Lecture 38:
Combining Turing Machines

Aims:

- To see more examples of Turing machines, and
- To see how more complex Turing machines can be built up from simpler ones.

38.1. Example Turing Machines

- **Example 1.** Consider a Turing machine where $\Sigma = \{a, b, \sqcup\}$ and $Q = \{q_0, q_1, h\}$ and δ is as follows:

δ	a	b	$\sqcup$
q_0	b, q_1	a, q_1	$\sqcup, h$
q_1	L, q_0	L, q_0	$\sqcup, h$

Class Exercise: What does this machine do?

- Here's a trace of a particular computation.

$$
\begin{array}{lcrlll c}
 & \langle & \sqcup aa, & b, & \sqcup, & q_0 & \rangle \\
\leadsto & \langle & \sqcup aa, & a, & \sqcup, & q_1 & \rangle \\
\leadsto & \langle & \sqcup a, & a, & a\sqcup, & q_0 & \rangle \\
\leadsto & \langle & \sqcup a, & b, & a\sqcup, & q_1 & \rangle \\
\leadsto & \langle & \sqcup, & a, & ba\sqcup, & q_0 & \rangle \\
\leadsto & \langle & \sqcup, & b, & ba\sqcup, & q_1 & \rangle \\
\leadsto & \langle & \sqcup, & \sqcup, & bba\sqcup, & q_0 & \rangle \\
\leadsto & \langle & \sqcup, & \sqcup, & bba\sqcup, & h & \rangle
\end{array}
$$

- **Example 2.** Consider a Turing machine where $\Sigma = \{a, Y, N, \sqcup\}$ and $Q = \{q_0, q_1, h\}$ and δ is as follows:

δ	a	Y	N	$\sqcup$
q_0	L, q_1	—	—	Y, h
q_1	L, q_0	—	—	N, h

Class Exercise: What does this machine do?

- **Homework:** Trace it for yourself on a couple of examples.

38.2. Combining Turing Machines

- Our examples so far have been somewhat unimpressive. But Turing machines are of ultimate generality: we can design Turing machines for every computable computational problem.
- To make life easier, we will now show how to combine simple Turing machines into more complex ones.
- We'll develop a graphical notation for these complex Turing machines, so that we don't get bogged down in details of transition functions.

38.2.1. The Basic Turing Machines

- **Symbol-writing machines:** For each symbol in Σ, we can build a machine that writes that symbol and halts.
  E.g. for a:

δ	a	b	$\ldots$	$\sqcup$
q_0	a, h	a, h	$\ldots$	a, h

  Call this machine W_a
  (and, similarly, $W_b, W_c, \ldots, W_\sqcup$)

- **Head-moving machines:** We can build a machine that moves one cell left or right and halts.
  E.g. for left:

δ	a	b	$\ldots$	$\sqcup$
q_0	L, h	L, h	$\ldots$	L, h

  Call this machine M_L
  (and, similarly, M_R)

38.2.2. Rules for Combining Machines

- If TM_1 and TM_2 are Turing machines, we can create a Turing machine which will first behave like TM_1 and then like TM_2.
- How?
  1. Change all state names in TM_2 so they don't clash with state names in TM_1.
  2. Change all halts in TM_1's transition table to the new name of the start state of TM_2.
  3. Append TM_2's transition table to the foot of TM_1's transition table.
- E.g. For $\Sigma = \{a, b, \sqcup\}$, let's combine W_a (a machine for writing an a) with M_L (a machine that moves its head one cell to the left).

δ	a	b	$\sqcup$
q_0	a, q_1	a, q_1	a, q_1
q_1	L, h	L, h	L, h

- In general, if TM_1 and TM_2 are combined in this way, we will write

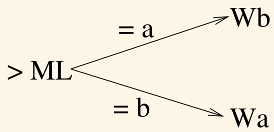
$$TM_1 \longrightarrow TM_2$$

  So this machine starts off in the initial state of TM_1, operates as per TM_1 until TM_1 would halt, then it launches TM_2 and operates as TM_2, until TM_2 would halt.
- We will also write $>$ to highlight the start of this combined machine.
- E.g. $> W_a \longrightarrow M_L$
- E.g. $> W_a \longrightarrow M_R \longrightarrow W_b \longrightarrow M_R \longrightarrow W_b \longrightarrow M_R \longrightarrow W_a$
- The connection between two Turing machines may depend on the symbol that is under the read/write head at the point when the first machine halts.

- We will depict this with a test alongside the arrow:

$$TM_1 \xrightarrow{test} TM_2$$

- E.g. $M_L \xrightarrow{=a} W_{\sqcup}$ This machine first moves left. Then, if there is an a under the read/write head, it overwrites it with a blank and then halts. If there had been any other symbol under the read/write head after moving left, it would have halted immediately.
- E.g. $M_L \xrightarrow{\in\{a,b\}} W_{\sqcup}$
- E.g. $M_L \xrightarrow{\neq a} W_{\sqcup}$
- E.g. $M_L \xrightarrow{\notin\{a,b\}} W_{\sqcup}$
- Multiple arrows are allowed, provided their tests are mutually exclusive.
- E.g.:



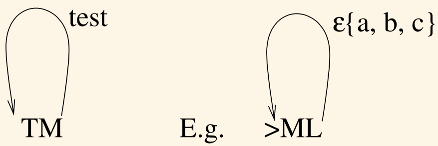
This machine first moves left. Then, if there is an a under the head, it writes a b and halts; if there is a b under the head, it writes an a and halts. If there were something else under the head, it would halt immediately after moving left.

