Valcomps

Tuesday, March 30, 2021 10:59 AM

If G is a CFG

Is $L(G) = \phi$?

This is decidable.

Is $L(G) = \Sigma^*$?

This is UNDECIDABLE!

Given a TM M and a word w $\langle M, w\rangle$ we can <u>effectively</u> <u>construct</u> a PDA (or a grammar) which recognizes the <u>COMPLEMENT</u> of the set of <u>valid</u> <u>computations</u> of M on w.

Effectively construct: I can describe the PDA explicitly without knowing in advance whether $M(w)\downarrow$.

What is a valid computation?

It is a string which describes all the steps taken by M as it processes w <u>until</u> <u>it</u> <u>halts</u>.

$VALCOMPS(M,w) = \phi$ iff M does not halt on w.

What are valcomps?

(1) A configuration of a TM is a description of its state, the word on the tape and the head pos.

Suppose the tape contains abbaab the state is q and the head is on the <u>third</u> cell; we write this as

$$a\,b\,q\,b\,a\,a\,b$$

the <u>name</u> of the state is written to the left of the cell where the head is positioned

(2) We use a special symbol #

assume $\# \notin Q \cup \Gamma$

This symbol separates consecutive configurations

e.g. suppose $\delta(q, b) = (q', a, R)$

$$a\,b\,q\,\underline{b}\,a\,a\,b \longrightarrow a\,b\,\underline{a}\,q'\,a\,a\,b$$

$$\cdots \# a\,b\,q\,b\,\underline{a\,a\,b \# a} \,b\,a\,q'\,a\,a\,b \# \cdots$$

A <u>PART</u> of a valid comp.

The start configuration looks like

$$\# q_0 a_1 \cdots a_n \#$$

$$w = a_1 \cdots a_n \in \Sigma^*$$

A <u>valid</u> <u>computation</u> for $\langle M, w\rangle$ is a sequence of configurations

$$\# \alpha_0 \# \alpha_1 \# \alpha_2 \# \cdots \# \alpha_N \# \quad \text{such that}$$

(i) α_0 is a start configuration

(ii) α_N is a halting configuration

(iii) α_{n+1} follows from α_n by the rules of the Turing machine.

$VALCOMPS(M,w)$

If $M(w)\uparrow$ then $VALCOMPS(M,w) = \phi$

Now I will describe a PDA (in outline) to recognize

$$\overline{VALCOMPS(M,w)}$$

If $VALCOMPS(M,w) = \phi$ then

$$\overline{VALCOMPS(M,w)} = \Delta^*$$

where $\Delta = \Gamma \cup Q \cup \{\#\}$

We can describe this PDA <u>without</u> <u>knowing</u> whether M halts on w.

If we can decide whether $\overline{VALCOMPS(M,w)} = \Delta^*$ we can answer the non-Halting problem.

$$\underline{\neg H_{TM}} \leq_m \underline{\{\langle G\rangle \mid L(G) = \Sigma^*\}}$$

5 conditions to be checked for membership in $VALCOMPS(M,w) \ni z$

(a) z begins and ends with # and between each successive pair of #'s we must have a non-empty string over $\Delta \setminus \{\#\}$ $\quad z = \# \alpha_0 \# \alpha_1 \# \alpha_2 \# \cdots \# \alpha_N \#$

(b) each α_i must contain <u>exactly</u> <u>one</u> symbol from Q

(c) α_0 must be a start config.

(d) α_N must be a halt config

(e) For each i $(\forall i)$ $\alpha_i \longrightarrow \alpha_{i+1}$ according to the rules of M.

Conditions (a), (b), (c) and (d) can be checked by a DFA.

If any of these conditions are violated accept z.

To check (e) is difficult but to check that (e) is violated is (relatively) easy.

$\exists$

Our pda will <u>GUESS</u> a place where $\alpha_i \longrightarrow \alpha_{i+1}$ is violated

CRUCIAL IDEA

if $\alpha_i \longrightarrow \alpha_{i+1}$ follows the rules

then α_i and α_{i+1} can only differ in a window of length 3.

e.g.: $\delta(q, a) = (p, b, L)$

$$a\,b\,\underline{a}\,q\,a\,b\,b\,a \longrightarrow a\,b\,p\,\underline{a}\,b\,b\,b\,a$$

We call 2 pairs of 3 symbol sequences CONSISTENT if

(i) they are identical and neither one contains the head

(ii) one or both contain the head and they respect the rules of the TM.

There are only finitely many such pairs and they can all be remembered in the state of the PDA.

We need the stack to find the corresponding positions

$$\cdots \# \underbrace{\quad}_{w_1} [XXX] \underbrace{\quad}_{w_2} \# \underbrace{\quad}_{w_1'} [YYY] \underbrace{\quad}_{w_2'} \# \cdots$$

The PDA guesses that this is where (e) breaks down.

It stacks w_1 on its stack, it remembers XXX in its state, it ignores w_2 and goes to the next #, then it pops the stack as it reads w_1' so that it finds the right place to compare. If the 2 3-letter sequences XXX, YYY do <u>not</u> match then <u>ACCEPT</u>.

DONE!

$$M(w)\uparrow \text{ iff } \overline{VALCOMPS(M,w)} = \Delta^*$$

$VALCOMPS2(M,w)$:

$$\# \alpha_0 \# \alpha_1^{REV} \# \alpha_2 \# \alpha_3^{REV} \# \cdots \# \alpha_N^{\odot} \#$$

$\alpha_N^{\odot} \to$ if N is odd reverse it
if N is even leave it as is.

One PDA can check

$\alpha_0 \to \alpha_1,\ \alpha_2 \to \alpha_3,\ \alpha_4 \to \alpha_5 \cdots$

Another PDA can check

$\alpha_1 \to \alpha_2,\ \alpha_3 \to \alpha_4, \cdots$

Between them they <u>CAN</u> check

$VALCOMPS2(M,w)$

This is NOT a 2 stack machine

$VALCOMPS2(M,w) =$

$$L(G_1) \cap L(G_2)$$

where G_1, G_2 are CFG's.

$$HP \leq_m \{\langle G_1, G_2\rangle \mid L(G_1) \cap L(G_2) \neq \phi\}$$

---

1. Given G a CFG is $L(G) = \Sigma^*$?
2. Given G_1, G_2, CFG's is $L(G_1) \cap L(G_2) \neq \phi$?

NEITHER QUESTION is decidable

1. co CE but not CE
2. CE but not co CE.