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, 03C60, 08A70

PII. S0097-5397(97)264317

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 N00014-93-1-1335, 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 \( \text{if } x = 0 \text{ then } a \text{ else (if } x = 0 \text{ then } b \text{ else } c) \) and that we have a nondeterministic choice operation \( \cup \) 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 \( g(U, \{0, 1\}) \) will yield either \( a \) or \( c \), whereas the set of possible results of \( g(U, \{0, 1\}) \) 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 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 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 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 models and power models which may severe problems with intuitive understanding. Proofs of the theorems are motivated by the ideas of 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 is denoted by \( W_\Sigma^\infty \). We always assume that \( S \), \( S^W_\Sigma \), is not empty.

If \( t \) is a set of sequents of atomic formulas \( \Sigma \)

denotes 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 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 models and power models which may severe problems with intuitive understanding. Proofs of the theorems are motivated by the ideas of the results from [24] where 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 \((S, F)\) of sorts \(S\) and operation symbols \(F\) (with argument and result sorts in \(S\)). 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 \(S, S^{W_{x}}\), 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 \((\text{LHS})\) of \(\Rightarrow\) is called the consequent and the right-hand side \((\text{RHS})\) the antecedent, 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) \(x\), \(\forall[x]\) will denote the set of variables in \(x\). If the variable set is not mentioned explicitly, we may also write \(x \in V\) to indicate that \(x\) is a variable.

An atomic formula in the consequent is either an equation, \(t = s\), or an inclusion, \(t \subseteq s\), of terms \(t, s \in W_{\Sigma, X}\). An atomic formula in the antecedent, written \(t \not= s\), will be interpreted as nonempty intersection of the \((\text{result})\) sets corresponding to \(t\) and \(s\). For a given specification \(SP = (\Sigma, \Pi)\), \(L(\text{SP})\) 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 \(V^{*}[t]\), a specification with plural arguments \(SP^{*}\), the corresponding extension of the language \(L\) by \(L^{*}\), etc.

3. A note on the intuitive interpretation. Multialgebraic semantics [10, 13] interprets specifications in some form of power structures where the \((\text{nondeterministic})\) operations correspond to set-valued functions. This means that a \((\text{ground})\) 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.

\footnote{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 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 set equality 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 \( t = t \) 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: \( s \triangleleft t \) 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.**

\[ S: \{\text{Nat}\}, \]

\[ F: 0: \rightarrow \text{Nat} \quad (\text{zero}) \]

\[ 
\begin{align*}
\text{succ} : & \quad \text{Nat} \rightarrow \text{Nat} \\
\end{align*}
\]

\[ \triangleleft: \text{Nat} \times \text{Nat} \rightarrow \text{Nat} \quad (\text{binary nondeterministic choice}) \]

\[ II: (1) \rightarrow 0 = 0 \\
(2) \rightarrow \text{succ}(x) = \text{succ}(x) \\
(3) 1 \triangleleft 0 \rightarrow \quad (\text{As usual}, \text{we abbreviate } s^n(0) \text{ as } n.) \\
(4) \rightarrow 0 \triangleleft 0 \cup 1 \quad \rightarrow 1 \triangleleft 0 \cup 1
\]

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. Axiom (3) makes 0 distinct from 1. Axioms (4) make then \( \sqcap \) 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 \( 0 \sqcap 1 \). In most models all other kinds of elements may be among its possible results as well, since no extension of the result set of \( 0 \sqcap 1 \) 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

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

the two occurrences of \( 0 \sqcup 1 \) return 0 or 1, but not really nondeterministic but they do not guarantee that either any application of \( 0 \sqcup 1 \) will return 0 or 1. Therefore, both 0 and 1 be among the results of any application of \( 0 \sqcup 1 \) and the specification inconsistent.

