CS 273 4/26/07

- Undecidability of CFG ambiguity from PCP

- Algorithms for automata and CFLs

$\text{Ambiguous}_{CFG} = \{ \langle G \rangle \mid G$ is a CFG and $L(G)$ is ambiguous$\}$

$\text{Ambiguous}_{CFG}$ is undecidable

Reduction is from PCP

# Post's Correspondence Problem (PCP)

$\Sigma$

$k$ dominos $\qquad t_j, b_j \in \Sigma^*$

$$\left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right] \; \cdots\cdots \; \left[\frac{t_k}{b_k}\right]$$

are there indices ~~i~~

$i_1, i_2, \ldots i_\ell \in \{1, 2, \ldots, k\}$ s.t.

$$t_{i_1} t_{i_2} \cdots t_{i_\ell} = b_{i_1} b_{i_2} \cdots b_{i_\ell} \; ?$$

Given dominos

$$\left[\frac{t_1}{b_1}\right]\left[\frac{t_2}{b_2}\right] \cdots\cdots \left[\frac{t_k}{b_k}\right]$$

first idea:

$$G = (V, \Sigma, R, S)$$

$$S \longrightarrow T \mid B$$

$$T \longrightarrow t_1 T \mid t_2 T \mid \cdots \mid t_k T \mid t_1 \mid t_2 \cdots \mid t_k$$

$$B \longrightarrow b_1 B \mid b_2 B \mid \cdots \mid b_k B \mid b_1 \mid b_2 \mid \cdots \mid b_k$$

add <u>new</u> symbols

$a_1\ a_2 \cdots a_k$ to $\Sigma$

$G = (V, \Sigma', R, S)$

where $\Sigma' = \Sigma \cup \{a_1, a_2, \ldots, a_k\}$

from PCP

$S \to T \mid B$

$T \to t_1 T a_1 \mid t_2 T a_2 \mid \cdots \mid t_k T a_k \mid$
$\mid t_1 a_1 \mid t_2 a_2 \mid \cdots \mid t_k a_k$

$B \to b_1 B a_1 \mid b_2 B a_2 \mid \cdots \mid b_k B a_k \mid$
$\mid b_1 a_1 \mid b_2 a_2 \mid \cdots \mid b_k a_k$

Suppose $T \overset{*}{\Rightarrow} w$

and $B \overset{*}{\Rightarrow} w$

Suppose $\text{Ambigous}_{CFG}$ is decidable.

Let $R$ be a decider for it

~~Let~~ create $M$ – a decider for PCP.

$M$: on input $x$ ~~b~~

1. check if $x$ is a valid PCP instance

2. From $x$ create $\langle G_x \rangle$ as described.

3. If $R$ accepts $\langle G_x \rangle$ accept $x$

4. If $R$ rejects $\langle G_x \rangle$ reject $x$.

# Algorithms

Decidability $\Longleftrightarrow$ Algorithms

Efficient algorithms

polynomial time solvability

on input of length $n$
algorithm runs in poly($n$)
time

$O(n)$ – linear time

$O(n^2)$ – quadratic

$O(n^3)$ – cubic

# Algorithms for automata

(1) Given DFA $M$ and $w$
is $w \in L(M)$ ?

$O(|w|)$ time

(a) input is $w$
(b) input is $\langle M, w \rangle$

(2) Given $w$ and NFA $M$
does $w \in L(M)$ ?

give $\langle M, w \rangle$

(a) convert $M$ into a DFA $M'$
run $w$ on $M'$

$O(2^s \cdot |w| \cdot s^2)$

$s$ = # of states of $M$

$O(|w| \cdot s + 2^s \cdot s^2)$

(b) $w = w_0 w_1 w_2 \cdots w_n$

$S_i$ = set of states of $M$
reachable from $q_0$
after reading
$w_0 w_1 \cdots w_i$

Given $S_i$
Compute $S_{i+1}$

$$S_{i+1} = E\left( \bigcup_{q \in S_i} \delta(q, w_{i+1}) \right)$$

$O(s^3)$

check if $S_n \cap F \neq \phi$

$O(|w| s^3)$

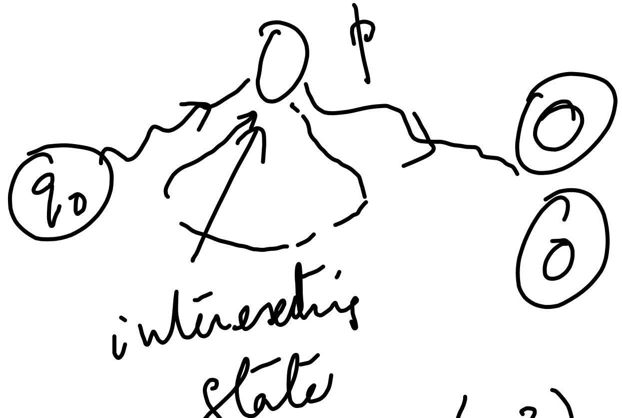
# Decision algorithms

Given $\langle M \rangle$ where $M$ is a DFA

① Is $L(M) = \phi$ ?

linear time
use BFS/DFS on
transition graph

② Is $L(M)$ finite ?




$O(s^3)$ time

③ $\langle M_1, M_2 \rangle$

is $L(M_1) = L(M_2)$ ?

$M_3$ s..

④ $\langle M_1, M_2 \rangle$

$L(M_1) \cap L(M_2) = \phi$ ?

~~$M_3$~~ create $M_3$ using product construction.

a. $L(M_3) = L(M_1) \cap L(M_2)$

check ~~$M_3$~~ if $L(M_3) = \phi$

③ $\langle M_1, M_2 \rangle$

$L(M_1) = L(M_2)$

create $M_3$ using product construction

s.t $L(M_3) = L(M_1) - L(M_2)$

check if $L(M_3) = \phi$

create $M_4$ using product construction s.t $L(M_4) = L(M_2) - L(M_1)$

check if $L(M_4) = \phi$

$L(M_1) = L(M_2)$ iff

$L(M_3) = \phi$ and $L(M_4) = \phi$

# Context Free Languages

## Decidable/Efficient

$\in$ CFG $\rightarrow$ PDA

PDA $\rightarrow$ CFG

$G \rightarrow$ CNF form

$O(n^2)$

$\boxed{w \in L(G)\ ? \quad O(n^3)\ \text{CYK}}$

$L(G) = \phi$ ?

$L(G)$ is finite?

## Undecidable

$ALL_{CFG}$

$L(G) = \Sigma^*$

$L(G)$ is ambiguous

$L(G_1) = L(G_2)$?

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

$L(G)$ is inherently ambiguous.

6 $L(G) = \phi$ ?

$G = (V, \Sigma, R, S)$

$L(G) = \{ w \mid S \overset{*}{\Rightarrow} w \}$

<u>Defn:</u> $A \in V$ is <u>generating</u> iff $\exists\, w \in \Sigma^*$ s.t. $A \overset{*}{\Rightarrow} w$

Compute <u>all</u> generating variables and check if $S$ is one of them.

$X = \Sigma$

repeat

for each $A \notin X$

if $A \rightarrow \alpha$ is a rule

and every symbol in $\alpha$ is

in $X$

add $A$ to $X$

Until $X$ does not change

Suppose $A \rightarrow BCa$

$B \overset{*}{\Rightarrow} x$ and $C \overset{*}{\Rightarrow} y$

$\Rightarrow A \overset{*}{\Rightarrow} xya$

$S \to AB$

$A \to a$

$X = \{a\}$

---

$X = \{a, A\}$

---