19 (1-2) 1993

LAMBDA CALCULUS AND TYPE THEORY. Preface.

Jerzy Tiuryn *I-II*

CONSERVATIVITY OF EQUATIONAL THEORIES IN TYPED LAMBDA CALCULI

Val Breazu-Tannen, Albert R. Meyer *1-49*

Abstract. In programming languages that feature unrestricted recursion, the equational theory corresponding to evaluation of data type expression must be distinguished from the classical theory of the data as given by, say, algebraic specifications. Aiming to preserve classical reasoning about the underlying data types, that is, for the equational theory of the programming language to be a *conservative extension* of the theory given by the data type specification, we investigate, alternative computational settings given by typed lambda calculi, specifically here by the Girard-Reynolds *polymorphic lambda calculus* ($\lambda^{\forall}$). We prove that the addition of just the $\lambda^{\forall}$-constructions to arbitrary specifications, as given by *algebraic* theories, and even *simply typed* lambda theories, is *conservative*. This suggests that polymorphic constructs and reasoning can be superimposed on familiar data-type definition features of programming languages without changing the behavior of these features.

Using purely syntactic methods, we give *transformational* proofs of these results for certain systems of equational reasoning. A new technique for analyzing polymorphic *equational proofs* is developed to this purpose.

Finally, we prove, with a semantics argument, that it is possible to combine arbitrary algebraic data type specifications and the $\lambda^{\forall}$-constructions into functional programming languages that both conserve algebraic reasoning about data and ensure, over arbitrary algebraically specified *observables*, a computing power equivalent to that of the pure $\lambda^{\forall}$. The corresponding problem for simply typed specifications remains open.

SUBSTITUTION UP TO ISOMORPHISM

Pierre-Louis Curien *51-85*

TYPE INFERENCE: SOME RESULTS, SOME PROBLEMS

Paola Giannini, Furio Honsell, Simona Ronchi Della Rocca *87-125*

Abstract. In these paper we investigate the type inference problem for a large class of type assignment systems for the λ-calculus. This is the problem of determining if a term has a type in a given system. We discuss, in particular, a collection of type assignment systems which correspond to the typed systems of Barendregt's "cube". Type dependencies being shown redundant, we focus on the strongest of all, Fω, the type assignment version of the system F^{ω} of Girard. In order to manipulate uniformly type inferences we give a syntax directed presentation of Fω and introduce the notions of scheme and of principal type scheme. Making essential use of them, we succeed in reducing the type inference problem for Fω to a restriction of the higher order semi-unification problem and in showing that the conditional type inference problem for Fω is undecidable. Throughout the paper we call attention to open problems and formulate some conjectures.

TYPE INFERENCE WITH EXTENDED PATTERN MATCHING AND SUBTYPES

Lalita A. Jategaonkar, John C. Mitchell *127-165*

Abstract. We study a type system, in the spirit of ML and related languages, with two novel features: a general form of record pattern matching and a provision for user-declared subtypes. Extended pattern matching allows a function on records to be applied to any record that contains a minimum set of fields and permits the additional fields of the record to be manipulated within the body of the function. Together, these two enhancements may be used to support a restricted object-oriented programming style. We define the type system using inference rules, and develop a type inference algorithm. We prove that the algorithm is sound with respect to the typing rules and that it infers a most general typing for every typable expression.

LAMBDA CALCULUS CHARACTERIZATIONS OF POLY-TIME

Daniel Leivant, Jean-Yves Marion *167-184*

Abstract. We consider typed λ-calculi with pairing over the algebra W of words over $\{0,1\}$, with a destructor and discriminator function. We show that the poly-time functions are precisely the functions (1) λ-representable using simple types, with abstract input (represented by Church-like terms) and concrete output (represented by algebra terms); (2) λ-representable using simple types, with abstract input and output, but with the input and output representations differing slightly; (3) λ-representable using polymorphic typing with type quantification ranging over multiplicative types only; (4) λ-representable using simple and list types (akin to ML style) with abstract input and output; and (5) λ-representable over the algebra of flat lists (in place of W), using simple types, with abstract input and output.

ON THE UNDECIDABILITY OF PARTIAL POLYMORPHIC TYPE RECONSTRUCTION

Frank Pfenning *185-199*

Abstract. We prove that partial type reconstruction for the pure polymorphic λ-calculus is undecidable by a reduction from the second-order unification problem, extending a previous result by H.-J. Boehm. We show further that partial type reconstruction remains undecidable even in a very small predicative fragment of the polymorphic λ-calculus, which implies undecidability of partial type reconstruction for λ^{ML} as introduced by Harper, Mitchell, and Moggi.

