

Prologプログラミングの学習を支援する構造化トレーサー†

2G-7

山崎 茂 榎本 圭孝 伊丹 誠 伊藤 紘二

東京理科大学 基礎工学部

1 はじめに

Prologプログラミングの初心者において、プログラムを理解するためには、インタプリタの働きを正しく把握しなければならない。しかし、初心者が通常のトレーサーやデバッガを用いても、インタプリタの働きを理解するのは大変に困難であるため、我々は、視覚的な工夫によりインタプリタのグローバルな働きを達成する構造化トレーサーの実現を試みている [1] [2]。今回は [2] の改良によって明らかになった具体的な仕様を提案する。

2 システムの構成

このシステムにおいて、トレース制御の役割を担うのは Prolog 部であり、ユーザーとのインターフェースを行なうものとして GUI 部がある。GUI 部においては XView ツールを使用し、キャンバス上でトレース状況を表示する。ユーザーによるトレース操作を受け付けるパネルボタンは以下のものからなる。

- **CREEP** 現在のゴールについて潜る。この時、前のステップと現在のステップを交互に表示することにより、引数の値の伝搬が明らかになる。
- **LEAP** 現在のゴールについて結果のみ出力する。
- **BACK FOR REDO** 前のゴールをやり直す。
- **OTHER SOLUTION** あるゴールについてトレースが終了し、そのゴールについて他の解を求めたい場合、最終ゴールを強制的に fail させる。

- **CLOSE** **CREEP** で展開された部分の表示を一時的に閉じる。
- **OPEN** **CLOSE** された部分を再度開ける。
- **TRACE EXIT** トレースを終了させる。

3 構造化トレーサー

Prologプログラミングにおいて単一化、バックトラック並びにカットの制御を把握するのは困難なものである。我々は、以下に示すような方法をトレーサーシステムに取り入れ、ユーザーがプログラムの動作を視覚的に理解できるように考えた。

1. ゴールについてある節と単一化が生じた時、その節のボディをゴールの下にはめ込むが、この時に単一化前後のトレース状況を交互に見せる。これにより、単一化により生じた値の伝搬が視覚的に判断できる。また、その横に単一化の対象となった節を並べて表示させ認識させる。
2. failが生じた時、どのゴールへバックトラック可能かを認識するのはプログラムの流れを把握する時には大切なことである。このためゴールと単一化が生じた時、単一化した節と残りの節番号を表示し、また現時点以降でバックトラックの対象となり得るゴールの先頭に # マークを表示する。このような視覚的な表示により、カットを通過する前後のバックトラックにおける動作もわかりやすくなる。
3. 表示データの量が実行毎に多くなってきた場合、表示の限界や見やすさを考慮し、今までに展開された部分の表示を一時的に閉じて、全体像を見やすくする。また、その部分は再度開くこともできる。

以上の機能を持つ構造化トレーサーのトレース実行画面を図1に示す。

†A Structured Tracer for Novice Prolog Programmers
Sigeru YAMAZAKI, Yoshitaka ENOMOTO, Makoto ITAMI
and Kohji ITOH
Department of Applied Electronics, Science University of
Tokyo
2641 Yamazaki, Noda, Chiba 278, Japan

