

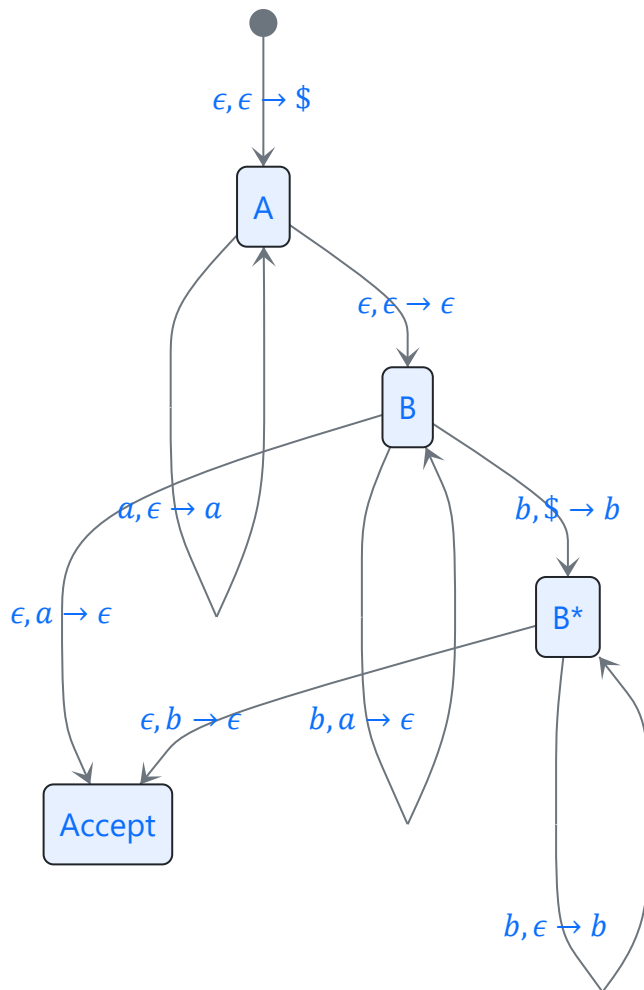
HW6CSDS343

1

Prove the following language is context free $E = \{a^i b^j | i \neq j\}$

A

Give a pushdown automaton that decides E



Proof

If $s \in L$ then there are either more a 's than b 's or vice versa.

Case 1: $s \in L$ where $i > j$

PDA will add all a 's to the stack, then pop a 's for each b . Since the number of b 's is less than the number of a 's we will run out of b 's and there will still be a 's on the stack. Since there are a 's on the

stack then we can move to the accept state and accept.

Case 2: $s \in L$ where $j > i$

PDA will add all a 's to the stack then pop a 's for each b . Since there are more b 's than a 's the PDA will pop all the a 's then start adding b 's only once the stack is empty. Once we reach the end of the string we will have b 's on the stack and move to the accept state and accept.

If $s \notin L$ then either $i = j$ or string is not of form $a^i b^j$

Case 3: s not of form $a^i b^j$

If at any point the string does not follow the form $a^i b^j$ then our machine will reject because any transition not labeled is a reject.

Case 4: $i = j$

PDA will add all a 's to the stack, then pop a 's for each b . Since the number of b 's is equal to the number of a 's we will clear the stack until it is just the dollar sign. From there, there is no transition that will lead to the accept state with just the $\$$ so our machine will reject.

Therefore since we have created a nondeterministic PDA that decides L then L must be context free.

B

Give a context free grammar for E

$$S \rightarrow aSb$$

$$S \rightarrow A$$

$$S \rightarrow B$$

$$A \rightarrow aA$$

$$A \rightarrow a$$

$$B \rightarrow bB$$

$$B \rightarrow b$$

Proof

Our CFG creates strings of the form $a^i b^j$ because there are no points where an a can be placed to the right of a b nor places to place a b to the left of an a . Initially our CFG creates strings with equal a 's and b 's, but then transitions into A or B which add one or more a / b 's respectively. Thus our CFG creates strings with unequal amounts of a 's and b 's with the format $a^i b^j$.

Thus L is a context free language because we created a CFG for L .

2

Prove the concatenation $(xy | x \in L_1 \wedge y \in L_2)$ of two context free languages L_1 and L_2 is context free

A

Assume you have the context free grammar for each language, give the context free grammar for the concatenation (create the CFG directly)

$S \rightarrow XY$ where X is the context free grammar for L_1 and Y is the context free grammar for L_2

Proof by contradiction

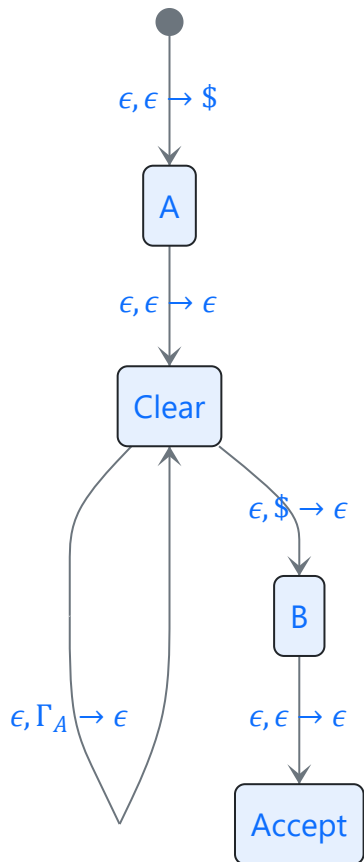
Suppose string $s \rightarrow S$ is not in L , by the definition of S , s must be split such that the first half of the string is in L_1 and the 2nd half is in L_2 . By definition of L then s is a concatenation of L_1 and L_2 thus $s \in L$ which is a contradiction.

