

Undecidable Problems for CFGs

CSCI 3130 Formal Languages and Automata Theory

Siu On CHAN

Chinese University of Hong Kong

Fall 2017

Decidable vs undecidable

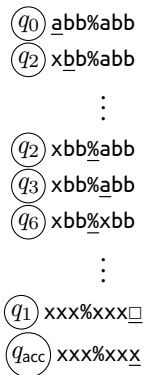
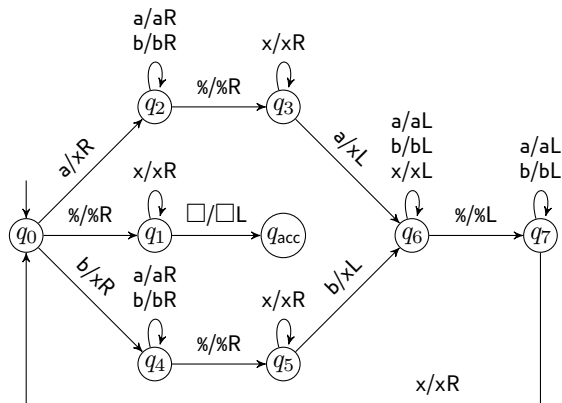
Decidable	Undecidable
DFA D accepts w	TM M accepts w
CFG G generates w	TM M halts on w
DFA D and D' accept same inputs	TM M accepts some input
	TM M and M' accept the same inputs

CFG G generates all inputs?

CFG G is ambiguous?

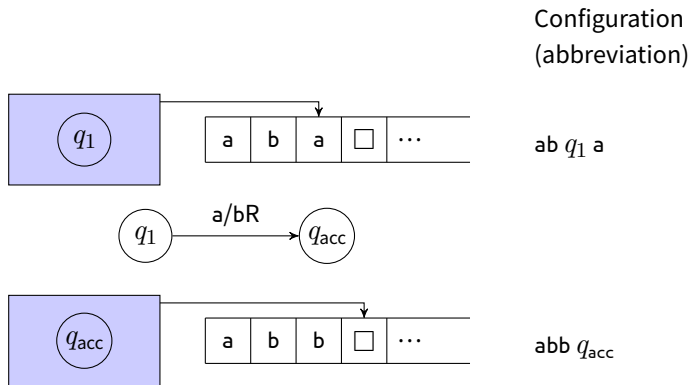
Representing computation

$$L_1 = \{w\%w \mid w \in \{a, b\}^*\}$$

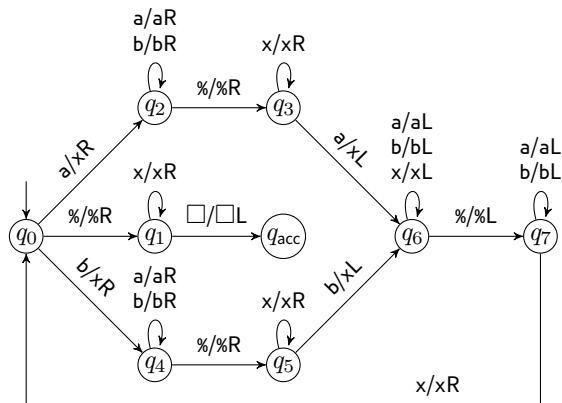


Configurations

A **configuration** consists of current state, head position, and tape contents



Computation histories



```

q0 abb%abb
x q2 bb%abb
  :
  :
xbb q2 %abb
xbb% q3 abb
xbb q2 %xbb
  :
  :
xxx%xxx q1
xxx%xx q_acc x
    
```

computation
history

Computation histories as strings

If M halts on w , the **computation history** of (M, w) is the sequence of configurations C_1, \dots, C_k that M goes through on input w

q_0	ab%ab	
x	q_2	b%ab
⋮		
xx%xx	q_1	
xx%x	q_{acc}	x

$$\# \underbrace{q_0 ab\%ab}_{C_1} \# x \underbrace{q_1 b\%ab}_{C_2} \# \dots \# xx\%x \underbrace{q_{acc} x}_{C_k} \#$$

The computation history can be written as a string h over alphabet $\Gamma \cup Q \cup \{\#\}$

accepting history: M accepts $w \iff q_{acc}$ appears in h
rejecting history: M rejects $w \iff q_{rej}$ appears in h

