A Strongly Complete Logic of Dense Time Intervals

Michał Walicki, Marc Bezem, Wojtek Szajnkenig
Department of Informatics, University of Bergen
PB. 7800, N-5020 Bergen, Norway
{michal,bezem}@ii.uib.no

Abstract

We discuss briefly the duality (or rather, complementarity) of system descriptions based on actions and transitions, on the one hand, and states and their changes, on the other. We settle for the latter and present a simple language, for describing state changes, which is parameterized by an arbitrary language for describing properties of the states. The language can be viewed as a simple fragment of step logic, admitting however various extensions by appropriate choices of the underlying logic. Alternatively, it can be seen as a very specific fragment of temporal logic (with a variant of 'until' or 'chop' operator), and is interpreted over dense (possibly continuous) linear time. The reasoning system presented here is sound, as well as strongly complete and decidable (provided that so is the parameter logic for reasoning about a single state). We give the main idea of the completeness proof and suggest a wide range of possible applications (action based descriptions, active logic, bounded agents), which is a simple consequence of the parametric character of both the language and the reasoning system.

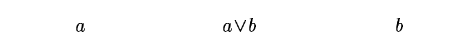
1 Introduction

In the description of reactive systems one has focused primarily on their capability to perform some specific *actions* (process algebras, labelled transition systems, CSP). For example, the famous vending machine can perform the actions of 'accepting a coin' and then 'dispense a coffee' an unspecified number of times. This is certainly a fruitful approach. However, one reason to be interested in actions (and maybe the only one) is that they change the *state* of the world, an agent's beliefs, or any other abstraction. A vending machine can be described equivalently as a device which can stay in a state of 'inactivity', from which it can pass to a state of 'having accepted a coin' and then to one in which it has 'dispensed a coffee'. The latter state may be identified with the state of inactivity if one, among other things, abstracts from the number of coins accepted and from the remaining amount of coffee. These views are in some sense dual, but we present an approach related to the second one, that is, we will describe systems in terms of evolving states. The states evolve over time and during consecutive time intervals certain specifications, that is, partial descriptions of the states, can be observed.

For instance, a description of a system might be as follows. At first an agent knows (or assumes) a. After an announcement he is no longer sure, and knows only $a \vee b$. Finally, after yet another event, he learns that b and retains this knowledge (for the rest of the time: here the scenario ends). This system is described as an expression consisting of a sequence of formulae, each partially describing the state(s), during three consecutive intervals:

$$a; a \vee b; b \tag{1.1}$$

The meaning of this expression could be described on a linear time scale as



Here, lines represent the 'duration' of the state(s) satisfying the formula which annotates it. One further abstraction is that we view 'duration' as something qualitative but not quantitative. Thus all intervals have been given the same length. Throughout the paper we will depict intervals with different lengths, for convenience, not to suggest different durations.

Viewing the above as a description of the result of some interactions (the announcement and the other event), it is natural to ask for possible consequences, and for an entailment relation between such descriptions in general. For example, a system

$$a \vee \neg b; a \vee b \tag{1.2}$$

can be viewed as a consequence of the previous one, with the last two intervals concatenated. Furthermore, $a \wedge b$ during a certain interval has both (1.1) and (1.2) as consequence by appropriately cutting the interval in three and two smaller ones, respectively.

Although the paper takes a logical approach, it is possible to interpret the systems above with the help of labelled transition systems. As a consequence of the focus on states instead of on actions, there is just one label. For example, the transition system corresponding to (1.1) is

$$\begin{tikzcd} a \arrow[loop above] \arrow[r] & a \vee b \arrow[loop above] \arrow[r] & b \arrow[loop above] \arrow[r] & final \end{tikzcd}$$

In our setting, 'being in a state satisfying a specification' is assumed to last for some time, whereas the transitions are instantaneous. The loops in the transition system express this assumption that considered intervals can always be split into smaller ones, that is, the density of the linear ordering modelling the time domain. Transitivity enables the concatenation of intervals. Traces should include the states and are therefore taken to be terminating reduction sequences. Systems are equivalent if they have a trace in common. Equivalent systems have actually infinitely many traces in common, but the trace sets may be different. For example, $a; a$ and a, represented respectively as

$$\begin{tikzcd} a \arrow[loop above] \arrow[r] & a \arrow[loop above] \arrow[r] & final & \text{and} & a \arrow[loop above] \arrow[r] & final \end{tikzcd}$$

are equivalent, but have different trace sets. The consequence relation between systems can also be expressed in terms of transition systems, see section 4.1.

The paper provides the answer to the main question about such systems and their descriptions: given a description (specification) of the sequence of states an agent (or a group of agents) is required to pass through, like (1.1), what other decriptions will be passed through in all cases? In other words, given a language of finite sequences of state-formulae, we ask for an axiomatization which would be strongly complete with respect to intervals of total, dense orderings. Our logic is parameterized by an arbitrary underlying logic of state-formulae. For the sake of illustration we will use propositional logic. Typical examples of possible applications can be obtained from epistemic contexts, like communication protocols or, generally, interacting agents. Due to space limitations, we do not give any more detailed examples, but sketch some possible applications – with particular emphasis on bounded agents – in the last section.

2 Language and Semantics

The logic is parameterized by an arbitrary underlying logic, u.l., which one might want to use for describing the single states. Hence, the language is parameterized by the language of the underlying logic, U.

Definition 2.1 (Sequence Language SL) *The language SL – containing sequence formulae over a parameter language U – is given by the following grammar:*

$$\sigma := U \mid \top \mid \sigma; \sigma$$

$\top$ stands for tautology of the underlying logic – this symbol is added only if it is missing in the underlying logic.

