# Singular and Plural Nondeterministic Parameters

Michal Walicki *)    Sigurd Meldal

**Abstract:** The article defines algebraic semantics of singular (call-time-choice) and plural (run-time-choice) nondeterministic parameter passing and presents a specification language in which operations with both kinds of parameters can be defined. Sound and complete calculi for both semantics are introduced. We study the relations between the two semantics and point out that axioms for operations with plural arguments may be considered as axiom schemata for operations with singular arguments.

**Keywords:** algebraic specification, nondeterminism, multialgebra, reasoning.

**AMS classifications:** 68Q65, 68Q60, 68Q10, 68Q55, 68T15.

## 1. Introduction

The notion of *nondeterminism* arises naturally in describing concurrent systems. Various approaches to the theory and specification of such systems, for instance, CCS [16], CSP [9], process algebras [1], event structures [26], include the phenomenon of nondeterminism. But nondeterminism is also a natural concept in describing sequential programs, either as a means of indicating a "don't care" attitude as to which among a number of computational paths will actually be utilized in a particular computation (e.g., [3]) or as a means of increasing the level of abstraction [14,25]. The present work proceeds from the theory of *algebraic specifications* [4, 27] and generalizes it so that it can be applied to describing nondeterministic operations.

In deterministic programming the distinction between call-by-value and call-by-name semantics of parameter passing is well known. The former corresponds to the situation where the actual parameters to function calls are evaluated and passed as values. The latter allows parameters which are function expressions, passed by a kind of Algol copy rule [21], and which are evaluated whenever a need for their value arises. Thus call-by-name will terminate in many cases when the value of a function may be determined without looking at (some of) the actual parameters, i.e., even if these parameters are undefined. Call-by-value will, in such cases, always lead to undefined result of the call. Nevertheless, the call-by-value semantics is usually preferred in the actual programming languages since it leads to clearer and more tractable programs.

*) We gratefully acknowledge the financial support received for this work from the Norwegian Council for Scientific and Industrial Research.

Following [20], we call the nondeterministic counterparts of these two notions *singular* (call-by-value) and *plural* (call-by-name) parameter passing. Other names applied to this, or closely related distinction, are *call-time-choice* vs. *run-time-choice* [2, 8], or *inside-out* (IO) vs. *outside-in* (OI) which reflect the substitution order corresponding to the respective semantics [5, 6]. In the context where one allows nondeterministic parameters the difference between the two semantics becomes quite apparent even without looking at their termination properties. Let us suppose that we have defined operation $g(x)$ as "**if** $x$=0 **then** $a$ **else** (**if** $x$=0 **then** $b$ **else** $c$)", and that we have a nondeterministic choice operation $\sqcup$. returning an arbitrary element from the argument set. The singular interpretation will satisfy the formula $\phi$: $g(x)$ = (**if** $x$=0 **then** $a$ **else** $c$), while the plural interpretation need not satisfy this formula. For instance, under the singular interpretation $g(\sqcup.\{0,1\})$ will yield either $a$ or $c$, while the set of possible results of $g(\sqcup.\{0,1\})$ under the plural interpretation will be $\{a,b,c\}$. (Notice that in a deterministic environment both semantics would yield the same results.) The fact that the difference between the two semantics occurs already in very trivial examples of terminating nondeterministic operations motivates our investigation.

We discuss the distinction between the singular and plural passing of nondeterministic parameters in the context of algebraic semantics focusing on the associated reasoning systems. The singular semantics is given by *multialgebras*, that is, algebras where functions are set-valued and where these values correspond to the sets of possibile results returned by nondeterministic operations. Thus, if $f$ is a nondeterministic operation, $f(t)$ will denote the set of possible results returned by $f$ when applied to $t$. We introduce the calculus NEQ which is sound and complete with respect to this semantics.

Although terms may denote sets the variables in the language range only over individuals. This is motivated by the interest in describing unique results returned by each particular application of an operation (execution of the program). It gives us the possibility of writing, instead of a formula $\Phi(f(t))$ which expresses something about the whole set of possible results of $f(t)$, the formula corresponding to $x \in f(t) \Rightarrow \Phi(x)$ which express something about each particular result $x$ returned by $f(t)$. Unfortunately, this poses the main problem of reasoning in the context of nondeterminism – the lack of general substitutivity. From the fact that $h(x)$ is deterministic (for each $x$, has a unique value) we cannot conclude that so is $h(t)$ for an arbitrary term $t$. If $t$ is nondeteministic, $h(t)$ may have several possible results. The calculus NEQ is designed so that it appropriately restricts the substitution of terms for singular variables.

Although operations in multialgebras are set-valued their carriers are usual sets. Thus operations map individuals to sets. This is not sufficient to model plural arguments. Such arguments can be understood as sets being passed to the operation. The fact that, under plural interpretation, $g(x)$ as defined above need not satisfy $\phi$ results from the two occurrences of $x$ in the body of $g$. Each of these occurrences corresponds to a repeated application of choice from the argument set $x$, that is, potentially, to a different value. In order to model such operations we take as the carrier of the algebra a (subset of the) power set – operations map sets to sets. In this way we obtain *power algebra* semantics. The extension of the semantics is reflected at the syntactic level by introduction of *plural variables* ranging over sets rather than over individuals. The sound and complete extension of NEQ is obtained by adding one new rule which allows for usual substitution of arbitrary terms for plural variables.

The structure of the paper is as follows. In sections 2-3 we introduce the language for specifying nondeterministic operations and explain the intuition behind its main features. In section 4 we define *multialgebraic* semantics for singular specifications and introduce a sound and complete calculus for such specifications. In section 5 the semantics is generalized to *power algebras* capable of modeling plural parameters, and the sound and complete extension of the calculus is obtained by introducing one additional rule. A comparison of both semantics in section 6 is guided by the similarity of the respective calculi. We identify the subclasses of multimodels and power models which may serve as equivalent semantics of one specification. We also highlight the increased complexity of the power algebra semantics reflecting the problems with intuitive understanding of plural arguments.

Proofs of the theorems are merely indicated in this presentation. It reports some of the results from [24] where the full proofs and other details can be found.