If a string s does not follow the rules of S then it cannot be a concatenation of 2 strings in L_1 and L_2 otherwise it would follow the rules of S

Therefore s is in L if it follows the grammar rule S given above. As such, since there is a grammar rule for L then L is a context free grammar.

B

Assume you have the pushdown automata for each language, give the pushdown automata for the concatenation (create the PDA directly)



A is the machine for L_1 , B is the machine for L_2

We map the accept state of the first automata to a clear state, which clears the stack, then move to the initial state of the second automata.

That is, we only move to the other machine if the first machine accepts. We only transition to the accept state if both machines accept. This machine only accepts if we end at the accept state and there is nothing left on the string.

Proof

If our PDA accepts the string, then there exists a midpoint such that the first half is accepted by A and the second half is accepted by B . This would indicate that the string is comprised of xy where $x \in L_1, y \in L_2$ which would mean our string is in L .

If our PDA rejects the string, then there is no such midpoint (since this is a nondeterministic machine). This would mean our string is not in L as there is no midpoint.

Therefore L is a context free grammar because there exists a PDA that decides L .

3

Prove that each of the following languages are not context free

A

$L_1 = \{w\bar{w} \mid w \in \{0, 1\}^*\}$ where \bar{w} is the bit complement of w

We can prove this language is not context free using proof by contradiction with the pumping lemma

Suppose L_1 is context free, then \exists a p s.t. all strings s in L_1 with $|s| > p$ can be divided into $s = uvxyz$ such that $|vxy| \leq p$ and $|vy| > 0$

Given the string $0^p 1^p 0^p 1^p$ and that $|vxy| \leq p$, there are 5 cases of vxy

1. $vxy = 0^*$ first
2. $vxy = 0^* 1^*$ first
3. $vxy = 1^*$ first
4. $vxy = 1^* 0^*$ first
5. $vxy = 0^*$ second
6. $vxy = 0^* 1^*$ second
7. $vxy = 1^*$ second
8. $vxy = 1^* 0^*$ second
9. $vxy = 0^*$ third
10. $vxy = 0^* 1^*$ third
11. $vxy = 1^*$ third

For all cases suppose the string uv^0xy^0z . This string should be in L_1 according to the pumping lemma but we can prove this is not true for all cases.

In case 1, our string is now of the form $0^{p-|vy|} 1^p 0^p 1^p$. This shifts the midpoint to the right by $|vy|/2$. Thus if we split the string in half we get $w = 0^{p-|vy|} 1^p 0^{p-|vy|/2}$ and the second half as $1^{p-|vy|/2} 0^p 1^p$. Since the second half is not the complement of w this string is not in L_1 .

Case 5 and 9 follow similar logic, with case 9 shifting the midpoint left.

In case 3, our string is now of the form $0^p 1^{p-|vy|} 0^p 1^p$. This shifts the midpoint to the right by $|vy|/2$. Thus if we split the string in half we get $w = 0^p 1^{p-|vy|} 0^{p-|vy|/2}$ and the second half as $1^{p-|vy|/2} 0^p 1^p$. Since the second half is not the complement of w this string is not in L_1 .

Case 7 and 11 follow similar logic, with case 11 shifting the midpoint left.

Cases 2 and 4 follow similar logic, that is decreasing the number of elements on the left by $|vy|$ and then shifting the midpoint to the right and creating strings 0101 and 101 with some number of 1's and 0's (less than or equal to p). Since the second half is not the complement of w this string is not in L_1 .

Similar logic for cases 8 and 10 except these cases decrease the number of elements on the right, shifting the midpoint left and creating strings 010 and 0101 (with some number of 1's and 0's less than or equal to p) which are not complements of each other.

Case 6 is at the midpoint, thus if it takes away more 0's than 1's the midpoint will shift to the right and follow similar logic as cases above. If it takes away more 1's than 0's the midpoint will shift to the left and follow similar logic as cases above. However if it takes away the same number of 0's and 1's the midpoint will not shift and we will split the string into halves $w = 0^p 1^p 0^{p-|vy|/2}$ and $1^{p-|vy|/2} 0^p 1^p$. Since the second half is not the complement of w this string is not in L_1 .

We have shown for all cases that uv^0xy^0z is not in L_1 , thus L_1 is not context free as it contradicts the pumping lemma

B

$$L_2 = \{a^m b^n c^{m \times n} | m, n \in \mathbf{Z}_{\geq 0}\}$$

We can prove this language is not context free using proof by contradiction with the pumping lemma

Suppose L_2 is context free, then \exists a p s.t. all strings s in L_2 with $|s| > p$ can be divided into $s = uvxyz$ such that $|vxy| \leq p$ and $|vy| > 0$

Given the string $a^p b^p c^{p \times p}$ and that $|vxy| \leq p$, there are 5 cases of vxy

1. $vxy = a^*$
2. $vxy = a^* b^*$
3. $vxy = b^*$
4. $vxy = b^* c^*$
5. $vxy = c^*$

In case 1 / 2 / 3, suppose the string uv^0xy^0z , then we have less a 's / b 's without changing the number of c 's, thus our new string is not in L_2 .

In case 5, suppose the string uv^0xy^0z , then we have less c 's without changing the number of a 's or b 's, thus our new string is not in L_2 .

In case 4, suppose the string uv^0xy^0z , then we linearly decrease the number of b 's and c 's, but is we decrease the number of b 's by one then we need to decrease the number of c 's by p . However we cannot decrease c by a factor of p because $|vxy| < p$, thus our new string is not in L_2 .

Therefore since all of our cases reject the pumping lemma, L_2 is not context free