In the sequel $\sigma, \sigma_1, \ldots$ denote (sequence) formulae of SL; $f, f_1 ... g, g_1 ...$ – atomic formulae, i.e., those without ; occurring inside. The formulae of our logic will be simple sequents, i.e., have the form $\sigma_1 \Vdash \sigma_2$. We will denote sequents using $q, q', \ldots$. Complexity of a sequence formulae/sequent refers to the number of _;_ occurring in it.

2.1 Semantics

The semantics is parameterized by the semantics of u.l. Sequence formulae are evaluated over a total, dense ordering which is left-closed and right-open. Given such an ordering $\mathcal{O} = (O, <)$, its points are (mapped to) models of the u.l. An SL-structure is a function

$$r : O \to Mod(u.l.)$$

Left-closedness models the "beginning" (of a computation, or its part), and left-openness its possibly unbounded character. In general, we will consider also subintervals of a whole order $\mathcal{O}$. We denote by $[a, b)$ a *left-closed right-open interval,* $[a, b) = \{o \in O : a \leq o < b\}$, of a given order $\mathcal{O}$. We do not consider empty intervals at all, so the notation $[a, b)$ always implicitly means $a < b$. The satisfaction relation is defined, in general, for any such interval.

Definition 2.2 (Satisfaction Relation) *Satisfaction of an SL-formula σ in an SL-structure, written $[a,b) \models_{\mathcal{O},r} \sigma$, is defined as follows:*

1. $[a, b) \models_{\mathcal{O},r} \top$ *for all* $[a, b)$
2. $[a, b) \models_{\mathcal{O},r} f \iff \forall o \in O : a \leq o < b \Rightarrow r(o) \models_{u.l.} f \quad (f \in u.l.)$
3. $[a, b) \models_{\mathcal{O},r} \sigma_1; \sigma_2 \iff \exists o \in O : a < o < b \;\&\; [a, o) \models_{\mathcal{O},r} \sigma_1 \;\&\; [o, b) \models_{\mathcal{O},r} \sigma_2$

We skip the subscripts in the notation, assuming always given O and r. In fact, we will concentrate on the case where the interval is actually the whole O, writing $r \models \sigma$. This is justified by the following equivalence between (2.1) and (2.2). For a semantical entailment, we write $\sigma_1 \models_{\mathcal{O},r} \sigma_2$ iff

$$\forall \mathcal{O} \forall r \forall [a, b) : [a, b) \models_{\mathcal{O},r} \sigma_1 \Rightarrow [a, b) \models_{\mathcal{O},r} \sigma_2 \tag{2.1}$$

Equivalently, we can consider only whole orderings (and not all subintervals):

$$\forall \mathcal{O} \forall r : O \models_{\mathcal{O},r} \sigma_1 \Rightarrow O \models_{\mathcal{O},r} \sigma_2 \tag{2.2}$$

It is an easy exercise to verify that $\_;\_$ is associative. One could view this operator as the until, **U**, of temporal logic over linear time (depending, however, on the details of the definition which may vary). Then, our language could be viewed as a very special subset of temporal logic, where a sequence formula

$$f_1; f_2; f_3; ...; f_n \quad \text{corresponds to} \quad f_1 \mathbf{U} \Big(f_2 \mathbf{U} (f_3 \mathbf{U} ... (f_{n-1} \mathbf{U} \Box f_n) ...) \Big),$$

where the final f_n (and only it) appears always and only under $\Box$, to remain true from then on.[1] Thus it is not surprising that we can express several temporal modalities, for instance:

1. $r \models f$ iff f holds always in r
2. $r \models \top; f$ iff f becomes eventually true and holds then forever
3. $r \models f; \top$ iff f holds initially for at least some time
4. $r \models f; \neg f$ iff f holds for some time, after which $\neg f$ holds forever

Example 2 above admits all the same r's as does 1 but, in addition, also all where f holds almost always, i.e., everywhere with the possible exception of some initial interval. Thus, $f \models \top; f$ but $\top; f \not\models f$. Dually, 3 also allows all models of 1 but also ones where f, holding initially, becomes false after some time, so $f \models f; \top$ but $f; \top \not\models f$. In 4, the requirement is for f to actually become false after some time, never to become true again.

This analogy to temporal logic (of dense linear time) concerns the limited semantic interpretation. However, unlike modal logics, in general, we will offer not only sound and complete, but also strongly complete reasoning, which is also decidable. (All these properties obtain relatively to their presence in the underlying logic.)

[1] Alternatively, one can almost identify $\_;\_$ with the chop operator, common in interval logics, [11, 9]. We have, however, only a very limited fragment of such logics.

The semantics is based on points but, nevertheless, it is strongly interval oriented. For the first, it does not include "point-intervals" (as single points are sometimes called in the interval semantics.) More significantly, although the satisfaction relation is defined relatively to satisfaction in single points, $\models_{u.l.}$, a formula satisfied only at a single point is not satisfied by any interval. For instance, consider $[0,2)$, with all $x \in [0,1) \cup (1,2) : r(x) \models a$ while $r(1) \not\models a$. Then $[0,2) \not\models a$ and $[0,2) \not\models \neg a$. There are some subintervals satisfying a, e.g., $[0,1) \models a$, but there is no subinterval of $[0,2)$ satisfying $\neg a$. This is related also to the phenomenon given in the following example.

