CS 381 Homework 8 solutions

October 19, 2000

Problem 18.1 Show $G = S \to aSb|bSa|SS|\epsilon$ generates the set of all strings over $\{a, b\}$ with equal number of a's and b'.

First we show inductively if $S \to^n x$ then x has a balanced number of a's and b's.

Our base case is a length 1 derivation. The only possible x with a length 1 derivation is $S \to \epsilon$, and clearly ϵ has the same number of a's as b's.

Next, we assume that if $S \to^k x$ for any $k < n$ then x has a balanced number of a's and b's. Now suppose $S \to^{n+1} x$. Then there are three cases:

- $S \to aS'b \to^n x$. In this case we know $S' \to^n y$, where y is balanced, by our inductive hypothesis. Thus $x = ayb$, and x is balanced as well.
- $S \to bS'a \to^n x$. In this case we know $S' \to^n y$, where y is balanced, by our inductive hypothesis. Thus $x = bya$, and x is balanced as well.
- $S \to S'S'' \to^n x$. In this case we know $S' \to^l y$ and $S'' \to^m z$, and $l + m = n$. We also know that $l > 0$ and $m > 0$ because it takes at least one step to reduce a non-terminal to a terminal. Thus $l < n$ and $m < n$, so our inductive hypothesis applies: y and z are balanced. Thus x is balanced.

Now we wish to show that $\forall\, x \in \{a, b\}^*, \#a(x) - \#b(x) = 0$ then there is a derivation $S \to^* x$. This time we will induct on the length of the string.

The base case is a string of length 0, or $x = \epsilon$. Then the production $S \to \epsilon$ generates x.

Inductively, we assume that a string x with $|x| = k$ is balanced for any even $k < n$. Now let us look at a string x with $|x| = n$, where n is even. Consider the graph of $\#a(y) - \#b(y)$ for y a prefix of x of length 0 to n. Since x is balanced, the graph must start and end at 0. There are three possibilities: The graph is everywhere positive, the graph is everywhere negative, or the graph is both positive and negative.

- The graph is all positive. This means that for a substring of length 1 we have $\#a(y) - \#b(y) = 1$, or that $x = azb$ for some balanced string z. In this case, we also know that $|z| < n$, so we know that there is a derivation $S' \to^* z$. Thus there is a derivation $S \to aS'b \to^* z$.
- The graph is all negative. This means that for a substring of length 1 we have $\#a(y) - \#b(y) = -1$, or that $x = bza$ for some balanced string z. In this case, we also know that $|z| < n$, so we know that there is a derivation $S' \to^* z$. Thus there is a derivation $S \to bS'a \to^* z$.
- The graph is both positive and negative. In this case, because the graph is discrete and varies by $+1$ or -1 at each step, we know it crosses 0 somewhere in the middle. Thus $x = yz$ for some y and z that are balanced (because their graphs both start and end at 0), and we also know $0 < |y| < |x|$ and $0 < |z| < |x|$, because the slope of the difference graph is ± 1. Thus there are derivations $S' \to^* y$ and $S'' \to^* z$, and thus the derivation $S \to S'S'' \to^* x$ is valid.

Problem 18.2 A CFG that produces PAREN_2 is: $S \to (S)|[S]|SS|\epsilon$.

Part 1: If $S \to^* x$ then $x \in \text{PAREN}_2$. We will induct on the length of the derivation. The base case is a length 1 derivation, and the only length 1 derivation producing a string is $S \to \epsilon$. $\epsilon \in \text{PAREN}_2$ by definition. We now inductively assume that if $S \to^k x$ for any $k \leq n$ then $x \in \text{PAREN}_2$.

- Case 1. $S \to (S') \to^n (x)$. By our inductive hypothesis, we know $x \in \text{PAREN}_2$, and by rule (ii), we know $(x) \in \text{PAREN}_2$.
- Case 2. $S \to [S'] \to^n [x]$. By our inductive hypothesis, we know $x \in \text{PAREN}_2$, and by rule (ii), we know $[x] \in \text{PAREN}_2$.
- Case 3. $S \to S'S'' \to^n xy$. By our inductive hypothesis, we know $x \in \text{PAREN}_2$ and $y \in \text{PAREN}_2$, and by rule (iii), we know $xy \in \text{PAREN}_2$.

