B-5

KL1 プログラムの実行状況の視覚化
Visualization of KL1 Program Execution

中島 利方 †
NAKAJIMA Toshimichi

六沢 一昭 ‡
ROKUSAWA Kazuaki

1. はじめに

並行論理型言語 KL1 は, 逐次型言語と異なり, 実行順序という概念を持たない. また, 1 つのゴールが複数のゴールを生成することがある. これは, 手続き型言語で例えると 1 つのサブルーチンが複数のサブルーチンを fork するようなものである. 従って, KL1 プログラムの実行状況をソースコードから想像することは困難である. このことから, その実行状況を把握するには視覚化が有効であると考える.

KL1 プログラムの実行状況を視覚化するシステムを作成した. 本システムは, KLIC [1] トレーサの出力結果を利用し, KL1 プログラムの振る舞いをゴール書き換えモデルに基づいた形で表示する. また, 巻き戻し機能及びスパイ機能を有する.

2. 並行論理型言語 KL1

KL1 は, Guarded Horn Clauses(GHC) に基づいて設計された並行論理型プログラミング言語である. KL1 プログラムは, 以下の形をした節の集合である.

ヘッド :- ガード | ボディ.

ヘッドは節の名前を示す. ガードは節の選択条件を表す. ボディは節が選択されたときに生成されるゴール群である.

KL1 プログラムの実行は, 以下の手順でゴールを書き換えることにより行われる.

1. ゴールプールからゴールを 1 つ取り出す (ゴール呼び出し操作).
2. 取り出したゴールに適合する節を探し,

   適合する節があったら, その節のボディのゴール群をゴールプールに入れる (リダクション操作).

   ゴールの引数が具体化されていないために適合する節が決まらない場合は, そのゴールをサスペンドゴールプールに入れる (サスペンド操作).

   適合する節がないなら, そのゴールは失敗する.

この実行モデルをゴール書き換えモデルと言う (図 1).

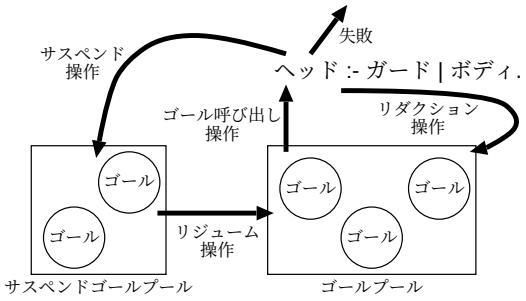

図 1: ゴール書き換えモデル

3. 視覚化システムの検討

プログラムの実行状況を把握するためには視覚化システムに何が必要かを検討した.

3.1 視覚化方法の検討

KL1 プログラムは, ゴール書き換えモデルに基づき実行される. そのため, このモデルのイメージを用いて実行状況を視覚化するのが自然であり, また直観的にわかりやすいと考えられる. しかし, ただモデルの図を静的に表示しただけでは, プログラムの実行という動的な振る舞いを表現するには不十分である.

そこで, アニメーション表示を行う. アニメーションにより, ゴールリダクションの様子やそれに伴うプールの状態の変化といったプログラムの振る舞いを自然に表現できる.

3.2 巻き戻し機能

デバッグのためにプログラムをトレースする際, 気になるところをもう一度トレースしたいと思うことがある. また, ブレークポイント設定の失敗などにより実行が行き過ぎてしまい, 調べたい部分をスキップしてしまうことがある. このような場合に, プログラムの実行を最初からやり直すのは面倒である. バグを見つけるまで何度も再実行することになり, 再実行のたびにブレークポイント設定をやり直さなければならないからである.

このような面倒を避けるため, 任意の時点まで戻して実行状況の視覚化を再開する巻き戻し機能を提供する. この機能により, プログラムの実行をやり直すことなく気になるところを確認することができる.

†千葉工業大学大学院 工学研究科 情報工学専攻
‡千葉工業大学 情報科学部 情報工学科

3.3 スパイ機能

デバッグの際, ある特定の部分に注目してトレースしたいということがよくある. このようなときには, ブレークポイントを設定してその部分までのトレースをスキップするということがよく行われる.

そこで, ゴールを指定してそのゴールが出現するまでトレースをスキップするスパイ機能を提供する. また, スパイ指定したゴールが出現するまで巻き戻すという機能も提供する. この機能により, どこまで巻き戻すかを指定することができる.