Example 2.3 *Assume that u.l. contains propositional disjunction. We may have that $[a,b) \models f \vee g$, while for every subinterval $[c,d) \sqsubseteq_I [a,b) : [c,d) \not\models f$ and $[c,d) \not\models g$.* [2] *Take, for instance, $[a,b)$ such that for any $o \in [a,b), r(o) = (f \vee g) \wedge (\neg f \vee \neg g)$ and, in addition, distribute the models of $\neg f$ densely between those of $\neg g$ and vice versa (i.e., $\forall o,p \in [a,b) : o < p \wedge r(o) \models \neg g \wedge r(p) \models \neg g \Rightarrow \exists q : o < q < p \wedge r(q) \models \neg f$, and the corresponding fact when $r(o), r(p) \models \neg f$.) Then $[a,b) \models f \vee g$ but nowhere, i.e., in no subinterval $[c,d) \sqsubseteq_I [a,b)$, we have that $[c,d) \models f$ or $[c,d) \models g$.*

2.2 Cuts

An equivalent definition of satisfaction can be expressed by saying that an interval $[a,b)$ satisfies a sequence formula $f_1; ...; f_n$ iff it is possible to cut the interval into n subintervals, each left-closed right-open and each satisfying the corresponding f_i. This is the content of the following definition.

Definition 2.4 *An n-*cut *C of $[a,b)$ is a partition of $[a,b)$ into n intervals*

$$[a,b)|_C = [a,o_1)[o_1,o_2)\ldots[o_i,o_{i+1})\ldots[o_n,b)$$

with $a < o_i < o_{i+1} < b$.

By $r|_\sigma$ we denote a cut of r which verifies σ, i.e., shows that $r \models \sigma$.[3]

Two cuts of an interval can be superimposed on each other yielding a (possibly) more refined cut.

Example 2.5 ($r|_{\sigma_1 \& \sigma_2}$) *Consider two cuts, each verifying $\sigma_1 = f_1; f_2; f_3$, respectively, $\sigma_2 = g_1; g_2; g_3$. A possible situation is the following:*

$$\begin{array}{ll} r|_{\sigma_1} & [\ f_1 \mid f_2 \quad f_2 \mid f_3\ ) \\ r|_{\sigma_2} & [\ g_1 \mid g_2 \mid g_3 \quad g_3\ ) \end{array}$$

The result of superimposing these two cuts is as shown below:

$$r|_{\sigma_1 \& \sigma_2} : \qquad \begin{array}{l} [\ f_1 \mid f_2 \mid f_2 \mid f_3\ ) \\ \ \ g_1 \mid g_2 \mid g_3 \mid g_3 \end{array}$$

The following definition formalizes the superposition of two cuts. It will be used only in the situations where each cut verifies a respective sequence formula and, moreover, when we are considering the sequent $\sigma_1 \Vdash \sigma_2$. Hence, although the constructions are symmetric, we mark the asymmetry $\sigma_1 \Vdash \sigma_2$ in the notation. The operation $Paths(\_)$ collects all possible ways of superimposing a cut verifying $\sigma_1 = f_1; ...; f_n$ and one verifying $\sigma_2 = g_1; ...; g_m$. In other words, whenever an interval satisfies both formulae, the superposition of the two cuts will satisfy some path in $Paths(\sigma_1 \Vdash \sigma_2)$, which is defined below. (We also have the opposite implication, see lemma 2.8.)

Definition 2.6 ($Paths()$) *For an arbitrary sequent q, we define $Paths(q)$ by induction on the complexity l of (number of _;_ in) q :*

$l = 0$: *i.e.,* $q = f \Vdash g - Paths(f \Vdash g) := \{[f \vdash g]\}$.

$l > 0$: a. $q = f_1; f_2; \ldots; f_n \Vdash g$

$Paths(q) := \{[f_1 \vdash g] - [f_2 \vdash g] - \ldots - [f_n \vdash g]\}$

[2] $[c,d) \sqsubseteq_I [a,b)$ denotes that $[c,d)$ is a *subinterval* of $[a,b)$, i.e., $[c,d) = \{o \in O : a \le c \le o < d \le b\}$.

[3] This is ambiguous, since there may be many different cuts of r all verifying σ.

b. $q = f \Vdash g_1; g_2; \ldots; g_m$
$Paths(q) := \{[f \vdash g_1] - [f \vdash g_2] - \ldots - [f \vdash g_m]\}$

c. $q = f_1; f_2; \ldots; f_n \Vdash g_1; g_2; \ldots; g_m$
i. $Paths(q) := \{[f_1 \vdash g_1]\} - Paths(f_2; \ldots; f_n \Vdash g_2; \ldots; g_m) \cup$
ii. $\{[f_1 \vdash g_1]\} - Paths(f_1; f_2; \ldots; f_n \Vdash g_2; \ldots; g_m) \cup$
iii. $\{[f_1 \vdash g_1]\} - Paths(f_2; \ldots; f_n \Vdash g_1; g_2; \ldots; g_m)$

Point c. exhausts the possible ways of overlapping of subsequent intervals. Starting with f_1 and g_1, we have the three possibilities illustrated in Figure 1, each corresponding to one of the cases listed under c.

$$\begin{array}{ccc} \begin{array}{l} f_1 \qquad | \; f_2; \ldots; f_n \\ \hline g_1 \; | \quad g_2; \ldots; g_m \end{array} & \begin{array}{l} f_1 \; | \quad f_2; \ldots; f_n \\ \hline g_1 \qquad | \; g_2; \ldots; g_m \end{array} & \begin{array}{l} f_1 \; | \; f_2; \ldots; f_n \\ \hline g_1 \; | \; g_2; \ldots; g_m \end{array} \\ \text{(ii)} & \text{(iii)} & \text{(i)} \end{array}$$

Figure 1: Possible overlapping of initial intervals.

