The current configuration means current state, tape contents, head location

- \textit{uqv}: \textit{q}: current state
  - \textit{uv}: current tape content
  - \textit{u}: left, \textit{v}: right

head: first of \textit{v}
Example of configuration 1

- $a, b, c \in \Gamma$, $u, v, \in \Gamma^*$ (i.e., strings from $\Gamma$)
- $q_i, q_j$: states
- if $\delta(q_i, b) = (q_j, c, L)$
  
  $uaq_i bv$ yields $uq_j acv$

- if $\delta(q_i, b) = (q_j, c, R)$
  
  $uaq_i bv$ yields $uacq_j v$
More about Configurations I

- start configuration: $q_0w$
- accepting configuration: $q_{accept}$
- rejecting configuration: $q_{reject}$
- A TM accepts $w$ if configurations $c_1 \cdots c_k$
  1. $c_1$: start configuration
  2. $c_i$ yields $c_{i+1}$
  3. $c_k$ accepting configuration
- Language: $L(M)$: strings accepted by $M$
A language is Turing-recognizable if it is recognized by a TM.

For a Turing machine, there are three possible outcomes:

- accept
- reject
- loop

If an input fails: reject or loop

This is difficult to decide.

We prefer a TM that never loops.

Deciders: only accept or reject.
A language is Turing-decidable if some TM decides it.

In Chapter 4 we will discuss decidable languages.
Consider the following language

\[ \{ w \# w \mid w \in \{0, 1\}^* \} \]

Fig 3.10
Example 3.9 II

0, 1 → R

q_2

# → R

q_4

x → R

q_7

0, 1 → L

q_6

# → L

0, 1, x → L

q_a

1 → x, L

q_5

x → R

q_3

# → R

q_8

1 → x, R

q_1

0 → x, R

q_2

# → R

q_4

x → R

q_7

0, 1 → L

q_6

# → L

0, 1, x → L

q_a

1 → x, L

q_5

x → R

q_3

# → R
Example 3.9

- Links to \(q_r\) are not shown
- Simulate 01#01

\[
\begin{align*}
q_1 & 01\#01 & xq_2 & 1\#01 & x1q_2 & \#01 & x1\#q_4 & 01 \\
x1 & q_6 \# x1 & xq_7 & 1\#x1 & q_7 & x1\#x1 & xq_1 & 1\#x1 \\
xxq_3 & \# x1 & xx & \# q_5 & x1 & xx & \# xq_5 & 1 & xx & \# q_6 & xx \\
xxq_6 & \# xx & xq_7 & x\# xx & xxq_1 & \# xx & xx & \# q_8 & xx \\
xx & \# xxq_8 & \sqcup & xx & \# xx & \sqcup & q_a
\end{align*}
\]
Example 3.9 IV

- Idea of the diagram:

\[ q_1 \rightarrow q_2 \rightarrow q_4 \rightarrow q_6 \]

Check 0 at the same position of the two strings

\[ q_1 \rightarrow q_3 \rightarrow q_5 \rightarrow q_6 \]

Check 1 at the same position of the two strings

- \( q_6 \): move left to the beginning of the second string
Example 3.9 V

- $q_7$: move left by

$$q_7 \xrightarrow{0,1 \rightarrow L} q_7$$

until finding the first 0, 1 not handled yet:

$$q_7 \xrightarrow{x \rightarrow R} q_1$$

- Thus $q_6$ and $q_7$ cannot be combined. At $q_6$,

$$x \rightarrow L$$

but at $q_7$

$$x \rightarrow R$$
Example 3.11

- \( C = \{ a^i b^j c^k \mid i \times j = k, i, j, k \geq 1 \} \)
- Procedure
  1. check if the input is \( a^+ b^+ c^+ \)
  2. back to start
  3. fix \( a \), for each \( b \), cancel \( c \)
  4. store \( b \) back, cancel one \( a \), go to step 3
- Too complicated to draw state diagram
- But one may wonder if TM can really do the above procedure
- Here are more details
Step 1 can be done by a DFA (as DFA is a special case of TM)
Step 2 can be done by using a special symbol in the beginning
Step 3 is similar to the procedure of handling $w \# w$
Now we see the concept of subroutines
Example 3.12

- $E = \{\#x_1\#x_2 \cdots \#x_l \mid x_i \in \{0, 1\}^*, x_i \neq x_j\}$
- Idea: sequentially compare every pair

$x_1x_2, x_1x_3, \ldots, x_1x_l$

$x_2x_3, \ldots, x_2x_l$

$x_{l-1}x_l$

- This description is rough. Let’s check more details

For $x_i, x_j$ mark ‘#’s of both strings by ‘.’

$\#x_1\#x_2\#x_3$: $x_1$ and $x_3$ being compared
Example 3.12 II

- Compare $x_i$ and $x_j$:
  - Can use a TM similar to that for $w \neq w$
  - We can copy $x_i, x_j$ to the end and do the comparison there