

KIIMA

並行論理プログラム自動修正系

網代 育大 上田 和紀

早稲田大学理工学部 情報学科 上田研究室

{ajiro,ueda}@ueda.info.waseda.ac.jp

<http://www.ueda.info.waseda.ac.jp/~ajiro/study-j.html>

はじめに

Kima は強型 (モード) 体系を用いることによって、**プログラムに関する仕様や宣言を与えることなく** KL1 プログラムの軽微な誤りを自動的に修正するシステムである。

- 強型体系 (**strong typing**) はデータ型の一貫性を保証し、多くの簡単な誤りを静的に検出する。
- 並行論理プログラミングにおける強モード体系 (**strong moding**) はデータフローの一貫性を保証し、多くの簡単な誤りを静的に検出する。
- このときのモード/型解析は多数の簡単なモード/型制約の制約充足問題であり、誤りを含むプログラムでは制約集合が充足不可能 (**non-well-moded**) となる。

並行論理プログラミング

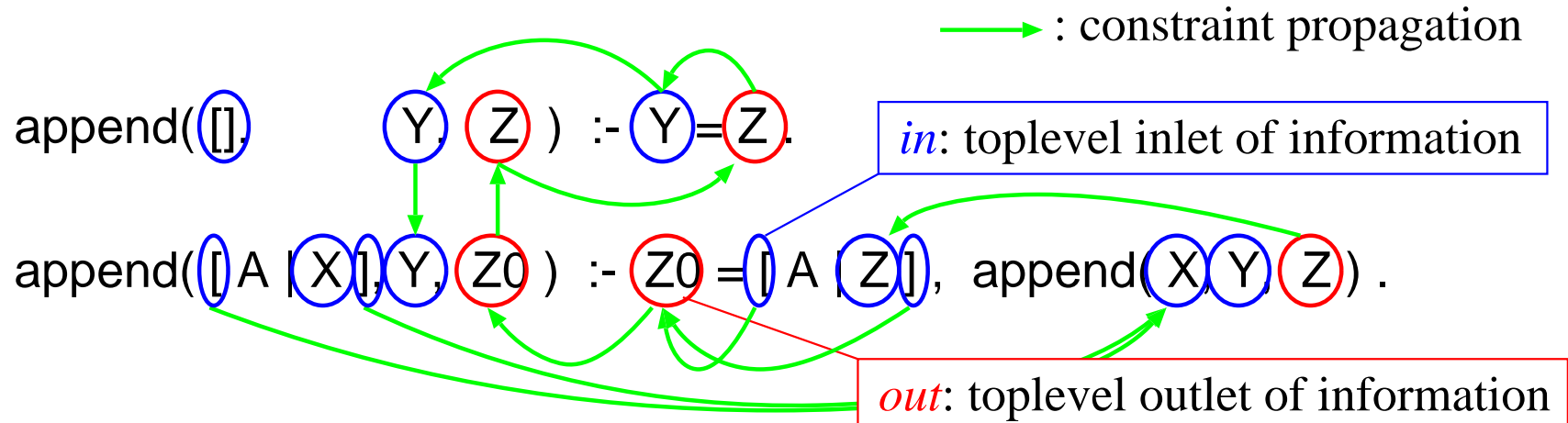
並行論理型言語 KL1 の特徴

- 並行・並列・分散計算に適している。
- 論理型言語 (Prolog) からバックトラックを除いた言語。
- 計算は (並列計算/論理プログラミングにおける) リダクションによって進む。
- プロセス間通信は、論理変数を通じて暗黙のうちに行なわれる。

KL1 プログラムは以下のような節の集合：

```
p(X,Z) :- true | q(X,Y)@node(0), r(Y,Z)@node(1).
```

モード解析による情報の流れの解析



- モード解析は多数の簡単なモード制約の制約充足問題である。
- モード制約は (i) プログラムテキスト中の記号出現と (ii) モードづけ規則によって課される。
- 正しいプログラムは制約集合が充足可能となり、**誤りを含んだプログラムでは、制約集合が充足不可能となることが多い。**
- データ型に関しても同様の方法で解析可能

