

Research Report

Keith L. Clark

KL1について

本メモは、10月の初めに ICOT KL1 Design Task Group と Dr. Clark, Dr. Gregory (Imperial College), および Dr. Shapiro (Weizmann Institute of Science) との間で行なわれた KL1 に関する議論をまとめたものである。

KL1 は並列実行のできる論理型言語である。guarded clause と commit operator で don't care nondeterminism をサポートし、集合表現により don't know nondeterminism をサポートする。また、プロセス自体を対象として扱うための機能としてメタ述語を備えている。さらに、モジュール構造化機能や文字列処理機能も含んでいる。以下に各項目ごとの議論内容についてまとめる。

1. 並列性

1. 逐次型AND のoperational な意味について

逐次型AND の commit operator による実現について、Dr. Clarkより通信の timing の semantics を変えるというコメントを得た。逐次型AND は記述の自然さおよび余分な並列性の制限という意味で非常に有効であるが、Dr. Shapiroは逐次性自体は read-only annotation で表現できるという立場から導入について反対意見を述べた。

2. 集合

1. 全解と lambda abstraction

KL1 では集合表現を、あるゴールを満足するすべての解の集まりを表すためと lambda abstraction のために用いる。また、集合表現として外延、内包双方に基づくものを許したが、これらの点について出席者の賛同を得た。

2. term と集合との unification の意味について

KL1 における、集合と term の unification を membership テストと見る観点について、logic を壊すという立場から反対意見を得た。この機能自体は Prolog/KR 等の言語にもあり強力な表現力をもつものであるが、導入にあたっては慎重な論理的考察が必要と思われる。

3. 基本操作として何を取るか

KL1 は集合を操作するための機能として、Member, Intersect, Enumerate等の比較的高いレベルの述語のみ用意したのであるが、これに対しDr. Clarkより 集合を単なるdata構造として見て操作するための基本操作が必要であるとの意見を得た。

3. メタ推論

1. simulateという概念

KL1 における simulate という概念はいろんな概念が集約されたものであり、primitive 述語としては high level すぎる。よりatomicな概念に分解すべきだとの意見を得た。simulateには次のような特徴が集約されている。

- ① context switching
- ② illogical communication
- ③ returning result capsule
- ④ more primitive meta operations

2. guard の中から通信をすることについて

上記の4項目の中で特に②の illogical communication について議論した。中心となった問題はguard part の中からのメッセージの発信である。通常guard partの中からのメッセージの発信はcommit operator の処理までは実質的に延期されるのであるが、これではOS等の実際的なプログラムを書く際に大変不都合であるので、KL1 ではこの問題を解決するために、illogical なI/O channel 変数を備えたsimulate述語を提案した。この同じ問題に対して、Dr. Clarkと Dr. Gregory はsimulateによく似た evaluate述語というものを提案した。また、Dr. Shapiroはシンプルな概念early commitmentを提案した。これらのいくつかの提案について今後の検討が必要である。

4. 文字列

1. パターン表現の問題

① logic との関係

提案したパターン表現(文脈依存パターン)は、logicalなものではないのではないかと、という疑問が出された(Dr. Clark, Dr. Gregory)。これに対しては、文脈依存パターンを用いたプログラムが、通常の論理プログラムへの明確な変換規則を持っていることをDr

Gregory に説明し、了解を得た。この性質は、Dr. Shapiro が、ユーザ言語を満たすべき criteria として示したものである。

ただ、パターン表現は、ユーザ言語に対する工夫であるから、KL 1 を機械語として位置づけるならば、それには導入しないことになる。

② 集合表現との関係

文字列パターンは、文脈依存パターンの考えかたによらなくても、集合機能によって表現できるのではないかと、このコメントがあった (Dr. Gregory)。たしかに、集合表現を用いて、パターンの論理的性質を記述することはできる。しかし、そこに計算の概念を組み込むのは容易でないように思える。つまり、文脈依存パターンで書いたときのような、効率的なマッチングアルゴリズムの表現が可能かどうか疑問である。