## 2. The specification language

A specification is a pair $(\Sigma,\Pi)$ where the signature $\Sigma$ is a pair $(\mathbf{S},\mathbf{F})$ of sorts $\mathbf{S}$ and operation symbols $\mathbf{F}$ (with argument and result sorts in $\mathbf{S}$). The set of terms over a signature $\Sigma$ and variable set $X$ is denoted by $W_{\Sigma,X}$. We always assume that, for every sort $S$, the set of ground words of sort $S$, $S^{W_\Sigma}$, is not empty. [1]

$\Pi$ is a set of sequents of atomic formulae written as $a_1,...,a_n \mapsto e_1,...,e_m$. The left hand side (LHS) of $\mapsto$ is called the *antecedent* and the right hand side (RHS) the *consequent*, and both are to be understood as sets of atomic formulae (i.e., the ordering and multiplicity of the atomic formulae do not matter). In general, we allow either antecedent or consequent to be empty, though $\emptyset$ is usually dropped in the notation. A sequent with exactly one formula in the consequent ($m$=1) is called a *Horn formula*, and a Horn formula with empty antecedent ($n$=0) is a *simple formula* (or a *simple sequent*).

[1] This restriction is motivated by the fact (pointed out in [7]) that admitting empty carriers requires additional mechanisms (explicit quantification) in order to obtain sound logic. We conjecture that similar solution can be applied in our case.

All variables occurring in a sequent are implicitly universally quantified over the whole sequent. A sequent is satisfied if, for every assignment to the variables, one of the antecedents is false or one of the consequents is true (it is valid iff the formula $a_1 \wedge \ldots \wedge a_n \Rightarrow e_1 \vee \ldots \vee e_m$ is valid).

For any term (formula, set of formulae) $\xi$, $\mathcal{V}[\xi]$ will denote the set of variables in $\xi$. If the variable set is not mentioned explicitly, we may also write $x \in \mathcal{V}$ to indicate that $x$ is a variable.

An atomic formula in the consequent is either an *equation,* $t=s$, or an *inclusion,* $t \prec s$, of terms $t, s \in W_{\Sigma,X}$. An atomic formula in the antecedent, written $t \frown s$, will be interpreted as non-empty intersection of the (result) sets corresponding to $t$ and $s$. For a given specification SP=$(\Sigma,\Pi)$, $\mathcal{L}$(SP) will denote the above language over the signature $\Sigma$.

The above conventions will be used throughout the paper. The distinction between the singular and plural parameters (introduced in the section 5) will be reflected in the notation by the superscript $^*$ : a plural variable will be denoted by $x^*$, the set of plural variables in a term $t$ by $\mathcal{V}^*[t]$, a specification with plural arguments SP$^*$, the corresponding extension of the language $\mathcal{L}$ by $\mathcal{L}^*$ etc.

## 3. A note on the intuitive interpretation

Multialgebraic semantics [10, 13] interprets specifications in some form of power structures where the (nondeterministic) operations correspond to *set-valued functions*. This means that a (ground) term is interpreted as a set of *possibilities* – it denotes the set of *possible* results of the corresponding operation. We, on the other hand, want our formulae to express *necessary* facts, i.e., facts which have to hold in *every evaluation* of a program (specification). This is achieved by interpreting terms as *applications* of the respective operations. Every two syntactic occurrences of a term $t$ will refer to *possibly distinct* applications of $t$. For nondeterministic terms this means that they may denote two distinct values.

Typically, equality is interpreted in a multialgebra as *set equality* [13, 23, 12]. For instance, the formula $\mapsto t=s$ means that *the sets corresponding to all possible results* of the operations $t$ and $s$ are equal. This gives a model which is mathematically plausible, but which does not correspond to our operational intuition. The (set) equality $\mapsto t=s$ does not guarantee that the result returned by some particular application of $t$ will actually be equal to the result returned by an application of $s$. It merely tells us that *in principle* (in all possible executions) any result produced by $t$ can also be produced by $s$ and vice versa.

Equality in our view should be a *necessary* equality which must hold in every evaluation of a program (specification). *It does not correspond to set equality, but to identity of 1-element sets*. Thus the simple formula $\mapsto t=s$ will hold in a multistructure M iff both $t$ and $s$ are interpreted in M as one and the same set which, in addition, *has only one element*. Equality is then a *partial equivalence relation* and terms $t$ for which $\mapsto t{=}t$ holds are exactly the deterministic terms, denoted by $D_{SP,X}$. This last equality indicates that arbitrary two applications of $t$ have to return the same result.

If it is possible to produce a computation where $t$ and $s$ return different results – and this is possible when they are nondeterministic – then the terms are not equal but, at best, *equivalent*. They are equivalent if they are capable of returning the same results, i.e., if they are interpreted as the same set. This may be expressed using the inclusion relation: $s \prec t$ holds iff the set of possible results of $s$ is included in the set of possible results of $t$, and $s \succ\!\prec t$ if each is included in the other.

Having introduced inclusion one might expect that a nondeterministic operation can be specified by a series of inclusions – each defining one of its possible results. However, such a specification gives only a "lower bound" on the admitted nondeterminism. Consider the following example:

*Example 3.1*

**S**: { Nat },

**F**: 0: $\to$ Nat (zero)

$s$_: Nat $\to$ Nat (successor)

_⊔_: Nat×Nat $\to$ Nat (binary nondeterministic choice)

**Π**: 1. $\mapsto 0{=}0$

2. $\mapsto s(x){=}s(x)$

3. $1 \frown 0 \mapsto$ (As usual, we abbreviate $s^n(0)$ as $n$.)

4. $\mapsto 0 \prec 0 \sqcup 1$ $\quad \mapsto 1 \prec 0 \sqcup 1$

□

The first two axioms make zero and successor deterministic. A limited form of negation is present in $\mathcal{L}$ in the form of sequents with empty consequent. Axiom 3. makes 0 distinct from 1. Axioms 4. make then ⊔ a nondeterministic choice with 0 and 1 among its possible results. This, however, ensures only that in every model both 0 and 1 can be returned by $0 \sqcup 1$. In most models all other kinds of elements may be among its possible results as well, since no extension of the result set of $0 \sqcup 1$ will violate the inclusions of 4. If we are satisfied with this degree of precision, we may stop here and use only Horn formula. All the results in the rest of the paper apply to this special case. But to specify an "upper bound" of nondeterministic operations we need *disjunction* – the multiple formulae in the consequents. Now, if we write the axiom:

5. $\mapsto 0 \sqcup 1{=}0,\ 0 \sqcup 1{=}1$

the two occurrences of $0 \sqcup 1$ refer to two arbitrary applications and, consequently, we obtain that either any application of $0 \sqcup 1$ equals 0 or else it equals 1, i.e., that ⊔ is not really nondeterministic, but merely underspecified. Since axioms 4. require that both 0 and 1 be among the results of $t$, the addition of 5. will actually make the specification inconsistent.

