**Arab Academy for Science &Technology and Maritime Transport (AASTMT)**
**College of Computing and Information Technology (CCIT)**
**Theory of Computation CS311 – Spring 2014**

**Dr. Manal Helal**
**Eng. Nada Mahmoud**

**Section 10 – 18th of May 2014**

1. For $\sum$ : {a, b}, design a Turing machine that accepts:

$L = \{a^n b^n : n \geq 1\}$

Intuitively, we solve the problem in the following fashion. starting at the Leftmost a, we check it off by replacing it with some symbol, say x. We then let the read-write head travel right to final the leftmost b, which in turn is checked off by replacing it with another symbol, say y. After that, we go left again to the leftmost a, replace it with an x, then move to the leftmost b and replace it with y, and so on. Traveling back and forth this way' we match each a with a corresponding b. If after some time no a's or b's remain, then the string must be in L.

working out the details, we arrive at a complete solution for which:

Q: $\{q_0, q_1, q_2, q_3, q_4\}$

F: $\{q_4\}$

$\sum$:{ a,b},

$\Gamma$: {a, b, x, y, □}

The transitions can be broken into several parts. The set

$\delta(q_0, a) = (q_1, x, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, y) = (q_1, y, R)$

$\delta(q_1, b) = (q_2, y, L)$

Replaces the leftmost a with an x, then causes the read-write head to travel right to the first b, replacing it with a y. when the y is written, the machine enters state $q_2$, indicating that an a has been successfully paired with a b.

The next set of transitions reverses the direction until an x is encountered, repositions that read-write head over the leftmost a, and returns control to the initial state.

$\delta(q_2, y) = (q_2, y, L)$

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, x) = (q_0, x, R)$

We are now back in the initial state $q_0$, ready to deal with the next a and b. After one pass through this part of the computation, the machine will have carried out the partial computation:

$q_0aa \ldots abb \ldots b \vdash xq_1a \ldots ayb \ldots b$

So that a single a has been matched with a single b. After two passes, we will have completed the partial computation

$q_0aa \ldots abb \ldots b \vdash^* xxq_1 \ldots ayy \ldots b$

And so on, indicating that the matching process is being carried out properly.

When the input is a string $a^nb^n$, the rewriting continues this way, stopping only when there are no more a's to be erased. When looking for the leftmost a, the read-write head travels left with the machine in state $q_2$. When an x is encountered, the direction is reversed to get the a. But now, instead of finding an a it will find a y. To terminate, a final check is made to see if all a's and b's have been replaced (to detect input where an a follows a b). This can be done by:

$\delta(q_0, y) = (q_3, y, R)$

$\delta(q_3, y) = (q_3, y, R)$,

$\delta(q_3, \square) = (q_4, \square, R)$.

If we input a string not in the language, the computation will halt in a non-final state. For example, if we give the machine a string $a^nb^m$, with $n > m$, the machine will eventually encounter a blank in state $q_1$. It will halt because no transition is specified for this case. Other input not in the language will also lead to a non-final halting state.

The particular input "aabb" gives the following successive instantaneous descriptions:

$q_0aabb \vdash \quad xq_1abb \vdash xaq_1bb \vdash xq_2ayb \vdash$

$q_2xayb \vdash xq_0ayb \vdash xxq_1yb \vdash$

$xxyq_1b \vdash xxq_2yy \vdash xq_2xyy \vdash$

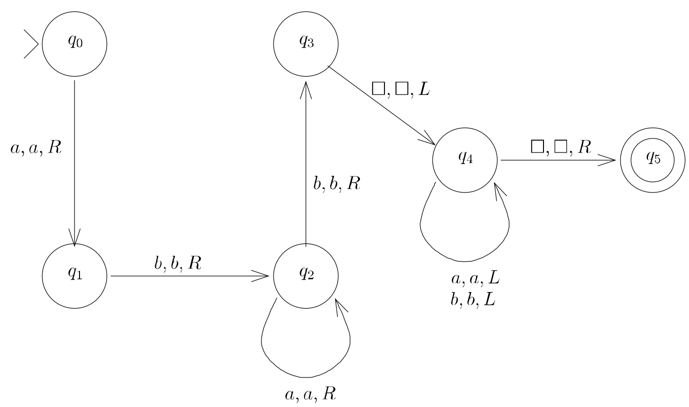
$xxq_0yy \vdash xxyq_3y \vdash xxyyq_3\square \vdash$

$xxyy\square q_4\square$.

At this point the Turing machine halts in a final state, so the string aabb is accepted.

Trace with "ab", and "aaabbb"

2. Consider the following Turing machine M:




(a) Give the computation path trace for initial configuration $q_0abab$.

(b) Give the computation path trace for initial configuration $q_0abba$.

(c) Give a regular expression for L(M).