August 1997

Singular and plural non-deterministic parameters

Sigurd Meldal
San Jose State University, sigurd.meldal@sjsu.edu

M. A. Walicki

Follow this and additional works at: https://scholarworks.sjsu.edu/computer_eng_pub

Part of the Computer Engineering Commons

Recommended Citation

This Article is brought to you for free and open access by the Computer Engineering at SJSU ScholarWorks. It has been accepted for inclusion in Faculty Publications by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.
SINGULAR AND PLURAL NONDETERMINISTIC PARAMETERS*

MICHAL WALICKI† AND 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 simultaneously. 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.

Key words. algebraic specification, many-sorted algebra, nondeterminism, sequent calculus

AMS subject classifications. 68Q65, 68Q60, 68Q10, 68Q55, 03C05, 08A70

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], and 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 the theory 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.

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

*Received by the editors March 9, 1994; accepted for publication (in revised form) August 7, 1995.
The research of the first author was partially supported by the Architectural Abstraction project under NFR (Norway), the CEC under ESPRIT-II Basic Research Working Group 6112 COMPASS, DARPA under ONR contracts N00014-92-J-1928 and N00014-93-J-1335, and Air Force Office of Scientific Research grant AFOSR-91-0354.
†Department of Informatics, University of Bergen, HiB, 5020 Bergen, Norway (michal.walicki@ii.uib.no, sigurd.meldal@ii.uib.no).
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 \( \phi: g(x) = (\text{if } x = 0 \text{ then } a \text{ else } c) \), whereas the plural interpretation need not satisfy this formula. For instance, under the singular interpretation \( g(\{0, 1\}) \) will yield either \( a \) or \( c \), whereas the set of possible results of \( g(\{0\}) \) 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 the 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 possible 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 expresses 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 nondeterministic, \( h(t) \) 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 and 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 deductive systems and power models which may serve also to highlight the increased complexity problems with intuitive understanding.

Proofs of the theorems are motivated by the results from [24] where the

2. The specification language

signature \( \Sigma \) is a pair \((S,F)\) of some result sorts in \( S \). The set of terms

by \( W_{S,X}. \) We always assume that \( S, S_{W_{X}}, \) is not empty.\(^1\)

If \( \xi \) is a set of sequents of atomic \( \Lambda \)-formulas, \( \xi \) is called a consequent, and both are to be understood with respect to the original operation. A singular consequent or consequent to be empty when it has exactly one formula as antecedent and a Horn formula with empty antecedent.

All variables occurring in a set \( \xi \) are variables of the whole sequent. A sequent is satisfied, if the antecedent is false or one of the sequent formulas \( a_1 \wedge \cdots \wedge a_n \Rightarrow e_1 \vee \cdots \vee e_m \) is valid.

For any term \((\text{formula set of terms}) \) \( \sigma \text{ at } t \text{ or } s \) in \( W_{S,X}. \) An arbitrary \( t \) may be interpreted as nonempty intersection of \( t \text{ and } s \text{ at } \). For a given specification \( SP = (\Sigma, W_{S,X}, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots) \) and signature \( \Sigma \).

The above conventions will between the singular and the plural aspects of the power models and the set of plural variables in a term \( s \) is reflected in the notation by the structure of the set of plural variables in a term \( s \). For the corresponding extension \( SP^\ast \).

3. A note on the intuitive interpretation

By interpreting specifications in some formal (operations corresponding to set-interpretations as sets of possibilities) operations correspond to set-interpretations as sets of possibilities. We, on the other hand, interpret facts, i.e., facts which have to hold. This is achieved by interpreting the operation. Every two syntactic occurrences of \( \xi \) are different. For nondeterministic terms the

\(^1\)This restriction is motivated by the correspondence of the respective deductive systems and power models which may serve also to highlight the increased complexity problems with intuitive understanding.