What we are trying to say with the disjunction of 5. is that *every application* of $0 \sqcup 1$ returns either 0 or 1, i.e., we need a means of identifying two occurrences of a nondeterministic term as referring to one and the same application. This can be done by *binding* both occurrences to a variable.The appropriate axiom will be:

5′. $x \frown 0 \sqcup 1 \mapsto x=0, x=1$

The axiom says: whenever $0 \sqcup 1$ returns $x$, then $x$ equals 0 or $x$ equals 1. Notice that such an interpretation presupposes that the variable $x$ refers to a *unique*, *individual* value. Thus bindings have the intended function only if they involve *singular* variables. (Plural variables, on the other hand, will refer to sets and not individuals, and so the axiom

5′′. $x^* \frown 0 \sqcup 1 \mapsto x^*=0, x^*=1$

would have a completely different meaning.) The singular semantics is the most common in the literature on algebraic semantics of nondeterministic specification languages [2, 8, 11], in spite of the fact that it prohibits unrestricted substitution of terms for variables. Any substitution must now be guarded by the check that the substituted term yields a unique value, i.e., is deterministic. We return to this point in the subsection on reasoning where we introduce a calculus which does not allow one, for instance, to conclude $0 \sqcup 1 = 0 \sqcup 1 \mapsto 0 \sqcup 1 = 0, 0 \sqcup 1 = 1$ from the axiom 5′ (though it could be obtained from 5′′).

## 4. The singular case: semantics and calculus

This section defines the multialgebraic semantics of specifications with singular arguments and introduces a sound and complete calculus.

### 4.1. Multistructures and multimodels.

**Definition 4.2 (Multistructures)**. Let $\Sigma$ be a signature. M is a $\Sigma$-*multistructure* if

(1) its carrier $|M|$ is an $\mathbf{S}$-sorted set and

(2) for every $f$: $S_1 \times \ldots \times S_n \rightarrow S$ in $\mathbf{F}$, there is a corresponding fuction $f^M{:}S_1{}^M \times \ldots \times S_n{}^M \rightarrow \mathcal{P}^+(S^M)$.

A function $\Phi$: $A \rightarrow B$ (i.e., a family of functions $\Phi_S$: $S^A \rightarrow S^B$, for every $S \in \mathbf{S}$) is a *multihomomorphism* from a $\Sigma$-multistructure A to B if

(H1) for each constant symbol $c \in \mathbf{F}$, $\Phi(c^A) \subseteq c^B$ and

(H2) for every $f$: $S_1 \times \ldots \times S_n \rightarrow S$ in $\mathbf{F}$ and $\underline{a}_1 \ldots \underline{a}_n \in S_1{}^A \times \ldots \times S_n{}^A$ :
$\Phi(f^A(\underline{a}_1 \ldots \underline{a}_n)) \subseteq f^B(\Phi(\underline{a}_i) \ldots \Phi(\underline{a}_n))$

If all inclusions in H1 and H2 are (set) equalities the homomorphism is *tight*, otherwise it is strictly loose (or just *loose*).

□

$\mathcal{P}^+(S)$ denotes the set of non-empty subsets of the set $S$. Operations applied to sets refer to their unique pointwise extensions. Notice that for a constant $c: \to S$, 2. indicates that $c^M$ can be a *set* of several elements of sort $S$.

Since multihomomorphisms are defined on individuals and not sets they preserve singletons and are $\subseteq$-monotonic. We denote the class of $\Sigma$-multistructures by MStr($\Sigma$). It has the distinguished word structure $MW_\Sigma$ defined in the obvious way, where each ground term is interpreted as a singleton set. We will treat such singleton sets as terms rather than 1-element sets (i.e., we do not take special pains to distinguish $MW_\Sigma$ and $W_\Sigma$). $MW_\Sigma$ is not an initial $\Sigma$-structure since it is deterministic and there can exist several homomorphisms from it to a given multistructure. We do not focus on the aspect of initiality and merely register the useful fact from [11]:

**Lemma 4.3**. M is a $\Sigma$-multistructure iff, for every set of variables $X$ and assignment $\beta$: $X \to |M|$, there exists a unique function $\beta[\_]$: $W_{\Sigma,X} \to \mathcal{P}^+(|M|)$ such that:

1. $\beta[x] = \{\beta(x)\}$
2. $\beta[c] = c^M$
3. $\beta[f(t_i)] = \bigcup\{f^M(y_i) \mid y_i \in \beta[t_i]\}$

□

In particular, for $X=\emptyset$, there is a unique interpretation function (not a multihomomorphism) $\mathcal{I}$: $W_\Sigma \to \mathcal{P}^+(|M|)$ satisfying the last two points of this definition.

As a consequence of the definition of multistructures, all operations are $\subseteq$-monotonic, i.e., $\beta[s] \subseteq \beta[t] \Rightarrow \beta[f(s)] \subseteq \beta[f(t)]$. Notice also that assignment in the lemma (and in general, whenever it is an assignment of elements from a multistructure) means assignment of individuals, not sets.

Next we define the class of *multimodels* of a specification.

**Definition 4.4 (Satisfiability)**. A $\Sigma$-multistructure M *satisfies* an $\mathcal{L}(\Sigma)$ sequent $\pi$

$$t_i \frown s_i \mapsto p_j = r_j,\ m_k \prec n_k,$$

written M $\models \pi$, iff for every $\beta$: X$\to$M we have

$$\bigwedge_i \beta[t_i] \cap \beta[s_i] \neq \emptyset \Rightarrow \bigvee_j \beta[p_j] \equiv \beta[r_j] \vee \bigvee_k \beta[m_k] \subseteq \beta[n_k]$$

where A$\equiv$B iff A and B are the same 1-element set.

An SP-*multimodel* is a $\Sigma$-multistructure which satisfies all the axioms of SP. We denote the class of multimodels of SP by MMod(SP).

□

The reason for using nonempty intersection (and not set equality) as the interpretation of $\frown$ in the antecedents is the same as using “elementwise” equality $\equiv$ in the consequents. Since we avoid set equality in the positive sense (in the consequents), the most natural negative form seems to be the one we have chosen. For deterministic terms this is the same as equality, i.e., deterministic antecedents correspond exactly to the usual (deterministic) conditions. For nondeterministic terms this reflects our interest in binding such terms: the sequent "$...s\frown t...\mapsto...$" is equivalent to "$...x\frown s,\ x\frown t...\mapsto...$". A binding "$...x\frown t...\mapsto...$" is also equivalent to the more familiar "$...x\in t...\mapsto...$", so the notation $s\frown t$ may be read as an abbreviation for the more elaborate formula with two $\in$ and a new variable $x$ not occurring in the rest of the sequent.

