FORMAL LANGUAGES, AUTOMATA AND THEORY OF COMPUTATION

EXERCISES IN CONTEXT-FREE LANGUAGES

2010

CONTEXT FREE LANGUAGES & PUSH-DOWN AUTOMATA

CONTEXT-FREE GRAMMARS, CFG

Problems Sudkamp

Problem 1. (3.2.1)
Which language generates the grammar G given by the productions

$$S \rightarrow aSa \mid aBa$$
$$B \rightarrow bB \mid b$$

Problem 2. (3.2.2)
Find a CFG that generates the language:

$$L(G) = \{ a^n b^m c^m d^{2n} \mid n \geq 0,\ m > 0 \}.$$

Problem 3. (3.2.4)
Find a CFG that generates the language

$$L(G) = \{ a^n b^m \mid 0 \leq n \leq m \leq 2n \}.$$

Problem 4. (3.2.5)
Consider the grammar

$$S \rightarrow abScB \mid \lambda$$
$$B \rightarrow bB \mid b$$

What language does it generate?

Problems Lewis-Papadimitriou

Construct context free grammars to accept the following languages.

Problem 5. (2.4b) {w | w starts and ends with the same symbol}

Problem 6. (2.4c) {w | |w| is odd}

Problem 7. (2.4d) $\{w \mid |w| \text{ is odd and its middle symbol is } 0\}$

Problem 8. $\{w\#x \mid w^R \text{ is a substring of } x, \text{ where } w, x \in \{a, b\}^*\}$

Problem 9. $\{0^n1^n \mid n>0\} \cup \{0^n1^{2n} \mid n>0\}$

Problem 10. $\{0^i1^j2^k \mid i\neq j \text{ or } j\neq k\}$

Problem 11. Binary strings with twice as many 1s as 0s.

Problems Linz

Problem 12. (134/8c): Find a Context-Free Grammar for the following language:

$$L = \{a^nb^mc^k : k = n + m \}$$

Problem 13. (134/8c):

$$L = \{a^nb^mc^k : k \neq n + m \} \text{ (Variation on a theme).}$$

Ambiguity

Problem 14. Explain why the grammar below is ambiguous.

$S \to 0A \mid 1B$
$A \to 0AA \mid 1S \mid 1$
$B \to 1BB \mid 0S \mid 0$

Problem 15. Given the following ambiguous context free grammar

$S \to Ab \mid aaB$
$A \to a \mid Aa$
$B \to b$

(a) Find the string s generated by the grammar that has two leftmost derivations. Show the derivations.
(b) Show the two derivation trees for the string s.
(c) Find an equivalent unambiguous context-free grammar.
(d) Give the unique leftmost derivation and derivation tree for the string s generated from the unambiguous grammar above.

PUSH-DOWN AUTOMATA, PDA

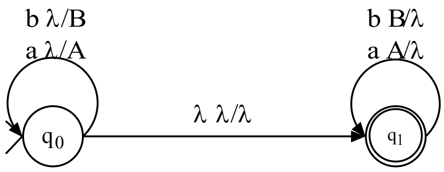
Sudkamp

Problem 1. CH8 8.1.3

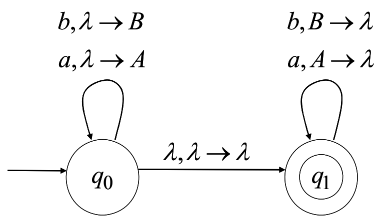
Let M be the PDA that accepts even-length palindromes over {a, b}. That is,

$$L(M) = \{ w\, w^R \mid w \in \{a, b\}^* \}$$

Where non-determinism allows the machine to "guess" when the middle of the string has been reached.



which in Linz notation looks like this:



(a, b/c stands for a, b→c)

a) Give the transition table of M.

b) Trace all computation of the strings *ab, abb, abbbb* in M.

c) Show that *aaaa, baab* $\in$ L (M).

d) Show that aaa, ab $\notin$ L (M).

Problem 2. CH8 3k

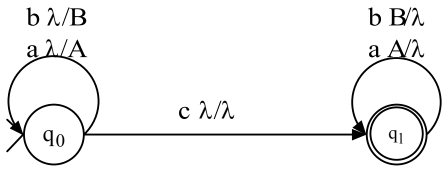
Construct a PDA that accepts the set of palindromes over $\{a, b\}$.

Problem 3. CH8 15a

Let M be the PDA in example 8.1.1. the PDA that accepts the language

$$L = \{wcw^R \mid w \in \{a,b\}^*\}.$$

The stack is used to record the string w as it is processed. Stack symbols A and B represent input a and b respectively.



A successful computation records the string w on the stack as it is processed. Once the c is encountered, the accepting state q_1 is entered and the stack contains a string representing w^R. The computation is completed by matching the remaining input with the elements of the stack.

Trace the computation that accepts *bbcbb.*

NPDA EXAMPLES

Sipser

Construct non-deterministic pushdown automata to accept the following languages. (Note: ε has the same meaning as λ)

Example 1. $\{1^n0^n \mid n > 0\}$

Example 2. $\{0^n1^{2n} \mid n \geq 0\}$

Example 3. $\{1^n0^n \mid n>0\}$ U $\{0^n1^{2n} \mid n \geq 0\}$

Example 4. $(0+1)^*$ - $\{ww \mid w \text{ in } \{0,1\}^*\}$ (complement of ww)

Linz

Example 5. $L = \{a^nb^{2n}: n \geq 0\}$

Example 6. $L = \{\omega\sigma\omega^R: \omega \in \{a,b\}^*, \sigma \in \{a,b\}\}$

Example 7. Construct the NPDA corresponding to the grammar

$S \rightarrow aABB \mid aAA$
$A \rightarrow aBB \mid a$
$B \rightarrow bBB \mid A$

(Problem 5 on page 194, Linz)

DPDA EXAMPLES

Sipser

Construct deterministic pushdown automata to accept the following languages.

Example 8. $\{10^n1^n \mid n>0\}$ U $\{110^n1^{2n} \mid n>0\}$

Example 9. Binary strings that contain an equal number of 1s and 0s.

Example 10. Binary strings with twice as many 1s as 0s.

Example 11. Binary strings that start and end with the same symbol and have the same number of 0s as 1s.

Linz

Example 12. Show that $L = \{a^n b^m : m \geq n + 2\}$ is deterministic.

(Problem 2 on page 199.)

Problems Linz

Problem 16. **(134/8c):** Find a Context-Free Grammar for the following language:

$$L = \{a^n b^m c^k : k = n + m\}$$

Problem 17. **(134/8c):**

$$L = \{a^n b^m c^k : k \neq n + m\} \text{ (Variation on a theme).}$$

Ambiguity

Problem 18. Explain why the grammar below is ambiguous.