Undecidable problems for CFGs

$$\text{ALL}_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG that generates all strings} \}$$

The language ALL_{CFG} is undecidable

We will argue that

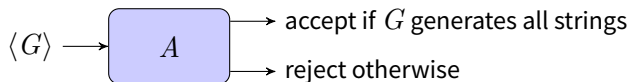
If ALL_{CFG} can be decided, so can $\overline{A_{\text{TM}}}$

$$\overline{A_{\text{TM}}} = \{ \langle M, w \rangle \mid M \text{ is a TM that rejects or loops on } w \}$$

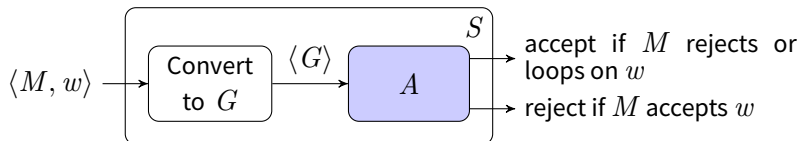
Undecidable problems for CFGs

Proof by contradiction

Suppose some Turing machine A decides ALL_{CFG}



We want to construct a Turing machine S that decides $\overline{A_{\text{TM}}}$



G generates all strings if M rejects or loops on w

G fails to generate some string if M accepts w

Undecidable problems for CFGs

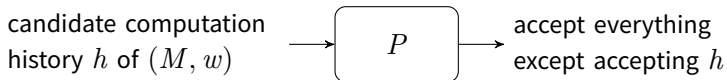


The **alphabet** of G will be $\Gamma \cup Q \cup \{\#\}$

G will generate all strings **except**
accepting computation histories of (M, w)

First we construct a PDA P , then convert it to CFG G

Undecidability via computation histories



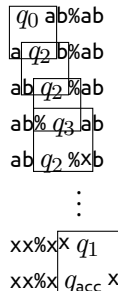
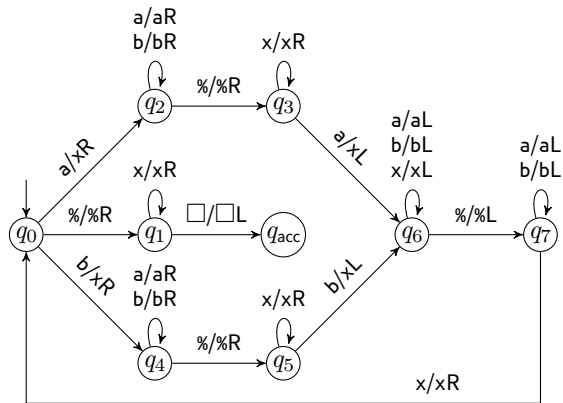
$\#q_0ab\%ab\#xq_1b\%ab\#\dots\#xx\%xq_{acc}x\# \Rightarrow$ Reject

$P =$ on input h (try to spot a **mistake** in h)

- ▶ If h is **not** of the form $\#w_1\#w_2\#\dots\#w_k\#$, **accept**
- ▶ If $w_1 \neq q_0w$ or w_k does **not** contain q_{acc} , **accept**
- ▶ If two consecutive blocks $w_i\#w_{i+1}$ do **not** follow from the transitions of M , **accept**

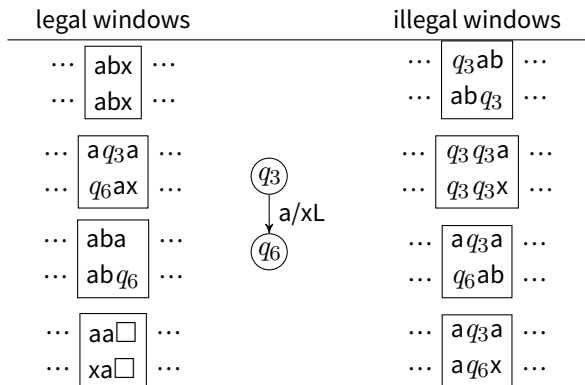
Otherwise, h must be an accepting history, **reject**