What we are trying to say is that \( 0 \sqcup 1 \) can be either 0 or 1; i.e., \( 0 \sqcup 1 \) is a nondeterministic term as referred to by binding both occurrences to the same variable.

\[ (5') x \triangleleft 0 \sqcup 1 \rightarrow x = 0 \]

The axiom says: whenever \( 0 \sqcup 1 \) occurs, such an interpretation presuppuses that its value. Thus bindings have just one occurrence (Plurals variables, on the other hand, have an axiom

\[ (5'') x^* \triangleleft 0 \sqcup 1 \rightarrow x^* = 0 \]

This would have a completely different semantics common in the literature on programming languages [2, 8, 11], in spite of the terms for variables. Any substituted term yields a unique result, i.e., in the subsection on reasoning, one, for instance, to conclude 0 = 1, (though it could be obtained from 0 = 1).

4. The singular case: Singular semantics.

**4.1. Multistructures and multihomomorphisms.**

**Definition 4.2 (Multistructure).**

If

1. its carrier [\( M \)] is an \( S \)-set
2. for every \( f: S^1 \times \cdots \times S^n \rightarrow T \)
   A function \( \Phi: A \rightarrow B \) (i.e., a functionial multihomomorphism from a \( \Sigma \)-set \( A \) to a \( \Sigma \)-set \( B \))

\[ (H1) \text{ for each constant symbol } a \in A, \Phi(a) = a \]

\[ (H2) \text{ for each function symbol } f : S^1 \times \cdots \times S^n \rightarrow T, \Phi(f^A(a_1, \ldots, a_n)) \subseteq f^B(\Phi(a_1), \ldots, \Phi(a_n)) \]

If all inclusions in \( H1 \) and \( H2 \) are strict, the multihomomorphism is called loose and otherwise it is strictly loose (or just loose).\( P^+(S) \) denotes the set of non-empty subsets of the carrier \( S \); if we assume that \( S \) is a set, \( p^+(S) \) refers to its unique pointwise extension. Thus the simple formula \( f \rightarrow 0 \sqcup 1 \) is included in the set \( P^+(S) \).

Since multihomomorphisms preserve singletons and are \( \Sigma \)-monad,
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:

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

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 axiom

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

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 \( 0 \cup 1 = 0 \cup 1 \mapsto 0 \cup 1 = 0, 0 \cup 1 = 1 \) 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 \( f: S_1 \times \ldots \times S_n \rightarrow S \) in \( F \), there is a corresponding function \( f^M: S^M_1 \times \ldots \times S^M_n \rightarrow P^+(S^M) \).

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

\[
\Phi(c^A) \subseteq c^B,
\]

\[
\Phi(f^A(a_1, \ldots, a_n)) \subseteq f^B(\Phi(a_1) \cdots \Phi(a_n)).
\]

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

\( P^+(S) \) 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 \( c: \rightarrow S(2) \) 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 MW₂ 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 MW₂ and W₂). MW₂ is not an initial Σ-structure since it is deterministic and there 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[-]: W₂,X → P⁺(|M|) such that

1. β(x) = {β(x)},
2. β[c] = c|M|,
3. β[f(t₁)] = ∪{β[f(yₗ)] | yₗ ∈ β[tₗ]}.

In particular, for X = ∅ there is a unique interpretation function (not a multihomomorphism) I: W₂ → P⁺(|M|) satisfying the last two points of this definition.

As a consequence of the definition of multistructures, all operations are ≤-monic, i.e., β[s] ≤ β[t] ⇒ β[f(s)] ≤ β[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(Σ) sequent π

$t₁ ∼ s₁ ↔ p_j = r_j, m_k < n_k$,

written M |= π iff for every β: X → M we have

$\bigwedge_i β[t_i] \cap β[s_i] \neq ∅ \Rightarrow \bigvee_j β[p_j] = β[r_j] \lor \bigvee_k β[m_k] \subseteq β[n_k].$

where A = 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 ∼ in the antecedents is the same as using “elementwise” equality = 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.

**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) $\rightarrow x = x, \quad x ∈ V,$

(R2) $Γ₁ \vdash Δ₁ : \Gamma₂ \vdash Δ₂,$

(R3) $Γ₁ \vdash Δ₁ : \Gamma₂ \vdash Δ₂,$

(R4) (a) $x \sim y \rightarrow x = y,$

(R5) $Γ, t \sim t \rightarrow Δ,$

(R6) (a) $Γ \rightarrow Δ,$

(R7) $Γ, t \rightarrow Δ,$

Γbart denotes Γ with b substituted for t in order.

