Languages and Machines

Chap 7: (p 247-249): 1abcd, 3acdj, 14 (pp 249-250): 17acd, 19abc

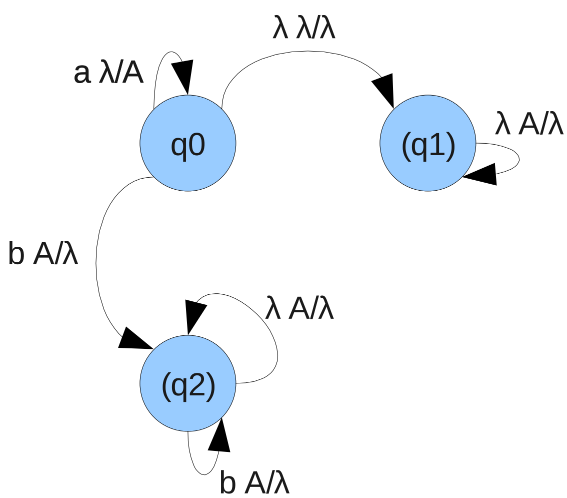
- Find a PDA that accepts all strings of a's and b's that are not of the form ww.
- Devise a Deterministic Pushdown Automaton that accepts strings with equally many a's and b's. Convert it to a context-free grammar.
- Prove that for any tree of depth d for which each node has at most n children, the number of leaves is at most n^d.

1

(a)

n a's followed by m b's where n<=m

(b)



(c)

[q0, aab, .\] |- [q0, ab, A] |- [q0, b, AA] |- [q2, .\, A] |- [q2, .\, .\]	[q0, abb, .\] |- [q0, bb, A] |- [q2, b, .\] |- fail	[q0, aba, .\] |- [q0, ba, A] |- [q2, a, .\] |- fail

(d)

[q0, aabb, .\] |- [q0, abb, A] |- [q0, bb, AA] |- [q2, b, A] |- [q2, .\, .\]	[q0, aaab, .\] |- [q0, aab, A] |- [q0, ab, AA] |- [q0, b, AAA] |- [q2, .\, AA] |- [q2, .\, A] |- [q2, .\, .\]

3.
(a)
Q = {q0, q1, q2}
F = {q0, q2}
d(q0, b, .\) = {[q2, .\]}
d(q0, a, .\) = {[q1, B]}
d(q1, b, B) = {[q2, .\]}
d(q1, a, .\) = {[q1, B]}
d(q2, b, .\) = {[q2, .\]}
d(q2, b, B) = {[q2, .\]}

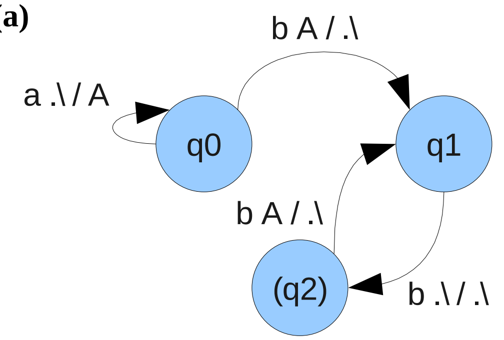
(c)
Q = {q0, q1, q2, q3}
F = {q0, q3}
d(q0, a, .\) = {[q1, I]}
d(q1, a, .\) = {[q1, I]}
d(q1, b, I) = {[q2, .\]}
d(q2, b, I) = {[q2, .\]}
d(q2, b, .\) = {[q2, J]}
d(q2, .\, .\) = {[q3, .\]}
d(q2, c, J) = {[q3, .\]}
d(q3, c, J) = {[q3, .\]}

(d)
Q = {q0, q1}
F = {q0}
d(q0, a, A) = {[q0, .\]}
d(q0, b, .\) = {[q1, A]}
d(q1, .\, .\) = {[q0, A]}

(j)
Q = {q0, q1}
F = {q0 ,q1}
d(q0, a, .\) = {[q0, A]}
d(q0, b, .\) = {[q0, B]}
d(q0, a, .\) = {[q1, .\]}
d(q0, b, .\) = {[q1, .\]}
d(q0, .\, .\) = {[q1, .\]}
d(q1, a, A) = {[q1, .\]}
d(q1, b, B) = {[q1, .\]}

14.

