

## The Emptiness Problem

$E_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$ .

**Theorem.**  $E_{\text{TM}}$  is undecidable.

**Proof** a TM  $R$  deciding  $E_{\text{TM}} \rightarrow$  a TM  $S$  deciding  $A_{\text{TM}}$   
 $S$ 's algorithm: On input  $x$ ,

1. **Check that**  $x = \langle M, w \rangle$ . If "fail" reject  $x$ .
2. Construct a machine  $M_1$  such that for each input  $y$ ,  
( $\star$ ) **simulates**  $M$  on  $w$  and **accepts**  $y$  if  $M$  has accepted.
3. **Simulate**  $R$  on  $\langle M_1 \rangle$ . Accept  $x$  if  $R$  has accepted and reject  $x$  otherwise. ■

## Reducibility

## Testing Whether a TM Accepts a Regular Language

$REGULAR_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is regular}\}$ .

**Theorem.**  $REGULAR_{\text{TM}}$  is undecidable.

**Proof** a TM  $R$  deciding  $REGULAR_{\text{TM}} \rightarrow$  a TM  $S$  deciding  $A_{\text{TM}}$   
 $S$ 's algorithm: on input  $x$ ,

1. **Check that**  $x = \langle M, w \rangle$ . If "fail" reject  $x$ .
2. Construct a machine  $M_1$ : Let  $a, b$  be two distinct symbols in the input alphabet  $\Sigma$  of  $M$ . On input  $y$ ,
  - (a) **Accept**  $y$  if  $y = a^n b^n$  for some  $n$ .
  - (b) Otherwise, simulate  $M$  on  $w$ , then **accept**  $y$  if and only if  $M$  on  $w$  has accepted.

## The Halting Problem

$HALT_{\text{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM and halts on input } w\}$ .

**Theorem.**  $HALT_{\text{TM}}$  is undecidable.

**Proof** We can construct a machine  $S$  for  $A_{\text{TM}}$  from a machine  $R$  for  $HALT_{\text{TM}}$ . On input  $x$ :

1. **Check that**  $x = \langle M, w \rangle$ . If "fail" reject  $x$ .
2. **Simulate**  $R$  on  $w$ . Reject  $x$  if  $R$  has rejected.
3.  $M$  is **guaranteed to halt on**  $w$ .  
**simulate**  $M$  on  $w$ . Accept  $x$  if  $M$  has accepted and reject  $x$  otherwise. ■ ■

## Linear Bounded Automata

A **linear bounded automaton** is a Turing machine wherein the head is not permitted to move beyond the region in which the input was written. If the head attempts to move beyond the region it is kept at the same position.

**Lemma.** *Let  $M$  be an LBA with  $q$  states and with a tape alphabet of size  $s$ . For every  $n \geq 1$ , for every input of length  $n$ , there are **precisely  $qn s^n$  possible configurations.***

## The Acceptance Problem for LBA

$A_{\text{LBA}} = \{ \langle M, w \rangle \mid M \text{ is a TM and accepts } w \text{ when restricted to be an LBA} \}$ .

**Theorem.**  $A_{\text{LBA}}$  is decidable.

**Proof** Let  $M$  be a TM with  $q$  states and  $s$  symbols in the tape alphabet and  $w$  be an input to  $M$  of length  $n$ . By the previous lemma, there are only  $qn s^n$  configurations, so if  $M$  on  $w$  accepts, it should do so within  $qn s^n$  steps. So we have only to simulate  $M$  on  $w$  for  $qn s^n$  steps. ■

## Proof (cont'd)

The machine  $M_1$  satisfies:

(\*)  $L(M_1) = \{ a^n b^n \mid n \geq 1 \}$  if  $M$  accepts  $w$  and  $\Sigma^*$  otherwise, so  $M$  accepts  $w$  if and only if  $L(M_1)$  is not regular.

3. Simulate  $R$  on  $\langle M_1 \rangle$ . **Accept  $x$  if  $R$  has rejected and reject  $x$  otherwise.** ■

## Testing Equivalence Between TMs

$E_{\text{TM}} = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2) \}$ .

**Theorem.**  $E_{\text{TM}}$  is undecidable.

**Proof** a TM  $R$  deciding  $E_{\text{TM}} \rightarrow$  a TM  $S$  deciding  $A_{\text{TM}}$   
 $S$ 's algorithm: on input  $x$ ,

1. **Check that  $x = \langle M, w \rangle$ .** If "fail" reject  $x$ .
2. Construct a machine  $M_1$  as in the previous proof. Construct a machine  $M_2$  that accepts  $\Sigma^*$ .

Then  $L(M_1) = L(M_2)$  if and only if  $M$  does not accept  $w$ .