Example 2.7 *$Paths(q)$, for q from example 2.5, are the following:*

$[f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$, *(1)*
$[f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(2)*
$[f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(3)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(4)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_1 \vdash g_3] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(5)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$, *(6)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(7)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_2] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(8)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(9)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$, *(10)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(11)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(12)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_3 \vdash g_1] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$ *(13)*

The path in line (2) corresponds to the cut we obtained in Example 2.5.

A structure r satisfies a path if it satisfies the corresponding sequence formula. For instance, an r satisfies path (2) from the above example iff $r \models f_1 \wedge g_1; f_2 \wedge g_2; f_2 \wedge g_3; f_3 \wedge g_3$.

The following lemma states the observation from example 2.5.

Lemma 2.8 *For all r, σ_1, σ_2 we have that $r \models \sigma_1$ and $r \models \sigma_2$ if and only if there exists a $\pi \in Paths(\sigma_1 \Vdash \sigma_2)$ such that $r \models \pi$.*

3 The reasoning system

Given a sequence formula σ (possibly only a single formula of the underlying logic) we let $\sigma*$ denote σ or an arbitrary extension of σ to a (longer) sequence formula (analogously for $*\sigma$).

Definition 3.1 *The calculus SEC contains the following rule schemata:*

$$\text{(lift)}\ \frac{}{f \Vdash g}\,[f \vdash g] \qquad (L)\ \frac{f* \Vdash \sigma}{f* \Vdash g; \sigma}\,[f \vdash g]$$

$$(E)\ \frac{\sigma_1 \Vdash \sigma_2}{f; \sigma_1 \Vdash g; \sigma_2}\,[f \vdash g] \qquad (R)\ \frac{\sigma \Vdash g*}{f; \sigma \Vdash g*}\,[f \vdash g]$$

If $*$ is empty, the formula to the left of it remains unchanged. The intuition behind these rules is straightforward and concerns the possible overlapping of subsequent intervals. It refers again to the Figure 1, now with (L) corresponding to (ii), (R) to (iii) and (E) to (i). In either case, validity of the conclusion requires validity of $f \vdash g$ (which is placed in the side-condition). The relation between the remaining parts depends on whether the left formula, f, "lasts longer" than g (L), "shorter than" g (R), or if both have equal duration

(E). Axioms are absent because side-conditions will give proof obligations determining if a given derivation is a proof. The rule (lift) terminates construction of a derivation (bottom-up). A derivation is defined in the standard way. The following gives a couple of examples.

Example 3.2 (Derivation) *Consider the sequent q from Example 2.5. The following are two of its possible derivation:*

$$\Delta_q : \quad \begin{array}{lcl} (lift) & \dfrac{}{f_3 \Vdash g_3} & [f_3 \vdash g_3] \\ (E) & \dfrac{f_3 \Vdash g_3}{f_2; f_3 \Vdash g_2; g_3} & [f_2 \vdash g_2] \\ (E) & \dfrac{f_2; f_3 \Vdash g_2; g_3}{f_1; f_2; f_3 \Vdash g_1; g_2; g_3} & [f_1 \vdash g_1] \end{array}$$

$$\Delta'_q : \quad \begin{array}{lcl} (lift) & \dfrac{}{f_3 \Vdash g_3} & [f_3 \vdash g_3] \\ (R) & \dfrac{f_3 \Vdash g_3}{f_2; f_3 \Vdash g_3} & [f_2 \vdash g_3] \\ (L) & \dfrac{f_2; f_3 \Vdash g_3}{f_2; f_3 \Vdash g_2; g_3} & [f_2 \vdash g_2] \\ (E) & \dfrac{f_2; f_3 \Vdash g_2; g_3}{f_1; f_2; f_3 \Vdash g_1; g_2; g_3} & [f_1 \vdash g_1] \end{array}$$

We denote by $Der(q)$ the set of all possible derivations of q. Side-conditions in a derivation constitute the proof obligations. Given a derivation Δ_q, we denote by $po(\Delta_q)$ the sequence of all side-conditions (in the bottom-up order). For the derivations from example 3.2, we have

$$\begin{aligned} po(\Delta_q) &= [f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_3] \\ po(\Delta'_q) &= [f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3] \end{aligned}$$

This operation is extended pointwise to the set of all derivation of a sequent, i.e., $po(Der(q)) = \{po(\Delta) \mid \Delta \in Der(q)\}$.

The following, hardly surprising, lemma states equivalence of the proof obligations obtained over all derivations of a sequent q and the possible overlappings of cuts verifying q.

Lemma 3.3 *For an arbitrary sequent $q : po(Der(q)) = Paths(q)$.*

Definition 3.4 *A derivation Δ_q of $q = f_1; ...; f_n \Vdash g_1; ...; g_m$ is a* proof *of q iff either*

$$\begin{array}{rl} \exists 1 \leq j \leq n : & f_j \vdash \bot \text{ (if } \bot \text{ exists in u.l.)} \quad \text{or} \\ \forall [f_i \vdash g_j] \in po(\Delta_q) : & f_i \vdash g_j \end{array}$$

It is always assumed that, in the underlying logic, $\vdash \top$ and for all $f \vdash \top$.

Example 3.5 *Assuming the underlying logic to be propositional, let the sequent q be as in our previous examples, specialized to actual formulae as follows: $\top; b; \neg b \Vdash \top; \top; \neg b$. Then the derivation Δ_q (from Example 3.2) is a proof of q, but the derivation Δ'_q is not. The latter fails to be a proof, because the side-condition $f_2 \vdash g_3$ is now $b \vdash \neg b$, which does not satisfy the second condition of the definition 3.4.*

Every rule, applied bottom-up, decreases complexity of the sequent. Hence every derivation Δ_q terminates in finitely many steps. Checking if the obtained $po(\Delta_q)$ is a proof is trivially decidable, provided that provability in the underlying logic is decidable. Hence we have a simple, but useful, fact.