For a justification of this, as well as other choices we have made here, the reader is referred to [24].

### 4.2. The calculus for singular semantics

In [24] we have introduced the calculus NEQ which is sound and complete with respect to the class MMod(SP). Its rules are:

R1: $\mapsto x{=}x \qquad\qquad x\in\mathcal{V}$

R2: $$\frac{\Gamma^x_{t_2} \mapsto \Delta^x_{t_2} \quad ; \quad \Gamma' \mapsto t_1{=}t_2, \Delta'}{\Gamma^x_{t_1}, \Gamma' \mapsto \Delta^x_{t_1}, \Delta'}$$

R3: $$\frac{\Gamma^x_{t_2} \mapsto \Delta^x_{t_2} \quad ; \quad \Gamma' \mapsto t_1{\prec}t_2, \Delta'}{\Gamma^x_{t_1}, \Gamma' \mapsto \Delta^x_{t_1}, \Delta'} \qquad x \text{ not in a RHS of } \prec$$

R4: a) $x\frown y \mapsto x{=}y$ b) $x\frown t \mapsto x{\prec}t \qquad x,y\in\mathcal{V}$

R5: $$\frac{\Gamma \mapsto \Delta, s{\preceq}t \quad ; \quad \Gamma', s\frown t \mapsto \Delta'}{\Gamma,\Gamma' \mapsto \Delta,\Delta'} \qquad \text{(CUT)} \qquad (\preceq \text{ stands for either } = \text{ or } \prec)$$

R6: a) $\dfrac{\Gamma \mapsto \Delta}{\Gamma \mapsto \Delta, e}$ b) $\dfrac{\Gamma \mapsto \Delta}{\Gamma, e \mapsto \Delta}$ (WEAK)

R7: $$\frac{\Gamma, x\frown t \mapsto \Delta}{\Gamma^x_t \mapsto \Delta^x_t} \qquad x\in\mathcal{V}-\mathcal{V}[t], \text{ at most one } x \text{ in } \Gamma \mapsto \Delta \qquad \text{(ELIM)}$$

$\Gamma^a_b$ denotes $\Gamma$ with $b$ substituted for $a$. Short comments on each of the rules may be in order.

The fact that '=' is a partial equivalence relation is expressed in R1. It applies only to variables and is sound because all assignments assign individual values to the (singular) variables.

R2 is a paramodulation rule allowing replacement of terms which may be deterministic (in the case when $t_1{=}t_2$ holds in the second assumption). In particular, it allows derivation of the standard substitution rule when the substituted terms are deterministic, and prevents substitution of nondeterministic terms for variables.

R3 allows "specialization" of a sequent by substituting for a term $t_2$ another term $t_1$ which is included in $t_2$. The restriction that the occurrences of $t_2$ which are substituted for don't occur in the RHS of $\prec$ is needed to prevent, for instance, the unsound conclusion $\mapsto t_3 \prec t_1$ from the premises $\mapsto t_3 \prec t_2$ and $\mapsto t_1 \prec t_2$.

R4 and R5 express the relation between $\frown$ in the antecedent and the equality and inclusion in the consequent. The axiom of standard sequent calculus, $e \mapsto e$, (i.e., $s \frown t \mapsto s \preceq t$) does not hold in general here because the antecedent corresponds to non-empty intersection of the result sets while the consequent to the inclusion ($\prec$) or identity of 1-element (=) result sets. Only for deterministic terms $s$, $t$, do we have that $s \frown t \mapsto s = t$ holds.

R5 allows us to cut both $\mapsto s = t$ and $\mapsto s \prec t$ with $s \frown t \mapsto \Delta$.

R7 eliminates redundant bindings, namely those that bind an application of a term occurring at most once in the rest of the sequent.

We will write $\Pi \vdash_{\mathrm{CAL}} \boldsymbol{\pi}$ to indicate that $\boldsymbol{\pi}$ is provable from $\Pi$ with the calculus CAL. When we need to write the sequent $\boldsymbol{\pi}$ explicitly this notation becomes sometimes awkward, and so we will optionally write it as $\frac{\Pi}{\boldsymbol{\pi}}\,{}_{\mathrm{CAL}}$.

The counterpart of soundness/completeness of the equational calculus is [24]:

**Theorem 4.5.** NEQ is sound and complete wrt. MMod(SP):

$$\mathrm{MMod(SP)} \models \boldsymbol{\pi} \ \text{ iff } \ \Pi \vdash_{\mathrm{NEQ}} \boldsymbol{\pi}.$$

*Proof idea:*

Soundness is proved by induction on the length of the proof $\Pi \vdash_{\mathrm{NEQ}} \boldsymbol{\pi}$. The proof of the completeness part is a standard, albeit rather involved, Henkin-style argument. The axiom set $\Pi$ of SP is extended by adding all $\mathcal{L}$(SP) formulae $\boldsymbol{\pi}$ which are consistent with $\Pi$ (and the previously added formulae). If the addition of $\boldsymbol{\pi}$ leads to inconsistency, one adds the negation of $\boldsymbol{\pi}$. Since empty consequents provide only a restricted form of negation, the general negation operation is defined as a set of formulae over the original signature extended with new constants. One shows then that the construction yields a consistent specification with a deterministic basis from which a model can be constructed.

□

We also register an easy lemma that the set-equivalent terms, $t \succ\!\prec s$, satisfy the same formulae:

**Lemma 4.6**. $t \succ\!\prec s$ iff, for any sequent $\boldsymbol{\pi}$, $\Pi \vdash_{\mathrm{NEQ}} \boldsymbol{\pi}^{z}_{t}$ iff $\Pi \vdash_{\mathrm{NEQ}} \boldsymbol{\pi}^{z}_{s}$.

□

## 5. The plural case: semantics and calculus

The singular semantics for passing nondeterminate arguments is the most common notion to be found in the literature. Nevertheless, the plural semantics has also received some attention. In the denotational tradition most approaches considered both possibilities [18, 19, 20, 22]. Engelfriet and Schmidt gave a detailed study of both – in their language, IO and OI – semantics based on tree languages [5], and continuous algebras of relations and power sets [6]. The unified algebras of Mosses and the rewriting logic of Meseguer [15] represent other algebraic approaches distinguishing these aspects.

We will define the semantics for specifications where operations may have both singular and plural arguments. The next subsection gives the necessary extension of the calculus NEQ to handle this generalized situation.

### 5.1. Power structures and power models

Singular arguments (such as the variables in $\mathcal{L}$) have the usual algebraic property that they refer to a unique value. This reflects the fact that they are evaluated at the moment of substitution and the result is passed to the following computation. Plural arguments, on the other hand, are best understood as textual parameters. They are not passed as a single value, but every occurrence of the formal parameter denotes a distinct application of the operation.