4. システムの実現方法

4.1 実行時情報の取得

本システムは, 実行時情報を KLIC のトレーサから取得して実行状況を視覚化する. KLIC のトレーサ出力の例を図 2 に示す. 1 行目は, ゴール番号 6 のゴール main:sum がゴールプールから取り出されたことを表し, 2 行目は, そのゴールがサスペンドしたことを表す.

```
1   6 CALL: main:sum(_12,0,_5)
2   6 SUSP: main:sum(_12,0,_5)
```

図 2: トレーサ出力の例

既存のトレーサを利用することにより, 処理系内部に手を加えることなく視覚化システムを実現することができた.

4.2 アニメーション表示の実現

本システムはゴールリダクションの様子を視覚化する. 具体的には, どのゴールに対して, どのような操作を行い, それによってプールの状態がどう変わったかをアニメーション表示する. これを行うには,「操作対象のゴール」と「どの操作が行われたか」, 及び「その時点でのプールの状態」をトレーサ出力から得る必要がある.

「操作対象のゴール」と「どの操作が行われたか」は, 1 つのゴールリダクションに関するトレーサ出力から得ることができる. 一方,「その時点でのプールの状態」はそれまでのゴールリダクションの集合によって決まる. 従って, 1 つひとつのゴールリダクションの結果を蓄積してプールの状態を求めていく必要がある.

このようにして KLIC トレーサから得た 3 つの情報を, 実行の始まりから順番に配列に格納する. そして, 視覚化の際, その配列から情報を取り出してアニメーション表示を行う.

5. 実行例

本システムの実行例を図 3 に示す.

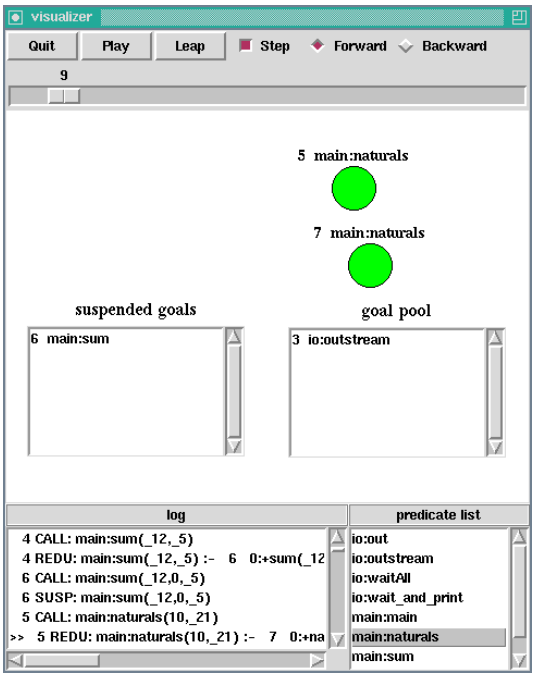

図 3: 視覚化システムの画面例

丸が, 現在リダクション中のゴールを表し, 丸の上に書かれているのがそのゴールの番号と名前である. 画面中央の左のボックス (suspended goals) はサスペンドゴールプールを, 右のボックス (goal pool) はゴールプールを示す. また, 画面下部の左のボックス (log) は KLIC トレーサの出力を, 右のボックス (predicate list) はプログラムに登場するゴールの名前を表示する.

この図は, ゴール番号 5 のゴール main:naturals が 7 番の main:naturals を生成した時点の画面である. ゴールプールには io:outstream がある. また, サスペンドゴールプールには main:sum がある. log ボックス中の >> はリダクション中のゴールを示す. また, predicate list ボックスの main:naturals のグレー表示はスパイ指定を表す.

6. おわりに

KLIC のトレーサ出力を利用して KL1 プログラムの実行状況を視覚化するシステムを作成した. 本システムはゴールリダクションの様子をアニメーション表示する. また, 巻き戻し機能及びスパイ機能を持つ.

本システムを利用することにより, 実行状況を容易に把握できることが期待される.

参考文献

[1] Chikayama,T.: “KLIC: A Portable Parallel Implementation of a Concurrent Logic Programming Language”, Parallel Symbolic Languages and Systems, pp.286-294, 1995.