Undecidable problems about CFL's

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

26 November 2011

Outline

Problems about CFL's

Problem (a)

Is it decidable whether a given CFG accepts a non-empty language?

Problems about CFL's

Problem (a)

Is it decidable whether a given CFG accepts a non-empty language?

Yes, it is. We can find out which non-terminals of G can derive a terminal string: i.e. there exists a derivation $X \stackrel{*}{\Rightarrow} w$ for some terminal string w.

- Maintain a set of "marked" non-terminals. Initially $N_{marked} = \emptyset$.
- Mark all non-terminals X such that $X \rightarrow w$ is a production in G.
- Repeat untill we are unable to mark any more non-terminals:
  - Mark X if there exists a production $X \rightarrow \alpha$ such that $\alpha \in (A \cup N_{marked})^*$.
- Return "Non-emtpy" if $S \in N_{marked}$, else return "Empty."

Problems about CFL's

Problem (b)

Is it decidable whether a given CFG accepts a finite language?

Problems about CFL's

Problem (b)

Is it decidable whether a given CFG accepts a finite language?

Yes, it is.

- Convert G to CNF.
- Check if there is a parse tree of depth $n+1$ where n is the number of non-terminals. $L(G)$ is infinite iff there is a parse tree of depth $n+1$ or more.

Problems about CFL's

Problem (c)

Is it decidable whether a given CFG G is universal. That is, is $L(G) = A^*$?

Problems about CFL's

Problem (c)

Is it decidable whether a given CFG G is universal. That is, is $L(G) = A^*$?

No, it is undecidable.

Undecidability of universality of a CFL

- We can reduce $\neg$HP to the problem of universality of a CFG:

$$\neg \mathrm{HP} \leq \text{Universality of CFG}.$$

- Given a TM M and input x, we can construct a CFG $G_{M,x}$ over an input alphabet Δ such that

  M does not halt on x iff $G_{M,x}$ is universal (i.e. $L(G_{M,x}) = \Delta^*$).

- Hence the problem is non-r.e.

Encoding computations of M on x

Let $M = (Q, A, \Gamma, s, \delta, \vdash, \flat, t, r)$ be a given TM and let $x = a_1 a_2 \cdots a_n$ be an input to it.
We can represent a configuration of M as follows:

$$\begin{array}{cccccc} \vdash & b_1 & b_2 & b_3 & \cdots & b_m \\ - & - & q & - & & - \end{array}$$

Thus a configuration is encoded over the alphabet $\Gamma \times (Q \cup \{-\})$.

Encoding computations of M on x

A computation of M on x is string of the form

$$c_0\#c_1\#\cdots\#c_N\#$$

such that

1. Each c_i is the encoding of a configuration of M.
2. c_0 is (encoding of) the start configuration of M on x.

$$\begin{array}{cccccc} \vdash & a_1 & a_2 & a_3 & \cdots & a_n \\ s & - & - & - & & - \end{array}$$

3. Each $c_i \xrightarrow{1} c_{i+1}$, and
4. c_N is a halting configuration (i.e. state component is t or r).

c_0	#	c_1	#	c_2	#	c_3	#	·······	#	c_N	#

Describing $Valcomp_{M,x}$

The language $Valcomp_{M,x}$ over the alphabet

$$\Delta = \Gamma \times (Q \cup \{-\}) \cup \{\#\}$$

can be described as the intersection of

- $L_1 = (C \cdot \#)^*$ where C is the set of valid encodings of configurations of M.
- $L_2 = \{c_0\# \cdots \#c_N\# \mid N \geq 1,\ c_i \overset{1}{\Rightarrow} c_{i+1}\}$.

Hence $\neg Valcomp_{M,x} = \overline{L_1} \cup \overline{L_2}$.
Claim: $\neg Valcomp_{M,x}$ is a CFL.

Problems about CFL's

Problem (d)

Is it decidable whether the intersection of 2 given CFG's is a CFL?

Problems about CFL's

Problem (d)

Is it decidable whether the intersection of 2 given CFG's is a CFL?

No, it is undecidable. Given M and x, describe 2 CFL's that accept computations of the form:

$$c_0 \# c_1 \# c_2 \# c_3 \# \cdots \# c_N \#$$

Problems about CFL's

Problem (e)

Is it decidable whether the complement of a given CFL is a CFL?

Problems about CFL's

Problem (e)

Is it decidable whether the complement of a given CFL is a CFL?

No, it is undecidable.
Exercise!