The fact that “=” is a partial order only to variables and is sound for (singular) variables.

(R2) is a paramodulation rule on deterministic (in the case where t is not deterministic) assignment in the lemma (and in general whenever it is an assignment of elements from a multistructure) means assignment of individuals, not sets.

The counterpart of soundness is the following.

**THEOREM 4.5.** NEQ is sound with respect to the class MMod(SP).

**Proof idea.** Soundness is proved by a two-step style argument. The axiom set Π is a sublanguage of the singular language, and therefore NEQ is sound with respect to the class MMod(SINGULAR AND I) for that sublanguage. The result follows from the completeness of NEQ with respect to the class MMod(SINGULAR AND I).
We will treat such singleton sets. We will not take special pains to discuss the structure since it is deterministic to a given multistructure. We do mention the useful fact from [11].

For every set of variables $X$ and assignment $\beta[-]$: $\mathcal{W}_{\Sigma, X} \rightarrow \mathcal{P}^+(\mathcal{M})$ such that

\[ (R2) \quad \frac{\Gamma_{t_2} \vdash \Delta_{t_2} ; \quad \Gamma' \vdash t_1 = t_2, \Delta'}{\Gamma_{t_1}, \Gamma' \vdash \Delta_{t_1}', \Delta'}, \]

\[ (R3) \quad \frac{\Gamma_{t_2} \vdash \Delta_{t_2} ; \quad \Gamma' \vdash t_1 < t_2, \Delta'}{\Gamma_{t_1}, \Gamma' \vdash \Delta_{t_1}', \Delta'}, \quad x \text{ not in a RHS of } <, \]

\[ (R4) \quad (a) \ x \rhd y \vdash x \equiv y, \quad (b) \ x \rhd t \vdash x \equiv t, \quad x, y \in \mathcal{V}, \]

\[ (R5) \quad \frac{\Gamma \vdash \Delta, s \leq t ; \quad \Gamma', s \rhd t \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'} \quad \text{(CUT)} \quad \leq \text{stands for either } = \text{ or } <, \]

\[ (R6) \quad (a) \ \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, e} \quad \text{(WEAK)}, \]

\[ (R7) \quad \frac{\Gamma, x \rhd t \vdash \Delta}{\Gamma_{t} \vdash \Delta_{t}}, \quad x \in \mathcal{V} - \mathcal{V}[t], \text{ at most one } x \text{ in } \Gamma \rightarrow \Delta \quad \text{(ELIM)}. \]

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

The fact that $\equiv$ 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 where $t_1 = t_2$ 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 $<$ is needed to prevent, for instance, the unsound conclusion $\vdash t_3 < t_1$ from the premises $\vdash t_3 < t_2$ and $\vdash t_1 < t_2$.

(R4) and (R5) express the relation between $\rhd$ in the antecedent and the equality and inclusion in the consequent. The axiom of standard sequent calculus, $e \rhd e$, (i.e., $s \rhd t \vdash s \equiv t$) does not hold in general here because the antecedent corresponds to nonempty intersection of the result sets while the consequent to the inclusion ($\leq$) or identity of one-element ($=$) result sets. Only for deterministic terms (in particular, variables) $s, t$ do we have that $s \rhd t \rightarrow s = t$ holds.

(R5) allows us to cut both $s \equiv t$ and $s \leq t$ with $s \rhd t \rightarrow \Delta$.

(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 $\Pi \vdash_{\text{CAL}} \pi$ 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.** NEQ is sound and complete with respect to MMod(SP): $MMod(SP) \models \pi$ 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 a standard, albeit rather involved, Henkin-style argument. The axiom set $\Pi$ of SP is extended by adding all $\mathcal{L}(\Sigma)$ formulas $\pi$. 

SINGULAR AND PLURAL NONDETERMINISTIC PARAMETERS 997

...
which are consistent with \( \Pi \) (and the previously added formulas). If the addition of \( \pi \) leads to inconsistence, one adds the negation of \( \pi \). Since empty consequences 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 \( t \sim s \) satisfy the same formulas.