PRIMITIVE RECURSION WITH EXISTENTIAL TYPES

Paweł Urzyczyn *201-222*

Abstract. We consider computability over abstract structures with help of primitive recursive definitions (an appropriate modification of Gödel's system T). Unlike the standard approach, we do not assume any fixed representation of integers, but instead we allow primitive recursion to be polymorphic, so that iteration is performed with help of counters viewed as objects of an abstract type **Int** of arbitrary (hidden) implementation. This approach involves the use of existential quantification in types, following the ideas of Mitchell and Plotkin. We show that the halting problem over finite interpretations is primitive recursive for each program involving primitive recursive definitions. Conversely, each primitive recursive set of interpretations is defined by the termination property of some program.

19 (3-4) 1993

ON A FRAMEWORK FOR LOGIC PROGRAMMING WITH INCOMPLETE INFORMATION

Hiroshi Sakai *223-234*

Abstract. A framework for *Logic Programming with Incomplete Information* is proposed. Recently, an important problem in the fields of knowledge bases, logic programming and AI is the problem of how effectively we use information which may have incompleteness, especially disjunctive information.

In order to express disjunctive information, we use the predicate symbol or^m, where the superscript m implies its arity, and we include any disjunctive information in or^m as arguments, which we call an *or-type atom*. Based on or-type atoms, we define a new language which we call an *or-type language* and give an interpretation for the new language. *Logic Program with Incomplete Information (LPII)* is defined by a subset of the or-type language.

First, in a special case our framework is reduced to that near-Horn prolog by Loveland and disjunctive logic programming by Minker.

Then, we consider a general case, which is seen as an advancement from Lipski's incomplete information system. In this case, we show the model intersection property, fixpoint theorem, the soundness and the completeness of a resulution which we call *Box*-resolution.

DEPENDENCIES IN RELATIONAL DATABASES ALGEBRAIC AND LOGICAL APPROACH

Cecylia M. Rauszer *235-274*

Abstract. Functional and multivalued dependencies are expressed in terms of special equivalence relations, called indiscernibility relations. Subsequently, an algebra of indiscernibility relations of a given relational database is introduced and examined. It is shown that a dependency holds in a database if and only if a. corresponding algebraic formula is equal to the greatest element in the algebra. In addition, a formal deductive system, called D-logic is defined. This system corresponds to a fragment of nonclasical logic. An equivalence theorem showing that D-logic describes, in an adequate way, properties of functional and multivalued dependencies is stated. D-logic and D-lattices are instrumental in proving several properties of dependencies. Examples showing how these techniques are valuable in proving theorems about dependencies are given.

RETRIEVAL SYSTEM AND DYNAMIC ALGORITHM LOOKING FOR AXIOMS OF NOTIONS DEFINED BY PROGRAMS

Andrzej Biela *275-301*

Abstract. In this paper we shall introduce a formal system of algorithmic logic which enables us to formulate some problems connected with a retrieval system which provides a comprehensive tool in automated theorem proving of theorems consisting of programs, procedures and functions. The procedures and functions may occur in considered theorems while the program of the above mentioned system is being executed. We can get an answer whether some relations defined by programs hold and we can prove functional equations in a dynamic way by looking for a special set of axioms /assumptions/ during the execution of system. We formulate RS-algorithm which enables us to construct the set of axioms for proving some properties of functions and relations defined by programs. By RS-algorithm we get the dynamic process of proving functional equations and we can answer the question whether some relations defined by programs hold. It enables us to solve some problems concerning the correctness of programs. This system can be used for giving an expert appraisement. We shall provide the major structures and a sketch of an implementation of the above formal system.

A FULLY PRECISE NULL EXTENDED NESTED RELATIONAL ALGEBRA

Mark Levene, George Loizou *303-342*

Abstract. The nested relational model extends the flat relational model by relaxing the first normal form assumption in order to allow the modelling of complex objects. Recently many extended algebras have been suggested for the nested relational model, but only few have incorporated null values into the attribute domains. Furthermore, some of the previously defined extended algebras are defined only over a subclass of nested relations, and all of them are difficult to use, since the user must know the detailed structure of the nested relations being queried.

Herein, we define an extended algebra for nested relations, which may contain null values, called the *null extended algebra*. The null extended algebra is defined over the general class of nested relations withnull values and, in addition, allows queries to be formulated without the user having to know the detailed structure of the nested relations being queried. In this sense, our *null extended join* operator of the null extended algebra is unique in the literature, since it joins two nested relations by taking into account all their common attributes at all levels of their structure, whilst operating directly on the two nested relations. All the operators of the null extended algebra are proved to be *faithful* and *precise*. The null extended algebra is a *complete extended algebra* in the context of nested relations, and, in addition, it includes the *null extended powerset* operator, which provides recursion and iteration facilities.

