ECS 120: Theory of Computation

Homework 7 Solution

Date: 5/24/06

[Problem 1.]

[Linz, Section 8.1, Exercise 5]

No. We prove this by the Pumping lemma for linear languages. Given m, let $w = a^{2^m} b^m$. w can be decomposed into $uvxyz$, where $|vxy| \leq m$ and $|vy| \geq 1$. Then the combination of v and y include:

- $vy = a^k$. Let $i = 2$, then $w_2 = a^{2^m+k} b^m$. Since $2^m < 2^m + k < 2^m + m < 2^{m+1}$, $w_1 \notin L$.
- $vy = b^k$. Let $i = 2$, then $w_2 = a^{2^m} b^{m+k}$. Since $2^{m+k} = 2^m \cdot 2^k > 2^m$, $w_1 \notin L$.
- $v = a^k$ and $y = b^l$. Let $i = 2$, then $w_2 = a^{2^m+k} b^{m+l}$. Since $2^m + k < 2^{m+1} \leq 2^{m+l}$, $w_2 \notin L$.
- Either v or y contains both a's and b's. Then by pumping up yields a string with the wrong sequence.

[Linz, Section 8.1, Exercise 7(j)]

No. We prove this by the pumping lemma for CFL. Given m, let $w = a^p b^0$ such that $p \geq m$ and p is prime. w can be decomposed into $uvxyz$, where $|vxy| \leq m$ and $vy \neq \lambda$. Suppose $|vy| = k$, then $w_i = a^{p+(i-1)k} b^0$. Let $i = p + 1$, then $w_{p+1} = a^{p+pk} b^0 = a^{p(1+k)} b^0 \notin L$.

[Linz, Section 8.1, Exercise 7(k)]

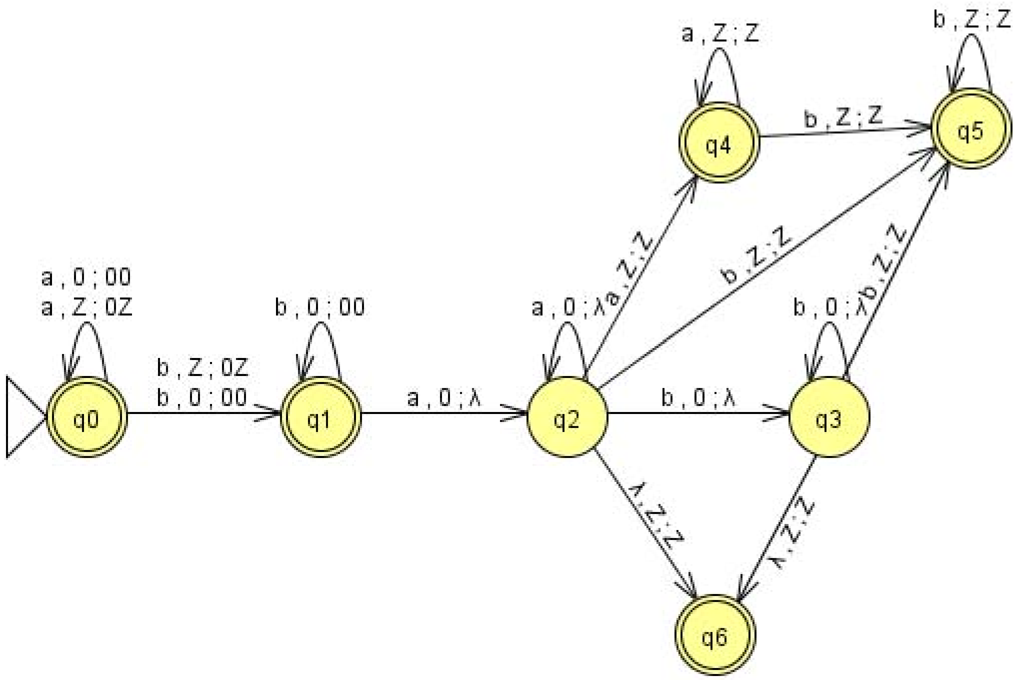
No. We prove this by the pumping lemma for CFL. Given m, let $w = a^p b^0$ such that $p \geq m$ and p is prime. w can be decomposed into $uvxyz$, where $|vxy| \leq m$ and $vy \neq \lambda$. Suppose $|vy| = k$, then $w_i = a^{p+(i-1)k} b^0$. Let $i = p + 1$, then $w_{p+1} = a^{p+pk} b^0 = a^{p(1+k)} b^0 \notin L$.