$S \to 0A \mid 1B$
$A \to 0AA \mid 1S \mid 1$
$B \to 1BB \mid 0S \mid 0$

PUMPING LEMMA: CONTEXT-FREE OR NOT?

Problem 1. Show that the following language on $\Sigma = \{a, b, c\}$ is not context-free.
$L = \{a^n b^j c^k : k = jn\}$.

Problem 2. Show that the following language on $\Sigma = \{a, b\}$ is not context-free:
$L = \{ ww^R a^{|w|} : w \in \{a, b\}^* \}$

Problem 3. (CH8 17c Sudkamp) Using the pumping lemma prove that the language below is not context-free.

$$L = \{a^i b^{2i} a^i \mid i \geq 0\}$$

Problem 4. (CH8 18a Sudkamp) Prove that the language

$L = \{a^i b^{2i} c^j \mid i, j \geq 0\}$ is context-free.

Problem 5. (17 page 220, Linz) We want to prove that the language

$L = \{a^n b^n : n \geq 0 \text{ and } n \text{ is not a multiple of } 5\}$

is a context-free language.

Problem 6. $\{1^k 0^i 1^i 0^j 1^j 0^k \mid i, j, k > 0\}$

Problem 7. $\{w\#x \mid w$ is a substring of x, where w, x are in $\{0,1\}^*\}$.

Problem 8. $\{0^i 1^i 0^j 1^i \mid i, j > 0\}$

Problem 9. The complement of $\{(0^n 1^n)^m \mid m, n > 0\}$.

Problem 10. 2.18 a. (Lewis-Papadimitriou): $A = \{0^n 1^n 0^n 1^n \mid n >= 0\}$.

Problem 11. 2.18 b (Lewis-Papadimitriou): $B = \{0^n \# 0^{2n} \# 0^{3n} \mid n >= 0\}$.

Problem 12. 2.18 c (Lewis-Papadimitriou):

$C = \{ w\#x \mid w$ is a substring of x, where $w, x \in \{a, b\}^*\}$.

SOLUTIONS

CONTEXT-FREE GRAMMARS, CFG

Sudkamp examples

Problem 1. **3.2.1**
Which language generates the grammar G given by the productions

$$S \to aSa \mid aBa$$
$$B \to bB \mid b$$

Solution

$L(G) = \{ a^n b^m a^n \mid n > 0, m > 0\}.$

The rule $S \to aSa$ recursively builds an equal number of a's on each end of the string. The recursion is terminated by the application of the rule $S \to aBa$, ensuring at least one leading and one trailing a. The recursive B rule then generates any number of b's. To remove the variable B from the string and obtain a sentence of the language, the rule $B \to b$ must be applied, forcing the presence of at least one b.

Problem 2. **3.2.2**
Find a CFG that generates the language

$L(G) = \{ a^n b^m c^m d^{2n} \mid n \geq 0, m > 0\}.$

Solution

The relationship between the number of leading a's and trailing d's in the language indicates that the recursive rule is needed to generate them. The same is true for b's and c's. Derivations in the grammar

$$S \to aSdd \mid A$$
$$A \to bAc \mid bc$$

generate strings in an outside-to-inside manner. The S rules produce the *a's* and *d's* while the A rules generate *b's* and *c's*. The rule $A \to bc$, whose application terminates the recursion, ensures the presence of the substring *bc* in every string in the language.

Problem 3. **3.2.4**

Find a CFG that generates the language

$L(G) = \{ a^n b^m \mid 0 \le n \le m \le 2n\}$.

Solution

$$S \to aSb \mid aSbb \mid \lambda$$

The first recursive rule of G generates a trailing *b* for every *a*, while the second generates two *b's* for each *a*. Thus there is at least one b for every *a* and at most two, as specified in the language.

Problem 4. **3.2.5**

Consider the grammar

$$S \to abScB \mid \lambda$$
$$B \to bB \mid b$$

What language does it generate?

Solution

The recursive *S* rule generates an equal number of *ab's* and *cB's*. The *B* rules generate b^+. In a derivation, each occurrence of B may produce a different number of b's. For example in the derivation

$$\begin{aligned} S &\Rightarrow abScB \\ &\Rightarrow ababScBcB \\ &\Rightarrow ababcBcB \\ &\Rightarrow ababcbcB \\ &\Rightarrow ababcbcbB \\ &\Rightarrow ababcbcbb, \end{aligned}$$

the first occurrence of *B* generates a single *b* and the second occurrence produces *bb*. The language of the grammar consists of the set $L(G) = \{ (ab)^n (cb^m)^n \mid n \ge 0,\ m > 0\}$.

Problems Lewis-Papadimitriou

Construct context free grammars to accept the following languages. $\Sigma = \{0, 1\}$

Problem 5. (2.4b) $\{w \mid w$ starts and ends with the same symbol$\}$

Solution

$$S \rightarrow 0A0 \mid 1A1$$
$$A \rightarrow 0A \mid 1A \mid \lambda$$

Problem 6. (2.4c) $\{w \mid |w|$ is odd$\}$

Solution

$$S \rightarrow 0A \mid 1$$
$$A \rightarrow 0S \mid 1S \mid \lambda$$

Problem 7. (2.4d) $\{w \mid |w|$ is odd and its middle symbol is $0\}$

Solution

$$S \rightarrow 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1$$

Problem 8. $\{w\#x \mid w^R$ is a substring of x, where $w, x \in \{a, b\}^*\}$

Solution

The following grammar generates language L, where S is the start variable.

$$S \rightarrow AT$$
$$A \rightarrow aAa \mid bAb \mid \#T$$
$$T \rightarrow aT \mid bT \mid \lambda$$

The strings in the language have the form $\mathbf{w\#uw^Rv}$, where u and v are strings of the form (a + b)* (any string made from symbols a and b). The variable T generates the strings u and v, while variable A generates the string $\mathbf{w\#uw^R}$ and the variable S generates the desired string $\mathbf{w\#uw^Rv}$.

Problem 9. $\{0^n1^n \mid n>0\} \cup \{0^n1^{2n} \mid n>0\}$

Solution

$S \rightarrow 0A1 \mid 0B11$
$A \rightarrow 0A1 \mid \lambda$
$B \rightarrow 0B11 \mid \lambda$

Problem 10. $\{0^i1^j2^k \mid i \neq j \text{ or } j \neq k\}$

Solution

$S \rightarrow AC \mid BC \mid DE \mid DF$
$A \rightarrow 0 \mid 0A \mid 0A1$
$B \rightarrow 1 \mid B1 \mid 0B1$
$C \rightarrow 2 \mid 2C$
$D \rightarrow 0 \mid 0D$
$E \rightarrow 1 \mid 1E \mid 1E2$
$F \rightarrow 2 \mid F2 \mid 1F2$

Problem 11. Binary strings with twice as many 1s as 0s.