4. A note on the intuitive interpretation

by interpreting specifications in some formal (operations corresponding to set-interpretations as sets of possibilities) operations correspond to set-interpretations as sets of possibilities. We, on the other hand, interpret facts, i.e., facts which have to hold. This is achieved by interpreting the operation. Every two syntactic occurrences of \( \xi \) are different. For nondeterministic terms the
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 $\mathcal{W}_X$. We always assume that, for every sort $S$, the set of ground words of sort $S$, $S^{\mathcal{W}_X}$, is not empty.

$\Pi$ is a set of sequents of atomic formulas written as $a_1, \ldots, a_n \rightarrow e_1, \ldots, e_m$. The left-hand side (LHS) of $\rightarrow$ is called the antecedent and the right-hand side (RHS) the consequent, and both are to be understood as sets of atomic formulas (i.e., the ordering and multiplicity of the atomic formulas 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).

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 $a_1 \land \cdots \land a_n \rightarrow e_1 \lor \cdots \lor e_m$ is valid).

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

An atomic formula in the consequent is either an equation, $t = s$, or an inclusion, $t \leq s$, of terms $t, s \in \mathcal{W}_X$. An atomic formula in the antecedent, written $t \rightarrow s$, will be interpreted as nonempty intersection of the (result) sets corresponding to $t$ and $s$. For a given specification $\mathcal{SP} = (\Sigma, \Pi)$, $\mathcal{L}(\mathcal{SP})$ will denote the above language over the signature $\Sigma$.

The above conventions will be used throughout the paper. The distinction between the singular and the 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 $\forall^*[t]$, a specification with plural arguments $\mathcal{SP}^*$, 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 formulas 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.

---

1This 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 a similar solution can be applied in our case.
Typically, equality is interpreted in a multialgebra as set equality \([13, 23, 12]\). For instance, the formula \(t = s\) 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 \(t = s\) 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 identity but to identity of one-element sets. Thus the simple formula \(t = s\) 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 \(\rightarrow t = t\) holds are exactly the deterministic terms, denoted by \(Dsp\). 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: \(s \triangleleft t\) holds iff the set of possible results of \(s\) is included in the set of possible results of \(t\), and \(s \triangleleft t\) 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.**

\[
\begin{align*}
S &: \{\text{Nat}\}, \\
F &: \text{Nat} \rightarrow \text{Nat} \\
&= \text{(zero)} \\
&= \text{(successor)} \\
&= \text{(binary nondeterministic choice)} \\
I &= (1) \rightarrow 0 = 0 \\
&= (2) \rightarrow s(x) = s(x) \\
&= (3) 1 \rightarrow 0 \\
&= (4) \rightarrow 0 < 0 \cup 1 \\
&\quad \quad \rightarrow 1 < 0 \cup 1
\end{align*}
\]

The first two axioms make zero and successor deterministic. A limited form of negation is present in \(L\) in the form of sequents with empty consequent. Axioms \(3\) makes 0 distinct from 1. Axioms \(4\) make then \(\rightarrow\) 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 \(0 \cup 1\). In most models all other kinds of elements may be among its possible results as well, since no extension of the result set of \(0 \cup 1\) will violate the inclusions of \(4\). If we are satisfied with this degree of precision, we may stop here and use only the 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 formulas in the consequents. Now, if we write the axiom

\[
(5) \rightarrow 0 \cup 1 = 0, 0 \rightarrow 1
\]

the two occurrences of \(0 \cup 1\) return either 0 or 1; i.e., \(0 \cup 1\) is a nondeterministic term as referred to by binding both occurrences to \(x\),

\[
(5') \rightarrow x \cup 0 \rightarrow x = 0, 0 \rightarrow x = 1
\]

The axiom says: whenever \(0 \cup 1\) returns either 0 or 1; i.e., \(0 \cup 1\) is a nondeterministic term as referred to by binding both occurrences to \(x\)

\[
(5'') \rightarrow x^* \cup 0 \rightarrow x^* = 0, 0 \rightarrow x^* = 1
\]

would have a completely different meaning common in the literature on program languages \([2, 8, 11]\), in spite of the common use of terms for variables. Any substituted term yields a unique result. The meanings in the subsection on reasoning, for instance, to conclude 0 is a singleton set, and are.

