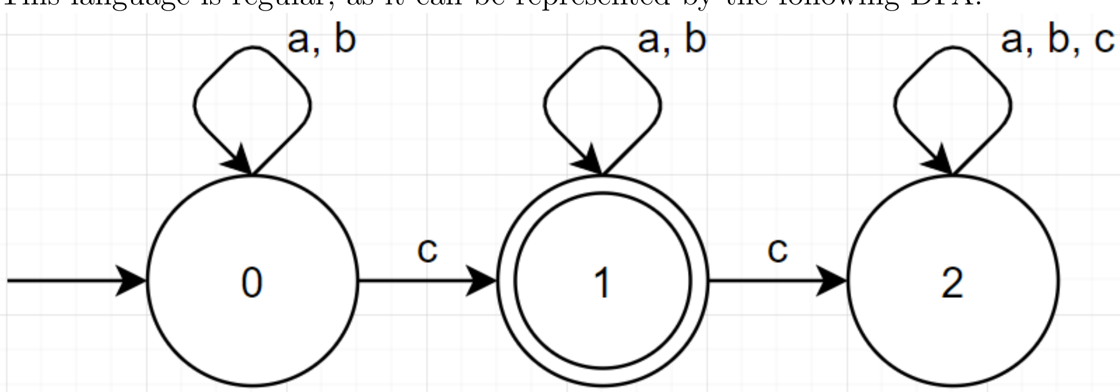
CSCI 3434, CU Boulder | Prof. Rafael Frongillo
Fall 2020 | Problem Set 4
Collaborators: None | Student: Alex Book

- **Do not use external sources** beyond the materials linked to on the course website to solve these problems.
- You are encouraged to work with ($\leq$ 2) other students, but you must **write your solutions independently**.
- Be sure to **list your collaborators** by name clearly at top of your submission, or “no collaborators” if none.
- Recall: letters correspond to the section of the book: [H]omeworks starting page 301, [M]iscellaneous [E]xercises starting page 315.
- Throughout, you may assume the languages $\{a^nb^n : n \geq 0\}$, $\{a^nb^m : n \geq m\}$, $\{a^nba^n : n \geq 0\}$ are context-free but nonregular, and the languages $\{a^nb^na^n : n \geq 0\}$, $\{a^nb^nc^n : n \geq 0\}$, $\{a^nba^nba^n : n \geq 0\}$ are not context-free.

1. Classification - Regular (S8)
ME.37(f,h,k,l,m) You can demonstrate a language is regular using the usual constructions (DFA,NFA,regexp) and do not need to prove that your construction works. You could also use closure properties and known regular languages. To show a language is non-regular, you can use either the pumping lemma or closure properties.

(f) $\{xcy \mid x, y \in \{a, b\}^*\}$

This language is regular, as it can be represented by the following DFA:

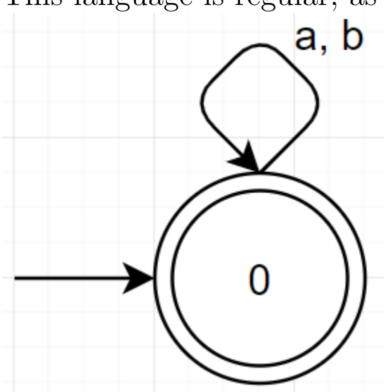


(h) $\{a^n b^m \mid n - m \leq 481\}$

Assume that the given language A is regular.
Therefore, $\sim A$ is also regular.
$\{a^*b^*\}$ is a known regular language, so $\sim A \cap \{a^*b^*\} = \{a^nb^m | n - m > 481\}$ must also be regular.
$\{b^{481}\}$ is regular, as it is finite.
Therefore, the concatenation $\{a^nb^m|n - m > 481\}\{b^{481}\}$ should also be regular.
However, $\{a^nb^m|n - m > 481\}\{b^{481}\} = \{a^nb^m|n \geq m\}$, which is a known non-regular language. So, A must be non-regular.

(k) $L((a^*b)^*a^*)$

This language is regular, as it can be represented by the following DFA:



(l) $\{a^nb^nc^n \mid n \geq 0\}$

Assume the given language A is regular.
For $k \geq 0$, let $x = \epsilon, y = a^k, z = b^kc^k$.
$y = uvw$ and $|v| \neq 0$, so v must contain at least one a.
Take $i = 0 : xuv^iwz = xuv^0wz = a^{k-|v|}b^kc^k \notin A$
Therefore, $A = \{a^nb^nc^n|n \geq 0\}$ is not regular.