Solution

$S \rightarrow \lambda \mid 0S1S1S \mid 1S0S1S \mid 1S1S0S$

Problem 12. Find a Context-Free Grammar for the following language:

$$L = \{a^n b^m c^k : k = n + m\}$$

Solution

Let G be the grammar with productions:

$S \to aSc \mid B$

$B \to bBc \mid \lambda$

Claim: $L(G) = L$

Proof:

Consider the following derivation:

$S \to^* a^n S c^n \to a^n B c^n \to^* a^n b^m B c^m c^n \to a^n b^m c^{(n+m)}$

(where the first $\to^*$ applies $S \to aSc$ n times, the second $B \to bBc$ m times)

Since all words in L(G) must follow this pattern in their derivations, it is clear that $L(G) \subseteq L$

Consider $w \in L$, $w = a^n b^m c^{(n+m)}$ for some $n, m \geq 0$

The derivation $S \to^* a^n S c^n \to a^n B c^n \to^* a^n b^m B c^m c^n \to a^n b^m c^{(n+m)}$
clearly produces w for any n, m.

$\therefore L \subseteq L(G)$

$\therefore L = L(G)$

$\therefore$ G is a CFG for L

Problem 13. $L = \{a^n b^m c^k : k \neq n + m\}$ (Variation on a theme).

Solution

Let G be the grammar with productions:

$S \to EcC \mid aAE \mid AU$

$A \to aA \mid \lambda$

$B \to bB \mid \lambda$

$C \to cC \mid \lambda$

$E \to aEc \mid F$

$F \rightarrow bFc \mid \lambda$

$U \rightarrow aUc \mid V$

$V \rightarrow bVc \mid bB$

Note: $L(E) = \{a^n b^m c^k : k = n + m\}$ (from 4(a) in Linz book)

Let $L1 = \{a^n b^m c^k : k < n + m\}$, $L2 = \{a^n b^m c^k : k > n + m\}$

Claim: $L(G) = L = L1 \cup L2$

Proof:

Consider the leftmost productions following $S \rightarrow EcC$

$S \rightarrow EcC \rightarrow * a^n b^m c^{(n+m)} cC \rightarrow * a^n b^m c^{(n+m)} cc^k C \rightarrow a^n b^m c^{(n+m)} cc^k = a^n b^m c^{(n+m+k+1)}$

(where the first $\rightarrow *$ follows from 4(a), the second from k repetitions of { $C \rightarrow cC$ })

Since $(n+m) < (n+m+k+1)$, $L(EcC) \subseteq L2 \subset L$

Consider the leftmost productions following $S \rightarrow aAE$

$S \rightarrow aAE \rightarrow * aa^k AE \rightarrow aa^k E \rightarrow * aa^k a^n b^m c^{(n+m)} c = a^{(n+k+1)} b^m c^{(n+m)}$

(where the first $\rightarrow *$ follows from k repetitions of { $A \rightarrow aA$ }, the second from 4(a))

Since $(n+m+k+1) > (n+m)$, $L(aAE) \subseteq L1 \subset L$

Consider the leftmost productions following $S \rightarrow AU$

$S \rightarrow AU \rightarrow * a^k AU \rightarrow a^k U \rightarrow * a^k a^n U c^n \rightarrow a^k a^n V c^n \rightarrow * a^k a^n b^m V c^m c^n \rightarrow a^k a^n b^m bBc^m c^n$

$\rightarrow * a^k a^n b^m bb^j Bc^m c^n \rightarrow a^k a^n b^m bb^j c^m c^n = a^{(k+n)} b^{(m+j+1)} c^{(m+n)}$

(where the first, second, third and fourth $\rightarrow *$ are k, n, m, and j repetitions of

{ $A \rightarrow aA$ }, { $U \rightarrow aUc$ }, { $V \rightarrow bVc$ }, and { $B \rightarrow bB$ }, respectively)

Since $((k+n)+(m+j+1)) > (m+n)$, $L(AU) \subseteq L1 \subset L$

$\therefore$ $L(G) = L(S) = L(EcC) \cup L(aAE) \cup L(AU) \subseteq L1 \cup L2 = L$

Consider $w \in L1$, $w = a^n b^m c^{(n+m+k)}$ for some n, m, k, with $k > 0$

Then w has the leftmost derivation:

$S \rightarrow EcC \rightarrow^* a^n b^m c^{(n+m)} cC \rightarrow^* a^n b^m c^{(n+m)} cc^{(k-1)} C \rightarrow a^n b^m c^{(n+m)} cc^{k(k-1)} = a^n b^m c^{(n+m+k)} = w$

$\therefore$ $L2 \subseteq L(G)$

Consider $w \in L2$, $w = a^{(n+j)} b^{(m+k)} c^{(n+m)}$ for some j, k, m, n, with $j+k > 0$

If $k = 0$, w has the leftmost derivation:

$S \rightarrow aAE \rightarrow^* aa^{(j-1)}AE \rightarrow aa^{(j-1)}E \rightarrow^* aa^{(j-1)}a^n Ec^n \rightarrow aa^{(j-1)}a^n Fc^n \rightarrow^* aa^{(j-1)}a^n b^m Fc^m c^n$

$\rightarrow aa^{(j-1)}a^n b^m c^m c^n = a^{(j+n)} b^{(0+m)} c^{(m+n)} = w$

$\therefore$ $L1\ (k = 0) \subseteq L(G)$

If $k > 0$, then w has the leftmost derivation:

$S \rightarrow AU \rightarrow^* a^j AU \rightarrow a^j U \rightarrow^* a^j a^n Uc^n \rightarrow a^j a^n Vc^n \rightarrow^* a^j a^n b^m Vc^m c^n \rightarrow a^j a^n b^m bBc^m c^n$

$\rightarrow a^j a^n b^m bb^{(k-1)} Bc^m c^n \rightarrow a^j a^n b^m bb^{(k-1)} c^m c^n = a^{(j+n)} b^{(m+k)} c^{(m+n)} = w$

$\therefore$ $L1\ (k > 0) \subseteq L(G)$

$\therefore$ $L1 \subseteq L(G)$

$\therefore$ $L = L1 \cup L2 \subseteq L(G)$

$\therefore$ $L = L(G)$

Ambiguity

Problem 14. Explain why the grammar below is ambiguous.