4. The singular case: Specification with a singlualr algebraic semantics of predicate

**4.1. Multistructures and Multihomomorphisms**

**Definition 4.2 (Multistructure).**

A function \(\Phi: A \rightarrow B\) (i.e., a multihomomorphism from a \(\Sigma\)-multia
gebra \(A\) to a \(\Sigma\)-multialgebra \(B\))

\[
\Phi(f^A(a_1, \ldots, a_n)) \subseteq f^B(b_1, \ldots, b_m)
\]

If all inclusions in \(H1\) and \(H2\) are strict, it is strictly loose (or just loose).

Since multihomomorphisms preserve singletons and are \(\Sigma\)-monadic,
SINGULAR AND PLURAL NONDETERMINISTIC PARAMETERS

995

\[ (5) \implies 0 \cup 1 = 0, \quad 0 \cup 1 = 1, \]

the two occurrences of \(0 \cup 1\) refer to two arbitrary applications and, consequently, we obtain that either any application of \(0 \cup 1\) equals 0 or else it equals 1, i.e., that \(\cup\) 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 \(0 \cup 1\) 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

\[ (5') \ x \sim 0 \cup 1 \iff x = 0, \ x = 1. \]

The axiom says: whenever \(0 \cup 1\) 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

\[ (5'') \ x^* \sim 0 \cup 1 \iff x^* = 0, \ x^* = 1 \]

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 \(0 \cup 1 = 0 \cup 1 \implies 0 \cup 1 = 0, \ 0 \cup 1 = 1\) 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 \(S\)-sorted set,
2. for every \(f: S_1 \times \cdots \times S_n \to S\) in \(F\), there is a corresponding function \(f^M: S_{1}^M \times \cdots \times S_{n}^M \to \mathcal{P}(SM)\).

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

1. for each constant symbol \(c \in F\), \(\Phi(c^A) \subseteq c^B\),
2. for every \(f: S_1 \times \cdots \times S_n \to S\) in \(F\) and \(a_1, \ldots, a_n \in S^A\), \(\Phi(f^A(a_1, \ldots, a_n)) \subseteq f^B(\Phi(a_1), \ldots, \Phi(a_n))\).

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 nonempty 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(Σ). It has the distinguished word structure MWE 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 one-element sets (i.e., we do not take special pains to distinguish MWE and WΣ). MWE is not an initial Σ-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 Σ-multistructure iff for every set of variables X and assignment β: X → |M|, there exists a unique function β[-]: WΣ,X → P+(|M|) such that

1. β[x] = {β(x)},
2. β[c] = cM,
3. β[f(t)] = \{f^M(y) | y ∈ β[t]\}.

In particular, for X = ∅ there is a unique interpretation function (not a multihomomorphism) I: WΣ → P+(|M|) satisfying the last two points of this definition.

As a consequence of the definition of multistructures, all operations are \(\leq\)-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 Σ-multistructure M satisfies an \(L(\Sigma)\) sequent \(\pi\)

\[ t_i \leftarrow s_i \leftarrow p_j \leftarrow r_j, m_k \leftarrow n_k, \]

written \(M \models \pi\) iff for every \(\beta: X \rightarrow M\) we have

\[ \bigwedge_i \beta[s_i] \cap \beta[t_i] \neq \emptyset \Rightarrow \bigvee_j \beta[p_j] = \beta[r_j] \lor \bigvee_k \beta[m_k] \subseteq \beta[n_k], \]

where \(A \models B\) iff A and B are the same one-element set.