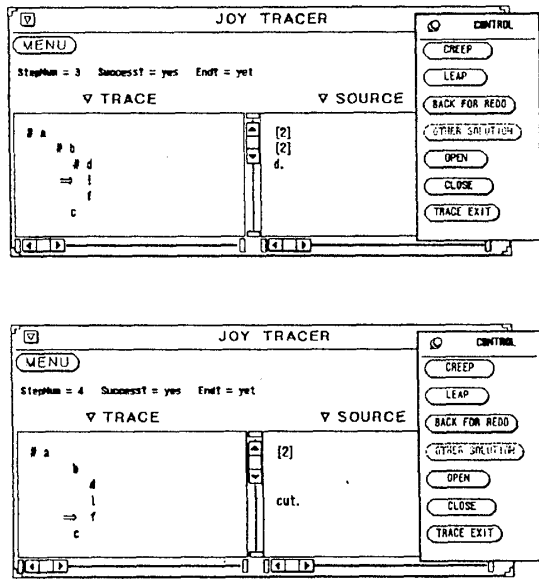


図 1: トレース実行画面

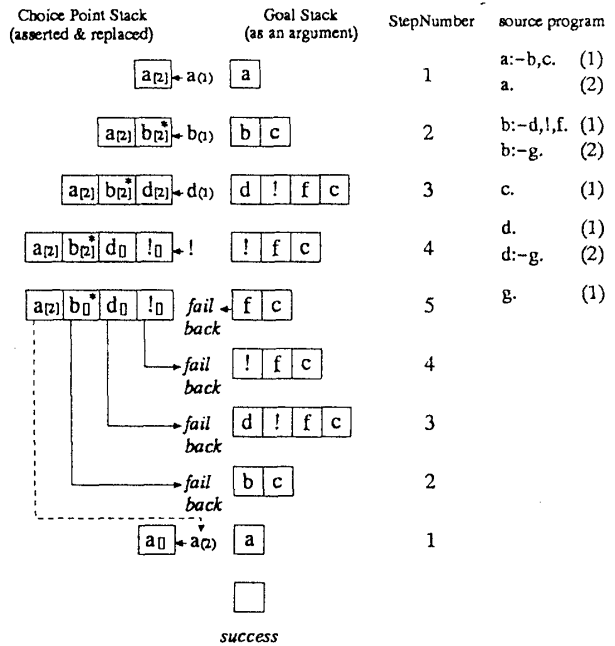


図 2: 仮想インタプリタ

4 仮想インタプリタ

Prolog 自体のインタプリタでは 3 節の機能を実現できない。そこで、図 2 のような仮想インタプリタを用意することで、トレース状況を視覚的かつ構造的に表現する。

- **Goal Stack** ゴールスタックを用いて、Prolog で単一化を行なう。例えば、ソースプログラムを見てトレースしたいゴールを a とすると、頭部が a であるようなデータを捜し出し、ゴールスタックに入っていた a の代わりに、b、c が置き換わる。以下同様なことを先頭ゴールスタックについて行ない、ステップ毎にゴールスタックを更新していく。
- **Choice Point Stack** ゴールについてある節と単一化代入が生じた時は、単一化代入の生じた先頭ゴールスタックを Choice Point Stack に積んでいくことにより、節の履歴をとっておく。バックトラックの制御に用いる。
- **Choice Point Goal Stack** トレース状況をキャンパス上に表示するためには、ゴールを根とし、ゴールにつれて単一化が行なわれた節のサブゴールを枝としてその各々のゴールを根とする部分木をそこへ接ぎ木することによって形成される木構造のリスト表現が必要である。これを Windowdata といい GUI 部へ送るトレースデータの元となるデータである。

これは Goal Stack と Choice Point Stack を用いて作られる Choice Point Goal Stack から得ることができる。

5 おわりに

図 2 のインタプリタを用いた構造化トレーサーにより、学習者は Prolog の特徴である単一化やバックトラックなどの制御の流れをトレーサー上で視覚的に理解することができる。今後の課題としては、[2] で提案したプロトタイプとインスタンスに基づくプログラミング支援との連携を実現することである。

参考文献

[1] 村松, 府川, 伊丹, 伊藤: “Prolog プログラミングの学習を支援する構造化トレーサー その 2,” 電子情報通信学会春季大会, A-304 (1993).

[2] Itoh, K., Itami, M., Fukawa, K., Muramatsu, J. and Enomoto, Y.: “A Workbench System for Novice Prolog Programmers: Visually-Structured Interactive Tracer and Prototype-Based Programming Support,” *Tran. IEICE, Information and Systems*, vol.E77-D, no.1 (1994).