Computation is local



Changes between configurations always occur around the head

Legal and illegal transitions windows



Implementing P

If two consecutive blocks $w_i\#w_{i+1}$ do **not** follow from the transitions of M , **accept**

#xb% q_3 ab
#xb q_5 %xb

For every position of w_i :

- Remember offset from # in w_i on stack

- Remember first row of window in state

After reaching the next #:

- Pop offset from # from stack as you consume input

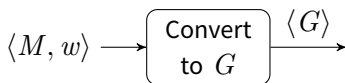
- Remember second row of window in state

If window is **illegal**, accept; Otherwise reject

The computation history method

$$\text{ALL}_{\text{CFG}} = \{\langle G \rangle \mid G \text{ is a CFG that generates all strings}\}$$

If ALL_{CFG} can be decided, so can $\overline{A_{\text{TM}}}$



G accepts all strings **except** accepting computation histories of (M, w)

We first construct a PDA P , then convert it to CFG G

Post Correspondence Problem

Input: A fixed set of tiles, each containing a pair of strings

bab	c	a	baa	a	bab
cc	ab	ab	a	baba	ϵ

Given an infinite supply of tiles from a particular set, can you match top and bottom?

a	baa	bab	c	c	bab	a
ab	a	ϵ	ab	ab	cc	baba

Top and bottom are both abaababccbaba

Undecidability of PCP

$PCP = \{ \langle T \rangle \mid T \text{ is a collection of tiles that contains a top-bottom match} \}$

Next lecture we will show (using computation history method)

The language PCP is undecidable

Ambiguity of CFGs

$$\text{AMB} = \{ \langle G \rangle \mid G \text{ is an ambiguous CFG} \}$$

The language AMB is undecidable

We will argue that

If AMB can be decided, then so can PCP

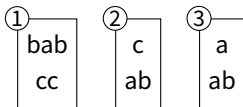
Ambiguity of CFGs

T (collection of tiles) \mapsto G (CFG)

If T can be matched, then G is ambiguous

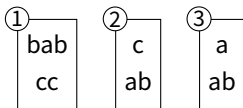
If T cannot be matched, then G is unambiguous

First, let's number the tiles



Ambiguity of CFGs

T (collection of tiles) \mapsto G (CFG)



Terminals: $a, b, c, 1, 2, 3$

Variables: S, T, B

Productions:

$S \rightarrow T \mid B$

$T \rightarrow babT_1$

$T \rightarrow cT_2$

$T \rightarrow aT_3$

$B \rightarrow ccB_1$

$B \rightarrow abB_2$

$B \rightarrow abB_3$

$T \rightarrow bab1$

$T \rightarrow c2$

$T \rightarrow a3$

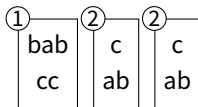
$B \rightarrow cc1$

$B \rightarrow ab2$

$B \rightarrow ab3$

Ambiguity of CFGs

Each sequence of tiles gives a pair of derivations



$S \Rightarrow T \Rightarrow \text{bab}T1 \Rightarrow \text{bab}cT21 \Rightarrow \text{bab}cc221$

$S \Rightarrow B \Rightarrow \text{cc}B1 \Rightarrow \text{ccab}B21 \Rightarrow \text{ccabab}221$

If the tiles **match**, these two derive the same string
(with different parse trees)

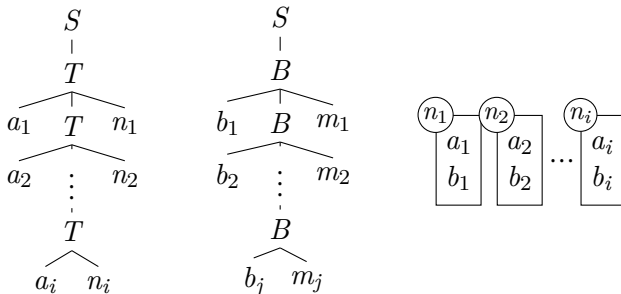
Ambiguity of CFGs

T (collection of tiles) \longmapsto G (CFG)

If T can be matched, then G is ambiguous ✓

If T cannot be matched, then G is unambiguous ✓

If G is ambiguous, then the two parse trees will look like



Therefore $n_1 n_2 \dots n_i = m_1 m_2 \dots m_j$, and there is a match