$S \rightarrow 0A \mid 1B$
$A \rightarrow 0AA \mid 1S \mid 1$
$B \rightarrow 1BB \mid 0S \mid 0$

Solution

The grammar is ambiguous because we can find strings which have multiple derivations:

$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 001S1 \Rightarrow 0011B1 \Rightarrow 001101$$
$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 0011S \Rightarrow 00110A \Rightarrow 001101$$

Problem 15. Given the following ambiguous context free grammar

$S \rightarrow Ab \mid aaB$
$A \rightarrow a \mid Aa$
$B \rightarrow b$

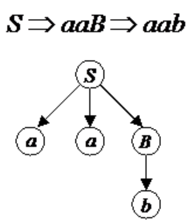
Solution

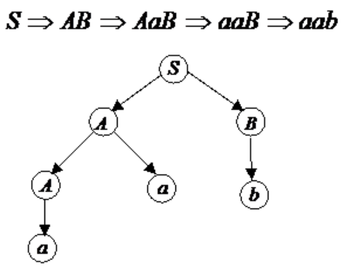
(a) Find the string s generated by the grammar that has two leftmost derivations. Show the derivations.

The string $s = aab$ has the following two leftmost derivations

$$S \Rightarrow aaB \Rightarrow aab$$
$$S \Rightarrow AB \Rightarrow AaB \Rightarrow aaB \Rightarrow aab$$

(b) Show the two derivation trees for the string s.
The two derivation trees of string *aab* are shown below.





(c) Find an equivalent unambiguous context-free grammar.
The equivalent unambiguous grammar is following

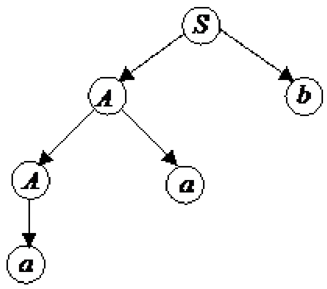
$$S \rightarrow Ab$$
$$A \rightarrow a \mid Aa$$

This grammar is not ambiguous because at any derivation step there is only one choice to make. This grammar is equivalent to the previous one, because both grammars generate the same language: all the strings that start with one or more a, and end with a single b.

(d) Give the unique leftmost derivation and derivation tree for the string s generated from the unambiguous grammar above.

With the new grammar the unique leftmost derivation and derivation tree of the string *aab* are shown below.

$$S \Rightarrow Ab \Rightarrow Aab \Rightarrow aab$$



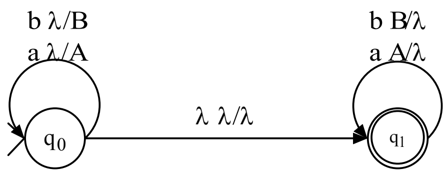
PUSH-DOWN AUTOMATA, PDA

Problem 1. CH8 8.1.3 Sudkamp

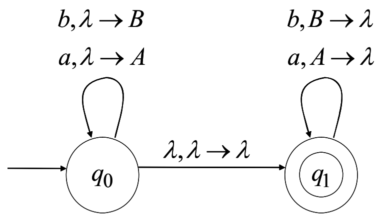
Let M be the PDA that accepts even-length palindromes over {a, b}. That is,

$$L(M) = \{ w\, w^R \mid w \in \{a, b\}^* \}$$

Where non-determinism allows the machine to "guess" when the middle of the string has been reached.



which in Linz notation looks like this:



(a, b/c stands for a, b→c)

a) Give the transition table of M.

b) Trace all computation of the strings *ab, abb, abbbb* in M.

c) Show that *aaaa, baab* $\in$ L (M).

d) Show that aaa, ab $\notin$ L (M).

Solution:

a) Give the transition table of M.

$$\delta(q_0, a, \lambda) = \{[q_0, A]\}$$
$$\delta(q_0, b, \lambda) = \{[q_0, B]\}$$
$$\delta(q_0, \lambda, \lambda) = \{[q_1, \lambda]\}$$
$$\delta(q_1, a, A) = \{[q_1, \lambda]\}$$
$$\delta(q_1, b, B) = \{[q_1, \lambda]\}$$

δ	a	b	λ
q_0	q_0 $\lambda/\mathbf{A}$	q_0 $\lambda/\mathbf{B}$	q_1 λ/λ
q_1	q_1 $\mathbf{A}/\lambda$	q_1 $\mathbf{B}/\lambda$	ϕ

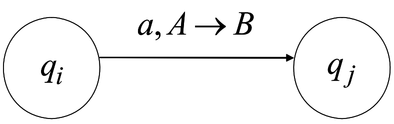
Notation: The PDA consults the current state, input symbol, and the symbol on the top of the stack to determine the machine transition. The transition function δ lists all possible transitions for a given state, symbol, and stack top combination. The value of the transition function

$\delta(q_i, a, A) = \{[q_j, B], [q_k, C]\}$

indicates that two transitions are possible when automaton is in state qi scanning an A with A on top of the stack. The transition

$[q_j, B] \in \delta(q_i, a, A)$

[new state, new stack top] ∈ δ(current state, current input, current stack top)



b) Trace all computation of the strings *ab, abb, abbbb* in M.

Computations for *ab*:

$[q_0, ab, \lambda]$	$[q_0, ab, \lambda]$	$[q_0, ab, \lambda]$
$\vdash [q_0, b, A]$	$\vdash [q_0, b, A]$	$\vdash [q_1, ab, \lambda]$
$\vdash [q_0, \lambda, BA]$	$\vdash [q_1, b, A]$	Rejected.
$\vdash [q_1, \lambda, BA]$	Rejected.	
Rejected.		

Computations for *abb:*

$[q_0, abb, \lambda]$	$[q_0, abb, \lambda]$	$[q_0, abb, \lambda]$	$[q_0, abb, \lambda]$
$\vdash [q_0, bb, A]$	$\vdash [q_0, bb, A]$	$\vdash [q_0, bb, A]$	$\vdash [q_1, abb, \lambda]$
$\vdash [q_0, b, BA]$	$\vdash [q_0, b, BA]$	$\vdash [q_1, bb, A]$	Rejected.
$\vdash [q_0, \lambda, BBA]$	$\vdash [q_1, b, BA]$	Rejected.	
$\vdash [q_1, \lambda, BBA]$	$\vdash [q_1, \lambda, A]$		
Rejected.	Rejected.		

Computations for *abbb:*

c) Show that *aaaa, baab* $\in$ L (M).

Both *aaaa, baab* are accepted by this PDA, so they are in L (M).

