HW8: One Way or Another

Solution Key

Include your full name*, CS login, and the problem number(s) on each piece of paper you hand in, and please staple your pages together before handing in.*

While collaboration is encouraged in this class, please remember not to take away notes from collaboration sessions other than your scheduled lab section.

Problem 1

Describe a pushdown automaton for the language $L = \{0^r 1^s \mid r \leq s \leq 2r\}$. An English description of what the PDA pushes onto the stack when and what it pops off when is completely acceptable.

A PDA that recognizes L would have a nondeterministic control unit that does the following (note that an "area" would contain multiple states):

- In the start area, if there is no more input, accept (this corresponds to $w = \varepsilon$). If the input is a 1, reject (this corresponds to a string starting with 1), and if it sees a 0, nondeterministically push either one or two 1's onto the stack, and transition to area 2.
- In area 2, if the input ends it rejects. If it sees a 0 it nondeterministically pushes either one or two 1's onto the stack, and go to the beginning of area 2. If it sees a 1, it pops a 1 off the stack and transitions to area 3.
- In area 3, if the input ends, check if the stack is empty. If so, accept, otherwise reject. If the input is a 0, reject. If the input is a 1, pop a 1 off the stack (or reject if the stack is empty), and go to the beginning of area 3.

Problem 2

Let $G = (\mathcal{N}, \mathcal{T}, \mathcal{R}, \mathrm{s})$ be the context-free grammar defined below:

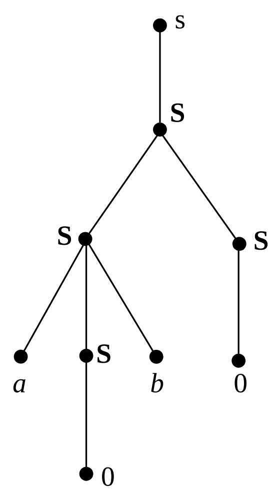
$$\begin{aligned}\mathcal{N} &= \{\mathbf{S}\}\\ \mathcal{T} &= \{a, b, 0\}\\ \mathcal{R} &= \{\mathbf{S} \to 0, \mathbf{S} \to \mathbf{SS}, \mathbf{S} \to a\mathbf{S}b\}\\ \mathrm{s} &= \mathbf{S}.\end{aligned}$$

Generate a parse tree for the string $a0b0$ using the above rules. Please show each intermediate step in your solution.

The following production rules parse $a0b0$:

$$\mathrm{s} \to \mathbf{S} \to \mathbf{SS} \to a\mathbf{S}b\mathbf{S} \to a0b\mathbf{S} \to a0b0.$$

This is the parse tree it generates:



The following questions are lab problems. *Please remember to prepare a solution for your assigned problem before going to your lab section.*

Lab Problem 1

Show that the following language $L = \{ww \mid w \in \{0,1\}^*\}$ is not context-free.

Hint: *The intersection of a context-free language and a regular language is context-free.*

Consider the language $L' = L \cap 0^*1^*0^*1^*$. Note that $L' = \{0^i1^j0^i1^j \mid i, j \geq 0\}$ and is context free if L is context free. If L is regular, then it has a pumping length m, so consider the string $w = 0^n1^n0^n1^n \in L$ where $n = 2^m$ where m is the pumping length. We can split $w = rstuv$ with $|stu| \leq 2^m$, so we know that stu is entirely contained in two consecutive of the four $0^n, 1^n, 0^n, 1^n$. If stu is entirely contained in one of the sets, then $rs^2tu^2v \notin L$ since one of the four is now two long. If s is in one and u is in another, then $rs^2tu^2v \notin L$ because two consecutive subsets are being extended, but the other corresponding subsets are not.

Therefore, we see that $0^n1^n0^n1^n \in L$ cannot be pumped in L', so L' is not context-free, and thus L is not context-free.

Lab Problem 2

Consider the following language L:

$$\begin{aligned} L = \Big\{ s_1\#s_2 \quad | \quad & s_1 \in \{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}\}^*, s_2 \in \{0,1\}^*, \\ & \textit{and if } s_1 = \begin{bmatrix}x_0\\y_0\end{bmatrix}\begin{bmatrix}x_1\\y_1\end{bmatrix}\cdots\begin{bmatrix}x_n\\y_n\end{bmatrix}, \textit{ then } s_2 = x + y \\ & \textit{where } x = \Sigma_{i=0}^n x_i 2^i \textit{ and }, y = \Sigma_{i=0}^n y_i 2^i \Big\} \end{aligned}$$

In other words, for a string $s_1\#s_2$, we interpret $s_1 = \begin{bmatrix}x\\y\end{bmatrix}$ as two binary integers x, y with least-significant bit first, and s_2 as a binary integer with most-significant bit first. $s_1\#s_2 \in L$ whenever $s_2 = x + y$ (i.e., the sum of the two numbers represented as pairs equals the number after the #).

To ensure that all additions have a unique representation, we require the following:

- $s_2 \in \varepsilon \cup 1\{0,1\}^*$ *(s_2 either is empty or begins with a 1.)*
- $s_1 \in \varepsilon \cup \{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}\}^*\{\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}\}$
  (x and y cannot both begin with 0s.)