Keywords: nested relations, null values, null extended algebra, faithful extended algebra operators, precise extended algebra operators

ON DEPENDENCE IN WILLE'S CONTEXTS

Jiri Novotný, Miroslav Novotný *343-353*

Abstract. A concept in Wille's context can be generated starting with a set of features. The problem of finding a minimal subset of the given set of features that generates the same concept as the given one is solved using a suitable dependence space and constructing reducts on one of its subsets. This result can be applied to information systems where it enables to cancel superfluous value of attributes.

Keywords: dependence, reduct, dependence space, context, information system, descriptor, decision table

DELETION SETS

Lila Kari, Alexandru Mateescu, Arto Salomaa, Gheorghe Paun *355-370*

Abstract. We discuss questions related to the cardinality, the effective construction, and decidability of the co-called deletion sets: the sets of strings obtained by erasing from a word the subwords which appear as elements of a given language.

UNDECIDABLE FRAGMENTS OF TERM ALGEBRAS WITH SUBTERM RELATION

Gabriele Marongiu, Sauro Tulipani *371-382*

Abstract. In these paper we give a new proof of undecidability results about term algebras with subterm relation. This proof yields a novel result which states that, in presence of the subterm relation and in signatures with symbols of arity greater than one, the $\sum_1$ theory of rational trees in properly included in the $\sum_1$ theory of infinite trees. Moreover, these theories are quite different; in fact, the first is r.e. and the second has degree not less than $\sum_1^1$. Here, in analogy with the arithmetical hierarchy, we call Δ_0 the prenex formulas whose quantifiers are bounded by the predicate $\leq$, and we call $\sum_1$ the formulas which are existential quantification of Δ_0 formulas.

THE SYNTAX OF PARALLELISM

Amihood Amir, Carl H. Smith *383-402*

Abstract. One of the problems associated with the introduction of parallel processors is the so called ”dusty deck” problem. A solution entails the development of optimizing compilers that transform programs previously written for a conventional serial processor into functionally equivalent programs that exploit the parallel processing capabilities of the new multiprocessor machines.

We introduce a function *Composition Model* that models parallel architectures as a hierarchy of syntactic function definitions. Position in the hierarchy is equivalent to parallel time complexity in the modelled architecture. Other parallel concepts such as global vs. local communications, concurrency or exclusivity of read and write, and the number of processors used in a computation, are modelled as well. We rigorously prove that a compiler that optimizes a program for parallelism on a CREW PRAM is not effectively computable, even if it is also given an optimal serial program for the same task and a time bounding function.

It turns out that the function composition model is similar to some traditional models, such as the Grzegorczyk Hierarchy. Our parallel interpretation of the Grzegorczyk Hierarchy offers new insights and admits a new cleaner and more elegant definition of the hierarchy with a single base class, as opposed to Grzegorczyk's three.

TIME AND DURATION IN NONINTERLEAVING CONCURRENCY

David Murphy *403-416*

Abstract. The purpose of this paper is to present a realtimed concurrency theory in the noninterleaving tradition. The theory is based on the occurrences of actions; each occurrence or event has a start and a finish. Causality is modelled by assigning a strict partial order to these starts and finishes, while timing is modelled by giving them reals.

The theory is presented in some detail. All of the traditional notions found in concurrency theories (such as conflict, confusion, liveness, and so on) are found to be expressible. Four notions of causality arise naturally from the model, leading to notions of *securing*. Three of the notions give rise to underlying event structures, demonstrating that our model generalises Winskel's.

Infinite structures are then analysed: a poset of finite structures is defined and suitably completed to give one containing infinite structures. These infinite structures are characterised as just those arising as limits of finite ones. Our technique here, which relies on the structure of time, is of independent interest.

Keywords: Concurrency, duration, event structure, noninterleaving models, real time.

THE SET OF PROBABILISTIC ALGORITHMIC FORMULAS VALID IN A FINITE STRUCTURE IS DECIDABLE WITH RESPECT ON ITS DIAGRAM

Wiktor Dańko *417-431*

Abstract. In the paper it is considered a probabilistic logic of programs PrAL, similar to the Feldman-Harel logic, constructed for expressing properties of iterative programs with operations x :=?, *either ... or ...* interpreted in a probabilistic way. A language of PrAL contains a sort of variables interpreted as real numbers, used to describe probabilities of behaviours of programs. The main result states that the set of formulas of PrAL valid in a finite structure is decidable with respect to the diagram of the structure.