$[q_0, aaaa, \lambda]$	$[q_0, baab, \lambda]$
$\vdash [q_0, aaa, A]$	$\vdash [q_0, aab, B]$
$\vdash [q_0, aa, AA]$	$\vdash [q_0, ab, AB]$
$\vdash [q_1, aa, AA]$	$\vdash [q_1, ab, AB]$
$\vdash [q_1, a, A]$	$\vdash [q_1, b, B]$
$\vdash [q_1, \lambda, \lambda]$	$\vdash [q_1, \lambda, \lambda]$
Accepted.	Accepted.

d) Show that aaa, ab $\notin$ L (M).

The computations for string *aaa:*

$[q_0, aaa, \lambda]$	$[q_0, aaa, \lambda]$	$[q_0, aaa, \lambda]$	$[q_0, aaa, \lambda]$
$\vert\text{-}[q_0, aa, A]$	$\vert\text{-}[q_0, aa, A]$	$\vert\text{-}[q_0, aa, A]$	$\vert\text{-}[q_1, aaa, \lambda]$
$\vert\text{-}[q_0, a, AA]$	$\vert\text{-}[q_0, a, AA]$	$\vert\text{-}[q_1, aa, A]$	Rejected.
$\vert\text{-}[q_0, \lambda, AAA]$	$\vert\text{-}[q_1, a, AA]$	$\vert\text{-}[q_1, a, \lambda]$	
$\vert\text{-}[q_1, \lambda, AAA]$	$\vert\text{-}[q_1, \lambda, A]$	Rejected.	
Rejected.	Rejected.		

The computations for *ab* are shown in part b).

There is none of the computations accepting the strings, so they are not in L (M).

Problem 2. CH8 3k Sudkamp p. 252

Construct a PDA that accepts the set of palindromes over $\{a, b\}$.

Solution:

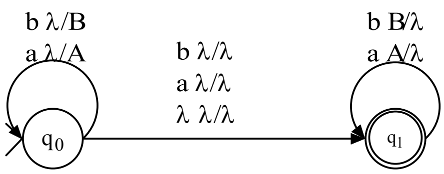
Construct the PDA:

M: $Q = \{q_0, q_1\}$

$\Sigma = \{a, b\}$

$\Gamma = \{A, B\}$

$F = \{q_1\}$



$\delta(q_0, a, \lambda) = \{[q_0, A], [q_1, \lambda]\}$

$\delta(q_0, b, \lambda) = \{[q_0, B], [q_1, \lambda]\}$

$\delta(q_0, \lambda, \lambda) = \{[q_1, \lambda]\}$

$\delta(q_1, a, A) = \{[q_1, \lambda]\}$

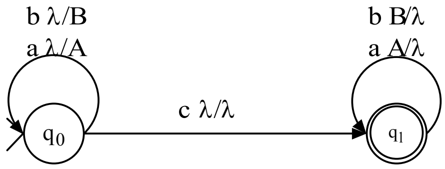
$\delta(q_1, b, B) = \{[q_1, \lambda]\}$

Problem 3. CH8 15a

Let M be the PDA in example 8.1.1. the PDA that accepts the language

$L = \{wcw^R \mid w \in \{a,b\}^*\}$.

The stack is used to record the string w as it is processed. Stack symbols A and B represent input a and b respectively.



A successful computation records the string w on the stack as it is processed. Once the c is encountered, the accepting state q_1 is entered and the stack contains a string representing w^R. The computation is completed by matching the remaining input with the elements of the stack.

Trace the computation that accepts *bbcbb.*

Solution:

$[q_0, bbcbb, \lambda]$

$\vdash [q_0, bcbb, B]$

$\vdash [q_0, cbb, BB]$

$\vdash [q_1, bb, BB]$

$\vdash [q_1, b, B]$

$\vdash [q_1, \lambda, \lambda]$

Accepted.

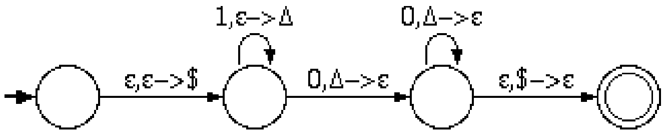
NPDA EXAMPLES

(Sipser)

Construct non-deterministic pushdown automata to accept the following languages. (Note: ε has the same meaning as λ)

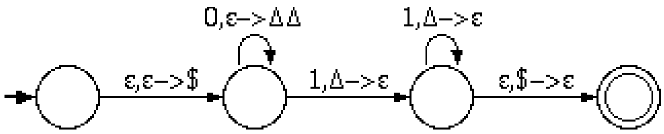
Example 1. $\{1^n 0^n \mid n>0\}$

Solution



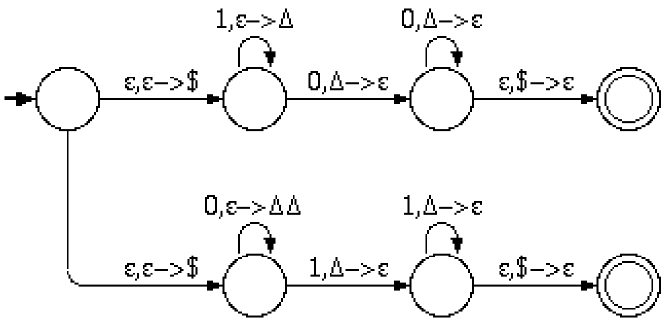
Example 2. $\{0^n 1^{2n} \mid n>=0\}$

Solution



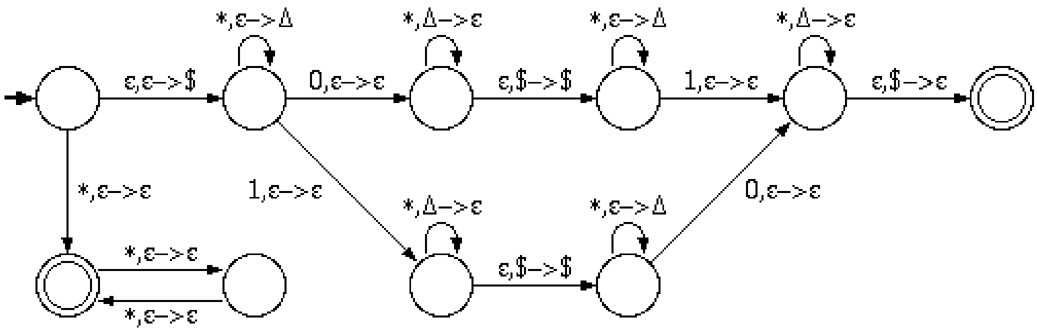
Example 3. $\{1^n 0^n \mid n>0\}$ U $\{0^n 1^{2n} \mid n>=0\}$

Solution



Example 4. (0+1)* - {ww | w in {0,1}*} (complement of ww)

