

THE POST CORRESPONDENCE PROBLEM

- It is a solitaire game played with a special set of dominoes.
 - Each half of each domino has a word on some alphabet A .
 - A Post Correspondence System is a finite set of such dominoes.
 - Example:
- | | | |
|----|----|----|
| a | bb | a |
| aa | b | bb |
- $A = \{a, b\}$
- right of the
- A move consists of placing a domino to the right of the dominoes already laid down with the understanding that the dominoes do not get used up. In other words, there is an ~~not~~ inexhaustible supply of dominoes for each domino in a given Post Correspondence System.
 - To win the game: You read all the letters in the upper halves of the dominoes as a single word. You do the same for the letters in the lower halves. The two words must be the same in order to win.
 - When a game is won, the word formed in either the upper halves or the lower halves is called a solution of the given Post Correspondence System.
 - What is a solution for the Post Correspondence System

a	bb	a
aa	b	bb

 ?

THEOREM: In general, determining whether or not a Post Correspondence System has a solution is unsolvable.

PROOF: You already know that, in general, the word problem for a semi-Thue process is unsolvable. That is, given the productions of a semi-Thue process Π and given two words u and v , we cannot devise a general purpose algorithm that will tell us whether $u \xrightarrow{\Pi} v$.

We will now show that for each derivation $u \xrightarrow{\star} v$ there exists a Post Correspondence System $P_{u,v}$ such that $P_{u,v}$ has a solution iff $u \xrightarrow{\star} v$.

- Take any semi-Thue process Π on the alphabet $\{a, b\}^*$.
- Add the productions $a \rightarrow a$ and $b \rightarrow b$ to Π . This allows us to write $w \xrightarrow{\Pi} w$ for every $w \in \{a, b\}^*$. (This is to ensure that a derivation $u \xrightarrow{\star} v$ is odd in length.)
- Let the productions be Π be denoted $g_i \rightarrow h_i$, $i = 1, 2, \dots, n$.
- The alphabet of $P_{u,v}$ consists of the 8 symbols: $a, b, \tilde{a}, \tilde{b}, [,], *, \tilde{*}$
- For any word $w \in \{a, b\}^*$, we write \tilde{w} for the word on the sub-alphabet $\{\tilde{a}, \tilde{b}\}$ that is obtained from w by replacing a by \tilde{a} and b by \tilde{b} .
- The Post Correspondence System $P_{u,v}$ consists of the following $2n+4$ dominoes:

l_w*	*	$\tilde{*}$	$]$	h_i	\tilde{h}_i
[*	*	\tilde{w}	\tilde{g}_i	g_i

$\rightarrow i = 1, 2, \dots, n$

- Note in particular that $P_{u,v}$ contains the following four dominoes

\tilde{a}	\tilde{a}	\tilde{b}	b
a	a	b	b
- The above allows us to use "macro" dominoes

\tilde{p}	p
p	p

 and

\tilde{p}	p
p	p

 for any $p \in \{a, b\}^*$

- Now let's assume that there exists the following derivation in Π

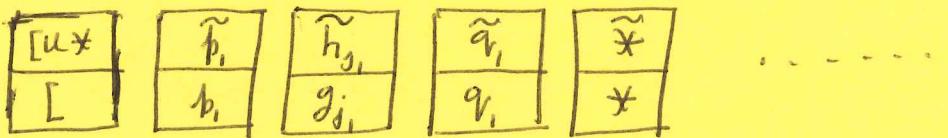
$$u = u_1 \xrightarrow{\pi} u_2 \xrightarrow{\pi} u_3 \xrightarrow{\pi} \dots \xrightarrow{\pi} u_m = v$$

- For the consecutive u_i 's we can obviously write

$$u_i = p_i g_{ji} q_i \quad u_{i+1} = p_i h_{ji} q_i$$

where we have assumed that the
 $u_i \Rightarrow u_{i+1}$ rewrite used
the j th production $g_j \rightarrow h_j$