An SP-multimodel is a Σ-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 \(\leftarrow\) 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 \leftarrow t ...” is equivalent to “... x \leftarrow s, x \leftarrow t ...”. A binding “... x \leftarrow t ...” is also equivalent to the more familiar “... x ∈ t ...”, so the notation \(s \leftarrow t\) may be read as an abbreviation for the more elaborate formula with two \(\epsilon\) 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 introduced the calculus NEQ which is sound and complete with respect to the class MMod(SP). Its rules are as follows:

(R1) \(\rightarrow x = x, \quad x \in \mathcal{V}\),

(R2) \(\frac{\Gamma_i \vdash s \equiv \epsilon}{\Gamma_i \vdash \epsilon \rightarrow \epsilon, X \leftarrow \epsilon} \)

(R3) \(\frac{\Gamma_i \vdash s \equiv \epsilon}{\Gamma_i, \Delta \vdash \epsilon \rightarrow \Delta_{i+1}} \)

(R4) \(\frac{\Gamma_i \vdash x \leftarrow x}{\Gamma_i \vdash \epsilon \rightarrow \epsilon, X \leftarrow \epsilon} \)

(R5) \(\frac{\Gamma_i \vdash \epsilon \rightarrow \Delta}{\Gamma_i, \Gamma'_i \vdash \epsilon \rightarrow \Delta} \)

(R6) \(\frac{\Gamma_i \vdash \epsilon \rightarrow \Delta}{\Gamma, \Gamma'_i \vdash \epsilon \rightarrow \Delta} \)

(R7) \(\frac{\Gamma_i \vdash \epsilon \rightarrow \Delta}{\Gamma_i \vdash \epsilon \rightarrow \Delta} \)

\(\Gamma_i\) denotes \(\Gamma\) with \(b\) substituted for \(t\) in order.

The fact that \(\equiv\) is a particularization of \(\epsilon\) to variables and is sound (singular) variables.

(R2) is a paramodulation rule. (R3) allows derivation of the statement, it allows derivation of the statement, deterministic and prevents substitution.

(R4) allows “specialization” of \(t\) which is included in \(t_2\), \(t_1\) substituted for don’t occur in the consequent, and inclusion in the consequent.

(R5) expresses then \(s \leftarrow t \leftarrow s \leftarrow t\) does not hold in the nonempty intersection of the respective term occurring at most once in the consequent.

We will write \(\Pi_{\epsilon}^{\text{CAL}} \pi\) to indicate the completeness of the calculus.

The counterpart of soundness is proved in [24].

Theorem 4.5. NEQ is sound and complete with respect to MMod(SP).

Proof idea. Soundness is proved by an indirect style argument. The axioms set \(\Pi_{\epsilon}^{\text{CAL}}\)
M _\Sigma_ defined in the obvious way, is a set. We will treat such singleton sets as singleton sets. We do not take special pains to distinguish singular structures since it is deterministic for a given multistructure. We do consider the useful fact from [11].

For every set of variables X and assignments \( \beta(X) \), we have:\n\[ W \Sigma, X \rightarrow P^+(|M|) \]

Arity interpretation function (not a multihomomorphism) for the two points of this definition.

Now, all operations are \( <\)-monotone (i.e., also that assignment in the lemma preserves the properties of a multistructure) means we have chosen. For deterministic terms this reflects our definition.

If structure M satisfies an \( L(\Sigma) \) sequent \( \pi \) and
\[ \bigwedge \beta[m_k] \leq \beta[n_k], \]

\( \exists X \subseteq \Sigma \) with the antecedent then \( X \) satisfies all the axioms of SP. We will write \( \Pi \vdash_{\text{CAL}} \pi \) to indicate that \( \pi \) is provable from \( \Pi \) with the calculus CAL.

The counterpart of soundness/completeness of the equational calculus is as follows [24].

**Theorem 4.5.** \( \text{NEQ} \) is sound and complete with respect to \( \text{MMod}(\text{SP}) \):

\[ \text{MMod}(\text{SP}) \models \pi \iff \Pi \vdash_{\text{NEQ}} \pi. \]

**Proof idea.** Soundness is proved by induction on the length of the proof \( \Pi \vdash_{\text{NEQ}} \pi \). The proof of the completeness part is standard, albeit rather involved, Henkin-style argument. The axiom set \( \Pi \) of SP is extended by adding all \( L(\Sigma) \) formulas \( \pi \).
which are consistent with \( \Pi \) (and the previously added formulas). If the addition of \( \pi \) leads to inconsistency, one adds the negation of \( \pi \). Since empty consequents provide only a restricted form of negation, the general negation operation is defined as a set of formulas 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 \( t \rightarrow s \) satisfy the same formulas.