Solution

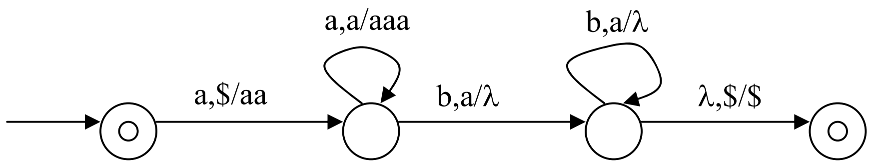


Linz

Equivalent with Sipser Example 2, in different notation. Remark: The notation a, b/c stands for a, b→c

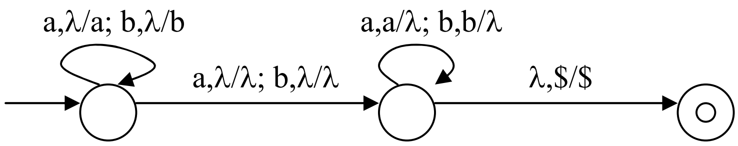
Example 5. $L = \{a^n b^{2n}: n \geq 0\}$

Solution



Example 6. $L = \{\omega\sigma\omega^R: \omega \in \{a,b\}^*, \sigma \in \{a,b\}\}$

Solution

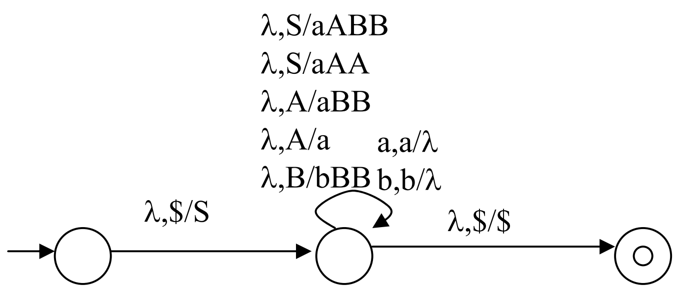


Example 7. Construct the NPDA corresponding to the grammar

$S \to aABB \mid aAA$
$A \to aBB \mid a$
$B \to bBB \mid A$

(Problem 5 one page 194, Linz)

Solution

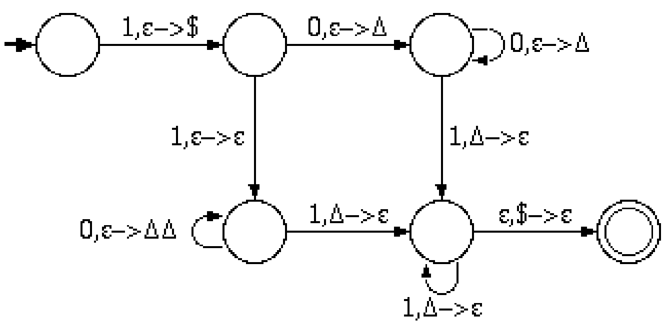


DPDA EXAMPLES (Sipser)

Construct deterministic pushdown automata to accept the following languages.

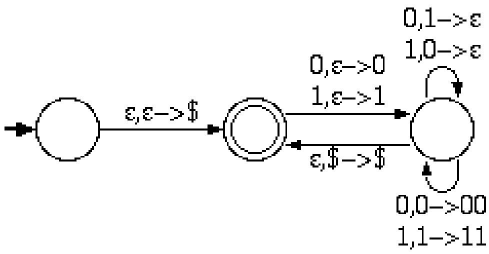
Example 8. $\{10^n1^n \mid n>0\} \cup \{110^n1^{2n} \mid n>0\}$

Solution



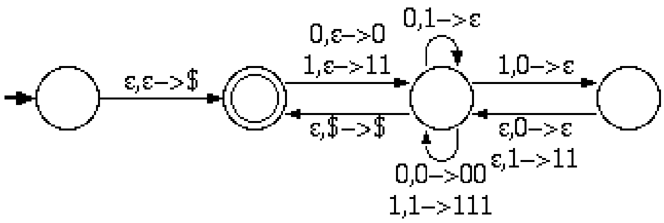
Example 9. Binary strings that contain an equal number of 1s and 0s.

Solution



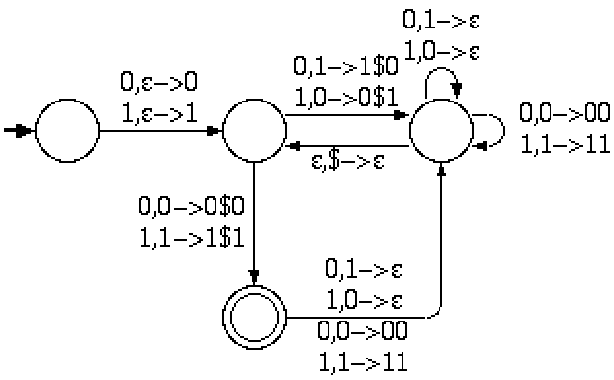
Example 10. Binary strings with twice as many 1s as 0s.

Solution



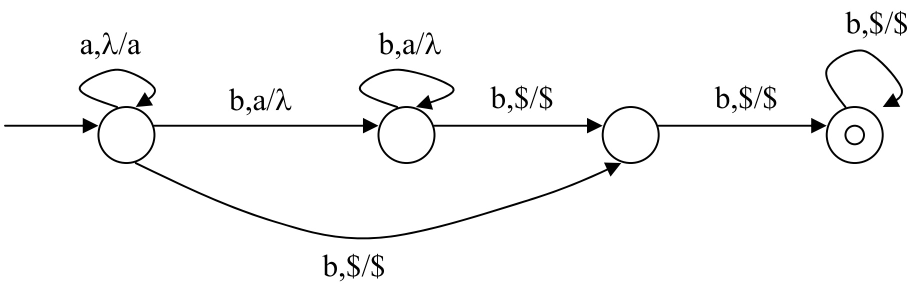
Example 11. Binary strings that start and end with the same symbol and have the same number of 0s as 1s.

Solution



Example 12. Show that $L = \{a^n b^m : m \geq n + 2\}$ is deterministic. (Problem 2 on page 199, Linz

Solution



Since there is a DPDA that accepts L, L is deterministic.

PUMPING LEMMA: CONTEXT-FREE OR NOT?

Problem 1. Show that the following language on $\Sigma = \{a, b, c\}$ is not context-free.

$L = \{a^n b^j c^k : k = jn\}$.

Solution

Assume that $L = \{a^n b^j c^k : k = jn\}$ is a context free language.