[Linz, Section 8.1, Exercise 8(c)]

L is context-free and can be accepted by the following context-free grammar:
$S \rightarrow aSb|A$
$A \rightarrow bAa|\lambda$

[Linz, Section 8.1, Exercise 8(d)]

L is context-free and can be accepted by the following push-down automaton.



[Linz, Section 8.1, Exercise 8(f)]

No.We prove by the pumping lemma for CFL. Given m, let $w = a^m b^m c^m$. w can be decomposed into $uvxyz$, where $|vxy| \leq m$ and $|vy| \geq 1$. Then the combination of v and y include:

- $vy = a^k$ or $vy = b^k$. Then by pumping up, $n_a(w_i) \neq n_b(w_i)$.
- $vy = c^k$. Then by pumping down, $n_a(w_0), n_b(w_0) > n_c(w_0)$.
- $v = a^k$ and $y = b^l$. Let $i = 2$, then $w_2 = a^{m+k} b^{m+l} c^m$. If $k \neq l$, then $n_a(w_0) \neq n_b(w_0)$. If $k = l$, then by pumping up $n_a(w_2), n_b(w_2) > n_c(w_2)$.
- $v = b^k$ and $y = c^l$. Let $i = 2$, then $w_2 = a^m b^{m+k} c^{m+l}$. Then by pumping up, $n_a(w_2) \neq n_b(w_2)$.
- v contains both a's and b's or y contains both b's and c's. Then by pumping up yields a string with the wrong sequence.

[Linz, Section 8.2, Exercise 10]

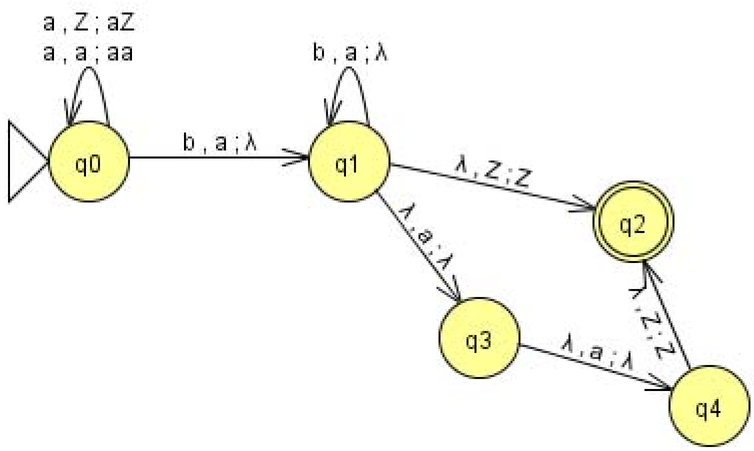
Here's a counter example: $L(a^n b^n c^m) \cap L(a^m b^n c^n) = L(a^n b^n c^n)$, where $n, m \geq 0$. $L(a^n b^n c^n)$ is not context-free (proved in Example 8.1, p.207), yet $L(a^n b^n c^m)$ and $L(a^m b^n c^n)$ are both context-free:

$L(a^n b^n c^m)$ can be accepted by the following grammer:
$S \rightarrow Sc|\lambda|A$

$A \rightarrow aAb|\lambda$

$L(a^m b^n c^n)$ can be accepted by the following grammer:
$S \rightarrow aS|\lambda|A$
$A \rightarrow bAc|\lambda$

[Problem 2.]



[Problem 3.]