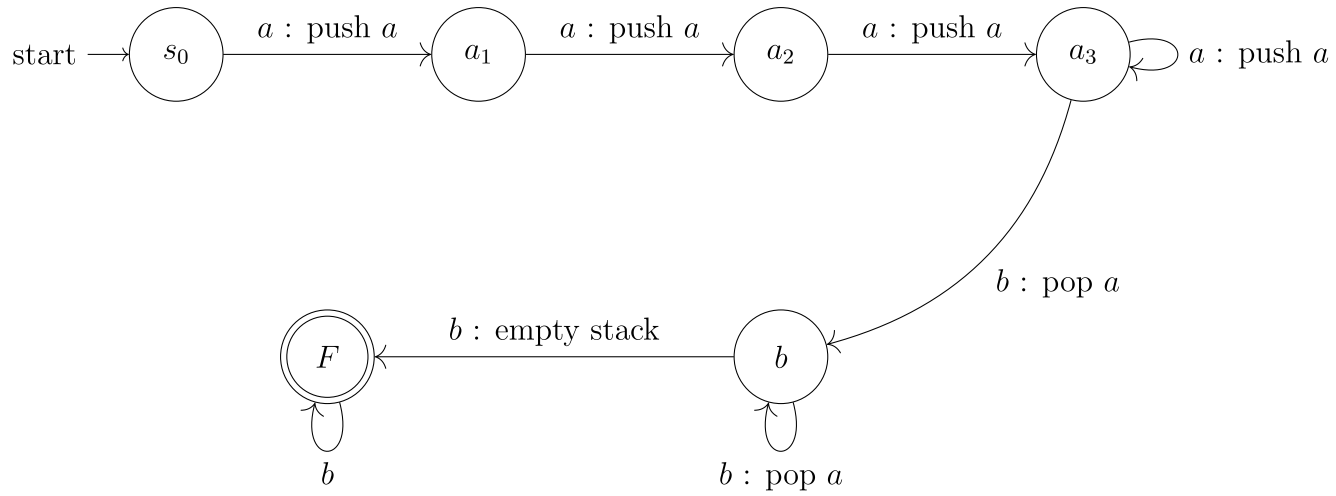
Problem 1.

Define a NPDA for the language $L = \{a^n b^m : m, n \in \mathbb{N}, n \geq 3, m > n\}$.

Solution

The NPDA here is pretty simple, $n \geq 3$ so we need at minimum 3 a's in our string, so read those first, we can then add as many a's as we want, so we optionally push those onto the stack. Once we are done reading those we need more b's, so start reading b's until we we have an empty stack. Once the stack is empty we know number of a's is equal to number of b's, so if we read one our condition is met and we can enter our final state. From there, there is no further limit on the number of b's (other than they must be finite of course), so we can continue to read b's with no consequences (and no stack modifications) once in the final state.



Problem 2.

Consider the language

$$L = \{w \in \{a, b\}^* : \text{the longest run of } a\text{'s in } w \text{ is longer than any run of } b\text{'s in } w\}$$

Prove that L is not context-free.

Solution

Proof. by contradiction
Assume L is context-free. Let $p \geq 1$ be the pumping length for L by the Pumping Lemma for Context-Free Languages.
Define $w = b^{p-1}a^p b^{p-1}$, note that the longest run of a's $= p$ and the longest run of b's $= p - 1$, thus $w \in L$. We now apply the pumping lemma on this string. Let $w = uvwxy$ by this fact. Note that as $|vwx| \leq p$ by construction of w, if $b \in vwx$ then all b's in vwx must come from one side of side of the string w. This is because a^p is long enough that vwx will not be able to reach the other set of b's without violating $|vwx| \leq p$. Therefore, when we pump, we are only modifying one run of b's (or none at all).

case $|vx|_a \geq |vx|_b$:
Pump for $n = 0$, thus we are removing at least 1 a from our string, we may or may not be removing any b's, but if we are removing b's, we are only removing them from one side of our string, the other side will remain untouched. By removing at least one a, our longest run of a's is now maximally $p - 1$. As noted before, one run of b's must be remaining untouched, thus our longest run of b's is still also $p - 1$. This violates the assumptions of our language, thus $uv^0wx^0y \notin L$ contradicting the pumping lemma.
case $|vx|_a < |vx|_b$:
Pump for $n = 2$, we have added at least one additional b over a's, if we added k a's, we know have a longest run of $p + k$, note that all a's are next to each other in our string. We have also pumped one side of b's, we have added at least $k + 1$ b's, so this particular run of b's is now minimally of size $p - 1 + k + 1 = p + k$, thus our longest run of b's is now $p + k$ equal to our longest run of a's. This violates the assumptions of our language thus $uv^2wx^2y \notin L$ contradicting the pumping lemma.
Both exhaustive cases lead to contradictions, thus our initial assumption must be incorrect and we conclude that L is not context-free.

□

Problem 3.

Prove or disprove that the following language is context-free:

$$L = \{\alpha 2 \beta : \alpha, \beta \in 1(0+1)^*, [\alpha]_2 < [\beta]_2\}$$

where $[x]_2$ is the numerical value of the string x interpreted as a positive number in base 2.

Solution

Proof. by contradiction
Assume L is context-free. Let $p \geq 1$ be the pumping length for L by the Pumping Lemma for Context-Free Languages.
Define $w = 10^{p-1}1^{p-1}0\ 2\ 10^{p-1}1^{p}$. Note that $[\alpha]_2 + 1 = [\beta]_2 \Rightarrow [\alpha]_2 < [\beta]_2$ therefore $w \in L$. We can now apply the pumping lemma to this string. Let $w = uvwxy$ by this fact.

Note that the number of bits in $\alpha = \beta = 2p$.
Further note that we must pump both sides of the string, if we pump only α, we can pump it larger than the β (as it will have more bits), if we pump only β, we simply pump it for $n = 0$ and it will be less than α (as it will have less bits). We can extend this assumption to state that both sides of our pumping (x and v) must be equivalent in size (and we must pump something) we will denote this size k, that is $|x| = |v| = k \geq 1$. Again, if this is not the case, we simply pump such that α has more bits, or pump $n = 0$ such that β has less. Also observe that 2 cannot be pumped as the string necessarily must only have one 2. From these observations, it must be the case that $2 \in w$ (else we would only be pumping α or only β) and $v \subset \alpha \wedge x \subset \beta$. We also cannot pump the initial 1 of β, as that will leave a β starting with a 0, which is not allowed. To pump with a leading 1, we need to follow v with a 1, but the next 1 in β cannot be reached, as we only have $p/2$ bits to work with ($|xv| \leq p \wedge |x| = |v| \Rightarrow |v| \leq p/2$) in β which has $p-1$ 0's before that 1 can be reached.

Note, that as $|vwx| \leq p$ we must then be pumping $1^{k-1}0$ in v and 0^k in x. Therefore, after pumping n times, our string now looks like

$$10^{p-1}1^{p-k}(1^{k-1}0)^n\ 2\ 1(0^k)^n 0^{p-k-1}1^p$$

Our strings still have the same length, but α has $p-1$ leading 0's (after the initial 1 bit), where as β now has $kn + p - k - 1 = k(n-1) + p - 1$, thus for $n > 1$ because α and β have the same number of bits, and α has it's second leading 1 bit before β will, $[\alpha]_2 > [\beta]_2$. Therefore the pumping lemma fails and the language is not context-free. □