We will allow both singular and plural parameter passing in one specification. The corresponding semantic distinction is between power set functions which are merely $\subseteq$-monotonic and those which also are $\cup$-additive.

In the language we merely introduce a notational device for distinguishing the singular and plural arguments. We allow annotating the sorts in the profiles of the operation by a superscript, like $S^*$, to indicate that an argument is plural.

Furthermore, we partition the set of variables into two disjoint subsets of singular, $X$, and plural, $X^*$, variables. $x$ and $x^*$ are to be understood as distinct symbols. We will say that an operation $f$ is *singular in the i-th argument* iff the i-th argument (in its signature) is singular. The specification language extended with such annotations of the signatures will be referred to as $\mathcal{L}^*$.

These are the only extensions of the language we need. We may, optionally, use superscripts $t^*$ at any (sub)term to indicate that it is passed as a plural argument. The outermost applications, e.g. $f$ in $f(\ldots)$, are always to be understood plurally, and no superscripting will be used at such places.

**Definition 5.7.** Let $\Sigma$ be a $\mathcal{L}^*$-signature. $A$ is a *$\Sigma$-power structure*, $A \in PStr(\Sigma)$, iff A is a (deterministic) structure such that:

1. for every sort S, the carrier $S^A$ is a (subset of the) power set $\mathcal{P}^+(S^-)$ of some basis

set $S^{-}$,

2. for every $f$: $S_1 \times \ldots \times S_n \to S$ in $\Sigma$, $f^A$ is a $\subseteq$-monotonic function $S_1{}^A \times \ldots \times S_n{}^A \to S^A$ such that, if the $i$-th argument is $S_i$ (singular) then $f^A$ is singular in the $i$-th argument.

□

The singularity in the $i$-th argument in this definition refers not to the syntactic notion but to its semantic counterpart:

**Definition 5.8.** A function $f^A$: $S_1{}^A \times \ldots \times S_n{}^A \to S^A$ in a power structure $A$ is *singular in the i-th argument* iff if it is $\cup$-additive in the i-th argument, i.e., iff for all $x_i \in S_i{}^A$ and all $.x_k \in S_k{}^A$ (for $k \neq i$), $f^A(.x_1 \ldots x_i \ldots x_n) = \bigcup \{f^A(x_1 \ldots \{x\} \ldots x_n) \mid x \in x_i\}$.

□

Thus, the definition of power structures requires that syntactic singularity be modeled by the semantic one.

Note the unorthodox point in the definition – we do not require the carrier to be the whole power set, but allow it to be a *subset* of some power set. Usually one assumes a primitive nondeterministic operation with the predefined semantics as set union. Then all finite subsets are needed for the interpretation of this primitive operator. Also, the join operation (under the set inclusion as partial order) corresponds exactly to set union only if all sets are present (see example 6.18). None of these assumptions seem necessary. Consequently, we do not assume any predefined (choice) operation but, instead, give the user means of specifying any nondeterministic operation (including choice) directly.

Let $\Sigma$ be a signature, $A$ a $\Sigma$-power structure, $X$ a set of singular and $X^*$ a set of plural variables, and $\beta$ an assignment $X \cup X^* \to |A|$ such that for all $x \in X$ $|\beta(x)| = 1$. (Saying "assignment" we will from now on mean only assignments satisfying this last condition.) Then, every term $t(x,x^*) \in W_{\Sigma,X,X^*}$ has a unique set interpretation $\beta[t(x,x^*)]$ in $A$ defined as $t^A(\beta(x),\beta(x^*))$.

**Definition 5.9 (Satisfiability).** Let $A$ be a $\Sigma$-power structure and $\pi$: $t_i \frown s_i \mapsto p_j = r_j$, $m_k \prec n_k$ be a sequent over $\mathcal{L}^*(\Sigma,X,X^*)$. *A satisfies* $\pi$, $A \models \pi$, iff for every assignment $\beta$: $X \cup X^* \to |A|$, we have that:

$$\bigwedge_i \beta[t_i] \cap \beta[s_i] \neq \emptyset \Rightarrow \bigvee_j \beta[p_j] \equiv \beta[r_j] \vee \bigvee_k \beta[m_k] \subseteq \beta[n_k]$$

A is a *power model* of the specification $SP=(\Sigma,\Pi)$, $A \in PMod(SP)$, iff $A \in PStr(\Sigma)$ and $A$ satisfies all axioms from $\Pi$.

□

Except for the change in the notion of an assignment, this is identical to the definition 4.4, which is the reason for retaining the same notation for the satisfiability relation.

## 5.2. The calculus for plural parameters

The calculus NEQ is extended with one additional rule:

R8: $$\frac{\Gamma \mapsto \Delta}{\Gamma^{x^*}_{t} \mapsto \Delta^{x^*}_{t}}$$

Rules R1-R7 remain unchanged, but now all terms $t_i$ belong to $W_{\Sigma,X,X^*}$. In particular, any $t_i$ may be a plural variable. We let NEQ$^*$ denote the calculus NEQ+R8.

The new rule R8 expresses the semantics of plural variables. It allows us to substitute an arbitrary term $t$ for a plural variable $x^*$. Taking $t$ to be a singular variable $x$, we can thus exchange plural variables in a provable sequent $\boldsymbol{\pi}$ with singular ones. The opposite is, in general, not possible because rule R1 applies only to singular variables. For instance, a plural variable $x^*$ will satisfy $\mapsto x^* \prec x^*$ but this is not sufficient for performing a general substitution for a singular variable. The main result concerning PMod and NEQ$^*$ is:

**Theorem 5.10.** For any $\mathcal{L}^*$-specification SP and $\mathcal{L}^*$(SP) sequent $\boldsymbol{\pi}$:

$$\text{PMod(SP)} \models \boldsymbol{\pi} \text{ iff } \Pi \vdash_{\text{NEQ}^*} \boldsymbol{\pi}$$

*Proof idea:*

The proof is a straightforward extension of the proof of theorem 4.5.

□

# 6. Comparison

Since plural and singular semantics are certainly not one and the same thing, it may seem surprising that essentially the same calculus can be used for reasoning about both. One would perhaps expect that PMod, being a richer class than MMod, will satisfy fewer formulae than the latter, and that some additional restrictions of the calculus would be needed to reflect the increased generality of the model class. In this section we describe precisely the relation between the $\mathcal{L}$ and $\mathcal{L}^*$ specifications (6.1) and emphasize some points of difference (6.2).

