

ECE 664

Lecture 14

Avi Kak

String Computations with Productions and Processes

- What is a production?

Given a pair of strings g and \bar{g} , the notation

$$g \rightarrow \bar{g}$$

means that when g appears as a substring in a longer string, it can be replaced by \bar{g} . We call $g \rightarrow \bar{g}$ a semi-Thue production, or just a production, or a rewrite rule.

- Obviously, then, if u is some word $r\bar{g}s$, we can rewrite u as $v = r\bar{g}s$. This fact is expressed with the notation

$$u \xrightarrow{P} v$$

where P denotes the production $g \rightarrow \bar{g}$.

- What is a semi-Thue process?

A semi-Thue process is a finite set of semi-Thue productions. If Π is a semi-Thue process and if there exists a production $P \in \Pi$ such that $u \xrightarrow{P} v$, we are also allowed to write $u \xrightarrow{\Pi} v$.

- What is a derivation of one string from another?

Given a semi-Thue process Π , if one can write down the following chain of productions

$$u = u_1 \xrightarrow{\Pi} u_2 \xrightarrow{\Pi} u_3 \dots \xrightarrow{\Pi} u_n = v$$

we then call the sequence u_1, u_2, \dots, u_n a derivation of v from u . When such a derivation exists, we write

$$u \xrightarrow[\Pi]{*} v$$

Simulation of a ~~Turing Machine~~ by a Semi-Thue Process

- Consider a Turing Machine M with alphabet $\{s_1, s_2, \dots, s_K\}$ and states q_1, q_2, \dots, q_n .
- We will denote the semi-Thue process that simulates M by $\Sigma(M)$.
- The alphabet of the semi-Thue process $\Sigma(M)$ will be

$s_0, s_1, s_2, \dots, s_K, q_0, q_1, \dots, q_n, q_{n+1}, h$

B M 's alphabet M 's states

to be used as a "root" symbol

to be used at a halt "state"

to be used as a beginning and end marker on Post words

- Each stage of the computation by M is characterized completely by the current tape configuration. Each tape configuration will be coded by a word on the alphabet $\Sigma(M)$ in the following manner:

$s_1 s_2 s_3 s_2 s_0 s_1 s_2$

$\uparrow q_4$

$\Rightarrow h s_1 s_2 s_3 q_4 s_2 s_0 s_1 s_2 h$

Post word

not unique

- In general, a word

$$h u q_i s_j v h \quad 0 \leq i \leq n+1$$

is called a Post word if u and v are words on the alphabet $\{s_0, s_1, \dots, s_K\}$.

- Next, we represent each quadruple of M by one or more productions in $\Sigma(M)$ using the following rules:

- ① if M contains $q_i s_j s_k q_l$ then $\Sigma(M)$ gets $q_i s_j \rightarrow q_l s_k$
- ② if M contains $q_i s_j R q_l$ then $\Sigma(M)$ gets $\begin{cases} q_i s_j s_k \rightarrow s_j q_l s_k & k=0, 1, \dots, K \\ q_i s_j h \rightarrow s_j q_l s_0 h \end{cases}$
- ③ if M contains $q_i s_j L q_l$ then $\Sigma(M)$ gets $\begin{cases} s_k q_i s_j \rightarrow q_l s_k s_j & k=0, 1, \dots, K \\ h q_i s_j \rightarrow h q_l s_0 s_j \end{cases}$
- ④ if for $i=1, \dots, n$ and $j=0, 1, \dots, K$, a particular $q_i s_j$ are NOT the first two entries of a quadruple of M , we place in $\Sigma(M)$ the production

$$q_i s_j \rightarrow q_{n+1} s_j$$

(that's why q_{n+1} represents the halt state of M)

- ⑤ finally, we place in $\Sigma(M)$ the production

$$\begin{aligned} q_{n+1} s_i &\rightarrow q_{n+1} \\ q_{n+1} h &\rightarrow q_h \\ s_i q_0 &\rightarrow q_0 \end{aligned}$$

our goal is that if M halts, we want to be able to derive q_h from the Post word corresponding to the initial tape configuration

Theorem: Let M be a nondeterministic Turing Machine. Then for each string u on the alphabet of M , M accepts u iff

$$h q_i s_0 u h \xrightarrow[\Sigma(M)]{*} h q_h h$$

Corresponds to the starting tape configuration

- The inverse of the production $g \rightarrow \bar{g}$ is the production $\bar{g} \rightarrow g$
- We will write $\Sigma(M)$ for the semi-Thue process which consists of the inverses of all the productions of $\Sigma(M)$.
- Let M be a nondeterministic Turing Machine. For each string u on the alphabet of M , M accepts u iff

$$h q_h h \xrightarrow[\Sigma(M)]{*} h q_i s_0 u h$$

The Word Problem

The word problem for a semi-Thue process Π is the problem of determining for any given pair u, v of words on the alphabet of Π whether

$$u \xrightarrow[\Pi]{*} v$$

Theorem: In general, the word problem is unsolvable because of the unsolvability of the word problem for $\Sigma(M)$ and $\Sigma(M)$ when the language accepted by M is r.e.

Proof:

Assume the word problem is solvable for $\Sigma(M)$. Then given two words u and v , we could devise an algorithm to test $u \xrightarrow[\Sigma(M)]{*} v$. The same algorithm could then be used to test whether $h q_i s_0 u h \xrightarrow[\Sigma(M)]{*} h q_h h$. That implies that we could use the same algorithm to test whether M would halt for a given input u . But since the language accepted by M is r.e., that is impossible.

- A semi-Thue process is called a Thue process if the inverse of each production in the process is also in the process.
- The notation $g \leftrightarrow \bar{g}$ means $g \rightarrow \bar{g}$ and $\bar{g} \rightarrow g$
- For a Turing Machine M , we write $\Theta(M) = \Sigma(M) \cup \Sigma(M)$

Thue process

For some reason, the unsolvability of the word problem is more disturbing than the unsolvability of the halting problem. It's best not to think about this unsolvability before your weekly zen meditation.

These rules guarantee that a Post word will only go into another Post word for a deterministic TM.