Week 4 Clicker Question 1

Let A and B be a context free languages, on Σ_A and Σ_B, respectively.
Let the start variable for a grammar for A be S_A, and the start variable for a grammar for B be S_B. Define $T = \{w_A w_b | w_A \in A \wedge w_B \in B\}$.
Is T a context free language?

A Yes

B No

Week 4 Clicker Question 2

Can this grammar generate strings longer than any given positive integer, m?

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

$S \rightarrow \epsilon \mid A$

$A \rightarrow BC \mid a$

$B \rightarrow CA \mid b$

$C \rightarrow AB \mid c$

A Yes

B No

Week 4 Clicker Question 3

Consider this grammar:

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

$S \to \epsilon \mid A$

$A \to BC \mid a$

$B \to CA \mid b$

$C \to AB \mid c$

Suppose you could use each rule at most once. What is the longest string you could generate?

- A 1
- B 2
- C 3
- D > 3

Week 4 Clicker Question 4

If there is a length of string such that longer strings have to have been generated by using at least one rule more than once, are we going to get a pumping lemma from this?

A True

B False

stop for a while

Exercise Convert Grammar to Chomsky Normal Form 5

$\Sigma \rightarrow Example1 | Example1 \cdot \Sigma \cdot Example1$
$Example1 \rightarrow AsAndBs$
$AsAndBs \rightarrow As \cdot Bs \cdot As$
$As \rightarrow As \cdot a \mid a \mid \epsilon$
$Bs \rightarrow Bs \cdot b \mid b \mid \epsilon$

1. Make a new start variable
2. Get rid of ϵ rules
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 6

$S_0 \rightarrow \Sigma$
$\Sigma \rightarrow Example1 | Example1 \cdot \Sigma \cdot Example1$
$Example1 \rightarrow AsAndBs$
$AsAndBs \rightarrow As \cdot Bs \cdot As$
$As \rightarrow As \cdot a \mid a \mid \epsilon$
$Bs \rightarrow Bs \cdot b \mid b \mid \epsilon$

1. **Make a new start variable**
2. Get rid of ϵ rules
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 7

$S_0 \rightarrow \Sigma$

$\Sigma \rightarrow Example1 | Example1 \cdot \Sigma \cdot Example1$

$Example1 \rightarrow AsAndBs$

$AsAndBs \rightarrow As \cdot Bs \cdot As$

$AsAndBs \rightarrow Bs \cdot As$

$AsAndBs \rightarrow As \cdot Bs$

$AsAndBs \rightarrow Bs$

$As \rightarrow As \cdot a \mid a$

$As \rightarrow a \mid a$

$Bs \rightarrow Bs \cdot b \mid b$

1. Make a new start variable
2. **Get rid of ϵ rules**
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 8

$S_0 \to \Sigma$

$\Sigma \to Example1 | Example1 \cdot \Sigma \cdot Example1$

$Example1 \to AsAndBs$

$AsAndBs \to As \cdot Bs \cdot As$

$AsAndBs \to As \cdot As$

$AsAndBs \to Bs \cdot As$

$AsAndBs \to As$

$AsAndBs \to As \cdot Bs$

$AsAndBs \to As$

$AsAndBs \to Bs$

$AsAndBs \to \epsilon$

$As \to As \cdot a \mid a$

$Bs \to Bs \cdot b \mid b$

1. Make a new start variable
2. **Get rid of ϵ rules**
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 9

$S_0 \rightarrow \Sigma$
$\Sigma \rightarrow Example1 | Example1 \cdot \Sigma \cdot Example1$
$Example1 \rightarrow AsAndBs$
$AsAndBs \rightarrow As \cdot Bs \cdot As \mid As \cdot As \mid Bs \cdot As \mid As \mid As \cdot Bs \mid As \mid Bs$
$AsAndBs \rightarrow \epsilon$
$As \rightarrow As \cdot a \mid a$
$Bs \rightarrow Bs \cdot b \mid b$

1. Make a new start variable
2. **Get rid of ϵ rules**
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 10

$S_0 \to \Sigma$

$\Sigma \to Example1 | Example1 \cdot \Sigma \cdot Example1$

$Example1 \to AsAndBs \mid \epsilon$

$AsAndBs \to As \cdot Bs \cdot As \mid As \cdot As \mid Bs \cdot As \mid As \mid As \cdot Bs \mid As \mid Bs$