Let $w = a^m b^m c^{m^2}$, $w \in L$

By the Pumping Lemma w can be decomposed as w = uvxyz with $|vxz| \le m$

and $|vy| \ge 1$ such that $uv^i xy^i z \in L, i \ge 0$

case 1

$$\underbrace{aaa\ldots a}_{uvxy}\underbrace{ab\cdots bc\cdots c}_{z}$$

If i=0, $uv^0xy^0z = a^{m-|vy|}b^m c^{m^2} \notin L$

case 2

$$\underbrace{aaa\ldots a}_{uv}\underbrace{ab}_{x}\underbrace{\ldots}_{y}\underbrace{bc\ldots}_{z}$$

If i=0, $uv^0xy^0z = a^{m-|v|}b^{m-|y|}c^{m^2} \notin L$

case 3

$$\underbrace{aaa\ldots ab}_{u}\underbrace{b\ldots b}_{vxy}\underbrace{bc\ldots c}_{z}$$

If i=0, $uv^0xy^0z = a^m b^{m-|vy|}c^{m^2} \notin L$

case 4

$$\underbrace{aaa\ldots ab}_{u}\underbrace{b\ldots bb}_{v}\underbrace{c}_{x}\underbrace{c\ldots c}_{yz}$$

If i=0, $uv^0xy^0z = a^m b^{m-|v|} c^{m^2-|y|} \notin L$

case 5

$$\underbrace{aaa\ldots ab\ldots bc}_{u}\underbrace{c\ldots c}_{vxyz}$$

If i=0, $uv^0xy^0z = a^m b^m c^{m^2-|vy|} \notin L$

case 6 *v* or *y* containing *ab* or *bc*

If I > 0, uv^ixy^iz would be $a\ldots ab\ldots ba\ldots ab\ldots b\ldots c\ldots c$

Or $a\ldots ab\ldots bc\ldots cb\ldots c$

Contrary to the assumption.

Language is not context free.

Problem 2. Show that the following language on $\Sigma = \{a, b\}$ is not context-free:

$L = \{ ww^R a^{|w|} : w \in \{a, b\}^* \}$

Solution

Assume that $L = \{ ww^R a^{|w|} : w \in \{a, b\}^* \}$is a context free language.

Let $w = a^m b^m$, then $ww^R a^{|w|} = a^m b^m b^m a^m a^{2m} \in L$

By the Pumping Lemma w can be decomposed as w = uvxyz with $|vxz| \le m$ and $|vy| \ge 1$ such that $uv^ixy^iz \in L, i \ge 0$

case 1

$$\underbrace{a\ldots ab\ldots b}_{uvxy}\underbrace{b\ldots ba\ldots aa\ldots a}_{z}$$

If i=0, $|w| = 2m - |vy|$ is less than $|w^R|$. So $uv^0xy^0z \notin L$.

case 2

$$\underbrace{a\ldots ab\ldots}_{uvxy}\underbrace{b\ldots ba\ldots aa\ldots a}_{z}$$

If i=0, $|w^R| = 2m - |vy|$ is less than $|w|$. So $uv^0xy^0z \notin L$.

case 3

$$\underbrace{a\ldots ab\ldots bb\ldots ba\ldots aa}_{u}\underbrace{b\ldots\ldots a}_{vxyz}$$

If i=0, $\left|a^{|w|}\right| = 2m - |vx| < |w|$. So $uv^0xy^0z \notin L$.

case 4

$$\underbrace{a\ldots ab}_{u}\underbrace{b\ldots\ldots b}_{vxy}\underbrace{ba\ldots aa\ldots a}_{z}$$

If i=0, $ww^R = 4m - |vy| < 2\left|a^{|w|}\right| = 4m$. So $uv^0xy^0z \notin L$.

case 5

$$\underbrace{a\ldots ab\ldots bb\ldots ba}_{u}\ldots\underbrace{a\ldots\ldots aa}_{vxy}\underbrace{a\ldots a}_{z}$$

If i=0, $w^R a^{|w|} = 4m - |vy| < 2|w| = 4m$. So $uv^0xy^0z \notin L$.

This is contrary to the assumption.

Language L is therefore not context free.

Problem 3. **CH8 17c Sudkamp**

Using the pumping lemma prove that the language below is not context-free.

$$L = \{a^i b^{2i} a^i \mid i \geq 0\}$$

Proof:

Assume that the language is context-free. By the pumping lemma, the string $w = a^k b^{2k} a^k$, where k is the number specified by the pumping lemma, can be decomposed into sub-string *uvwxy* that satisfy the condition *length* (*vwx*) ≤ *k*, *length (v)* + *length (x)* > 0, and $uv^i wx^i y \in L, (i \geq 0)$.

Since the *length* (*vwx*) ≤ *k*, the *vwx* can have the following possibilities:

Case 1:

It contains *a*-s and *b*-s ($a^i b^j$ or $b^i a^j$, $0 < i + j \leq k$. Since *length* (*vwx*) ≤ *k,* it cannot have *a*-s at both side of *b*-s,). Then uv^2wx^2y contains a…b…a…b…a. It will not be in L any more.

Case 2:

It is contained in one of a^k, b^{2k} or second a^k. Since the repetition in uv^2wx^2y will increase one of the string's lengths of a^k, b^{2k} or second a^k, the ratio of 1:2:1 will not be satisfied. Then $uv^2wx^2y \notin L$.

There is no decomposition *L* that satisfies the condition of pumping lemma. Therefore, L is not context free.

Problem 4. CH8 18a

Prove that the language $L = \{a^i b^{2i} c^j \mid i, j \geq 0\}$ is context-free.

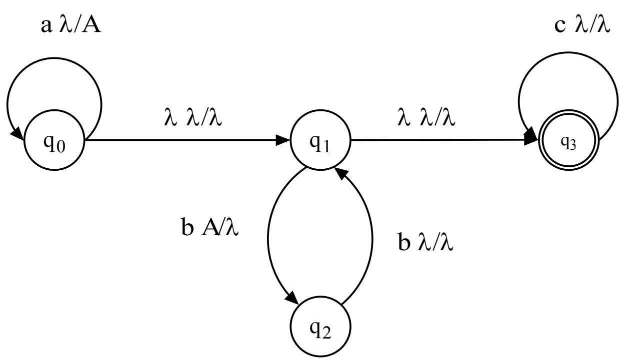
Proof:

A context-free language can be expressed by a context-free grammar. The context-free grammar below expresses the language L.

G:

$$S \to AB$$
$$A \to aAbb \mid \lambda$$
$$B \to cB \mid \lambda$$

Also, if a language L can be accepted by a PDA, it will be a context-free language. We can construct a PDA to accept language L as below.



Therefore, the language L is context-free.

Problem 5. 17 page 220, Linz

We want to prove that the language L = $\{a^n b^n : n \geq 0$ and n is not a multiple of 5$\}$ is a context-free language.