(m) {syntactically correct PASCAL programs}

A syntactically correct PASCAL program would need to have balanced parentheses. However, the language of strings with balanced parentheses is defined in the book (p. 131) to be non-regular. Therefore, a syntactically correct PASCAL program is also non-regular.

2. Designing CFGs (S13)
AT problems! (One is ME.73(a).)
Problems completed on Automata Tutor.

3. Normal forms (S12)
AT problems! NOTE: for full credit, **submit your Greibach grammar with your problem set solutions**. AT does not handle Greibach, so it will only test whether the grammar generates the correct language, not whether it is in Greibach normal form.
Problems completed on Automata Tutor.
GNF for 3b:
S $\rightarrow$ d T | a U
T $\rightarrow$ a S
U $\rightarrow$ a V | a
V $\rightarrow$ b W
W $\rightarrow$ b V | b

4. Classification - Context-free (S16)
Classify each language as regular (REG), nonregular but context-free (CF), or not context-free (NCF). Justify your answer. (Note: you can show NCF using closure properties; use of the CFL pumping lemma is optional.)

a. $\{a^k b^\ell a^m b^n : k = m \text{ or } \ell = n\}$

The language $\{a^n b a^n | n \geq 0\}$ is defined as being context-free but non-regular. The language $\{ab^m a | m \geq 0\}$ is clearly a regular language, and is thus also context-free. Therefore, the union of these languages must be context free.
$\{a^n b a^n | n \geq 0\} \cup \{ab^m a | m \geq 0\} = \{a^n b^m a^n | n, m \geq 0\}$
The language $\{b^j | j \geq 0\}$ is clearly a regular language, and is thus also context-free.
Therefore the following concatenation must be context-free:
$\{a^n b^m a^n | n, m \geq 0\}\{b^j | j \geq 0\} = \{a^n b^m a^n b^j | n, m, j \geq 0\}$
This resultant language can be alternately written as: $\{a^k b^\ell a^m b^n : k = m\}$

Swapping a's and b's in the previously-mentioned CFL $\{a^n b^m a^n | n, m \geq 0\}$ yields the following:
$\{b^n a^m b^n | n, m \geq 0\}$
The language $\{a^j | j \geq 0\}$ is clearly a regular language, and is thus also context-free.
Therefore the following concatenation must be context-free:
$\{a^j | j \geq 0\}\{b^n a^m b^n | n, m \geq 0\} = \{a^j b^n a^m b^n | n, m, j \geq 0\}$
This resultant language can be alternately written as: $\{a^k b^\ell a^m b^n : \ell = n\}$

So, the following union must be context-free:
$\{a^k b^\ell a^m b^n : k = m\} \cup \{a^k b^\ell a^m b^n : \ell = n\} = \{a^k b^\ell a^m b^n : k = m \text{ or } \ell = n\}$

Assume $\{a^k b^\ell a^m b^n : k = m \text{ or } \ell = n\}$ is regular. $\{a^* b^* a^*\}$ is a known regular language. So, the intersection of the two languages must be regular:
$\{a^k b^\ell a^m b^n : k = m \text{ or } \ell = n\} \cap \{a^* b^* a^*\} = \{a^k b^\ell a^m | k = m\}$
The language $\{b\}$ is regular, as it is finite. So, the following intersection must be regular: $\{a^k b^\ell a^m | k = m\} \cap \{b\} = \{a^k b a^m | k = m\} = \{a^n b a^n | n \geq 0\}$
However, $\{a^n b a^n | n \geq 0\}$ is a known non-regular language, so $\{a^k b^\ell a^m b^n : k = m \text{ or } \ell = n\}$ must be non-regular.

Therefore, $\{a^k b^\ell a^m b^n : k = m \text{ or } \ell = n\}$ is non-regular but context-free (CF).

b. $\{a^k b^\ell a^m b^n : k = m \text{ and } \ell = n\}$

Assume the given language is context-free.
For $k \geq 0$, let $z = a^k b^k a^k b^k$ such that $z = uvwxy, vx \neq \epsilon,$ and $|vwx| < k$.

Take $i = 2$:
If either v or x contains both a's and b's, then $uv^iwx^iy = uv^2wx^2y$ wouldn't follow the required form of $a^*b^*a^*b^*$.

If v and x both contain only zero or more a's (however either v or x would be required to contain at least one a, as $vx \neq \epsilon$), then $uv^iwx^iy = uv^2wx^2y$ would have one group of a's be larger than the other.