2. representation

プロセス間共有変数 (stream) としての長い文字列と、普通の term としての文字列の両方を考慮する必要があるということに合意した。

5. モジュール

1. メタ推論の world との関係

なぜ、モジュールと世界というふたつの概念が必要なのか、という質問が来た。これに対し、モジュールは世界の最小構成要素であること (それ自体世界でもある)、および最小構成要素をそれ以外の世界と区別するときには新たな用語が必要であることを説明した。

2. モジュールの名前と本体をどう対応づけるか

モジュールの名前と本体との関係づけを誰が管理するのか、という質問が来た。これに対し、

- ・ モジュールの管理のため、およびモジュールが他のモジュールに言及するためには、モジュールの名前を利用することが好都合であること
- ・ 現段階では、モジュールの名前は各モジュールの構成要素として考えていることを述べた。

ただ、モジュールに名前をつける機能を、機械語レベルでいかにサポートするかについては、さらに検討を要する。

3. 普通の programming で階層構造はあるか

Micro-PROLOG の経験では、モジュールの階層構造は不要で、flat な構造で充分であった、というコメントが Dr. Clark からあった。

たしかに、対話的プログラミングにおけるソフトウェア部品としてのモジュールならば、階層構造はいらぬかもしれない。だが、KL 1 のモジュールは世界と深い関係にあり、その世界は、メタ推論において階層化して使うこともありうる。したがって、いままでのモジュールの使われ方だけから機能を決定するわけにはいかない。

CURRICULUM VITAE

Keith Leonard Clark

Date of Birth: 29 March 1943

Nationality: British

Higher Education

- 1961-64 Hatfield College, Durham University, England
(Undergrate degree in Mathematics)
- 1964-66 Chuchill College, Cambridge University, England
(Undergraduate degree in Logic and Philosophy)
- 1968-69 Imperial College, London University, England
(Postgraduate Diploma Course in Computer Science)

Degrees

- 1964 B.A.(Durham University)
First Class Honours in Mathematics
- 1966 B.A.(Cambridge University)
Upper Second Class Honours in Moral Sciences Tripos
- 1970 Diploma of Imperial College
Diploma Dissertation in Computer Science
- 1980 PhD (London University)
Thesis Title: Predicate Logic as a Computational Formalism

Employment

- 1966-67 Lecturer in Mathematics
City Polytechnic. London
- 1967-68 School Teacher in Mathematics
Voluntary Service Overseas. Sierra Leone
- 1969-79 Lecturer in Computer Science
Queen Mary College. London University
- 1975-77 Research Fellow in Computing
Imperial College. London University
(on leave-of-absence from Queen Mary College)
- 1979-83 Senior Lecturer in Computing
Imperial College, London University
- 1980- Director of Logic Programming Associates Ltd
- 1983- SERC Senior Research Fellow, Imperial College, London University

Visiting Appointments

- 1977 Visiting Associate Professor
 Information Sciences
 University of California at Santa Cruz
- 1979 Visiting Professor
 Systems and Information Sciences
 Syracuse University, Syracuse, NY

Research Interests

Declarative programming, particularly the use of predicate logic for program specification, non-deterministic programming and parallel programming.

The implementation of logic programming languages on sequential and parallel machines and their interface with data base machines.

Logic, augmented with control annotations, as a language for parallel programming.

Transformation and deductive synthesis of logic programs.

The use of the logic programming language PROLOG for expert systems and deductive data bases.

Research Grants

- 1978-80 Logic Programming
 SERC Grant of £21000 (joint with Robert Kowalski)
 One research assistant
- 1981-84 Logic for Expert Systems
 SERC Grant of £125000
 Two research assistants
- 1982-85 A Relational Language for Parallel Programming
 SERC Grant of £38000
 One research assistant
- 1982-85 Logic Data Bases
 SERC Grant of £102000 (joint with Robert Kowalski)
 Two research assistants

Conference Organisation

- 1980 Co-chairman Logic Programming Workshop
 Debrecen, Hungary
- 1982 Member Program Committee
 Logic Programming Conference
 Marseilles, France