RECENT TECHNICAL REPORTS

INSTITUT DE RECHERCHE D'INFORMATIQUE ET D'AUTOMATIQUE

Available from: IRIA
Service des publications
B.P. 5
78150 Le Cheshav
Rocquencourt, France

(Papers are written in the language of their title and abstract.)

RL#1 SUR L'INTERPRETATION DES SCHEMAS DE PROGRAMMES MONADIQUES
M. Nivat (December 1972)

Le présent article constitue une tentative de definition algébrique de la fonction calculée par un schéma de programme recursif ou non, pour une interprétation donnée. Une construction analogue à la nôtre fait l'objet ce l'article de DANA SCOTT "Lattice of flow diagrams" (5) qui a été la principale source d'inspiration pour notre travail. Mais si la construction que l'on trouvera dans ces pages et celle de DANA SCOTT conduisent à des définitions identiques, elles sont pourtant d'inspirations différentes dans la mesure où les structures algébriques mises en jeu ne sont pas les mêmes. Ce que nous avons surtout voulu faire, c'est expliciter l'analogie formelle entre schémas de programmes et systèmes d'équations algébriques, et de ce fait l'accent se trouve déplacé de la structure d'ordre qui sous-tend la construction de DANA SCOTT, vers la structure de magma dont se trouve muni l'esemble des schémas de programme.

RL# 4 OPTIMAL DISK MERGE PATTERNS
M. Schlumberger - J. Vuillemin
January 1973

Our purpose is to study the optimal way to merge n initially sorted runs, stored on a disk like device, into a unique sorted file. This problem is equivalent to finding a tree with n leaves which minimizes a certain cost function. (See Knuth (1)).

We shall study some properties of those optimal trees, in the hope of finding efficient ways for constructing them.

In particular, if all the initial runs have the same length, an algorithm for constructing the best merge pattern is described; its running time is proportional to n^2 and it requires space proportional to n.

A special case is also analyzed in which the problem is solved in time and space proportional to n, which provides some insight into the asymptotic behavior optimal merge trees.

RL#6 A PRELIMINARY THEORY FOR PARALLEL PROGRAMS
GILLES KAHN
January 1973

In his thesis, D. ADAMS introduced a model of parallel computation which is convenient to represent a wide class of parallel algorithms. Later, D. SEROR pointed out the interest of ADAMS' model to represent the tasks of an operating system.

We show here how ideas due to D. SCOTT allow a systematic treatment of the properties of this model of parallel computation.

RL#7 APPLICATION OF PENALTY TECHNIQUE TO SOLVE A SCHEDULING PROBLEM AND COMPARISON WITH COMBINATORIAL METHODS
PIERRE NEPOMIASTCHY
January 1973

The problem studied here is the "general n/m job shop problem" in the terminology of CONWAY. It is well known that this problem cannot be solved exactly using combinatorial methods for a large number of tasks, even using a very large computer. Here, by application of the penalty technique, we transform the scheduling problem into the minimization without constraints of a function of A + 1 real variables, where A is the number of tasks to schedule. This function is not convex, but we give an algorithm which is able to find a near-optimal solution in a small computational time, and we prove the convergence of the approximate method. Then, we compare our numerical results with those we have obtained using the branch-and-bound algorithm (see, for example, SAID ASHOUR) applied to the same example. Our algorithm is very economical in storage utilization and its computational time grows approximately as A^3 (and not as A!, as in combinatorial methods).

RL#8 A MECHANIZATION OF TYPE THEORY
GERARD P. HUET
February 1973

A refutational system of logic for a language of order ω is presented. This language is a slight modification of Church's λ- calculus with types. The system is complete, in the sense that a refutation of a set of sentences exists if and only if this set does not possess a general Henkin model. The main rule of inference is a generalization of Robinson's resolution to type theory, which allows us to get rid of the substitution rule.