3. **Simulate  $R$  on  $\langle M_1, M_2 \rangle$ .** Accept  $x$  if  $R$  has rejected and reject  $x$  otherwise. ■

## The Equivalence Problem About CFG

$ALL_{CFG} = \{ \langle G \rangle \mid G \text{ is a CFG and } LG(G) = \sigma^* \}$ .

**Theorem.**  $ALL_{CFG}$  is undecidable.

**Proof** For a string  $x = \langle M, w \rangle$  such that  $M$  is a Turing machine and  $w$  is an input to  $M$ , let  $L_x$  be the set of all  $\#D_1\#\dots\#D_m\#$  for which there exist  $C_1, \dots, C_m$  such that:

1.  $C_1, \dots, C_m$  are configurations of  $M$ ,
2.  $C_1$  is the initial configuration of  $M$  on  $w$ ,
3.  $C_m$  is an accepting configuration of  $M$  on  $w$ ,
4. for every  $i$ ,  $2 \leq i \leq m$ ,  $C_i$  is the next configuration of  $C_{i-1}$ , and
5. for every  $i$ ,  $1 \leq i \leq m$ ,  $D_i = C_i$  if  $i$  is odd and  $D_i = C_i^R$  otherwise.

CS215, Lecture 9 © 1999 Mitsunori Ogihara

11

### Proof (cont'd)

Then  $L_x$  is empty if and only if  $M$  does not accept  $w$ .

So,  $\overline{L_x} = \Sigma^*$  if and only if  $M$  accepts  $w$ .

$\overline{L_x}$  is a CFL.

a TM  $R$  deciding  $ALL_{CFG} \rightarrow$  a TM  $S$  deciding  $A_{TM}$   
 $S$ 's algorithm: on input  $x$ ,

1. **Check that  $x = \langle M, w \rangle$ .** If "fail" reject  $x$ .
2. Construct a CFG  $G$  for  $\overline{L_x}$ .
3. **Simulate  $R$  on  $\langle G \rangle$ .** Accept  $x$  if  $R$  has accepted  $\langle G \rangle$  and reject  $x$  otherwise. ■

CS215, Lecture 9 © 1999 Mitsunori Ogihara

12

## The Emptiness Problem About LBA

$E_{LBA} = \{ \langle M \rangle \mid M \text{ is a TM and accepts no input viewed as an LBA } \}$ .

**Theorem.**  $E_{LBA}$  is undecidable.

**Proof** a TM  $R$  deciding  $E_{LBA} \rightarrow$  a TM  $S$  deciding  $A_{TM}$   
For a string  $x = \langle M, w \rangle$  such that  $M$  is a Turing machine and  $w$  is an input to  $M$ , let  $L_x$  be the set of all strings of the form  $\#C_1\#C_2\#\dots\#C_m\#$  such that

1.  $C_1, \dots, C_m$  are configurations of  $M$ ,
2.  $C_1$  is the initial configuration of  $M$  on  $w$ ,
3.  $C_m$  is an accepting configuration of  $M$  on  $w$ , and
4. for every  $i$ ,  $1 \leq i \leq m - 1$ ,  $C_{i+1}$  is the next configuration of  $C_i$ .

CS215, Lecture 9 © 1999 Mitsunori Ogihara

9

### Proof (cont'd)

Then  $L_x$  can be decided by an LBA.

$L_x \neq \emptyset$  if and only if  $M$  accepts  $w$ .

$S$ 's algorithm: on input  $x$ ,

1. **Check that  $x = \langle M, w \rangle$ .** If "fail" reject  $x$ .
2. Construct a TM  $B$  that accepts  $L_x$  as an LBA.
3. Simulate  $R$  on  $\langle B \rangle$ . Reject  $x$  if  $R$  has accepted  $\langle B \rangle$  and accept  $x$  otherwise. ■

CS215, Lecture 9 © 1999 Mitsunori Ogihara

10

## The Equivalence Problem (cont'd)

Define  $EQ_{CFG} = \{ \langle G, H \rangle \mid G \text{ and } H \text{ are CFGs that generate the same language} \}$ .

**Corollary.**  $EQ_{CFG}$  is undecidable.

**Proof** a TM  $R$  deciding  $EQ_{CFG} \rightarrow$  a TM  $S$  deciding  $ALL_{CFG}$   
 $S$ 's algorithm: On input  $x$ ,

1. **Check that  $x = \langle G \rangle$ .** If "fail" reject  $x$ .
2. Construct a grammar  $H$  for  $\Sigma^*$ .
3. **Simulate  $R$  on  $\langle G, H \rangle$ .** Accept  $x$  if  $R$  has accepted and reject  $x$  otherwise. ■