**Lemma 4.6.** \( t \rightarrow s \) iff, for any sequent \( \pi \), \( \Pi \vdash \text{NEQ } \pi \) iff \( \Pi \vdash \text{NEQ } \pi \). \( \square \)

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 [17] 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 signatures will be referred to as \( \mathcal{L}^{*} \), 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 \( f(...) \), are always to be understood plurally, and no superscripting will be used at such places.

**Definition 5.1.** Let \( \Sigma \) be a \( \mathcal{L}^{*} \)-signature. \( A \) is a \( \mathcal{L}^{*} \)-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 \( P^{+}(S) \) of some basis set \( S^{+} \),
2. for every \( f: S_{1} \times \cdots \times S_{n} \rightarrow S \) in \( \Sigma \), \( f^{A} \) is a \( \subseteq \)-monotonic function \( S_{1}^{A} \times \cdots \times S_{n}^{A} \rightarrow S^{A} \) 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 ith argument is the most common notion to but to its semantic counterparts.

**Definition 5.2.** A function \( f \in \Sigma \) is singular in the \( i \)th argument iff \( x_{i} \in S_{i}^{A} \) and all \( x_{k} \in S_{k}^{A} \) (for \( k \neq i \)) (for some \( x \in x_{i} \)).

Thus, the definition of plural arguments can be modeled by the semantic one.

Note the unorthodox point of using the whole power set but allow it to pass primitive nondeterministic operations without all finite subsets are needed for a model of the join operation (under the same logic unless all are present otherwise.). Consequently, we also pass the formulas instead, give the user means of expressing this (by choice) directly.

Let \( \Sigma \) be a signature, \( A \) an \( \mathcal{L}^{*} \)-power model of \( \Sigma \) (for \( \Sigma \), \( \mathcal{L}^{*} \) a set of plural variables, and \( m_{k}, n_{k} \) be a signature), and \( x \in X: |\beta(x)| = 1 \). (Saying as \( \beta(x_{1}, x_{2}) \) to satisfy this last condition.)

We will allow both singular and plural parameter passing in one specification. The next subsection gives the necessary extension of the calculus NEQ to handle this generalized situation.

**Definition 5.3 (Satisfiability).** Let \( m_{j}, n_{j} \) be a signature, \( A \) a power model of \( \Sigma \), and \( A \) satisfies all axioms from \( \Sigma \).

Except for the change in the rule 4.4, which is the reason for rule 5.3.

5.2. **The calculus for power structures**

with one additional rule:

**Rule (R8):** \( \Pi \vdash \Delta \)

Rules (R1)-(R7) remain unchanged, in particular, any \( t \) may be a plural argument in rule R8.

The new rule (R8) expresses that one may substitute an arbitrary term by another. In particular, any \( t \) may be a plural argument.

**Theorem 5.4.** For any \( \mathcal{L}^{*} \)-power structure \( A \in PStr(\Sigma) \),

\( PMod(\Sigma) \rightarrow PMod(\Sigma) \).
999 SINGULAR AND PLURAL NONDETERMINISTIC PARAMETERS

The singularity in the 
ith argument in this defi-
ition refers not to the syntax-
mic notion but to its semantic
counterpart.

DEFINITION 5.2. A function \( f^A: S_1^A \times \cdots \times S_n^A \rightarrow S^A \) in a power structure \( A \) is singular in the 
ith argument iff it is \( \cup \)-additive in the 
ith argument, i.e., iff for all 
\( x_i \in S_i^A \) and all \( x_k \in S_k^A \) (for \( k \neq i \)),

\[
\{ f^A(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \mid x_i \in x_i \}.
\]

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 6.8). 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 variables and
\( X^* \) a set of plural variables, and \( \beta \) an assignment \( X \cup X^* \rightarrow |A| \) 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 \( t(t(x, x^*)) \) in \( A \) defined as \( t^A(\beta(x), \beta(x^*)) \).