Some examples to help you understand the definition of the language:

- $\# \in L$

- $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\#1000 \in L$ *because* $101 + 011 = 1000$
- $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}\#100 \notin L$ *because* x *and* y *both begin with extraneous 0s.*
- $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\#01000 \notin L$ *because* s_2 *begins with extraneous 0s.*
- $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\#1001 \notin L$ *because* $101 + 011 \neq 1001$

a. *Prove that* L *is not regular.*

b. *Design a CFG for* L.

a. Suppose L is regular, and p is its pumping length. Consider the string $s = \begin{bmatrix}0\\0\end{bmatrix}^p \begin{bmatrix}1\\0\end{bmatrix}\#10^p$. This string is in L because x and y do not both begin with extraneous 0s, s_2 does not begin with extraneous 0s, and $10\cdots0 + 00\cdots0 = 10\cdots0$, where $0\cdots0$ has p 0s.

Since $s \in L$, there are some x, y, z such that $s = xyz$, $|xy| \leq p$, $|y| > 0$, and for all $i \geq 0$, $xy^iz \in L$. Since $|xy| \leq p$ and $|y| > 0$, $y = \begin{bmatrix}0\\0\end{bmatrix}^k$ for some $k > 0$. Then $xyyz \in L$:

$$\begin{bmatrix}0\\0\end{bmatrix}^{p+k} \begin{bmatrix}1\\0\end{bmatrix}\#10^p \in L, \text{ for } k > 0.$$

But here $x = 2^{p+k}$, and $s_2 = 2^p$, and so $x + y > s_2$, so $xyyz \notin L$, so L is not regular.

b. Here is a CFG for L:

$$
\begin{array}{llll}
S \to \# & & & \\
S \to \begin{bmatrix}0\\1\end{bmatrix} S\,1 & & S \to \begin{bmatrix}1\\0\end{bmatrix} S\,1 & \\
S \to \begin{bmatrix}0\\0\end{bmatrix} U\,0 & & S \to \begin{bmatrix}1\\1\end{bmatrix} C\,0 & \\
 & & & \\
U \to \begin{bmatrix}0\\1\end{bmatrix} S\,1 & & U \to \begin{bmatrix}1\\0\end{bmatrix} S\,1 & \\
U \to \begin{bmatrix}0\\0\end{bmatrix} U\,0 & & U \to \begin{bmatrix}1\\1\end{bmatrix} C\,0 & \\
 & & & \\
C \to \#1 & & & \\
C \to \begin{bmatrix}0\\1\end{bmatrix} C\,0 & & C \to \begin{bmatrix}1\\0\end{bmatrix} C\,0 & \\
C \to \begin{bmatrix}0\\0\end{bmatrix} S\,1 & & C \to \begin{bmatrix}1\\1\end{bmatrix} C\,1 &
\end{array}
$$

Lab Problem 3

Throughout this problem, $L_{01} = \{0^n 1^n \mid n \geq 0\}$.

a. *Consider a Turing Machine T that does the following:*

- *T takes as input $\langle M, w\rangle$—a description of a Turing Machine M*[1] *and a string w.*
- *T prints to its tape as output a description of a new machine M' where $L(M') = L_{01}$ if M accepts w, and $L(M') = \emptyset$ otherwise.*

Give an English description of what such a T would do to produce the desired output from input $\langle M, w\rangle$, and explain why the M' described in its output satisfies the desired properties.

Hint: *Keep in mind that since T is just printing a* description *of another Turing Machine it will always halt, whereas M' does not always have to.*

b. *Define the language A_{TM} as follows:*

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a Turing machine that accepts } w\}$$

[1] A description of a TM describes the operations of the TM in a way that allows another TM to simulate the described TM.

Note that A_{TM} is undecidable. Using part (a) and this fact, show that the language $B = \{\langle M \rangle \mid L(M) = L_{01}\}$ is not decidable. You need to show that if B is decidable, then so is A_{TM}. In order to do this, suppose that there exists a Turing machine R that decides B. Then build a Turing machine that decides A_{TM}, using as subroutines both R and the Turing machine you designed in part (a).

(a) The TM T acts as follows: on input $\langle M, w \rangle$, T will print to its tape the description of a machine M', which is described below. T halts always as it is only printing out a description.

We saw in class that L_{01} is decidable by a TM, let its decider be called M_L. Define M' as follows: On any input s, M' will simulate M on w. If M accepts w, M' will simulate M_L on s and will do as M_L; else it will reject.

If M accepts w, $L(M') = L_{01}$ as required since in this case M' simulates M_L. If M rejects w or does not halt on w, M' will also not accept s, in which case M' accepts nothing and so $L(M') \neq L_{01}$ since L_{01} is not the empty language.

(b) Assume for the purposes of contradiction that a decider R exists such that $L(R) = \{\langle M \rangle | L(M) = L_{01}\}$. Using R and T from above, a decider $M_{A_{TM}}$ for A_{TM} can be constructed as follows: On input (M, w), $M_{A_{TM}}$ passes (M, w) to T. T returns $\langle M' \rangle$ such that $L(M') = L_{01}$ if and only if M accepts w. $M_{A_{TM}}$ then passes $\langle M' \rangle$ to R, and accepts if and only if R accepts and rejects otherwise.

$M_{A_{TM}}$ will decide A_{TM}. The only time $M_{A_{TM}}$ accepts is when R accepts. Since R is assumed to decide $B = \{\langle M \rangle | L(M) = L_{01}\}$, this happens when $L(M') = L_{01}$. And $L(M') = L_{01}$ iff M accepts w. Whenever, $M_{A_{TM}}$ does not accept, it will reject since both T and R always halt.

Since A_{TM} is undecidable, this is a contradiction.