Proposition 3.6 *If relation $\vdash$ is decidable, then so is $\Vdash$.*

The next lemma reflects the desired property that we mentioned in the Introduction, i.e., that we do not want to differentiate between f and $f; f$.

Lemma 3.7 *The following rules are admissible.*

$$(\text{id}\Vdash) \frac{* f \star \Vdash \sigma}{* f; f \star \Vdash \sigma} \qquad (\Vdash\text{id}) \frac{\sigma \Vdash * g \star}{\sigma \Vdash * g; g \star}$$

Thus, we can ignore all adjacent duplicates in the considered sequence formulae. (This fact is used to simplify the proof of lemma 3.3.)

Assuming soundness of the underlying logic, one verifies relatively easily soundness of SEC. Lemma 3.3 is of crucial importance in the proof of completeness, and we now sketch the main steps of this proof.

3.1 Completeness

We assume completeness of the underlying logic. We also restrict our attention to the dense ordering of non-negative rationals, $\mathcal{Q}$, but constructions and the final result generalize to arbitrary dense orderings.

Given an interval $[a,b) \sqsubseteq_I \mathcal{Q}$, and $k \geq 2$ models $c_0, ..., c_{k-1}$, one can distribute them densely, i.e., so that for any two points $a \leq o_1 < o_2 < b$, the subinterval $[o_1, o_2)$ contains all models. We register this fact without proof.

Lemma 3.8 *$k \geq 2$ models $c_0, ..., c_{k-1}$ can be distributed densely over an interval $D = [a,b) \sqsubseteq_I \mathcal{Q}$ so that (the image of) every non-empty subinterval $X \sqsubseteq_I D$ contains all models $c_0, ..., c_{k-1}$.*

Theorem 3.9 *When u.l. is strongly complete, then so is SEC.*

PROOF: We show that every unprovable sequent has a counter-model. So assume a sequent q with no proof: $f_1; \ldots; f_n = \sigma_1 \Vdash \sigma_2 = g_1; \ldots; g_m$ (with no adjacent duplicates). By definition 3.4 combined with lemma 3.3, this means that $\forall \pi \in Paths(q)$:

- $\forall [f_i \vdash g_j] \in \pi : f_i \not\vdash \bot$ – all f_i are consistent, AND
- $\exists [f_i \vdash g_j] \in \pi : f_i \not\vdash g_j$.

We construct a model of $f_1; ...; f_n$ by constructing an interval $r_i \models f_i$ for every $1 \leq i \leq n$. We use rationals, so for every $i < n$, we let $r_i = [i-1, i)$, while $r_n = [n-1, \infty)$. For every r_i we assign the models as follows.

For every pair f_i, g_j, the proof obligation $f_i \vdash g_j$ occurs on some derivation path. For each f_i, we collect all g_j's (from all derivation paths) such that $f_i \not\vdash g_j$. (If for some f_i, there are no such g_j's, then we let r_i contain any model of f_i (existing by $f_i \not\vdash \bot$.) We construct r_i as follows:

Since $f_i \not\vdash g_j$ (for each chosen g_j) so, by completeness of u.l., $f_i \not\models g_j$, so we have a counter-model, m_{ij}, for each such pair. We collect all such m_{ij} for a given f_i and distribute them densly in the interval r_i. By lemma 3.8, we then have that, for all j for which we have a counter-model $m_{ij} : r_i \not\models g_j$ (and $\forall s \sqsubseteq_I r_i : s \not\models g_j$).

Concatenating all the intervals $r = r_1; r_2; ...; r_n$, we obtain $r \models f_1; ...; f_n$, with the cut $r_i \models f_i$, which we now fix.

If now $r \models g_1; ...; g_m$ (*) then, by Lemma 2.8, there exists a path $\pi \in Paths(\sigma_1 \Vdash \sigma_2)$ such that for every node $[f_i \vdash g_j] \in \pi$, the respective subinterval $r_{ij} \models f_i \wedge g_j$. However, by lemma 3.3, $Paths(q) = po(Der(q))$, i.e., π comprises the proof obligations from one of the derivation paths for q. Since no such path is a proof, it contains a node $[f_i \vdash g_j]$ where $f_i \not\vdash g_j$. But then also $\forall s \sqsubseteq_I r_i : s \not\models g_j$ – contradicting (*). $\square$

Example 3.10 *Consider the following (unprovable) sequent $q = \sigma_1 \Vdash \sigma_2$:*

$$a; a \vee b; b \Vdash a \vee b; a; b$$

The $Paths(q)$ are obtained as they were in example 2.7, and are here listed with the formulae f_i, g_j instantiated appropriately:

$[a \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash b],$ *(1)*
$[a \vdash a \vee b] - [a \vee b \vdash a] - [a \vee b \vdash b] - [b \vdash b],$ *(2)*
$[a \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash a] - [b \vdash b],$ *(3)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash b] - [b \vdash b],$ *(4)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vdash b] - [a \vee b \vdash b] - [b \vdash b],$ *(5)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash a] - [b \vdash b],$ *(6)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash a] - [a \vee b \vdash b] - [b \vdash b],$ *(7)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash a] - [b \vdash a] - [b \vdash b],$ *(8)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [b \vdash a] - [b \vdash b],$ *(9)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash b],$ *(10)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [a \vee b \vdash a] - [a \vee b \vdash b] - [b \vdash b],$ *(11)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash a] - [b \vdash b],$ *(12)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [b \vdash a \vee b] - [b \vdash a] - [b \vdash b]$ *(13)*