- How is the transition table for this machine built?
  - Rename the states in W_b and W_a to avoid clashes.

- Change halts in M_L. Any halts in the a column are changed to the renamed start state of W_b. Any halts in the b column are changed to the renamed start state of W_a.
- Append the tables together.

δ	a	b	$\sqcup$
q_0	$L, \not{h} q_1$	$L, \not{h} q_2$	L, h
$\not{q_0} q_1$	b, h	b, h	b, h
$\not{q_0} q_2$	a, h	a, h	a, h

- Loops are allowed
- E.g.:



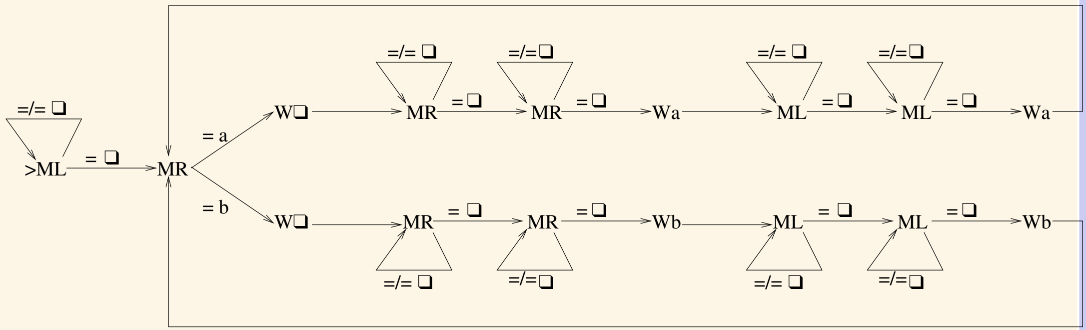
TM is executed. When it would halt, if the test is true, it returns to state q_0 instead. In the example, the machine moves left repeatedly, for as long as there is an a, b or c under the read/write head. When the symbol under the read/write head is not one of a, b or c, it halts. It is usual to include a test, otherwise you have an infinite loop.

- How is the transition table for this machine built?

– Change halts in M_L. Any halts in the a column are changed to the start state of M_L. Similarly any halts in the b and c columns.

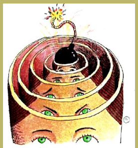
δ	a	b	c	$\sqcup$
q_0	$L, \not{h} q_0$	$L, \not{h} q_0$	$L, \not{h} q_0$	L, h

- Using this graphical notation, we can more easily specify more complex Turing machines.
- (Remember: this graphical notation is just a shorthand for specifying Turing machines properly. It saves us the tedious, pains-taking effort of writing down transition tables.)
- **Example 3.** Consider this Turing machine where $\Sigma = \{a, b, \sqcup\}$:



If the tape contains a string containing only a's and b's, this machine copies the string.

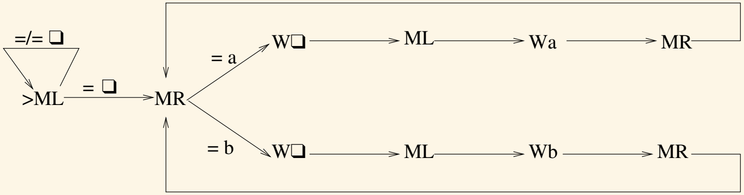
Consider initial configuration $\langle \sqcup ab, b, \sqcup \rangle$ (showing only the tape, ignoring the state). First the head is moved left until it reaches a blank: $\langle \sqcup, \sqcup, abb\sqcup \rangle$. Then it is moved



Module Home Page
Title Page
◀◀ ▶▶
◀ ▶
Page 8 of 9
Back
Full Screen
Close
Quit

right one, so we're now over the leftmost non-blank: $\langle \sqcup, a, bb\sqcup \rangle$. Since this is not a blank (of course), it is overwritten by a blank: $\langle \sqcup, \sqcup, bb\sqcup \rangle$. We scan right to the first blank, and then right again to the second blank: $\langle \sqcup bb\sqcup, \sqcup, \sqcup \rangle$. And we write an a: $\langle \sqcup bb\sqcup, a, \sqcup \rangle$. We then scan left until we reach a blank. And then we scan left again until we reach the next blank: $\langle \sqcup, \sqcup, bb\sqcup a\sqcup \rangle$. And the a is then rewritten: $\langle \sqcup, a, bb\sqcup a\sqcup \rangle$. Then we loop back to the machine that moves us right by one cell: $\langle \sqcup a, b, b\sqcup a\sqcup \rangle$. And now the process repeats: the b will be erased, written out at the second blank to the right, and rewritten in its original position. Then we move onto the next b. Ultimately, we move one cell right for the final time to obtain: $\langle \sqcup abb, \sqcup, abb\sqcup \rangle$. Since the symbol under the head is a blank, we do not take the transition from M_R. We halt.

- **Example 4.** Consider this Turing machine where $\Sigma = \{a, b, \sqcup\}$:



Homework: What does this machine do?

Acknowledgements

In preparing this material, I have used [Jun] and [LP81].

Clip Art (of head with bomb) licensed from the Clip Art Gallery on DiscoverySchool.com.

References

[Jun] A. Jung. Models of Computation (Course Notes).

[LP81] H.R. Lewis and C.H. Papadimitriou. *Elements of the Theory of Computation.* Prentice Hall, 1981.