If v and x both contain only zero or more b's (however either v or x would be required to contain at least one b, as $vx \neq \epsilon$), then $uv^iwx^iy = uv^2wx^2y$ would have one group of b's be larger than the other.

If v and v contain zero or more of different characters (however either v or x would be required to contain at least one character, as $vx \neq \epsilon$), then $uv^iwx^iy = uv^2wx^2y$ would have one group of a's larger than the other and/or one group of b's larger than the other.

Therefore, in any case, $uv^iwx^iy = uv^2wx^2y \notin \{a^k b^\ell a^m b^n : k = m \text{ and } \ell = n\}$.

So, $\{a^k b^\ell a^m b^n : k = m \text{ and } \ell = n\}$ is not context-free (NCF).

5. Proofs - Context-free (S17)
H5.2

2. Prove that the CFG

$$S \rightarrow aSb \mid bSa \mid SS \mid \epsilon$$

generates the set of all strings over $\{a, b\}$ with equally many a's and b's. (*Hint*: Characterize elements of the set in terms of the graph of the function $\#b(y) - \#a(y)$ as y ranges over prefixes of x, as we did in Lecture 20 with balanced parentheses.)

First we will show that the given CFG G creates balanced strings.

Proof by induction
Base Case
By the definition of $\xrightarrow[G]{0}$, $S \xrightarrow[G]{0} S$. S is trivially balanced.

Inductive Hypothesis
Assume that for $S \xrightarrow[G]{k} \alpha$, α is balanced.

Inductive Step
$S \xrightarrow[G]{k+1} \alpha$ can be represented as $S \xrightarrow[G]{k} \beta \xrightarrow[G]{1} \alpha$.
By the inductive hypothesis, β is balanced.
There are four possible cases to address for $\beta \xrightarrow[G]{1} \alpha$. In any case, there exist some $\beta_1, \beta_2 \in (N \cup \Sigma)^*$ such that $\beta = \beta_1 S \beta_2$.

a. If $S \rightarrow \epsilon$ was applied, $\alpha = \beta_1\beta_2$. Since β is balanced, so is α.

b. If $S \rightarrow SS$ was applied, $\alpha = \beta_1 SS \beta_2$. Since β is balanced, so is α.

c. If $S \rightarrow aSb$ was applied, $\alpha = \beta_1 aSb \beta_2$. Since β is balanced, so is α.

d. If $S \rightarrow bSa$ was applied, $\alpha = \beta_1 bSa \beta_2$. Since β is balanced, so is α.

Since all of the possible cases result in a balanced string after the last production, we can say that because β is balanced, $\beta \xrightarrow[G]{1} \alpha$ gives us a balanced α.

Therefore, the given CFG G creates balanced strings.

Now we will show that any balanced string can be created by the given CFG G.
We want to show that if x is a balanced string, then $S \xrightarrow[G]{*} x$.
Base case: $|x| = 0$, so $x = \epsilon$ and $S \xrightarrow[G]{*} x$ using $S \to \epsilon$.
Inductive Hypothesis: Assume that for a balanced string x, we have $S \xrightarrow[G]{*} x$.
Inductive Step: For $|x| > 0$, we break the argument into two cases:

a. There exists a balanced proper prefix y of x such that $0 < |y| < |x|$.
In case (a), we have $x = yz$ for some z, $0 < |z| < |x|$, and z is balanced:

$$a(z) = a(x) - a(y) = b(x) - b(y) = b(z)$$

By the inductive hypothesis, we have $S \xrightarrow[G]{*} y$ and $S \xrightarrow[G]{*} z$.
Then we can derive x starting with the production $S \to SS$, then deriving y from the first S, then deriving z from the second S :

$$S \xrightarrow[G]{1} SS \xrightarrow[G]{*} yS \xrightarrow[G]{*} yz = x$$

b. No such prefix exists.
In case (b), we have $x = azb$ or $x = bza$. z is balanced:

$$a(z) = a(x) - 1 = b(x) - 1 = b(z)$$

By the inductive hypothesis, we have $S \xrightarrow[G]{*} z$. We then can respectively combine this derivation with the productions $S \to aSb$ and $S \to bSa$.

$$S \xrightarrow[G]{1} aSb \xrightarrow[G]{*} azb = x$$
$$S \xrightarrow[G]{1} bSa \xrightarrow[G]{*} bza = x$$

Therefore, if x is a balanced string, then $S \xrightarrow[G]{*} x$.