On every path there exists a node with unprovable obligation, either $[a \vee b \vdash a]$ or $[a \vee b \vdash b]$ (as *well as $[b \vdash a]$ or $[a \vdash b]$). Hence $\sigma_1 \not\Vdash \sigma_2$. The counter-model will be built from three intervals, $r = [0,1)[1,2)[2,\infty)$, where $\forall o \in [0,1) : r(o) = a \wedge \neg b$ (a boolean structure assigning* `true` *to a and* `false` *to b), $\forall o \in [2,\infty) : r(o) = b \wedge \neg a$, while in $[1,2)$ we distribute densly the counter-models for $a \vee b \vdash b$ (namely $a \wedge \neg b$) and for $a \vee b \vdash a$ (namely $\neg a \wedge b$). This will ensure that $[1,2) \models a \vee b$, and so $r \models a; a \vee b; b$, as we have the following situation*

$$r = \begin{array}{c|c|c} r_1 & r_2 & r_3 \\ \hline a \wedge \neg b & a \vee b & \neg a \wedge b \end{array}$$

A cut verifying $a \vee b; a; b$ must first comprise some subinterval verifying $a \vee b$ and then some verifying a. But the latter can occur only within r_1, as $r_3 \models \neg a$, while every subinterval $s \sqsubseteq_I r_2 : s \not\models a$. But then, the rest of r will not satisfy b, since all subintervals $s \sqsubseteq_I r_2 : s \not\models b$. In short, $r \not\models a \vee b; a; b$.

4 Related systems and applications

The presented system, being interval-based, seems to require a comparison with other interval logics. However, since it possesses a number of desirable properties usually missing in such logics, it promises also potential for applications, and we comment briefly such possibilities. (An overview of other interval logics can be found in [9].) Relations to other frameworks – transition systems, 4.1, action-based descriptions, 4.2, and active logic, 4.3 – indicate also possible applications in the areas where these other techniques are applied.

4.1 Transition systems

As mentioned in the introduction, we can define for every sequence formula $\sigma = f_1; f_2; \ldots; f_n$ a transition system (or: rewrite system):

$$\begin{tikzcd} f_1 \arrow[loop above] \arrow[r] & f_2 \arrow[loop above] \arrow[r] & \cdots \arrow[r] & f_n \arrow[loop above] \arrow[r] & final \end{tikzcd}$$

The *meaning* of σ can now be given as the set of all rewrite sequences of this transition system. Here some care has to be taken. First, we only consider rewrite sequences that end in the final state. Second, we consider the states modulo equivalence in the underlying logic. Third, we adopt some notation of formal language theory and denote the rewrite sequences as words $f_1^{p_1} f_2^{p_2} \ldots f_n^{p_n}$ and call them *traces* (note that we denote only the states, and not the transitions). We use Kleene $^+$ to denote one or more iterations. With these points in mind we define the semantics of σ as follows:

$$[\![\sigma]\!] = f_1^+ \ldots f_n^+$$

For example, $[\![a \wedge b; b \wedge a; c]\!] = (a \wedge b)^+(b \wedge a)^+ c^+ = \{(a \wedge b)^p c^q \mid p > 1, q > 0\}$. Equivalence of systems can now be defined as follows.

Definition 4.1 *$\sigma_1 \cong \sigma_2$ if $[\![\sigma_1]\!] \cap [\![\sigma_2]\!]$ is non-empty.*

Indeed, $\cong$ is an equivalence relation: reflexivity and symmetry are trivial, and transitivity follows after a moment's reflection on the regular expressions involved. The following lemma characterizes the semantic entailment in terms of transition systems. ($\sigma(i)$ denotes the i-th "state" of σ.)

Lemma 4.2 *$\sigma_1 \models \sigma_2$ if and only if*

$$\exists\, \sigma_1' \cong \sigma_1,\ \sigma_2' \cong \sigma_2 : |\sigma_1'| = |\sigma_2'| \ \&\ \forall\, i = 1, \ldots, |\sigma_1'| : \sigma_1'(i) \models_{u.l.} \sigma_2'(i)$$

The correspondence between the traces σ_1' and σ_2' reflects the existence of a joint path $\pi \in Paths(\sigma_1 \Vdash \sigma_2)$ from lemma 2.8, which verifies $\sigma_1 \models \sigma_2$.

4.2 Representing actions

Suppose we want to model an action of sending a message m over a secure channel from (agent) A to B, $send(A, m, B)$. We model the environment as another agent E, and security of the channel means that E cannot see what is communicated between A and B. As the underlying logic, we use here some variant of epistemic logic, where $\mathbf{K}_X(y)$ stands for the statement that agent X knows y (has y available). Communication of m from A to B is modeled by a rewrite rule, which defines it in terms of the effects on the consequtive states:

$$send(A, m, B) \;\; \rightsquigarrow \;\; s_1 \; ; \; s_2$$

Now, for instance, a secure communication corresponds to the fact that, starting in a state where A, but not E, knows m, we pass to a state where both A and B, but still not E, know m. This means that as s_1, s_2 we use:

$$s_1 = \mathbf{K}_A(m) \wedge \neg \mathbf{K}_E(m)$$

$$s_2 = \mathbf{K}_A(m) \wedge \neg \mathbf{K}_E(m) \wedge \mathbf{K}_B(m).$$

A reliable communication, i.e., one which is not only secure, but where A can also be sure that B obtains his message, is modeled simply by adding additional conjunct to the resulting state:

$$s_1 = \mathbf{K}_A(m) \wedge \neg \mathbf{K}_E(m)$$

$$s_2 = \mathbf{K}_A(m) \wedge \neg \mathbf{K}_E(m) \wedge \mathbf{K}_B(m) \wedge \mathbf{K}_A(\mathbf{K}_B(m)).$$