**Lemma 4.6.** \( t \sim s \) iff, for any sequent \( \pi, \Pi \not\models_{NEQ} \pi_t \) iff \( \Pi \not\models_{NEQ} \pi_s \).

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 Moses [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 operation by a superscript, like \( S_t \), 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 \( \Sigma \)-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 \( P^+(S^-) \) of some basis set \( S^- \),
2. for every \( f: S_1 \times \cdots \times S_n \to S \in \Sigma \), \( f^A \) is a \( \subseteq \)-monotonic function \( S_1^A \times \cdots \times S_n^A \to S^A \) 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 ith argument is a deterministic notion but to its semantic counterpart.

**Definition 5.2.** A function \( \beta: X \to \mathcal{P}(S) \) is singular in the \( i \)th argument \( S_i \) if \( x_i \in S_i^A \) and all \( x_k \in S_k^A \) (for \( k \neq i \), \( x_k \in x_k \)).

Thus, the definition of plural parameter passing is modeled by the semantic one.

Note the unorthodox point of view that one only needs a whole power set but allow it to be restricted to the primitive nondeterministic operation (i.e., the join operation (under the set union) on the power sets). The presence of this definition is necessary. Consequently, we do not restrict the signatures and allow instead, give the user means of passing both singular and plural arguments (as a choice).

Let \( \Sigma \) be a signature, \( A \) an \( \mathcal{L}^* \)-structure, \( X^* \) a set of plural variables, and \( x \in X : |\{x| \beta(x)\}| = 1 \). Saying \( (x, x^*) \beta \) satisfies this last condition.)

**Definition 5.3 (Satisfiability).** Let \( \beta \) be a signature, \( s_i \) a set of singular variables, \( x \) a set of plural variables, \( \beta \) a set of \( \beta \)-monotonic functions \( \beta : X \to \mathcal{P}(S) \), and \( \beta \) an interpretation \( \beta[[t]] \) in a power model \( A \).\n
**Lemma 5.2.** The calculus for \( \mathcal{L}^* \) is extended with one additional rule:

\[
(R8) \quad \frac{\Gamma \vdash \Delta}{\Gamma^* \vdash \Delta^*}
\]

Rules (R1)-(R7) remain unchanged. In particular, any \( t_i \) may be a plural term with rule R8.

The new rule (R8) expresses that if one substitutes an arbitrary term \( t \) for a plural variable \( x \), then one can thus exchange it with any other plural variables. The opposite is, in general, not true for singular variables. For instance, \( \{x, y\} \neq \{x, z\} \) is not sufficient for performing this exchange.

**Theorem 5.4.** For any \( \mathcal{L}^* \) interpretation \( \beta \) and \( \Delta \) of \( \mathcal{L}^* \),

\[
PMo(\Delta^*) \iff PMo(\Delta)
\]
The singularity in the ith argument in this definition refers not to the syntactic notion but to its semantic counterpart.

**Definition 5.2.** A function \( f^A : S_1^A \times \cdots \times S_n^A \to S^A \) in a power structure \( A \) is singular in the ith argument \( \iff \) it is \( U \)-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 \( k \neq i \)), \( f^A(x_1, \ldots, x_i, \ldots, x_n) = \bigcup \{ f^A(x_1, \ldots, x_i, \ldots, x_n) \mid x_i = x_i \}. \)

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 operation. 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 \( X \cup X^* \to |A| \) 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 \( f^A(\beta(x), \beta(x^*)) \).

**Definition 5.3 (Satisfiability).** Let \( A \) be a \( \Sigma \)-power structure and \( \pi : t_i \mapsto s_i \mapsto p_j = r_j, m_k \leq n_k \) 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

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

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 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:

\[
\frac{\Gamma \vdash A}{\Gamma^* \vdash \Delta^*}
\]

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

**Theorem 5.4.** For any \( \mathcal{L}' \)-specification \( SP \) and \( \mathcal{L}'(SP) \)-sequent \( \pi \):

\[ PMod(SP) \models \pi \iff \Pi \vdash_{\text{NEQ}^*} \pi. \]
We now ask whether, or under what conditions, singular and plural semantics are interchangeable as the models of a particular logic. The one-way transition is trivial. If we have a model \( M \) of the singular logic, we can view every state \( m \) as a unary relation, and then a multistructure \( M^* \) is obtained by taking the power set of \( M \). Conversely, if we have a multistructure \( M^* \), we may consider every multistructure \( M \) such that the diagram in Figure 6 is commutative.