- By keeping one eye on the above derivation, we now start playing the game.
- We lay down the following dominoes for the $u_1 \Rightarrow u_2$ rewrite:



- In this compound move, we made the bottom half of the ~~last four~~ ^{five} dominoes catch up with the top half of the first domino.
- The next set of dominoes we lay down must now catch up in the bottom half with the additional string $\tilde{u}_2 *$ created in the top halves of the last four dominoes above.

and so on

- This results in the following solution for P_{uv} : $[u_1 * \tilde{u}_2 \tilde{*} u_3 * \dots * \tilde{u}_{m-1} \tilde{*} u_m]$
- see the text for the reverse of this proof.

GRAMMARS

- A phrase-structure grammar is just a semi-True process in which the letters of the alphabet are separated into two disjoint sets called the variables (V) and the terminals (T).
- One of the variables $S \in V$ is singled out and called the start symbol.
- Convention: lower-case for terminals, upper case for variables.

Let Π be a grammar with V , T , and $S \in V$ as the start symbol. We define

$$L(\Pi) = \{u \in T^* \mid S \xrightarrow{*} u\}$$

as the language generated by Π .

Theorem: Let U be a language accepted by a nondeterministic Turing Machine M . Then there exists a grammar Π such that $U = L(\Pi)$.

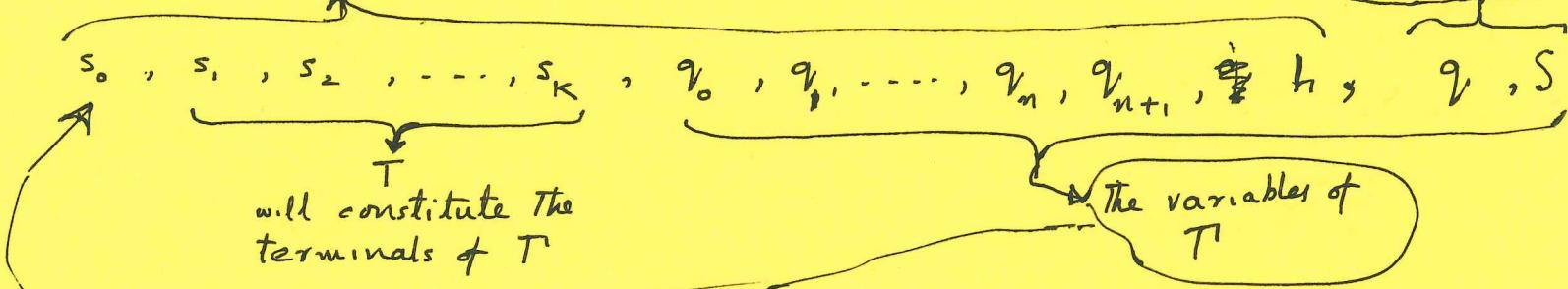
Proof: Let T be the alphabet of M . $T = \{s_1, s_2, \dots, s_K\}$. $U \subseteq T^*$.

- We will construct a grammar Π by modifying $\Sigma(M)$. Recall $\Sigma(M)$ is obtained from $\sum(M)$ by reversing all the productions.

- The alphabet of $\Sigma(M)$:

requires iff proof

two additional symbols for the alphabet of Π



- The productions of Π are the productions of $\Sigma(M)$ plus

- Let M accept $u \in T^*$. Then

$$S \xrightarrow{\pi} h q_0 h \xrightarrow{\pi} h q_1 s_0 h \xrightarrow{\pi} q_1 u h \xrightarrow{\pi} u q_1 h \Rightarrow u$$

- see text for the converse.

$$\begin{aligned} S &\rightarrow h q_0 h \\ h q_1 s_0 h &\rightarrow q_1 \\ q_1 s &\rightarrow s q \\ q_1 h &\rightarrow 0 \end{aligned}$$