Solution

Consider the following two languages:

L_1 ={w : w is made from a's and b's and the length of w is a multiple of ten}

$L_2 = \{a^n b^n: n \geq 0\}$

Let L_1' denote the complement of L. We have that $L = L_1' \cap L_2$.

It is easy to see that L_1 is a regular language, since we can easily build a finite automaton with 10 states that accepts any string in this language.

Since L_1 is regular it must be that L_1' is regular too, since regular languages are closed under complement (Theorem 4.1).

The language L_2 is context-free (the grammar is: $S \rightarrow aSb \mid \lambda$).

Therefore, the language $L = L_1' \cap L_2$ is also context-free, since context-free languages are closed under regular intersection (Theorem 8.5).

Problem 6. $\{1^k 0^i 1^i 0^j 1^j 0^k \mid i, j, k > 0\}$

Solution

CF. Here is a grammar that will generate the language.

$S \rightarrow 1A0$
$A \rightarrow 1A0 \mid B$
$B \rightarrow CC$
$C \rightarrow 0D1$
$D \rightarrow 0D1 \mid \lambda$

Problem 7. $\{w\#x \mid w$ is a substring of x, where w, x are in $\{0,1\}^*\}$.

Solution

Not CF. Assume for the purpose of contradiction that it is. Then let the pumping length be p. Consider the string $s=0^p1^p\#0^p1^p$, which is in the language. If we decompose s into s=uvwxy as in the statement of the pumping lemma, there are three cases to consider.

- If vwx is contained in the first half of the string s, then pumping up even once (that is, taking the string uv^2wx^2y) will give us a string where the first half is longer than then second half, which means it can't be a substring of the second half.

- If vwx is contained completely in the second half, then when we pump down (that is, take the string uwy). Then the second half will be shorter than the first half (since |vwx|>=1) and so the pumped down string will not be in the language.
- If vwx overlaps with the symbol #, then it must be the case that the # is contained in w, or else pumping up would give too many # symbols. So either the v will be a string of 1's or the x will be a string of 0's or both (We need the fact that |vwx| <= p here). If v is a nonempty string of 1's then pumping up will give the left half more 1's then the right side, so the left side will not be a subset of the right side. If x is a nonempty string of zeros then pumping down will give the right side fewer zeros than the left so the left side will not be a subset of the right side.

In any case we come to the conclusion that no matter how we choose the decomposition, the conditions of the pumping lemma will be violated. So the language cannot be regular.

Problem 8. $\{0^i 1^i 0^j 1^i \mid i, j > 0\}$

Solution

Not CF. Suppose it is. Then let the pumping length be p. Consider the string $s=0^p1^p01^p$, which is in the language. If we decompose s into s=uvwxy as in the statement of the pumping lemma, then an argument like the previous one shows that if vwx is anything but the lone zero between the ones, then pumping up will give you something that is not in the language. On the other hand if vwx is the lone zero, then pumping down will give you. $0^p1^p1^p$, which is not in the language. This contradicts the pumping lemma, so therefore the language is not context free.

Problem 9. The complement of $\{(0^n1^n)^m | m, n > 0\}$.

Solution

CF. Here is a grammar that generates the language, with some comments on the side to explain what each non-terminal represents.

$S \to ABA \mid$	
$A \to \lambda \mid AC$	
$C \to 01 \mid 0C1;$	0^n1^n
$B \to 0B1 \mid D \mid F;$	$0^m1^n m \neq n$
$D \to 0 \mid 0D;$	one or more 0s
$F \to 1 \mid 1F;$	one or more 1s

Why this works: Since R=(00*11*)(00*11*)* is regular, the complement ~R includes all strings which are not even of the correct form. We can Union 0*1* complement... nondeterministically guess which pair mismatches.

Problem 10. 2.18 a. (Lewis-Papadimitriou)

$A = \{0^n1^n0^n1^n \mid n >= 0\}$.

Solution

Not CF. Let p be the pumping length given by the pumping lemma. We show that s = $0^p1^p0^p1^p$ cannot be pumped. Let s = uvxyz.

If either v or y contain more than one type of alphabet symbol, s\` = $uv^2xy^2z$ does not contain the symbols in the correct order. Hence s\` $\notin$ A.

If both v and y contain (at most) one type of alphabet symbol, uv^2xy^2z contains runs of 0's and 1's of unequal length. Hence s\` $\notin$ A.

Because s cannot be pumped in any way without violating the pumping lemma,

A is not CF.

Problem 11. 2.18 b.

$B = \{0^n\#0^{2n}\#0^{3n} \mid n >= 0\}$.

Solution

Not CF. Let p be the pumping length given by the pumping lemma. We show that $s = 0^p\#0^{2p}\#0^{3p}$ cannot be pumped. Let s = uvxyz.

Neither v nor y can contain #, otherwise $s` = uv^2xy^2z$ contains more than two #s.

Therefore, if we divide x into three segments by #s: 0^{1p}, 0^{2p}, 0^{3p}, at least one of the segments is not contained within either v or y. Hence, $s` = uv^2xy^2z$ is not in B because the 1 : 2 : 3 length ratio of the segments is not maintained. Hence $s` \notin B$.

Because s cannot be pumped in any way without violating the pumping lemma,

B is not CF.

Problem 12. 2.18 c.

$C = \{ w\#x \mid w$ is a substring of x, where $w, x \in \{a, b\}^*\}$.

Solution

Not CF. Let p be the pumping length given by the pumping lemma.

We show that $s = a^pb^p\#a^pb^p$ cannot be pumped. Let s = uvxyz.

Neither v nor y can contain #,

otherwise $s` = uv^0xy^0z$ does not contain # and therefore $s` \notin C$.

If both v and y are nonempty and occur on the LHS of the #, the string $s` = uv^2xy^2z \notin C$ because it is longer on the LHS than on the RHS. Similarly, if both strings occur on the RHS, the string $s` = uv^0xy^0z \notin C$ because, again, the LHS is longer than the RHS.

If only one of v and y is nonempty ($|vy| > 0$, therefore v and y can not be empty at the same time), treat them as if they both occurred on the same side of the # as above.

The only remaining case will be where both v and y are nonempty and v is on the LHS, y on the RHS. However, $|vxy| <= p$, therefore, v can only contain b's and y can only contain a’s. Hence s` = $uv^2xy^2z \notin C$ because it contains more b’s on the LHS than on the RHS.

Since s cannot be pumped in any way without violating the pumping lemma, C is not CF.

Referenser

1. Linz, An Introduction to Formal Languages and Automata, Jones & Bartlett 2000
2. Sipser, Introduction to the Theory of Computation, PWS 1997
3. Sudkamp, Languages and Machines, Addison Wesley 1998
4. Lewis-Papadimitriou, Elements of the Theory of Computation, Prentice Hall 1998