DEFINITION 5.3 (Satisfiability). Let \( A \) be a \( \Sigma \)-power structure and \( \pi : t_i \rightarrow s_i \mapsto
p_j = r_j, m_k < n_k \) be a sequent over \( L^*(\Sigma, X, X^*) \). \( A \) satisfies \( \pi \), \( A \models \pi \), iff for every
assignment \( \beta : X \cup X^* \rightarrow |A| \), we have that

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

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

Except for the change in the notion of an assignment, this is identical to 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 \( \text{NEQ} \) is extended
with one additional rule:

\[
(\text{R8}) \quad \Gamma \vdash \Delta \quad \Gamma \vdash \Delta^* \quad \text{if } \quad \Delta^* \subseteq \Delta^*.
\]

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 \( \text{NEQ}^* \) denote the calculus \( \text{NEQ} + \text{R8} \).

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 \( \pi \) 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 \( \rightarrow x^* \prec x^* \), but this
is not sufficient for performing a general substitution for a singular variable. The
main result concerning \( \text{PMod} \) and \( \text{NEQ}^* \) is as follows.

THEOREM 5.4. For any \( L^* \)-specification \( SP \) and \( L^*(\Sigma, SP) \) sequent \( \pi : \)

\[
\text{PMod}(SP) \models \pi \text{ iff } \Pi \vdash_{\text{NEQ}^*} \pi.
\]
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 formulas 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 \( L \) and \( L^* \) specifications (section 6.1) and emphasize some points of difference (section 6.2).

6.1. The “equivalence” of both semantics. The following example illustrates a strong sense of equivalence of \( L \) and \( L^* \).

**Example 6.1.** Consider the following plural definition:

\[ \rightarrow f(x^*) < \text{ if } x^* = x^* \text{ then } 0 \text{ else } 1. \]

It is “equivalent” to the collection of definitions

\[ \rightarrow f(t) < \text{ if } t = t \text{ then } 0 \text{ else } 1 \]

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 \( B \), \( [\mathcal{P}(A) \rightarrow \mathcal{P}(B)] \simeq [\mathcal{P}(A) \rightarrow \mathcal{P}^+(B)] \), 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.

**Lemma 6.2.** Let \( SP \) be a singular specification (i.e., all operations are singular in all arguments), let \( A \in \text{MStr}(SP) \), and let \( \pi \) be a sequent in \( L(SP) \). Then \( A \models \pi \) iff \( A^* \models \pi^* \) and so \( A \in \text{MMod}(SP) \) iff \( A^* \in \text{PMod}(SP) \).

Call an \( L^* \) sequent \( \pi \) p-ground (for pluraly ground) if it does not contain any plural variables.

**Theorem 6.3.** Let \( SP^* = (\Sigma^*, \Pi^*) \) be an \( L^* \) specification. There exists a (usually infinite) \( L \) specification \( SP = (\Sigma, \Pi) \) such that

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

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

The axioms \( \Pi \) are obtained from \( \Pi^* \) as in Example 6.1. For every \( \pi^* \in \Pi^* \) with plural variables \( x_1^* \cdots x_n^* \), let \( \pi = (\pi^* : x_1^* \cdots x_n^* \mid t_1 \cdots t_n \in W_{\Sigma, X}) \). Obviously, for any \( \pi \in L(SP) \) if \( \Pi \vdash \text{NEQ} \pi \) then \( \Pi^* \vdash \text{NEQ}^* \pi \). If \( \Pi^* \vdash \text{NEQ}^* \pi \) then the proof can be simulated in \( \text{NEQ} \). Let \( \pi'(x^*) \) be the last sequent used in the \( \text{NEQ}^* \)-proof which contains plural variables \( x^* \) and the sequent \( \pi' \) is the next one obtained by (R8). Build the analogous \( \text{NEQ}^* \)-proof tree with all plural variables replaced by the terms which occupy their place in \( \pi' \). 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 completeness of \( \text{NEQ} \) and \( \text{NEQ}^* \) imply the conclusion of the theorem. \( \square \)
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 \( \in \) 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 6.2. It is not necessarily the case that MMod*(SP) \( \models \Pi^* \), as the following argument illustrates.

