ICOT Technical Report: TR-286, 287

TR-286, 287

TR 286-並列動作系における推論方式について
TR 287-KL1の並列処理－疎結合マルチプロセッサにおける負荷分配方式の評価－

286-高橋和子，竹内彰一，清水広之(三菱電機)
287-米山　貢，杉江　衛，菅谷正弘(日立)
後藤厚宏

July, 1987

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

6L-2

並列動作系における推論方式について

竹内彰一　高橋和子　清水広之
三菱電機株式会社　中央研究所

1．はじめに

一般に複数の構成要素が並列に動作する系では動作が一意的に定まらずダイナミックに様々な可能性が生じる場合が多い。committed choice型の並列論理型言語でこのような系の上における組み合わせ問題や協調型プラン生成などの問題を記述して実行すると、一旦ある節が選択されるとそれ以外の節を選択する可能性はすべて棄却されてしまう。しかし問題の性質を考慮すると、すべての可能な場合における推論を行ない全解を得る方が望ましい。従って、並列動作系上の問題を扱うためには以下の機能を満たす手法が必要であり、近年いくつかの研究がなされている[1][2][3]。

(1) 問題に内在する複数の可能性や非決定性などを簡潔に表現できる。
(2) あらゆる場合におけるシミュレーション・探索等の推論が効率良く行なえる。

本稿では、これらの機能を満たす一手法として、並列動作系上の問題を並列問題解決用言語ＡＮＤＯＲ－Ⅱで記述し、それを『色付きストリーム』に基づいてＦＧＨＣ(Flat Guarded Horn Clauses)へコンパイルして実行する方式について述べる。

2．ＡＮＤＯＲ－Ⅱによる問題の記述

ＡＮＤＯＲ－Ⅱは、ＡＮＤ並列性を有するＦＧＨＣにＯＲ並列性を付加することによって全解探索機能を導入した並列問題解決用言語である。例として図１の半加算器回路の故障診断問題を取り上げて説明する。この回路は入力A, B に対してそれらの和S と桁上げC を計算して出力する。

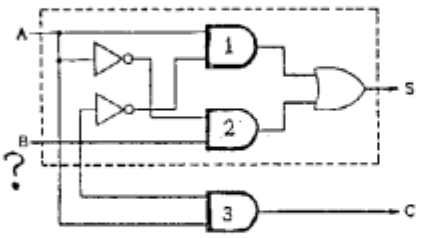

図１　半加算器回路

図２にこの回路のトップレベルの記述を示す。

```
:- mode circuit(+,-).
:- and-relation circuit/2.
circuit(In1,In2,Out) :- true  |
    inverter(In1,Out1),
    inverter(In2,Out2),
    and-gate-d([In1,Out2],Out3,State1),
    and-gate-d([Out1,In2],Out4,State2),
    and-gate-d([In1 ,In2],Out5,State3),
    or-gate([Out3,Out4],Out6),
    Output=[ [Out6,Out5],
     [(and1,State1),(and2,State2),(and3,State3)] ].
```

図２　半加算器回路のＡＮＤＯＲ－Ⅱによる記述

第１節はモード宣言部である。各引数の入出力モードは決定的であり、各ゴールの入力は必ずグラウンドでなければならないという制限を持つ。第２節は関係宣言部であり、第３節は定義部である。各述語はＡＮＤ関係かＯＲ関係のいずれか一方で定義される。ＡＮＤ関係の述語の定義節はＧＨＣに準じ、ガード部とボディ部から成る。ＯＲ関係の述語の定義節はコミット・オペレータを含まず、この節を並べることによって様々な可能性が列挙できる。簡単のため、この回路に含まれる素子のうちＡＮＤゲートのみが故障の可能性があると仮定すると、各ＡＮＤゲートは以下のように記述される（図３）。

```
:- mode and-gate-d(+,-,-).
:- or-relation and-gate-d/3.
and-gate-d (In,Out,State) :-          (1)
    and-gate(correct,In,Out), State=correct.
and-gate-d (In,Out,State) :-          (2)
    and-gate(error,  In,Out), State=error.
```

図３　ＯＲ関係の述語

これに対して決定的な動作をするものはＡＮＤ関係の述語として定義される（図４）。

```
:- mode inverter(+,-).
:- and-relation inverter/2.
inverter(0,Out) :- true  | Out:=1.
Inverter(1,Out) :- true  | Out:=0.
```

図４　ＡＮＤ関係の述語

On A Reasoning Method of Concurrent Systems
Akikazu TAKEUCHI, Kazuko TAKAHASHI and Hiroyuki SHIMIZU
MITSUBISHI Electric Corp.