An insecure channel makes it possible for E to intercept the message. Analyzing security aspects, one wants to address the worst case scenario, and therefore lets E actually intercept all messages:

$$s_1 = \mathbf{K}_A(m) \wedge \neg \mathbf{K}_E(m)$$

$$s_2 = \mathbf{K}_A(m) \wedge \mathbf{K}_E(m) \wedge \mathbf{K}_B(m).$$

Once a series of actions is rewritten as a series of their effects, σ_1, we can apply our logic for deriving its consequences, by asking for σ_2 such that $\sigma_1 \Vdash \sigma_2$. The point of these examples is to illustrate the flexibility of the proposed setting for handling a virtually unlimited variety of possible action types. This flexibility is achieved by *not* axiomatizing any actions, but merely by representing actions in terms of their effects on the states.

Remark 4.3 *Notation $\mathbf{K}_A(m)$ suggests that we can use some variant of modal epistemic logic, i.e.,* S4 *or* S5*. This is correct, but we should take some precautions in formulating the proof obligations (side-conditions in our reasoning system). For instance, we might want to prove that, given that $a \to b$, an agent A knowing first a, will eventually (be able to) know b, i.e., $\top; \mathbf{K}_A(b)$. The last step in a derivation of such a description would amount to the following:*

$$\frac{}{(a \to b) \wedge \mathbf{K}_A(a) \Vdash \mathbf{K}_A(b)} \; [(a \to b) \wedge \mathbf{K}_A(a) \vdash \mathbf{K}_A(b)] \tag{4.1}$$

In modal logic, strong completeness requires that the premisses are included under the $\Box$-modality (here $\mathbf{K}$), and the premise $a \to b$ does not satisfy this condition. The general statement of this fact is as follows (e.g., exercise 1.5.3 in [3]):

$$\Gamma \models^g \phi \iff \mathbf{K}^*(\Gamma) \models^l \phi \tag{4.2}$$

where $\models^g$ is the global logical consequence (which we are using), while $\models^l$ is local logical consequence, for which we have strongly complete reasoning systems.[4] *In the logics where $\mathbf{K}(\phi) \leftrightarrow \mathbf{K}(\mathbf{K}(\phi))$, we can simplify the right hand side of (4.2) to $\Gamma, \mathbf{K}(\Gamma) \models^l \phi$, or even drop Γ when axiom T is present. Consequently, when using modal epistemic logic, like* S4 *or* S5*, we would have to utilize strongly complete versions of the reasoning system, based on (4.2). For proving the side-condition of (4.1), we would obtain a proof ending with the following transition:*

[4] Statement (4.2) can be relativised to arbitrary classes of frames which are closed under 'reachable subframes', i.e., classes M where, when $(W, R) \in \mathsf{M}$ and (W_w, R_w) is a subframe of (W, R) obtained by taking $W_w = \{w' \mid R^*(w, w')\}$ for some fixed $w \in W$ and restricting R to this subset, then also $(W_w, R_w) \in \mathsf{M}$. This closure property is enjoyed by the typical modal logics like S4 or S5.

$$\frac{\begin{array}{c}\vdots\\ \hline \mathbf{K}_A(a \to b) \wedge \mathbf{K}_A(a) \vdash \mathbf{K}_A(b)\end{array}}{(a \to b) \wedge \mathbf{K}_A(a) \vdash \mathbf{K}_A(b)}$$

4.3 Active logic

Active logic has been developed to reason about the changing states of beliefs, [6, 2]. Although our framework does not aim at equal completeness of modelling, several aspects of active logic fall naturally into it (e.g., allowing for nonmonotonicity, temporally evolving beliefs). Other aspects addressed by active logic have been in our case factored out and delegated to the choice of the underlying logic. Thus, for instance, to model bounded agents, we only have to choose an appropriate logic for such agents; to treat contradictory beliefs, some form of paraconsistent logic could be chosen. We illustrate some of these aspects below.

Active logic operates with the explicit notion of (discrete) time points, and application of rules involves always increase of time. (This simple idea is present in the predecessor of active logic, step logic [4, 5, 7], and in its decidable fragment, [1].) For instance:

$$\frac{\mathsf{n} : f \ , \ f \to g}{\mathsf{n}+1 : g}$$
– if, at step n, f and $f \to g$ are known/available
– then, at step $\mathsf{n+1}$, g can become known/available

Semantical differences notwithstanding, the following rule, admissible in our system, can be used to model exactly this aspect of stepwise reasoning:

$$(step) \ \frac{\sigma \Vdash *f\star}{\sigma \Vdash *f; g\star} \, [f \vdash g]$$

Thus, for instance, the sequence starting with (i) the knowledge that birds fly, $\phi = \forall x(B(x) \to F(x))$, and, after some time, (ii) learning that t is a bird, after which (iii) ϕ, namely, the fact that birds can fly is forgotten, is expressed with the (appropriate fragment of) first-order logic as the underlying logic, as given on the left of $\Vdash$. From this we obtain the proof that the fact $F(t)$ was (possibly) known at some point:

$$\phi; \phi \wedge B(t); B(t) \ \ \Vdash \ \ \phi; \phi \wedge B(t); \phi \wedge B(t) \wedge F(t); B(t) \ \ \Vdash \ \ \top; F(t); \top$$

The modelling is more abstract than in active logic where one simply counts the applications of rules. This is possible here as well, but we can also allow much coarser granularity admitting transitions to arbitrary consequences. (Of course, counting of steps happens in our case only at the meta-level. Inclusion of the time-step, step-sequence, etc. into the agent language, so central in step and active logics, would require appropriate choice of the underlying logic.)