Kima の提示した修正案の内訳

修正案1 – 意図通りの正しい修正案

例えば、`append([1,2,3],[4,5,6],Out)` のように呼び出すと 2つの入力リストをつなげたリスト `[1,2,3,4,5,6]` を出力。

修正案2 – 意図通りではないが、ちゃんと動作をする

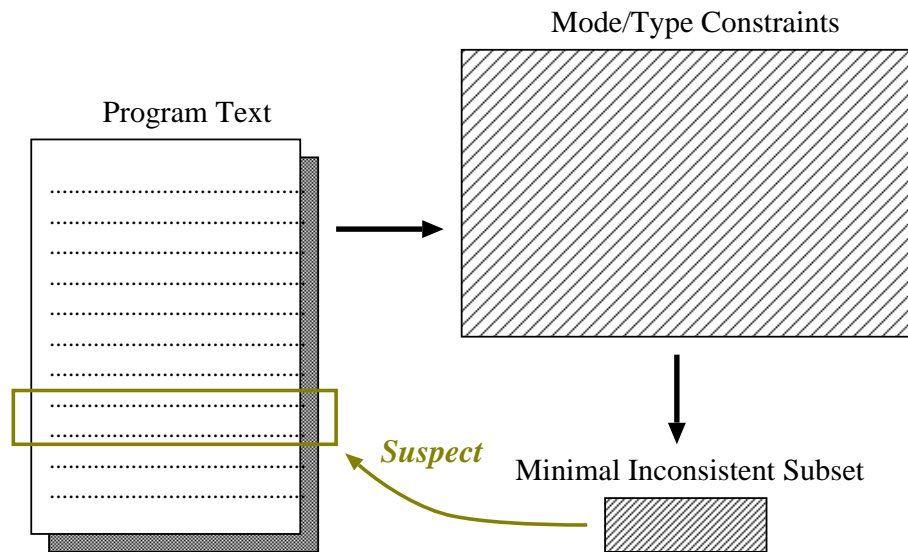
上と同様に、`append([1,2,3],[4,5,6],Out)` として呼び出すと `[1,4,2,5,3,6]` を出力。つまり、入力である 2つのリストを交互にマージする。

Kima はプログラムに関する明示的な仕様や宣言をまったく必要としないが、それらしいプログラムを見つけて提示することができる!

Kima の動作原理

基本アルゴリズム

1. モード/型制約の矛盾する極小部分集合の計算による、誤り箇所候補の特定
2. 以下の手続きの繰返しによる generate-and-test
 - (a) 特定した場所の周辺の機械的な書換え
 - (b) 書換えたプログラムのモード/型の再計算



矛盾する極小部分集合 (minimal inconsistent subset) はプログラムサイズに比例する手間で計算可能。また、極小部分集合のサイズは、プログラムサイズと関係なく小さい(およそ10以下)。

今後の目標 – 自動プログラミングへの応用

入出力例を与えることによって、変数の記述を自動化

クイックソートプログラム `qsort` の外枠と入出力例

```
:- instance qsort(+ [5,2,7,3], - [2,3,5,7], + []).
qsort/3 :- X = [] | = /2.
qsort/3 :- X = [X|X] |
    part/4, qsort/3, X = [X|X], qsort/3.
part(-, [], S, L) :- true | S = [], L = [].
part(A, [X|Xs], S0, L) :- A >= X | S0 = [X|S], part(A, Xs, S, L).
part(A, [X|Xs], S, L0) :- A < X | L0 = [X|L], part(A, Xs, S, L).
```

⇓ 探索による自動生成

```
qsort([], Ys0, Ys) :- true | Ys = Ys0.
qsort([X|Xs], Ys0, Ys3) :- true |
    part(X, Xs, S, L), qsort(S, Ys0, Ys1), Ys1 = [X|Ys2], qsort(L, Ys2, Ys3).
```