Since operations in \( M^* \) are definable on the power set of operations \( M \), it is not necessary to simulate all \( M \) multistructures by \( M^* \). Let \( M^* \) be the multistructure obtained from \( M \) by taking the power set of \( M \). 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 \( A \in \text{PMod}(SP) \) iff \( A^* \in \text{PMod}(SP) \).

By taking the power set of an operation \( \Sigma \) we obtain an operation \( \Sigma^* \) such that for every \( P \in \Sigma \) and every \( x \in X \), we have \( \Sigma^* P(x) = \{ P(x) \} \).

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

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

**Proof.** Let \( \Sigma^* \) be obtained from \( \Sigma \) by adding all \( \Sigma^* \) symbols to \( \Sigma \). 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 \( x^1, \ldots, x^n \), let \( \pi = (\pi^* t_1 \ldots t_n) \in W_{\Sigma, X} \). 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 \( \pi' \) be the last sequent used in the \( \text{NEQ}^* \)-proof which contains plural variables \( x^* \) and the sequent \( \pi' \) be 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. □
The extension of the proof of Theorem 6.3.

While the multiset axiomas are certainly not one and only the same calculus can be used to prove that PMod, being a rich class of power structures, implies that some additional axioms are obtained. We therefore need the increased generality of the definition of the relation between the \( L \) and \( L^* \) of difference (section 6.2).

The following example illustrates this. Let \( \pi^* \) be a set of multiset axioms.

**Example 6.4.** Let \( M^* \in \text{MMMod}(SP) \) have infinite carrier, \( \pi^* \in \Pi^* \) be

\[
\beta(x^*) = \{m_1, \ldots, m_l\}
\]

be an assignment such that \( \beta(x^*) \) is a set which is not denoted by any term in \( \mathcal{W}_E, x \). Let \( \beta_i \) be an assignment equal to \( \beta \) except that \( \beta_i(x^*) = \{m_i\} \), i.e., \( \beta = \cup_i \beta_i \). Then \( M^* \models \beta[\pi^*] \) if

\[
M^* = \beta_i[\beta_i(x^*)] \iff \beta_i[\beta_i(x^*)] = \beta_i[\beta_i(x^*)] \iff \beta_i[\beta_i(x^*)] = \beta_i[\beta_i(x^*)]
\]

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

\[
M^* = \beta_i[\beta_i(x^*)] \iff \beta_i[\beta_i(x^*)] = \beta_i[\beta_i(x^*)] \iff \beta_i[\beta_i(x^*)] = \beta_i[\beta_i(x^*)].
\]

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 \( M^* \in \text{MMMod}(SP) \).

Thus, the intuition that the multirelations 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 \( *: \text{MMMod}(SP) \rightarrow \text{PMod}(SP) \) from Lemma 6.2.

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

1. \( \| M \| \subseteq \mathcal{P}(M) \) is such that
   - (a) for every \( n \in \| M \| : \{n\} \subseteq \| M \| \),
   - (b) for every \( m \in \| M \| \) there exists a \( t \in \mathcal{W}_E, n \in \| M \| \) such that:
     \[
     t^M(n) = m.
     \]
2. The operations in \( \| M \| \) can be then defined by:

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

Since \( M \in \text{MMMod}(SP) \), 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

\[
\| \text{PMod}(SP^*) \| = \Pi, \text{ i.e., } \| \text{PMod}(SP^*) \| \subseteq \text{MMMod}(SP),
\]

\[
\| \text{MMMod}(SP) \| = \Pi^*, \text{ i.e., } \| \text{MMMod}(SP) \| \subseteq \text{PMod}(SP^*)
\]
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 6.7. The specification

\[
S: \{ \text{S} \},
\]
\[
F: \{ \text{\textup{\texttt{\texttt{\_\_\_}}} : \text{S}^* \rightarrow \text{S} } \},
\]
\[
\text{II}: \{ \text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_} : \text{S}^* \rightarrow \text{S} } \}
\]

defines $\text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_}}$ as the choice operator: for any argument $t$, $\text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_}} t$ 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 $\text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_}}$ observationally equivalent to $t$, and Lemma 4.6 allows us to remove any occurrences of $\text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_}}$ 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 $f: \text{S} \rightarrow \text{S}$, 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 intention of choosing an element for nondeterministic choice is the formalization of the power set construction, $\mathcal{P}^+(\text{Set}(S))$ to $\mathcal{P}^+(S)$. Assuming to the power set construction, $\mathcal{P}^+(\text{Set}(S))$, with a power set of a power set $\mathcal{P}^+(S)$, we cannot let the same variables $x$, $y$, $z$, $p$, $\text{\texttt{\texttt{\_\_\_}}} x$, $y$, $z$, $p$ be used.

Example 6.8. The following functions (wrt. $\prec$):

\[
S: \{ \text{S} \},
\]
\[
F: \{ \text{\textup{\texttt{\texttt{\_\_\_}}} : \text{S} \times \text{S} \rightarrow \text{S} } \},
\]
\[
\text{II}: \{ \text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_} : \text{S} \times \text{S} \rightarrow \text{S} } \}
\]

Axiom (2) although using singular variables with respect to all terms. (Note that $s \prec p$ would have a different, and in general, variable whenever $s \prec t < p$ and $s \prec t > p$ (see Figure 2). Violating our assumption shows the validity of the form $s \prec p \leftrightarrow t \cup s \prec p$.

Thus, in any model of the above specification, then natural to consider $\cup$ as the (deterministic) or non-deterministic operations. On the other hand, $\cup$ same as set union, we have to respect the singular nature of the model. For instance, the following:

\[
S^\Lambda = \{ \{1\}, \{2\} \}
\]
\[
\text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_}}^\Lambda \text{ defined as } \text{\textup{\texttt{\texttt{\texttt{\_\_\_}}} } \text{\_\_\_}}\text{\_\_\_}
\]

will be a model of the specification.

7. Conclusion. We have considered singular (deterministic choice) and plural (run-time-choice) forms of the central results reported in this
class of power models of a plural levels of some singular specification is essentially the same because able sets which can be referred to variables provide access to arbi-
J utilize this fact to give a more

variables provide access to arbitrary variables to utilize this fact to give a more

are several difficulties. Obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obviously, obvious

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) -- 4 S. Semantically, this would then be a function from p+(Set(S)) to 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.

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 | interpreted as set inclusion will correspond to set union (cf. [17]).

Example 6.8. The following specification makes binary choice the join operation wrt. < :

S: \{ S \},
F: \{ \sqcup : S \times S \to S \},
II: \{ (1) \leftarrow x^* \prec x^* \sqcup y^* \leftarrow y^* \prec x^* \sqcup y^* 
(2) x \prec z^*, y \prec z* \leftarrow 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^* \prec z^*, y^* \prec z* \leftarrow x^* \sqcup y^* \prec z^* would have a different, and in this context unintended, meaning.) We can show that whenever \leftarrow t < p and \leftarrow s < p hold (for arbitrary terms) then so does \leftarrow t \sqcup s < p (see Figure 2). 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 < p, s < p \leftarrow t \sqcup s < p.

Thus, in any model of the specification from Example 6.8 \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 must have 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

FIG. 2.

\[ R8 \frac{x \prec z^*, y \prec z^* \leftarrow x \sqcup y \prec z^*}{\to \larrow t < p \to p \sqcup p < p} R3 \]
\[ R7 \frac{y \prec p \to p \sqcup y < p}{\to t \prec p \to t \sqcup p < p} R3 \]
\[ R7 \frac{y \prec p \to p \sqcup y < p}{\to p \sqcup p < p} R3 \]
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


