Turing Machines

Özlem Çetinoğlu

CA215 - Languages and Computability

December 7th, 2009

Chomsky Hiearchy

Type	**Grammar**	**Language**	**Automata**
3	Finite State	Regular	Finite
2	Context-Free	C-F	Pushdown
1	Context-Sensitive	C-S	Linear-Bounded
0	General Rewrite	Unrestricted	Turing Machines

General Rewrite Grammars

Unrestricted languages are specified with a general rewrite grammar.

Formally, a general rewrite grammar G is a quadruple (T, N, S, P), where:

- T is the set of *terminal* symbols,
- N is the set of *nonterminals*,
- S is the *start symbol*, $S \in N$,
- P is a finite set of *productions* of the form $\alpha \rightarrow \beta$, where $\alpha, \beta \in (T \cup N)^*$ and α contains at least one nonterminal

Turing Machines

None of the automata which we have seen so far are general models of computation. *Turing machines* (`TM`) provide such a general model.

Unrestricted languages are recognized by a Turing Machine.

Turing machines extend the idea of linear bounded automata by having an **infinite** length of tape. The symbol $<$ is used to mark the leftmost bound of the tape beyond which the tape head cannot move and which cannot be overwritten.
There is no rightmost bound on the tape.

Turing Machines

As for LBA, Turing Machines will perform one of the actions $\mathcal{A} \in \{Y, N, L, R\}$ at each step, where:

- Y denotes "Yes", accept the input string
- N denotes "No", do not accept the input string
- L denotes "Left", move the read-write head one space to the left
- R denotes "Right", move the read-write head one space to the right

TM Definition

A Turing machine is a 5-tuple $M = (Q, \Sigma, \Gamma, q_0, \delta)$, where:

Q is a finite set of *states*,

Σ is an *input alphabet*,

Γ is a *tape alphabet*,

$q_0 \in Q$ is the *initial state*,

δ is the *transition function* $Q \times (\Gamma \cup \{<\}) \to Q \times (\Gamma \cup \{<\}) \times \mathcal{A}$.

If $((q, a), (q', b, action)) \in \delta$, then when in state q with a at the current read position on the tape, M may replace a with b on the tape, perform the specified *action*, and enter state q'.

The symbol "$\#$" is used to denote a blank tape square.

M accepts $w \in \Sigma^*$ iff it starts with configuration $(q_0, \underline{w}\#)$ and the action Y is taken.

TM Definition

Notation:

Configuration: $(s, aab\underline{a}a)$

Initial configuration: $(q_0, \underline{w}\#)$

Halted configuration: Configuration when action Y or N is performed.

Hanging configuration: No transition for the current state and current input symbol.

A *computation* is a sequence of configurations for some $n \geq 0$. Such a computation is of length n.

M is said to *halt* on input w iff $(q_0, \underline{w}\#)$ yields some halted configuration.

M is said to *hang* on input w if $(q_0, \underline{w}\#)$ yields some hanging configuration.

Turing Machine Example 1

$M_1 = (Q, \Sigma, \Gamma, q_0, \delta)$ where:

- $Q = \{s_0, s_1\}$
- $\Sigma = \{1\}$
- $\Gamma = \{1, \#\}$
- $q_0 = s_0$
- $$\delta = \begin{matrix} ((s_0, 1), (s_0, 1, R)) \\ ((s_0, \#), (s_1, 1, L)) \\ ((s_1, 1), (s_1, 1, Y)) \end{matrix}$$

This a Turing Machine that will add 1 to the input in the unary notation.

e.g., 4 is 1111 in the unary notation, add1(4)= 11111

Turing Machine Example 1

$M_1 = (Q, \Sigma, \Gamma, q_0, \delta)$ where:

- $Q = \{s_0, s_1\}$
- $\Sigma = \{1\}$
- $\Gamma = \{1, \#\}$
- $q_0 = s_0$
- $\delta = \begin{matrix} ((s_0, 1), (s_0, 1, R)) \\ ((s_0, \#), (s_1, 1, L)) \\ ((s_1, 1), (s_1, 1, Y)) \end{matrix}$

Example: 1111

$(s_0, \underline{1}111\#) \vdash (s_0, 1\underline{1}11\#) \vdash (s_0, 11\underline{1}1\#) \vdash (s_0, 111\underline{1}\#) \vdash (s_0, 1111\underline{\#}) \vdash$
$(s_1, 111\underline{1}1\#) \vdash$ halting with a yes

Turing Machine Example 2

$M_2 = (Q, \Sigma, \Gamma, q_0, \delta)$ where:

- $Q = \{s_0\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, \#\}$
- $q_0 = s_0$
- $\delta = \begin{array}{l} ((s_0, a), (s_0, a, L)) \\ ((s_0, b), (s_0, b, R)) \\ ((s_0, \#), (s_0, \#, Y)) \\ ((s_0, <), (s_0, <, R)) \end{array}$

Example: *aa*

$(s_0, \underline{a}a\#) \vdash (s_0, \underline{<}aa\#) \vdash (s_0, \underline{a}a\#) \vdash (s_0, \underline{<}aa\#) \vdash (s_0, \underline{a}a\#) \vdash \ldots$

Turing Machine Example 2

$M_2 = (Q, \Sigma, \Gamma, q_0, \delta)$ where:

- $Q = \{s_0\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, \#\}$
- $q_0 = s_0$
- $\delta = \begin{array}{l} ((s_0, a), (s_0, a, L)) \\ ((s_0, b), (s_0, b, R)) \\ ((s_0, \#), (s_0, \#, Y)) \\ ((s_0, <), (s_0, <, R)) \end{array}$

Example: *aa*

$(s_0, \underline{a}a\#) \vdash (s_0, \underline{<}aa\#) \vdash (s_0, \underline{a}a\#) \vdash (s_0, \underline{<}aa\#) \vdash (s_0, \underline{a}a\#) \vdash \ldots$

This is an example of a Turing machine which may not halt.

Turing Machine Example 3

$M_3 = (Q, \Sigma, \Gamma, q_0, \delta)$ where:

$Q = \{s_0, s_1, s_2, s_3, s_4\}$

$\Sigma = \{a\}$ $\Gamma = \{a, x, \#\}$ $q_0 = s_0$

$$\delta = \begin{array}{ll}
((s_0, a), (s_1, \#, R)) & ((s_3, a), (s_2, x, R)) \\
((s_0, x), (s_0, x, N)) & ((s_3, x), (s_3, x, R)) \\
((s_0, \#), (s_0, \#, N)) & ((s_3, \#), (s_3, \#, N)) \\
((s_1, a), (s_2, x, R)) & ((s_4, a), (s_4, a, L)) \\
((s_1, x), (s_1, x, R)) & ((s_4, x), (s_4, x, L)) \\
((s_1, \#), (s_1, \#, Y)) & ((s_4, \#), (s_1, \#, R)) \\
((s_2, a), (s_3, a, R)) & \\
((s_2, x), (s_2, x, R)) & \\
((s_2, \#), (s_4, \#, L)) &
\end{array}$$

This will accept all strings of a's whose length is a power of 2.

Turing Machine Example 3

$\delta =$

1. $((s_0, a), (s_1, \#, R))$
2. $((s_0, x), (s_0, x, N))$
3. $((s_0, \#), (s_0, \#, N))$
4. $((s_1, a), (s_2, x, R))$
5. $((s_1, x), (s_1, x, R))$
6. $((s_1, \#), (s_1, \#, Y))$
7. $((s_2, a), (s_3, a, R))$
8. $((s_2, x), (s_2, x, R))$
9. $((s_2, \#), (s_4, \#, L))$
10. $((s_3, a), (s_2, x, R))$
11. $((s_3, x), (s_3, x, R))$
12. $((s_3, \#), (s_3, \#, N))$
13. $((s_4, a), (s_4, a, L))$
14. $((s_4, x), (s_4, x, L))$
15. $((s_4, \#), (s_1, \#, R))$