4.4 Bounded agents

Boundedness of agents, which underlies the stepwise model of reasoning, can be captured in our setting simply by using appropriate (semantics of the) underlying logic. For instance, one might apply the logic of finite agents from [10], with the collection of formulae as its syntactic models of the subsequent states. With this logic one can express that an agent knows a, b, $\mathbf{K}_A(a, b)$, as well as that he knows *at most* b, c, d, written $\overline{\mathbf{K}}_A(b, c, d)$. Then we can prove that an agent who first (i) knows a, then (ii) knows a, b, c and then (iii) at most b, c, d, eventually does not know a:

$$\mathbf{K}_A(a); \mathbf{K}_A(a, b, c); \overline{\mathbf{K}}_A(b, c, d) \Vdash \top; \neg\mathbf{K}_A(a)$$

More intricate and practical examples of applications of the logic presented here, and its combination with the logic for bounded agents, can be found in [14], or obtained by contacting one of the authors.

More generally, one can model bounded agents by distinguishing the explicit (limited) and the implicit (potentially unlimited) knowledge, cf. [13, 8]. Here, we would interpret the left hand side of $\Vdash$ as the description of the actual sequence of states of *explicit* knowledge of an agent (each with a finite model). The derivable right hand sides of $\Vdash$ represent then possible *implicit* consequences of such transitions. For instance, using the (step) rule, we can unfold possible consequences of the following initial state:

$$\mathbf{K}_A(a \vee b, \neg a) \ \ \Vdash \ \ \mathbf{K}_A(a \vee b, \neg a); \mathbf{K}_A(a \vee b, a \vee b \vee c, \neg a); \mathbf{K}_A(a \vee b, a \vee b \vee c, \neg a, b)$$

The meaning is that the states on the right of $\Vdash$ are reachable from the (initial) state on the left by finitely many deduction steps. With such an interpretation, as the underlying logic one might even choose any variant of modal epistemic logic, and yet avoid the omniscience problems inherent to these logics. This particular application of our logic is, in fact, the approach taken in the logic of algorithmic knowledge, [12].

5 Conclusions and Future Work

We have presented a temporal logic with a single binary modality, roughly corresponding to **U**ntil, and interpreted it over dense linear time. It is parameterized by an *arbitrary* underlying logic for description of the states. Our logic inherits metaproperties of the underlying logic: its is strongly complete/sound/decidable whenever the underlying logic is. We have also suggested the possible way of handling actions in our framework, simply by defining them through their effects on the states of the system. What is missing at the present stage, are detailed examples. The parametric character of our logic offers possibility of applying it to a wide variety of contexts, but the details and usefulness of such applications remain to be investigated.

Also, we would like to adjust the logic to a wider variety of orderings. A simple restriction of the (L) rule yields a sound system for *all* total orderings. Although we expect it to be complete, the proof of the fact is still missing. For discrete orderings, the relation $\models$ is decidable, but the problem of constructing a natural reasoning system remains open. This problem turns out to be surprisingly difficult and is currently under investigation.

References

[1] Natasha Alechina, Brian Logan, and Mark Whitsey. A complete and decidable logic for resource-bounded agents. In *3-rd Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*. ACM Press, 2004.

[2] Michael L. Anderson, Walid Gomaa, John Grant, and Don Perlis. On the reasoning of real-world agents: toward a semantics of active logic. In *7-th Annual Symposium on the Logical Formalization of Commonsense Reasoning*, Corfu, Greece, 2005.

[3] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[4] Jennifer Drapkin and Don Perlis. A preliminary excursion into step-logics. In *SIGART International Symposium on Methodologies for Intelligent Systems*, 1986.

[5] Jennifer Elgot-Drapkin. *Step-logic: reasoning situated in time*. PhD thesis, University of Maryland, 1988.

[6] Jennifer Elgot-Drapkin, Sarit Kraus, Michael Miller, Madhura Nirkhe, and Don Perlis. Active logics: A unified formal approach to episodic reasoning. Technical Report CS-TR-3680 and UMIACS-TR-99-65, University of Maryland, 1999.

[7] Jennifer Elgot-Drapkin, Michael Miller, and Don Perlis. Memory, reason and time: the step-logic approach. In R. Cummins and J Pollock, editors, *Philosophy and AI: Essays at the Interface*. MIT Press, Cambridge, Mass., 1991.

[8] Ronald Fagin and Joseph Y. Halpern. Belief, awareness, and limited reasoning. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 480–490, 1985.

[9] Valentin Goranko, Angelo Montanari, and Guido Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1-2):9–54, 2004.

[10] Thomas Ågotnes. *A Logic of Finite Syntactic Epistemic States*. PhD thesis, Department of Informatics, University of Bergen, 2004.

[11] Joseph Y. Halperm and Yoav Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.

[12] Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Algorithmic knowledge. In Ronald Fagin, editor, *5-th Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'94)*. Morgan Kaufman, 1994.

[13] Hector J. Levesque. A logic of implicit and explicit belief. In *National Conference on Artificial Intelligence (AAAI-84)*, pages 198–202, 1984.

[14] Wojciech Szajnkenig. *Sequence Logic*. PhD thesis, Department of Informatics, University of Bergen, [forthcoming].