] Martin Caminada. Comparing two unique extension semantics for formal argumentation: Ideal and eager. In *BNAIC 2007*, pages 81–87, 2007.
- [5] Sylvie Coste-marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *Proc. 8th European Conf. on Symbolic and Quantitative Approaches to Reasoning With Uncertainty (ECSQARU), volume 3571 of LNAI*, pages 317–328. Springer-Verlag, 2005.
- [6] Pierre Duchet. Graphes noyau-parfaits, II. *Annals of Discrete Mathematics*, 9:93–101, 1980.

- [7] Pierre Duchet and Henry Meyniel. Une généralisation du théorème de Richardson sur l'existence de noyaux dans les graphes orientés. *Discrete Mathematics*, 43(1):21–27, 1983.
- [8] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
- [9] P.M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(1015):642 – 674, 2007.
- [10] Sjur Dyrkolbotn and Michał Walicki. Propositional discourse logic. (*submitted*). www.ii.uib.no/~michal/graph-paradox.pdf.
- [11] Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.
- [12] Victor Neumann-Lara. Seminúcleos de una digráfica. Technical report, Anales del Instituto de Matemáticas II, Universidad Nacional Autónoma México, 1971.
- [13] Emilia Oikarinen and Stefan Woltran. Characterizing strong equivalence for argumentation frameworks. *Artificial Intelligence*, 175(14–15):1985–2009, 2011.
- [14] Moses Richardson. Solutions of irreflexive relations. *The Annals of Mathematics, Second Series*, 58(3):573–590, 1953.
- [15] John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944 (1947).