## 6.1. The "equivalence" of both semantics

The following example illustrates a strong sense of equivalence of $\mathcal{L}$ and $\mathcal{L}^*$.

*Example 6.11*

Consider the following plural definition:

$\mapsto f(x^*) \prec$ if $x^*=x^*$ then 0 else 1

It is "equivalent" to the collection of definitions

$\mapsto f(t) \prec$ if $t$=$t$ then 0 else 1

for all terms $t$.

□

In the rest of this section we will clarify the meaning of this “equivalence”.

Since the partial order of functions from a set $A$ to the power set of a set $B$ is isomorphic to the partial order of additive (and strict, if we take $\mathcal{P}$ (all subsets) instead of $\mathcal{P}^+$) functions from the power set of $A$ to the power set of $B$, $[A \to \mathcal{P}(B)] \simeq [\mathcal{P}(A) \to_{\cup} \mathcal{P}(B)]$, we may consider every multistructure $A$ to be a power structure $A^*$ by taking $|A^*| = \mathcal{P}^+(A)$ and extending all operations in $A$ pointwise. We then have the obvious

**Lemma 6.12.** Let SP be a singular specification (i.e., all operations are singular in all arguments), let $A \in$MStr(SP), and $\boldsymbol{\pi}$ be a sequent in $\mathcal{L}$(SP). Then $A \models \boldsymbol{\pi}$ iff $A^* \models \boldsymbol{\pi}$, and so $A \in$MMod(SP) iff $A^* \in$PMod(SP).

□

Call an $\mathcal{L}^*$ sequent $\boldsymbol{\pi}$ *p-ground* (for plurally ground) if it does not contain any plural variables.

**Theorem 6.13.** Let SP$^*$=($\Sigma^*$,$\Pi^*$) be an $\mathcal{L}^*$ specification. There exists a (usually infinite) $\mathcal{L}$ specification SP=($\Sigma$,$\Pi$) such that

1. $W_{\Sigma,X} = W_{\Sigma^*,X}$
2. for any p-ground $\boldsymbol{\pi} \in \mathcal{L}^*(\text{SP}^*)$ • PMod(SP$^*$)$\models \boldsymbol{\pi}$ iff MMod(SP)$\models \boldsymbol{\pi}$.

*Proof:*

Let $\Sigma$ be $\Sigma^*$ with all “$^*$” symbols removed. This makes 1. true. Any p-ground $\boldsymbol{\pi}$ as in 2. is then a $\boldsymbol{\pi}$ over the language $\mathcal{L}(\Sigma,X)$.

The axioms $\Pi$ are obtained from $\Pi^*$ as in the example 6.11. For every $\boldsymbol{\pi}^* \in \Pi^*$ with plural variables $x_1{}^*...x_n{}^*$, let $\boldsymbol{\pi} = \{ \boldsymbol{\pi}^*{}^{x_1^* ... x_n^*}_{t_1 ... t_n} \mid t_1...t_n \in W_{\Sigma,X} \}$.

Obviously, for any $\boldsymbol{\pi} \in \mathcal{L}$(SP) if $\Pi \vdash_{\text{NEQ}} \boldsymbol{\pi}$ then $\Pi^* \vdash_{\text{PEQ}} \boldsymbol{\pi}$. If $\Pi^* \vdash_{\text{NEQ}^*} \boldsymbol{\pi}$ then the proof can be simulated in NEQ. Let $\boldsymbol{\pi}'(x^*)$ be the last sequent used in the NEQ$^*$-proof which contains plural variables $x^*$, and the sequent $\boldsymbol{\pi}'$ be the next one obtained by R8. Build the analogous NEQ-proof tree with all plural variables replaced by the terms which occupy their place in $\boldsymbol{\pi}'$. The leaves of this tree will be instances of the $\Pi^*$ axioms with plural variables replaced by the appropriate terms, and all such axioms are in $\Pi$.

Then soundness and completness of NEQ and NEQ$^*$ imply the conclusion of the theorem.

□

We now ask whether, or under which conditions, the classes PMod and MMod are interchangeable as the models of a specification. Let SP$^*$, SP be as in the theorem. The one way transition is trivial. Axioms of SP are p-ground so PMod(SP$^*$) will satisfy all these axioms by the theorem. The subclass $\downarrow$PMod(SP$^*$)$\subseteq$PMod(SP$^*$) where, for every P$\in\downarrow$PMod(SP$^*$), all operations are singular, will yield a subclass of MMod(SP).

For the other direction, we have to observe that the restriction to p-ground sequents in the theorem is crucial because plural variables range over arbitrary – also undenotable – sets. Let MMod$^*$(SP) denote the class of power structures obtained as in lemma 6.12. It is not necessarily the case that MMod$^*$(SP)$\models\mathbf{\Pi}^*$ as the following argument illustrates.

*Example 6.14*

Let $M^*\in$MMod$^*$(SP) have infinite carrier, $\boldsymbol{\pi}^*\in\mathbf{\Pi}^*$ be $t_i\frown s_i \mapsto p_j{=}r_j,\ m_k{\prec}n_k$ with $x^*\in\mathcal{V}[\boldsymbol{\pi}^*]$, and let $\beta$: $X\cup X^* \to |M^*|$ be an assignment such that $\beta(x^*)=\{\underline{m}_1\ldots\underline{m}_l\ldots\}$ is a set which is not denoted by any term in $W_{\Sigma,X}$. Let $\beta_l$ be an assignment equal to $\beta$ except that $\beta_l(x^*)=\{\underline{m}_l\}$, i.e., $\beta=\bigcup_l\beta_l$. Then

$M^* \models \beta[\boldsymbol{\pi}^*]$ iff

$$M^* \models \beta[t_i] \cap \beta[s_i] \neq \emptyset \quad\Rightarrow\quad \beta[p_j]\equiv\beta[r_j] \vee .. \vee \beta[m_k]\subseteq\beta[n_k] \quad \text{iff}$$

(a) $\quad M^* \models \bigcup_l\beta_l[t_i] \cap \bigcup_l\beta_l[s_l] \neq \emptyset \Rightarrow \bigcup_l\beta_l[p_j] \equiv \bigcup_l\beta_l[r_j] \vee .. \vee \bigcup_l\beta_l[m_k] \subseteq \bigcup_l\beta_l[n_k]$

since operations in $M^*$ are defined by pointwise extension. $M^*\in$MMod$^*$(SP) implies that, for all $l$

(b) $\quad M^* \models \beta_l[t_i] \cap \beta_l[s_l] \neq \emptyset \Rightarrow \beta_l[p_j] \equiv \beta_l[r_j] \vee .. \vee \beta_l[m_k] \subseteq \beta_l[n_k]$

But (b) does not necessarily imply (a). In particular, even if for all $l$, all intersections in the antecedent of (b) are empty, those in (a) may be non-empty. So we are not guaranteed that $M^*\in$PMod(SP$^*$).