$As \to As \cdot a \mid a$

$Bs \to Bs \cdot b \mid b$

1. Make a new start variable
2. **Get rid of ϵ rules**
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 11

$S_0 \rightarrow \Sigma$
$\Sigma \rightarrow Example1 | Example1 \cdot \Sigma \cdot Example1 \mid \epsilon$
$Example1 \rightarrow AsAndBs$
$AsAndBs \rightarrow As \cdot Bs \cdot As \mid As \cdot As \mid Bs \cdot As \mid As \mid As \cdot Bs \mid As \mid Bs$
$As \rightarrow As \cdot a \mid a$
$Bs \rightarrow Bs \cdot b \mid b$

1. Make a new start variable
2. **Get rid of ϵ rules**
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 12

$S_0 \to \Sigma \mid \epsilon$
$\Sigma \to Example1 | Example1 \cdot \Sigma \cdot Example1$
$Example1 \to AsAndBs$
$AsAndBs \to As \cdot Bs \cdot As \mid As \cdot As \mid Bs \cdot As \mid As \mid As \cdot Bs \mid As \mid Bs$
$As \to As \cdot a \mid a$
$Bs \to Bs \cdot b \mid b$

1. Make a new start variable
2. Get rid of ϵ rules
3. **Get rid of unit rules**
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 13

$S_0 \rightarrow \Sigma \mid \epsilon$
$\Sigma \rightarrow Example1 \cdot \Sigma \cdot Example1$
$Example1 \rightarrow AsAndBs$
$\Sigma \rightarrow AsAndBs$
$AsAndBs \rightarrow As \cdot Bs \cdot As \mid As \cdot As \mid Bs \cdot As \mid As \mid As \cdot Bs \mid As \mid Bs$
$As \rightarrow As \cdot a \mid a$
$Bs \rightarrow Bs \cdot b \mid b$

1. Make a new start variable
2. Get rid of ϵ rules
3. **Get rid of unit rules**
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 14

$S_0 \to \Sigma \mid \epsilon$
$\Sigma \to Example1 \cdot \Sigma \cdot Example1$
$Example1 \to AsAndBs$
$\Sigma \to AsAndBs$
$AsAndBs \to As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$
$As \to As \cdot a \mid a$
$AsAndBs \to As \cdot a \mid a$
$Bs \to Bs \cdot b \mid b$
$AsAndBs \to Bs \cdot b \mid b$

1. Make a new start variable
2. Get rid of ϵ rules
3. **Get rid of unit rules**
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 15

$S_0 \to \Sigma \mid \epsilon$
$\Sigma \to Example1 \cdot \Sigma \cdot Example1$
$\Sigma \to AsAndBs$
$AsAndBs \to As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$
$Example1 \to As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$
$As \to As \cdot a \mid a$
$AsAndBs \to As \cdot a \mid a$
$Example1 \to As \cdot a \mid a$
$Bs \to Bs \cdot b \mid b$
$AsAndBs \to Bs \cdot b \mid b$
$Example1 \to Bs \cdot b \mid b$

1. Make a new start variable
2. Get rid of ϵ rules
3. **Get rid of unit rules**
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 16

$S_0 \rightarrow \Sigma \mid \epsilon$

$\Sigma \rightarrow Example1 \cdot \Sigma \cdot Example1$

$AsAndBs \rightarrow As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$Example1 \rightarrow As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$\Sigma \rightarrow As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$As \rightarrow As \cdot a \mid a$

$AsAndBs \rightarrow As \cdot a \mid a$

$Example1 \rightarrow As \cdot a \mid a$

$\Sigma \rightarrow As \cdot a \mid a$

$Bs \rightarrow Bs \cdot b \mid b$

$AsAndBs \rightarrow Bs \cdot b \mid b$

$Example1 \rightarrow Bs \cdot b \mid b$

$\Sigma \rightarrow Bs \cdot b \mid b$

1. Make a new start variable
2. Get rid of ϵ rules
3. **Get rid of unit rules**
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 17

We're going to get rid of "unreachable" rules:

$S_0 \to \Sigma \mid \epsilon$

$\Sigma \to Example1 \cdot \Sigma \cdot Example1$

$AsAndBs \to As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$Example1 \to As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$\Sigma \to As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$As \to As \cdot a \mid a$

$AsAndBs \to As \cdot a \mid a$

$Example1 \to As \cdot a \mid a$

$\Sigma \to As \cdot a \mid a$

$Bs \to Bs \cdot b \mid b$

$AsAndBs \to Bs \cdot b \mid b$

$Example1 \to Bs \cdot b \mid b$

$\Sigma \to Bs \cdot b \mid b$

1. Make a new start variable
2. Get rid of ϵ rules
3. Get rid of unit rules
4. Put remaining rules in correct form

Exercise Convert Grammar to Chomsky Normal Form 18

$S_0 \rightarrow \Sigma \mid \epsilon$

$\Sigma \rightarrow Example1 \cdot \Sigma \cdot Example1$

$Example1 \rightarrow As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$\Sigma \rightarrow As \cdot Bs \cdot As \mid As \cdot As \cdot As \mid As \cdot Bs$

$As \rightarrow As \cdot a \mid a$

$Example1 \rightarrow As \cdot a \mid a$

$\Sigma \rightarrow As \cdot a \mid a$

$Bs \rightarrow Bs \cdot b \mid b$

$Example1 \rightarrow Bs \cdot b \mid b$

$\Sigma \rightarrow Bs \cdot b \mid b$

1. Make a new start variable
2. Get rid of ϵ rules
3. Get rid of unit rules
4. **Put remaining rules in correct form**

Exercise Convert Grammar to Chomsky Normal Form 19

We'll just see one instance of this process:

$S_0 \rightarrow \Sigma \mid \epsilon$

$\Sigma \rightarrow Example1 \cdot u_1$

$u_1 \rightarrow \Sigma \cdot Example1$

...

1. Make a new start variable
2. Get rid of ϵ rules
3. Get rid of unit rules
4. **Put remaining rules in correct form**

That's the end of the exercise.

Clickers 20

The pumping lemma does not tell us anything about finite, context-free languages.

A True

B False

Clickers 21

One way to calculate the threshold value p for the length of a string that must pump, uses the branching factor b for parse trees in the context free grammar. This number, b, can be any positive integer, unless the grammar has been put into a special form, such as Chomsky Normal Form, so a very large pumping length can be achieved with a very shallow tree.

A True

B False

Clickers 22

What's the truth about pumping lemmas?

A A long-enough string has to pump in both regular and context-free languages.

B If we reuse a state, or a substitution rule, once, we can reuse it an infinite number of times.

C Reusing a state or a rule implies there is a cycle in the machine or the grammar, and when such a cycle exists, there is a path that does not use the cyle at all, so we can "pump down", to the $i = 0$ case.

D One counterexample string is sufficient to prove a language is not context-free, but there must be no way at all that this string can be pumped.

E All of the above.

Clickers 23

Which assertion about pumping lemmas is false?

A When a string from a context free language is partitioned into parts $uvxyz$, $|v|$ is permitted to be 0.

B When a string from a context free language is partitioned into parts $uvxyz$, y is permitted to be 0.

C The section x does not get pumped.

D When a string from a context free language is partitioned into parts $uvxyz$, the parts having no length constraints are u, x and z.

E The lengths of the segments v and y do not have to be the same.

Clickers 24

A string can be pumped up or down as many times as we like.

A True

B False

Stop this for a while.

W5 25

Choose the strongest true statement

A Some regular languages can be recognized by pushdown automata.

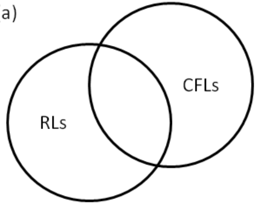
B Every regular language can be recognized by some pushdown automaton.

C Every regular language can be recognized by some pushdown automaton without use of the stack.

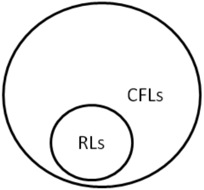
W5 26

Which Venn diagram best represents the classes of languages we have studied?