Example: aa

$(s_0, \underline{a}a\#) \vdash (s_1, \#\underline{a}\#) \vdash (s_2, \#x\underline{\#}) \vdash (s_4, \#\underline{x}\#) \vdash (s_4, \underline{\#}x\#) \vdash$
$(s_1, \#\underline{x}\#) \vdash (s_1, \#x\underline{\#}) \vdash$ halting state with a yes

Turing Machine Example 3

$$\delta = \begin{array}{ll}
1.\ ((s_0, a), (s_1, \#, R)) & 10.\ ((s_3, a), (s_2, x, R)) \\
2.\ ((s_0, x), (s_0, x, N)) & 11.\ ((s_3, x), (s_3, x, R)) \\
3.\ ((s_0, \#), (s_0, \#, N)) & 12.\ ((s_3, \#), (s_3, \#, N)) \\
4.\ ((s_1, a), (s_2, x, R)) & 13.\ ((s_4, a), (s_4, a, L)) \\
5.\ ((s_1, x), (s_1, x, R)) & 14.\ ((s_4, x), (s_4, x, L)) \\
6.\ ((s_1, \#), (s_1, \#, Y)) & 15.\ ((s_4, \#), (s_1, \#, R)) \\
7.\ ((s_2, a), (s_3, a, R)) & \\
8.\ ((s_2, x), (s_2, x, R)) & \\
9.\ ((s_2, \#), (s_4, \#, L)) &
\end{array}$$

Example: *aaaa*

$(s_0, \underline{a}aaa\#) \vdash (s_1, \#\underline{a}aa\#) \vdash (s_2, \#x\underline{a}a\#) \vdash (s_3, \#xa\underline{a}\#) \vdash (s_2, \#xax\underline{\#}) \vdash$
$(s_4, \#xa\underline{x}\#) \vdash (s_4, \#x\underline{a}x\#) \vdash (s_4, \#\underline{x}ax\#) \vdash (s_4, \underline{\#}xax\#) \vdash$
$(s_1, \#\underline{x}ax\#) \vdash (s_1, \#x\underline{a}x\#) \vdash (s_2, \#xx\underline{x}\#) \vdash (s_2, \#xxx\underline{\#}) \vdash$
$(s_4, \#xx\underline{x}\#) \vdash^* (s_4, \underline{\#}xxx\#) \vdash (s_1, \#\underline{x}xx\#) \vdash^* (s_1, \#xxx\underline{\#}) \vdash$ halting state with a yes

Turing Machine Example 3

$\delta =$

1. $((s_0, a), (s_1, \#, R))$
2. $((s_0, x), (s_0, x, N))$
3. $((s_0, \#), (s_0, \#, N))$
4. $((s_1, a), (s_2, x, R))$
5. $((s_1, x), (s_1, x, R))$
6. $((s_1, \#), (s_1, \#, Y))$
7. $((s_2, a), (s_3, a, R))$
8. $((s_2, x), (s_2, x, R))$
9. $((s_2, \#), (s_4, \#, L))$
10. $((s_3, a), (s_2, x, R))$
11. $((s_3, x), (s_3, x, R))$
12. $((s_3, \#), (s_3, \#, N))$
13. $((s_4, a), (s_4, a, L))$
14. $((s_4, x), (s_4, x, L))$
15. $((s_4, \#), (s_1, \#, R))$

Example: *aaa*

$(s_0, \underline{a}aa\#) \vdash (s_1, \#\underline{a}a\#) \vdash (s_2, \#x\underline{a}\#) \vdash (s_3, \#xa\underline{\#}) \vdash$ halting state with a no