12. Turing Machines/Combinations of

- We claimed that Turing machines are a formal model of computers.

  > They are capable of performing any computation that can be carried out by any computer that ever existed or can ever be built.

- This lecture justifies this claim by showing how complex machines can be constructed from simple ones so that a complex computation can be carried out.

- The next lectures will further justify the claim by showing that attempts to generalize Turing machines do not lead to more powerful machines.

Simple machines

- W_a: writes a in the current square and halts.

  $K = \{q\}$, $s = q$.
  For any $b \in \Sigma$, $\delta(q, b) = (h, a)$.

  Will be written as a instead of W_a.

- V_L: moves head one square to the left and halts.

  $K = \{q\}$, $s = q$.
  For any $b \in \Sigma$, $\delta(q, b) = (h, L)$.

  Will be written as L instead of V_L.

- V_R: moves head one square to the left and halts.

  $K = \{q\}$, $s = q$.
  For any $b \in \Sigma$, $\delta(q, b) = (h, R)$.

  Will be written as R instead of V_R.

Combination rules:

- Assumption:

  The machines being combined are assumed to share the same alphabet but do not share any states.

- Rule 1:

  Operation of the combined machine:

  Start in M_1.

  Run M_1 until it halts.

  Then run M_2 with the output of M_1 as the input to M_2 and the final head position of M_1 as the initial head position of M_2.

  The combined machine halts iff M_2 halts. The output of the combined machine is that of M_2.

  Sometimes, the combined machine is written as $M_1 M_2$.

- Rule 2:

Operation of the combined machine:

- Start in M_1 and run M_1 until it halts.
- Let σ be the symbol of the current square.
- If $\sigma = a$, run M_2 with the output of M_1 as the input to M_2 and the final head position of M_1 as the initial head position of M_2.
  The combined machine halts iff M_2 halts. The output of the combined machine is that of M_2.
- If $\sigma = b$, run M_3 with the output of M_1 as the input to M_3 and the final head position of M_1 as the initial head position of M_3.
  The combined machine halts iff M_3 halts. The output of the combined machine is that of M_3.

12. Turing Machines/Combinations of

Examples:

- $\Sigma = \{\#, a, b, c\}$.
- The machine $R_{\#}$ given below moves right until it finds a $\#$.

- Shorthand: a, b, c on arc, $\sigma \neq \#$ on arc, and $\overline{\#}$ on arc.

- The machine $R_{\overline{\#}}$ given below moves right until it finds non-blank square.

12. Turing Machines/Combinations of

- The machine $L_{\#}$ given below moves left until it finds a $\#$.

- The machine $L_{\overline{\#}}$ given below moves left until it finds a non-blank square.

12. Turing Machines/Combinations of

- A machine that moves two squares to the right.

- A machine that scans to the right until it finds a non-blank square and then copies that symbol onto the square to its left.

- S_L: left shifting machine.

  transforms $\#w\underline{\#}$ into $w\underline{\#}$

  Figure 4-12 here

Pseudocode interpretation.

- Move to the first $\#$.
- Move one square to the right.
- While the current symbol $\sigma \neq \#$,
  - Move one square to the left.
  - Write σ.
  - Move one square to the right.
  - Move one square to the right.
- Move the to left until the current symbol is $\#$.

- C: Copying machine.

  transforms $\#w\underline{\#}$ into $\#w\#w\underline{\#}$

Figure 4-11 here

Pseudocode

- Move to the first $\#$.
- Move one square to the right.
- While the current symbol $\sigma \neq \#$,
  - Write $\#$.
  - Move to the right until two $\#$'s seen and write σ.
  - Move to the left until two $\#$'s seen and write σ.
  - Move one square to the right.
- Move to the right until one $\#$ seen.

12. Turing Machines/Combinations of

- $L = \{w \in \{a, b, c\}^*$:w has equal number of a's, b's, and c's$\}$.
- Accepted by the following Turing machine:

Figure 4-15 here.

Pseudocode: Input $\#w\underline{\#}$.

For $i = 0$ to infinity

- Scan input from right to left until an a seen. Replace it by d.

  If no a seen, halt if all d's seen and hang otherwise.
- Scan input from right to left until an b seen. Replace it by d. If no b seen, hang.
- Scan input from right to left until an c seen. Replace it by d. If no c seen, hang.

12. Turing Machines/Combinations of

The following machine computes $f(m,n) = mn$, $m,n \leq 0$.

Figure 4-17

Examples: $\#III\#II\#$, $\#\#II\#$

Theorem 1 *All Turing-decidable languages are Turing-acceptable.*

Proof: If a language is decided by a Turing machine M, the following machine accepts L.

Figure 6-1

Theorem 2 *Complement of a Turing decidable language is also Turing decidable.*

Proof: If a language L is decided by Turing machine M, then its complement $\overline{L}$ is decided by the following machine.

Figure 6-2 here

Note: Some Turing-acceptable languages are **not** Turing-decidable.