□

Thus, the intuition that the multimodels are contained in the power models is not quite correct. To ensure that no undenotable sets from $M^*$ can be assigned to the plural variables we redefine the lifting operator $^*$: MMod(SP)$\to$ PMod(SP) from 6.12.

**Definition 6.15.** Given a singular specification SP, and $M\in$MMod(SP), we denote by $\uparrow$M the following power structure

1) $|\uparrow\mathrm{M}| \subseteq \mathcal{P}^+(|\mathrm{M}|)$ is such that

a) for every $\underline{n}\in|M|$: $\{\underline{n}\}\in|\uparrow\mathrm{M}|$,

b) for every $\underline{m}\in|\uparrow\mathrm{M}|$ there exist a $t\in W_{\Sigma,X}$, $\underline{n}\in|\mathrm{M}|$, such that: $t^M(\underline{n}) = \underline{m}$,

2) the operations in $\uparrow$M can be then defined by: $f(\underline{m})^{\ M} = f(t(\underline{n}))^M$.

□

Then, for any assignment $\beta$: $X^*\to|\uparrow\mathrm{M}|$ there exists an assignment $\theta$: $X^*\to W_{\Sigma,X}$ (1b), and an assignment $\alpha$: $X \rightarrow |M|$ (1a) such that $\beta(x^*) = \alpha[\theta(x^*)]$ (2), i.e., such that the following diagram commutes:

$$\begin{tikzcd}
x^* \arrow[r, "\beta"] \arrow[d, "\theta"'] & \upharpoonleft M \\
W_{\Sigma,x} \arrow[r, "\alpha[_]"] & M \arrow[u, "\upharpoonleft"'] \\
x \arrow[u] \arrow[ur, "\alpha"'] &
\end{tikzcd}$$

Since $M \in$MMod(SP) it satisfies all the axioms $\Pi$ obtained from $\Pi^*$ and the commutativity of the diagram gives us the second part of:

**Corollary 6.16.** Let SP$^*$ and SP be as in the theorem 6.13. Then

$$\downharpoonleft \mathrm{PMod}(\mathrm{SP}^*) \models \Pi \quad \text{i.e., } \downharpoonleft \mathrm{PMod}(\mathrm{SP}^*) \subseteq \mathrm{MMod}(\mathrm{SP})$$
$$\upharpoonleft \mathrm{MMod}(\mathrm{SP}) \models \Pi^* \quad \text{i.e., } \upharpoonleft \mathrm{MMod}(\mathrm{SP}) \subseteq \mathrm{PMod}(\mathrm{SP}^*)$$

□

The corollary makes precise the claim that the class of power models of a plural specification SP$^*$ may be seen as a class of multimodels of some singular specification SP, and vice versa. The reasoning about both semantics is essentially the same because the only difference concerns the (arbitrary) undenotable sets which can be referred to by plural variables.

## 6.2. Plural specification of choice

Plural variables provide access to arbitrary sets. In the following example we attempt to utilize this fact to give a more concise form to the specification of choice.

*Example 6.17*

The specification

**S**: { S }

**F**: { $\sqcup.\_ : S^* \rightarrow S$ }

**Π**: { $\mapsto \sqcup.x^* \succ\!\!\prec x^*$ }

defines $\sqcup.$ as the choice operator – for any argument $t$, $\sqcup.t$ is capable of returning any element belonging to the set interpreting $t$.

□

The specification may seem plausible but there are several difficulties. Obviously, such a choice operation would be redundant in any specification since the axiom makes $\sqcup.t$ observationally equivalent to $t$, and lemma 4.6 allows us to remove any occurrences of $\sqcup.$ from the (derivable) formulae. Furthermore, observe how such a specification confuses the issue of nondeterministic choice. Choice is supposed to take a *set* as an argument and return *one element* from the set, or, perhaps, to convert an argument of type "set" to a result of type "individual". This is the intention of writing the specification above. But power algebras model all operations as functions between power sets and such a "conversion" simply does not make sense. The only points where conversion of a set to an individual takes place is when a term is passed as a singular argument to another operation. If we have an operation with a singular argument $f\colon \mathrm{S}\to\mathrm{S}$, then $f(t)$ will make (implicitly) the choice from $t$.

This might be particularly confusing because one tends to think of plural arguments as sets and mix up the semantic sets (i.e., the elements of the carrier of a power algebra) and the syntactic ones (as expressed by the profiles of the operations in the signature). As a matter of fact, the above specification does not at all express the intention of choosing an element from the set. In order to do that it would have to give choice the signature $Set(S)\to S$. Semantically, this would be then a function from $\mathcal{P}^+(Set(S))$ to $\mathcal{P}^+(S)$. Assuming that semantics of $Set(S)$ will somehow correspond to the power set construction, this makes things rather complicated, forcing us to work with a power set of a power set. Furthermore, since $Set(S)$ and $S$ are different sorts we cannot let the same variable range over both as was done in the example above.

The above example and remarks illustrate some of the problems with the intuitive understanding of plural parameters. Power algebras – needed for modelling such parameters – significantly complicate the model of nondeterminism as compared to multialgebras.

On the other hand, plural variables allow us to specify the "upper bound" of nondeterministic choice without using disjunction. The choice operation can be specified as the join which under the partial ordering $\prec$ interpreted as set inclusion will correspond to set union (cf. [17]).

