

計算機言語

1 はじめに

モデル検査とは、形式言語でいうところの、言語の包含関係をみること。更には、実行可能列を木 (tree language) で表して、その包含関係をみること。

word language よりも tree language のほうが表現力が高い

例えば word language で {abc, abd} と表されるものは、tree language では 次の異なる 2 つある。

- (a b (c d))
- (a (b c) (b d))

2 高階モデル検査

HORS (grammar for generating an infinite tree) とかいう木。

木の型を o とする。これに対して高階な書き換え規則、

- order 1: $o \rightarrow o$
- order 2: $(o \rightarrow o) \rightarrow o$

次のプログラムを考える。

```
let f x =  
  if *  
    then close x  
    else { read x; f x }  
in f (open "foo")
```

ここで $*$ は非決定的な真偽値である。

このプログラムは次のような規則によって木に書き換えることが可能であるが、この規則は結局 CPS 変換に過ぎない。

```
(F x k) -> (+ (c x k) (r x (F x k)))  
(S d) -> (F (o "foo") d)
```

一引数だった関数 f を、引数と継続を取るシンボル F に置き換えてる。スタートはシンボル S である。また $+$ は非決定的な枝分かれを意味する。

2.1 combinational with predicate abstraction and CEGAR

ラムダと再帰で書かれるプログラムのことを、関数型プログラムとここでは言う。関数型プログラムについて、述語を与えることで、真偽値のみを扱うようなプログラムに変換する。一般にこの変換は非可逆。そしてその変換後のプログラムは容易に木に変換でき、高階モデル検査を行うことができる。

- 検査が成功した場合、プログラムに問題のないことが保証され、
- 失敗した場合はプログラムに問題があるか、あるいは与えた述語が悪い。

2.2 データ圧縮

データを出力するプログラムというのは、それ自体が圧縮だとみなせる。そこでは、解答とはプログラムを評価することである。データの性質を調べるのに、解凍することなく、できれば嬉しい。

例えば、次のプログラムはある run-length に相当する。

```
((lambda (f x) (f (f (f x)))) (lambda (x) (a (b x))) e)
```

これを解凍すると、(a (b (a (b (a (b e))))) となる。

また、圧縮という言葉が期待する性質は次であろう。

生データ D 、圧縮データ C 、解釈系 M に対して

$$\text{size}(D) \geq \text{size}(C) + \text{size}(M)$$

2.3 高階再帰スキーム

型付き木の決定的な書き換え規則の集合のことを言う。

木の型に対してオーダーを次のように数える。

$$\begin{aligned} \text{order } o &= 0 \\ \text{order } (\kappa_1 \rightarrow \kappa_2) &= \max\{\text{order } \kappa_1 + 1, \text{order } \kappa_2\} \end{aligned}$$

特に、オーダーが 0 のものを正規言語、1 のものを文脈自由文法、2 のものを index 言語という。

2.4 L-labeled tree

枝に自然数のインデックスが与えられ、インデックスの列からノードに付与されたラベルが一意に定まる木のことをそういう。下で定める T である。

- L is set of labels
- $\text{Pos} \in \mathbb{Z}_+$
- $T \subset \text{Hom}(\text{Pos}^* \rightarrow L)$

すなわち、あるノードからその子に対して n 本の枝が生えてるなら、左から、 $1, 2, \dots, n$ のインデックスを付与して、インデックスの列 $((), (1), (2, 1))$ を与えると、ラベル (a, b, c) が定まる。

この T が木であることを満たすために次の3つが必要。

- $\text{dom } T$ は prefix に就いて閉じている。
- $s \in \text{Pos}^*, i \in \text{Pos}. si \in \text{dom } T \implies sj \in \text{dom } T$ where $0 < j < i$
- $\text{dom } T \neq \emptyset$

2つ名言を言う。

木が有限である $\iff T$ の定義域が有限

finite branch を持つ \iff ある $s \in \text{Pos}^*$ について、 $\{i : si \in \text{dom } T\}$ が無限濃度を持つ。

2.5 ranked alphabet

アルファベットとは、

$$\Sigma = (L, \text{arity})$$

のこと。ここで、 L はラベルの集合、 arity は $L \rightarrow \mathbb{N}$ の型を持つ関数。

つまり、 arity によって、ラベルは一意的に自然数を持っている。つまり、そのラベルを与えられたノードはその自然数だけ子供を持つ。

Σ -labeled ranked tree はそのような木のことを言う。