RL#10 ON COMPUTING THE TIME COMPLEXITY OF TRANSITIVE CLOSURE ALGORITHMS
JOEL COFFY
February 1973

Two different algorithms to compute the transitive closure of a directed graph have been presented by WARSHALL & PURDOM, in the chronological order. PURDOM establishes also that for graphs selected at random with equal probability among those which satisfy a given hypothesis P, the time complexity t(n) of WARSHALL's algorithm satisfies:

$$\lim_{n \to \infty} \frac{n^2}{t(n)} = 0 \qquad (1)$$

where n is the number of vertices of the graph. The purpose of this note is to prove that the hypothesis P above is satisfied by all graphs, thus establishing the generality of (1).

RL#23 A UNIFICATION ALGORITHM FOR TYPE THEORY
G. P. HUET
July 1973

A semi-decision algorithm is presented, and its correctness proved, to search for unification of formulas in ω order type theory.

It is shown that the search space is a lot simpler than the one for research of most general unifiers. This allows our algorithm to have good directionality and convergence properties.

RL#24 CORRECT AND OPTIMAL IMPLEMENTATIONS OF RECURSION IN A SIMPLE PROGRAMMING LANGUAGE
J. VUILLEMIN
July 1973

The object of this paper is to study the mechanism of recursion in a simple, LISP-like programming language, where the only mean of iteration is through recursion. The theory of computation developed in SCOTT (6) provides the framework of our study. We show how the implementations of recursion which deserve to be called «correct» can be characterized semantically, and demonstrate a general criterion for the correctness of an implementation. We then describe an implementation of recursion which is both correct and optimal in a general class of sequential languages, and therefore constitutes an attractive to both «call-by-name» and «call-by-value»

RL#33 GRAPH PARITIONING AND CONSTRUCTING OPTIMAL DECISION TREES ARE POLYNOMIAL COMPLETE PROBLEMS
L. HYAFIL and R. L. RIVEST
October 1973

The problem of partitioning the nodes of a graph into subsets of bounded size so as to minimize the number of edges connecting nodes from distinct subsets is shown to be polynomial complete. The problem of constructing a decision tree which minimizes the expected number of tests required to identify an object is also shown to be polynomial complete. (Every polynomial complete problem (for example, traveling salesman, graph coloring) can be solved in polynomial time if one of them can.)

RL#37 ALGORITHMS D'EQUIVALENCE ET DE REDUCTION A DES EXPRESSIONS MINIMALES DANS UNE CLASSE D'EQUATIONS RECURSIVES SIMPLES
B. COURCELLE, G. KAHN and J. VUILLEMIN
November 1973

Nous décrivons ici un algorithme pour décider de l'équivalence dans un domaine dont les éléments sont définis par des schémas d'équations au point fixe. Cet algorithme est utilisé pour trouver diverses représentations minimales des éléments du domaine.

RL#39 DESIGN OF OPTIMAL MERGE ON DIRECT ACCESS DEVICES
L. HYAFIL, F. PRUSKER and J. VUILLEMIN
November 1973

Our goal in this paper is to provide a model and its mathematical analysis for optimal merging on direct access devices such as disks, drums, magnetic-cards,... thus allowing one to make rational design decisions. We present various algorithms and heuristics for efficiently computing optimal merge patterns, when the runs generated in the initial sorting phase have reasonably equal lengths.

The analysis which we provide for optimal merges gives ways to choose the best merge-pattern in a given configuration, evaluate the effect of having longer initial runs, and decide between a faster disk or a larger core memory.

RL#40 GENERALISED IMMUNE SETS
P. FLAJOLET and J. M. STEYAERT
November 1973

A generalisation of the concept of immunity in recursive function theory is investigated: β being a class of sets, a set is β immune if it has no infinite subset in β .

Such sets are constructed when we take for β the usual complexity classes: classes defined by nesting loops, complexity classes in any abstract measure and Turing machine tape or time bounded classes. Considering the elements of these classes as "easy", we thus construct sets which have only "hard" non trivial subsets.

Unsolvability results are derived which can also be interpreted as properties about program compactness in subrecursive programming languages of different powers.