Valid Computation Histories

Claim 1. $\overline{\text{VALCOMPS}(M, x)}$ is a CFL.

Claim 2. We can construct a CFG G for $\overline{\text{VALCOMPS}(M, x)}$ from the description of M and x.

Observation. $L(G) = \Delta^*$ iff M does not halt on x.

Reduction. $\overline{\text{HP}} \leq_m \underbrace{\{G \mid G \text{ is a CFG and } L(G) = \Delta^*\}}_{\text{not r.e.}}$.

Conditions for a string $z \in \Delta^*$ to be a valid computation history of M on x:

1. z must begin and end with a $\#$. It must be of the form $\#\alpha_0\#\alpha_1\#\cdots\#\alpha_N\#$,
2. each α_i is a string of symbols of the form

   a   a

         or

   \-   q

   where exactly one symbol of α_i has an element of Q on the bottom and others have $-$, and only the leftmost has a $\vdash$ on top,
3. α_0 represents the start configuration of M on x,
4. a halt state, either t or r appears somewhere in z. That is, α_N is a halting configuration,
5. $\alpha_i \xrightarrow[M]{1} \alpha_{i+1}$ for $0 \le i \le N-1$.

$$A_i = \{x \in \Delta^* \mid x \text{ satisfies conditions } (i)\}, \quad 1 \le i \le 5$$

$VALCOMPS(M,x) = \bigcap_{1 \le i \le 5} A_i$; $\overline{VALCOMPS(M,x)} = \bigcup_{1 \le i \le 5} \overline{A_i}$

Claim. Sets A_1, A_2, A_3, A_4 are regular sets.

- z must begin and end with a $\#$.

  Observation. A_1 is the regular set $\#\Delta^*\#$.

- each α_i is a string of symbols of the form

$$\begin{array}{c} a \\ - \end{array} \quad \text{or} \quad \begin{array}{c} a \\ q \end{array}$$

  where exactly one symbol of α_i has an element of Q on the bottom and others have $-$, and only the leftmost has a $\vdash$ on top,

  Suffices to check that between every two $\#$'s there is exactly one symbol with state q on the bottom and $\vdash$ occurs on the top immediately after each $\#$ (except the last) and nowhere else.

  Observation. A_2 is the regular set.

- α_0 represents the start configuration of *M* on *x*,

  Observation. A_3 is the regular set

- a halt state, either *t* or *r* appears somewhere in *z*. That is, α_N is a halting configuration,

  Observation. Suffices to check that *t* or *r* appears somewhere in the string.

Condition 5. $\alpha_i \xrightarrow[M]{1} \alpha_{i+1}$ for $0 \leq i \leq N-1$.

Claim. $\overline{A_5}$ is a CFL.

Condition 5. $\alpha_i \xrightarrow[M]{1} \alpha_{i+1}$ for $0 \le i \le N-1$.

$$\cdots \# \vdash a\ b\ a\ b\ b\ a\ b\ a \quad \# \vdash a\ b\ a\ a\ b\ a\ b\ a\ \# \cdots$$
$$\quad -\ -\ -\ -\ q\ -\ -\ -\ - \qquad\quad -\ -\ -\ p\ -\ -\ -\ -\ -$$

if $\quad \delta(q,b) = (p,a,L)$

Note. α_i & α_{i+1} should agree on most symbols except a few near the current head position

How to check if $\alpha \xrightarrow[M]{1} \beta$?

Check all 3 element substring u of α and the corresponding substring v of β
↳ occuring at the same distance from #

$$\underbrace{\begin{matrix} a & b & b \\ - & q & - \end{matrix}}_{\text{in } \alpha} \quad \text{and} \quad \underbrace{\begin{matrix} a & a & b \\ p & - & - \end{matrix}}_{\text{in } \beta} \Bigg\} \text{ are consistent with } \delta. \quad \delta(q,a) = \delta(p,b,L).$$

The pair $\underbrace{\begin{matrix} a & b & b \\ - & - & - \end{matrix}}_{\text{in } \alpha}$ and $\underbrace{\begin{matrix} a & b & b \\ - & - & - \end{matrix}}_{\text{in } \beta}$

How to check that $\alpha \xrightarrow[M]{1} \beta$ does not hold?

Check if there is a length 3 substring of α s.t the corresponding length 3 substring of β is not consistent with δ.

$$\#\alpha_0\#\alpha_1\#\alpha_2\#\cdots\#\alpha_N\#$$

Condition 5. $\alpha_i \xrightarrow[M]{1} \alpha_{i+1}$ for $0 \le i \le N-1$.

Claim. $\overline{A_5}$ is a CFL. There exists a NPDA M such that $L(M) = \overline{A_5}$

Definition of M.

- Guess α_i non-deterministically.
- Guess a length 3 substring u in α_i, check that the corresponding substring v in α_{i+1} is not consistent.

How do you identify the "corresponding" substring v in α_{i+1}? Using the stack

- Push the prefix of α_i till u in the stack.
- Store u in the finite control ($|u| = 3$)
- Pop the symbols in α_{i+1} after # to match the length of the prefix to find the corresponding substring v in α_{i+1}.