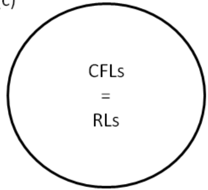
(a)



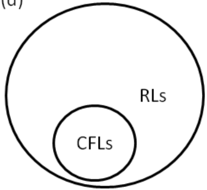
(b)



(c)



(d)



(e) None of the above

Note: CFLs = Context-Free Languages, RLs = Regular Languages

W5 27

When designing a pushdown automaton, is it required to have the same alphabet for stack symbols, Γ, as for input symbols, Σ?

A Yes

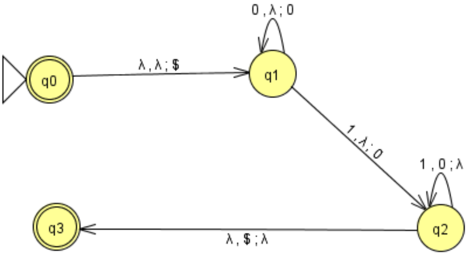
B No, except the symbol ϵ will be in common

C No commonality is necessary at all, not even ϵ

D Apart from ϵ, it depends on the language being processed by the PDA.

W5 28

What if a random number had been pushed onto the stack before this process started?



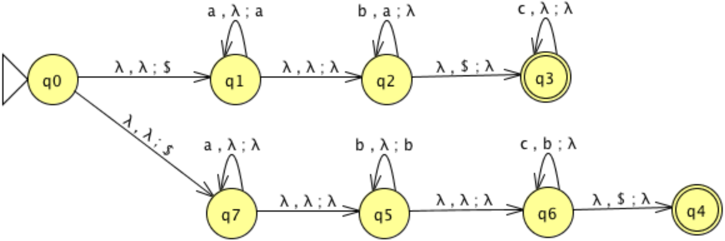
A The automaton would recognize the same language, because any further input after the $ is reached exits the accept state.

B The existing content on the stack would interfere with the ϵ, ϵ transition from q_0 to q_1.

C The q_1 to q_1 transition loop would operate for each 0 in the random number.

D The 1's in the random number would exercise the q_1 to q_2 transition.

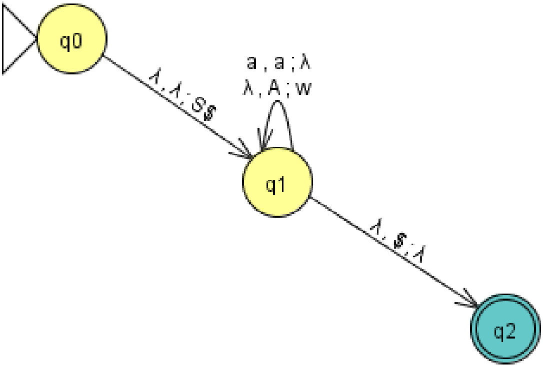
W5 29



Pick the most inclusive, correct answer.

- A Counts the a’s, b’s and c’s in the input
- B Counts the a’s, b’s and c’s in the input, matching the count of a’s with b’s
- C Counts the a’s, b’s and c’s in the input, matching the count of a’s with c’s.
- D Counts the a’s, b’s and c’s in the input, matching the count of a’s with b’s, and a’s with c’s.
- E Counts the a’s, b’s and c’s in the input, matching the count of a’s with b’s, and b’s with c’s.

W5 30



What does this machine do for $A \to \epsilon$?

A Moves to the accept state.

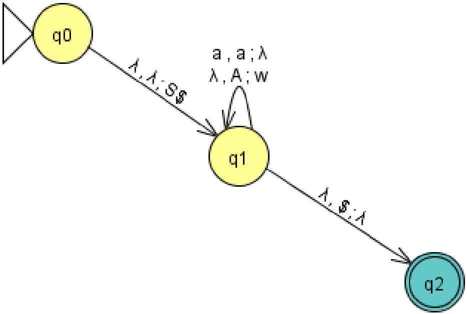
B Ceases the thread.

C Pops the stack.

D Looks at the next token of input.

E This is a vacuous question because $A \to \epsilon$ never occurs.

W5 31



What does this machine do for $A \to abc$?

A Pushes a then b then c

B Pushes c then b then a

C Pops A then pushes a then b then c

D Pops A then pushes c then b then a

E Pops A, then pushes from the right, only the variables, matching terminals against input characters rather than pushing them onto the stack.