*Example 6.18*

The following specification makes binary choice the join operation wrt. $\prec$ :

**S**: { S }

**F**: { $\_\sqcup\_ : \mathrm{S}\times\mathrm{S} \to \mathrm{S}$ }

**Π**: { 1. $\mapsto x^* \prec x^*\sqcup y^* \qquad \mapsto y^* \prec x^*\sqcup y^*$

2. $x\frown z^*, y\frown z^* \mapsto x\sqcup y \prec z^*$ }

□

Axiom 2, although using singular variables $x, y$, does specify the minimality of $\sqcup$ with respect to all terms. (Notice that the axiom $x^*\frown z^*, y^*\frown z^* \mapsto x^*\sqcup y^* \prec z^*$ would have a different, and in this context unintended, meaning.) We can show that whenever $\mapsto t\prec p$ and $\mapsto s\prec p$ hold (for arbitrary terms) then so does $\mapsto t\sqcup s\prec p$.

$$\dfrac{\dfrac{\dfrac{\dfrac{x\frown z^*,\ y\frown z^* \mapsto x\sqcup y \prec z^*}{x\frown p,\ y\frown p \mapsto x\sqcup y\prec p}\,\mathrm{R8}}{y\frown p\mapsto p\sqcup y\prec p}\,\mathrm{R7}}{\mapsto p\sqcup p\prec p}\,\mathrm{R7} \qquad \dfrac{\mapsto s\prec p \qquad \dfrac{\mapsto t\prec p \qquad \mapsto p\sqcup p\prec p\sqcup p}{\mapsto t\sqcup p\prec p\sqcup p}\,\mathrm{R3}}{\mapsto t\sqcup s\prec p\sqcup p}\,\mathrm{R3}}{\mapsto t\sqcup s\prec p}\,\mathrm{R3}$$

Violating our formalism a bit, we may say that the above proof shows the validity of the formula stating the expected minimality of join: $t\prec p, s\prec p \mapsto t\sqcup s\prec p$.

Thus, in any model of the specification from 6.18 $\sqcup$ will be a join. It is then natural to consider $\sqcup$ as the basic (primitive) operation used for defining other nondeterministic operations. Observe also that in order to ensure that join is the same as set union, we have to require the presence of *all* (finite) subsets in the carrier of the model. For instance, the power structure A with the carrier

$S^A = \{ \{1\},\{2\},\{3\},\{1,2,3\} \}$ and

$\sqcup^A$ defined as $x^A \sqcup^A y^A = \{1,2,3\}$ whenever $x^A \neq y^A$

will be a model of the specification although $\sqcup^A$ is not the same as set union.

## 7. Conclusion

We have defined the algebraic semantics for singular (call-time-choice) and plural (run-time-choice) passing of nondeterministic parameters. One of the central results reported in the paper is soundness and completeness of two new reasoning systems, NEQ and NEQ*, respectively, for singular and plural semantics. The plural calculus NEQ* is a minimal extension of NEQ which merely allows unrestricted substitution for plural variables. This indicated a close relationship between the two semantics. We have shown that plural specifications have equivalent (modulo undenotable sets) singular formulations if one considers the plural axioms as singular axiom schemata.


## Acknowledgments

We are grateful to Manfred Broy for pointing out the inadequacy of our original notation and to Peter D. Mosses for the observation that in the presence of plural variables choice may be specified as join with Horn formulae.


REFERENCES


[1] Bergstra, J.A., Klop, J.W., "Algebra of communicating processes", *Proceedings of CWI Symposium on Mathematics and CS*, 89 - 138, Oct. 6-7 1986.

[2] Clinger, W., "Nondeterministic call by need is neither lazy nor by name", *Proc. ACM Symp. LISP and Functional Programming*, 226-234, 1982.

[3] Dijkstra, E.W., *A Discipline of Programming*, Prentice-Hall, 1976.

[4] Ehrig, H., Mahr, B., *Fundamentals of Algebraic Specification*, vol. 1, Springer, 1985.

[5] Engelfriet, J., Schmidt, E.M., "IO and OI. 1", *Journal of Computer and System Sciences*, vol. 15, 328-353, 1977.
[6] Engelfriet, J., Schmidt, E.M., "IO and OI. 2", *Journal of Computer and System Sciences*, vol. 16, 67-99, 1978.
[7] Goguen, J.A., Meseguer, J., "Completeness of Many-Sorted Equational Logic", *SIGPLAN Notices*, vol. 16, no. 7, 1981.
[8] Hennessy, M.C.B., "The semantics of call-by-value and call-by-name in a nondeterministic environment", *SIAM J. Comput.*, vol. 9, no. 1, 1980.
[9] Hoare, C.A.R., *Communicating Sequential Processes*, Prentice-Hall International Ltd., 1985.
[10] Hußmann, H., *Nondeterministic Algebraic Specifications*, Ph.D. thesis, Fakultät für Mathematik und Informatik, Universität Passau, 1990.
[11] Hußmann, H., *Nondeterminism in Algebraic Specifications and Algebraic Programs*, Birkhäuser, 1993.
[12] Kaplan, S., "Rewriting with a Nondeterministic Choice Operator", *Theoretical Computer Science*, vol. 56, 37-57, 1988.
[13] Kapur, D., *Towards a theory of abstract data types*, Ph.D. thesis, Laboratory for CS, MIT, 1980.
[14] Meldal, S., "An Abstract Axiomatization of Pointer Types", in Proceedings of *The 22nd Annual Hawaii International Conference on System Sciences,* IEEE Computer Society Press, 1989.
[15] Meseguer, J., "Conditional rewriting logic as a unified model of concurrency", *TCS*, no. 96, 73-155, 1992.
[16] Milner, R., *Calculi for Communicating Systems*, LNCS vol. 92, Springer, 1980.
[17] Mosses, P.D., "Unified Algebras and Institutions", in Proceedings of *LICS'89, Fourth Annual Symposium on Logic in Computer Science,* 1989.
[18] Ore, C.E.S., *Introducing Girard's quantitative domains; the quantitative domains as a model for nondeterminism*, Ph.D. thesis, Dept. of Informatics, University of Oslo, 1988.
[19] Plotkin, G., "Domains", 1983, lecture notes, Dept. of Computer Science, University of Edinburgh.
[20] Søndergaard, H., Sestoft, P., *Non-Determinacy and Its Semantics*, Tech. Rep. 86/12, Datalogisk Institut, Københavns Universitet, January 1987.
[21] Schwartz, R.L., "An axiomatic treatment of ALGOL 68 routines", in Proceedings of *Sixth Colloquium on Automata, Languages and Programming,* vol. 71, Springer, 1979.
[22] Smyth, M.B., "Power domains", *J. of Computer and System Sciences*, vol. 16, 1978.
[23] Subrahmanyam, P.A., "Nondeterminism in Abstract Data Types", in *Automata, Languages and Programming*, LNCS, vol. 115, Springer, 1981.
[24] Walicki, M., *Algebraic Specifications of Nondeterminism*, Ph.D. thesis, University of Bergen, Department of Informatics, 1993.
[25] Walicki, M., Meldal, S., *Multialgebras, Power Algebras and Complete Calculi of Identities and Inclusions*, 10th ADT-W, 1994. [to appear in LNCS, Recent Trends in Data Type Specification]
[26] Winskel, G., "An introduction to event structures", LNCS, vol. 354, Springer, 1988.
[27] Wirsing, M., "Algebraic Specification", in *Handbook of Theoretical Computer Science*, vol. B, The MIT Press, 1990.