(a)



(b)

$\{a^n(bb)^n \mid n >= 0\}$

(c)

M' = M U
d(q0, a, A) = {[q0, AA]} -- pushes an A
d(q1, b, A) = {[q2, A]} -- does nothing to stack

Transition	**Rule**
	S -> <q0, .\, q2>
d(q0, a, .\) = {[q0, A]}	<q0, .\, q0> -> a<q0, A, q0> <q0, .\, q1> -> a<q0, A, q1> <q0, .\, q2> -> a<q0, A, q2>
d(q0, a, A) = {[q0, AA]}	<q0, A, q0> -> a<q0, A, q0><q0, A, q0> <q0, A, q0> -> a<q0, A, q1><q1, A, q0> <q0, A, q1> -> a<q0, A, q0><q0, A, q1> <q0, A, q1> -> a<q0, A, q1><q1, A, q1> <q0, A, q0> -> a<q0, A, q2><q2, A, q0> <q0, A, q2> -> a<q0, A, q0><q0, A, q2> <q0, A, q1> -> a<q0, A, q2><q2, A, q1> <q0, A, q2> -> a<q0, A, q1><q1, A, q2> <q0, A, q2> -> a<q0, A, q2><q2, A, q2>
d(q0, b, A) = {[q1, .\]}	<q0, A, q0> -> b<q1, .\, q0> <q0, A, q1> -> b<q1, .\, q1> <q0, A, q2> -> b<q1, .\, q2>
d(q1, b, .\) = {[q2, .\]}	<q1, .\, q0> -> b<q2, .\, q0> <q1, .\, q1> -> b<q2, .\, q1> <q1, .\, q2> -> b<q2, .\, q2>
d(q1, b, A) = {[q2, A]}	<q1, A, q0> -> b<q2, A, q0> <q1, A, q1> -> b<q2, A, q1> <q1, A, q2> -> b<q2, A, q2>
d(q2, b, A) = {[q1, .\]}	<q2, A, q0> -> b<q1, .\, q0> <q2, A, q1> -> b<q1, .\, q1> <q2, A, q2> -> b<q1, .\, q2>
	<q0, .\, q0> -> .\ <q1, .\, q1> -> .\ <q2, .\, q2> -> .\

(d)
[q0, aabbbb, .\] |- [q0, abbbb, A] |- [q0, bbbb, AA] |- [q1, bbb, A] |- [q2, bb, A] |- [q1, b, .\] |- [q0, .\, .\]
success

(e)
S => <q0, .\, q2>
=> a<q0, A, q2>
=> aa<q0, A, q2><q2, A, q2>
=> aabb<q2, .\, q2><q2, A, q2>
=> aabb<q2, A, q2>
=> aabbbb<q2, .\, q2>
=> aabbbb

17.

(a)

L = {a^i | i is a perfect square}

Assume L is context free. By theorem 7.4.1, the string z = $a^{k\wedge 2}$, where k is the number specified by the pumping lemma, can be decomposed into substrings *uvwxy* that satisfy the repetition properties.

length(uv^2wx^2y) =
length(uvwxy) + length(vx) =
k^2 + length(vx) <
$k^2 + 2k + 1$ = [because length(vwx) <= k]
$(k+1)^2$

Therefore, on pumping v and x once, we find that the resulting string has length greater than k^2 but less than $(k+1)^2$, and so is not a perfect square.

(c)

L = {$a^ib^{2i}a^i$ | i>=0}

Assume L is context free. By theorem 7.4.1, the string z = $a^kb^{2k}a^k$, where k is the number specified by the pumping lemma, can be decomposed into substrings uvwxy that satisfy the repetition properties.

Consider the possibilities for the substrings v and x:

i) vwx `elem` a* or vwx `elem` b*

ii) vwx `elem` a*b* or vwx `elem` b*a*

In case (ii), pumping v and x would cause the ordering to be broken.

In case (i), pumping v and x would cause only one or two substrings to increase in length, rather than all three.

(d)

L = {$a^ib^jc^k$ | 0 < i < j < k < zi }

Assume L is context free. By theorem 7.4.1, the string z = $a^kb^xc^y$ where k is the number specified by the pumping lemma, can be decomposed into substrings uvwxy that satisfy the repetition properties.

Consider the possibilities for the substrings v and x:

v and x must contain only a's, because that is the only possible decomposition when the beginning of the string is a^k and the length of vwx <= k. Pumping a's will eventually cause them to exceed the b's, which is not allowed.

19.

(a) Since L_1 is the concatenation of {a^ib^i} and {c^jd^j}, both of which are context free as proved in the examples, L_1 is context free.

(b) Since L_2 is the concatenation of the context free languages {a^j}, {b^ic^i}, and {d^k}, then L_2 is also context free.

(c) By theorem 7.5.2, since L_1 and L_2 are themselves both context free, their intersection is not.