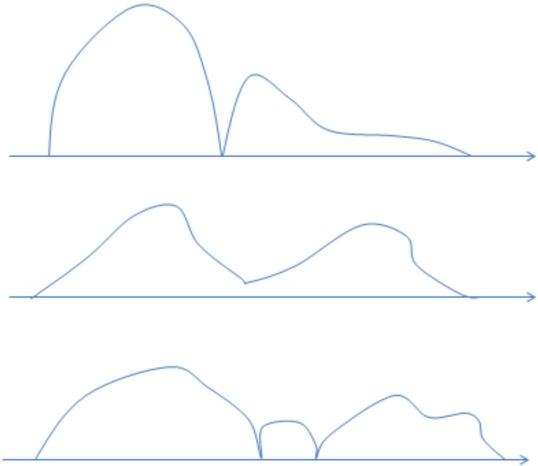
W5 32

A transition on the state q_{loop} is $a, a \to \epsilon$. What does this mean?

- A Takes an "a" from the input, only if there is an "a" on the stack, and nothing else.
- B Takes an "a" from the input and places it on the stack.
- C Takes an "a" from the input, pops an "a" from the stack, and nothing else.
- D Takes an "a" from a substitution rule and compares it with an input token.
- E Pops an "a" from the stack, only if the next character on the input is "a", and nothing else.

stop here for a while

W5 33



Which of the following stack height trajectories is possible?

A A, B and C

B A and C

W5 34

What about $A_{pq} \to aA_{rq}$?

A It is included in $aA_{rs}b$.

B It does not occur, because we want no net change in the stack.

C It is included in $A_{pr}A_{rq}$.

D In Chomsky Normal Form, we would not see this case.

E It is not as helpful to consider as $A_{pq} \to aA_{rs}b$

W5 35

We can say that $A_{pq} \rightarrow A_{pr}A_{rq}$ is handled, just because we are using an inductive proof on the length of the derivation, and both A_{pr} and A_{rq} necessarily have shorter, integer derivation lengths, and we have a base case.

A True

B False

W5 36

Is proof by induction totally sleek or what?

A Totally sleek

B What

C Confusing

W5 37

What does it take to persuade that if $A_{pq} \rightarrow a \cdot A_{rs} \cdot b$ generates x then x can bring P from p with an empty stack to q with an empty stack?

- A If the machine had a transition from p to r that did not read the stack but did push the first symbol of x, and the machine had a transition from s to q that popped the first symbol of x, the construction of the grammar would have created this rule.
- B The number of derivation steps for A_{rs} is necessarily smaller.
- C One of the states r or s pushes and the other pops.
- D A and B but not C.
- E A and B and C

W5 38

Are $A_{pq} \to A_{pr}A_{rq}$ and $A_{pq} \to a \cdot A_{rs} \cdot b$ really all the cases that there are?

A Yes

B No

W5 39

How is it that $A_{pq} \to A_{pr}A_{rq}$ and $A_{pq} \to a \cdot A_{rs} \cdot b$ are really all the cases that there are?

A Each transition either pushes or pops but not both.

B Any PDA can be converted into the form that each transition either pushes or pops but not both.

C Any PDA can be converted into the form where the stack is empty at the beginning and the end, while continuing to recognize the same language.

D Preparing the system to be analyzable is an important step in analysis.

E All of the above.

W5 40

Which of the following is, or are, reasons why regular languages are a proper subset of context free languages?

A Nondeterministic pushdown automata are more powerful than deterministic pushdown automata.

B Regular languages have a finite set of symbols, and can be constructed from languages containing single symbols using union, concatenation and star, and this is also true of context free languages.

C Any DFA can be converted into a PDA by adding a stack.

D A and C but not B.

E B and C but not A.

W5 41

Which of the following is the most relevant to the proof for $C = \{a^i b^j c^k | 0 \leq i \leq j \leq k\}$ is not context free?

A Borderline instances can be a good choice for a proof.

B When pumping, we can pump more times up than we necessarily can pump down.

C Every string in a context free language has to be able to be pumped in at least one way, where a way is an assignment of the elements u,v,x,y,z in the pumping lemma.

D Two cases form a partition, if one is the complement of the other, with respect to the universe of discourse.

E Some languages are more easily proved context free than others.