**Example 6.4.** Let \( M^* \in \text{MMod}^*(SP) \) have infinite carrier, \( \pi^* \in \Pi^* \) be \( t_i \sim s_i \mapsto p_j = r_j \), \( m_k < n_k \) with \( x^* \in V[\pi^*] \), and \( \beta: X \cup X^* \rightarrow [M^*] \) be an assignment such that \( \beta(x^*) = \{ m_1 \ldots m_l \} \) is a set which is not denoted by any term in \( W_{\Sigma,X} \). Let \( \beta_1 \) be an assignment equal to \( \beta \) except that \( \beta_1(x^*) = \{ m_1 \} \), i.e., \( \beta = \cup \beta_1 \). Then \( M^* \models \beta[\pi^*] \) iff

\[
M^* \models \beta[t_i] \cap \beta[s_i] \neq \emptyset \Rightarrow \beta[p_j] \equiv \beta[r_j] \vee \ldots \vee \beta[m_k] \leq \cup \beta[n_k]
\]

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

(1) \( M^* \models \beta[t_i] \cap \beta[s_i] \neq \emptyset \Rightarrow \beta[p_j] = \beta[r_j] \vee \ldots \vee \beta[m_k] \leq \cup \beta[n_k] \)

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 nonempty. So we are not guaranteed that \( M^* \in \text{PMod}(SP^*) \).

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 \( *: \text{MMod}(SP) \rightarrow \text{PMod}(SP) \) from Lemma 6.2.

**Definition 6.5.** Given a singular specification \( SP \) and \( M \in \text{MMod}(SP) \), we denote by \( \mid M \) the following power structure:

\[
\begin{align*}
(1) & \mid M \subseteq \mathcal{P}^+(\mid M) \\
(2) & \text{for every } \pi^* \in \Pi^* \text{ with } t_i \sim s_i \mapsto p_j = r_j \text{ there exists } t \in W_{\Sigma,X}, \pi \in [M] \text{ such that:}
\end{align*}
\]

\[
t^M(\pi) = m.
\]

Then, for any assignment \( \beta: X^* \rightarrow \mid M \) there exists an assignment \( \theta: X^* \rightarrow W_{\Sigma,X} \) and an assignment \( \alpha: X \rightarrow \mid M \) such that \( \beta(x^*) = \alpha \theta(x^*) \) (2), i.e., such that the diagram in Figure 1 commutes.

Since \( M \in \text{MMod}(SP) \), it satisfies all the axioms \( \Pi \) obtained from \( \Pi^* \) and the commutativity of the figure gives us the second part of the following.

**Corollary 6.6.** Let \( SP^* \) and \( SP \) be as in Theorem 6.3. Then

\[
\begin{align*}
1 \text{ PMod}(SP^*) \models \Pi, \text{ i.e., } 1 \text{ PMod}(SP^*) \subseteq \text{MMod}(SP), \\
1 \text{ MMod}(SP) \models \Pi^*, \text{ i.e., } 1 \text{ MMod}(SP) \subseteq \text{PMod}(SP^*).
\end{align*}
\]
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 6.7.** The specification

$S: \{ S \}$,

$F: \{ \cup_: S^* \rightarrow S \}$,

$II: \{ \rightarrow \cup_.x^* \rightarrow x^* \}$

defines $\cup_.$ as the choice operator: for any argument $t$, $\cup_.t$ 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 $\cup_.t$ observationally equivalent to $t$, and Lemma 4.6 allows us to remove any occurrences of $\cup_.$ from the (derivable) formulas. 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 behind writing the specification above. But power algebras model all operations as functions on 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 $f: S \rightarrow S$, 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, an intention of choosing an element actually is also expressed as set union (cf. Figure 2). Violating our requirement shows the validity of the form $s < p \rightarrow t u s < p$.