3. 計算モデル

ＡＮＤＯＲ－ⅡはＡＮＤ／ＯＲ両並列性を有する。即ち論理積の関係にあるすべてのゴールを並列に実行するというＡＮＤ並列性と、ある述語の論理和の関係にある節のうち適用可能なすべての節によるゴールのリゾリューションを並列に実行するというＯＲ並列性を有する。論理積のゴールのリゾリューションはＧＨＣと同様コミット・オペレータにより１つの節に限定され、ある節が選択されると他の節を選択する可能性は棄却される。これに対して、ＯＲ関係のゴールはコミット・オペレータを持たない複数の節で定義されており、ヘッド・ユニフィケーション可能なすべての節が並列に実行され、それぞれ独立した世界に分岐する。即ちリゾリューションが１つの節に限定されることなく、すべての可能性が保持される。このＯＲ関係を用いることで全解探索プログラムを容易に書くことができる。

例えばcircuit の第３ゴールのand-gate-dのリゾリューションを考える。図３で(1),(2) どちらの節もヘッド・ユニフィケーションが可能なのでどちらが選択されるかによって２つの可能世界が生じる（図５）。これらの可能世界には各々独自の色が付加されており、各世界における推論は互いに独立かつ並列に行なわれる。

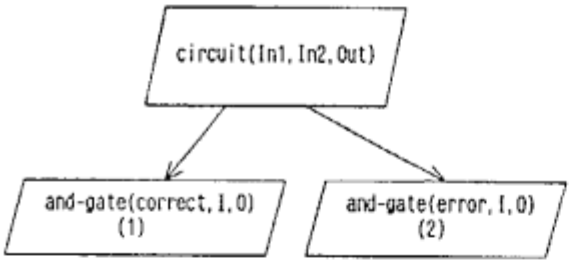

図５　circuit の計算モデルの一部

4. 実行系

ＡＮＤＯＲ－Ⅱで記述されたプログラムは『色付きストリーム』によるコンパイル方式に基づいてＦＧＨＣに変換される。

ＯＲ関係によって複数の世界が生じる場合、出力変数には各世界におけるリゾリューションの結果とそれが定義されている世界に付随する色のペアがストリームとして流れる。このようなストリームを受け取ったゴールは個々のデータを順に取り出して処理をする。また、１つのゴールに複数のストリーム型データが流れる場合は（図２における第６ゴールor-gate など）それらの色の整合性を調べ、同一世界に属するもの同志に対してのみリゾリューションを行なう。以上のような処理をするプロセスをshell と呼ぶ。コンパイラが行なう主要な処理はshell の必要なゴールの検出と色の整合性をチェックする部分の生成である。

5. 実行例

例として２で述べた半加算器回路の故障診断がいかに実行されるかを示す。ここでの故障診断とは論理回路の入出力データの一部を観測値として与え、論理回路内の故障素子及び未知入力値に対する全解を求めるものであり、以下のようにＡＮＤＯＲ－Ⅱで記述される（図６）。

```
:- mode test(+,-).
:- and-relation test/2.
test([DataIn,DataOut],Answer) :- true |
    setInput(DataIn,Input),
    circuit(Input,Output),
    checkOutput(Input,Output,DataOut,Answer).
```

図６　故障診断のＡＮＤＯＲ－Ⅱによる表現

このプログラムにおいて入力生成部(setInput)ではＯＲ関係を使って定義した未知入力に対して場合分けを行ない、回路解析部(circuit) ではこれらの入力に対してあらゆる場合のシミュレーションを実行し、出力チェック部(check Output) ではシミュレーションの結果得られたすべての出力値の中で観測値と一致するものを抽出する。今、B を未知入力とし、A,S,C としてそれぞれ1,0,0 が観測されたとすると、

```
test( [ [1,?], [0,0] ], Answer ).
```

を実行することによって故障診断が行われ、その結果

```
Answer  : B=0, and1 behaves incorrectly
          B=1, and3 behaves incorrectly
```

という２つの解が存在することがわかる。

6. むすび

並列動作系上の問題を並列問題解決用言語ＡＮＤＯＲ－Ⅱで記述し、それを『色付きストリーム』に基づいてＦＧＨＣへコンパイルして実行する方式について述べた。変換後のプログラムはＡＮＤ／ＯＲ両並列性を実現する。この方式では問題に内在する複数の可能性や非決定性などを簡潔に表現でき、あらゆる可能な場合におけるシミュレーション・探索等の推論が並列に効率良く行なえる。

謝辞

本研究は第５世代コンピュータ・プロジェクトの一環として行われた。日頃御指導をいただいているＩＣＯＴ古川研究所次長及び伊藤第１研究室室長に感謝致します。

参考文献

[1]K.Ueda, "Making Exhaustive Search Programs Deterministic," pp.270-282, Proc.of 3rd Int. Conf. on Logic Programming, LNCS225, Springer-Verlag, 1986.

[2]H.Tamaki, "Stream-Based Compilation of Ground I/O Prolog into Committed Choice Languages," pp.376-393, ICLP87, 1987.

[3] 奥村・松本, "レイヤードストリームを用いたプログラミング," pp.223-232, Proc. of 5th Logic Programming Conference, 1987.