Thus, in any model of the specification then natural to consider $\cup_.$ as the modality for nondeterministic operations. Considering it the same as set union, we have to remove the $\cup_.$ from the model. For instance, the model

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

$\cup^A$ defined as $\cup_.^A$ will be a model of the specification.

### 7. Conclusion

We have defined two approaches to nondeterministic choice (singular (run-time-choice) and plural (run-time-choice)) and plural (run-time-choice) and the central results reported in the
class of power models of a plural parameters. Thus, a class of power models of some singular specification is essentially the same because variables provide access to arbitrary sets which can be referred to.

We utilize this fact to give a more formal definition of choice.

variables provide access to arbitrary sets which can be referred to.

t is capable of returning any element of the carrier of a power set, whereas the choice from the set or, perhaps, to convert an arbitrary set into a profile. This is the intention behind plural variables. Thus, we can model all operations as functions 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, we would have to give choice the signature \( \text{Set}(S) \rightarrow S \). Semantically, this would then be a function from \( \mathcal{P}^{+}(\text{Set}(S)) \) to \( \mathcal{P}^{+}(S) \). Assuming that semantics of \( \text{Set}(S) \) 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 \( \text{Set}(S) \) and \( S \) are different sorts, we cannot let the same variable range over both as was done in the example above.

Example 6.7 and remarks illustrate some of the problems with the intuitive understanding of plural parameters. Power algebras, needed for modeling such parameters, significantly complicate the model of nondeterminism as compared with 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 6.8. The following specification makes binary choice the join operation wrt. \( \prec \):

\[
\begin{align*}
S & \colon \{ \{ S \} \}, \\
F & \colon \{ \cup, \cup^* : S \times S \rightarrow S \}, \\
\Pi & \colon \{ (1) \quad x^* \prec x^* \cup y^* \quad \rightarrow \quad y^* \prec x^* \cup y^* \\
(2) \quad x^* \prec z^*, y^* \prec z^* \quad \rightarrow \quad x^* \cup y^* \prec z^* \}.
\end{align*}
\]

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

Thus, in any model of the specification from Example 6.8 \( \cup \) will be a join. It is then natural to consider \( \cup \) 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

\[
S^A = \{ \{1\}, \{2\}, \{3\}, \{1, 2, 3\} \}
\]

\( \cup^A \) defined as \( x^A \cup^A y^A = \{1, 2, 3\} \) whenever \( x^A \neq y^A \)

will be a model of the specification although \( \cup^A \) 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 formulas.

REFERENCES

[24] M. Walicki, Algebraic Specimations, University of Be
[27] M. Wirsing, Algebraic speci
The MIT Press, Cambri
SINGULAR AND PLURAL NONDETERMINISTIC PARAMETERS

S. MELDAL

for singular and plural semantics.

NEQ which merely allows unre-

lated a close relationship between

lications have equivalent (modulo

ners the plural axioms as singular

Broy for pointing out the in-

esses for the observation that in

as join with Horn formulas.


dating processes, in Mathematics and


nor by name, Proc. ACM Symposium


Specification, Vol. 1, Springer-Verlag,


Sorted Equational Logic, SIGPLAN

call-by-name in a nondeterministic

Prentice-Hall International Ltd., En-

Ph.D. thesis, Fakultät für Mathe-

ifications and Algebraic Programs,


D. thesis, Laboratory for Computer

in Proc. 22nd Annual Hawaii Inter-

Computer Society Press, Piscataway, NJ,


Notes in Computer Science, Vol. 92,

LICS '89, Fourth Annual Symposium

The Quantitative Domains as a

ematics, University of Oslo, Norway,

Computer Science, University of Edin-

and Its Semantics, Technical report

rutines, in Proc. Sixth Colloquium

Springer-Verlag, Basel, Switzerland,


Data types, in Automata, Languages

science, Vol. 115, Springer-Verlag, Basel,

Switzerland, 1981.