Part 2: If $x \in \text{PAREN}_2$ then there is a derivation $S \to^* x$. We will induct on the reason that $x \in \text{PAREN}_2$. Base case: $\epsilon \in \text{PAREN}_2$, and $S \to \epsilon$. We assume that if k rules or fewer are used to put a particular string in PAREN_2 then we have a derivation $S \to x$.

- Case 1. $(x) \in \text{PAREN}_2$ because $x \in \text{PAREN}_2$. In this case we use the production $S \to (S')$, and we know $S' \to^* x$ by our inductive hypothesis.
- Case 2. $[x] \in \text{PAREN}_2$ because $x \in \text{PAREN}_2$. In this case we use the production $S \to [S']$, and we know $S' \to^* x$ by our inductive hypothesis.
- Case 3. $xy \in \text{PAREN}_2$ because $x \in \text{PAREN}_2$ and $y \in \text{PAREN}_2$. In this case we use the production $S \to S'S''$, and we know $S' \to^* x$ and $S'' \to y$ by our inductive hypothesis.

Problem 19.1

Next verse, same as the first.

$S \to aB|bA$ $A \to aS|bAA|a$ $B \to bS|aBB|b$

Step 1. If $S \to^* x$ then x is balanced. We have the following three inductive hypothesis to help us: $S \to^* x \Rightarrow \#a(x) = \#b(x)$, $A \to^* x \Rightarrow \#a(x) = \#b(x) + 1$, $B \to^* x \Rightarrow \#a(x) = \#b(x) - 1$. The base cases are easily verified: $A \to a$, and $\#a(a) = \#b(a) + 1$; $B \to b$ and $\#a(b) = \#b(b) - 1$, and $S \to aB \to ab$ and $\#a(ab) = \#b(ab)$ (similar for ba). Inductively:

- $S \to aB \to^n x$: $x = ay$ for some y, and we know $B \to^n y$, so we know $\#a(y) = \#b(y) - 1$. So $\#a(x) = 1 + \#a(y) = 1 + \#b(y) - 1 = \#b(y)$.
- $S \to bA \to^n x$: $x = by$ for some y, and we know $B \to^n y$, so we know $\#a(y) = \#b(y) + 1$. So $\#b(x) = 1 + \#b(y) = 1 + \#a(y) - 1 = \#a(y)$.
- $A \to aS \to^n x$: $x = ay$ for some y, and we know $S \to^n y$, so we know $\#a(y) = \#b(y)$. So $\#a(x) = 1 + \#a(y) = 1 + \#b(y)$.
- $A \to bAA \to^n x$: $x = byz$ for some y, z, and we know $A \to^l y$ and $A \to^m z$, so we know $\#a(y) = \#b(y) + 1$ and $\#a(z) = \#b(z) + 1$. So $\#b(x) = 1 + \#b(y) + \#b(z) = 1 + \#a(y) - 1 + \#a(z) - 1 = \#a(x) - 1$.
- $B \to bS \to^n x$: $x = by$ for some y, and we know $S \to^n y$, so we know $\#a(y) = \#b(y)$. So $\#b(x) = 1 + \#b(y) = 1 + \#a(y)$.
- $B \to aBB \to^n x$: $x = ayz$ for some y, z, and we know $B \to^l y$ and $B \to^m z$, so we know $\#b(y) = \#a(y) + 1$ and $\#b(z) = \#a(z) + 1$. So $\#a(x) = 1 + \#a(y) + \#a(z) = 1 + \#b(y) - 1 + \#b(z) - 1 = \#b(x) - 1$.

Step 2. If x is balanced then there is a derivation $S \to^* x$.

Part of our hypothesis will also be that if $\#a(x) - \#b(x) = 1$ then there is a production $A \to^* x$, and if $\#a(x) - \#b(x) = -1$ then there is a production $B \to^* x$.

Base cases: ab can be generated by $S \to aB \to ab$, and ba can be generated by $S \to bA \to ba$. a can be generated by $A \to a$, and b can be generated by $B \to b$.

Inductive cases: Assume that any balanced string of length n or smaller can be generated by S, and any off-by-one string of length $n - 1$ or smaller can be generated by A or B, depending on which way it's off. Now consider a string x of length $n + 2$. There are several cases:

- $x = aay$ for some y with 2 more b's than a's. This means that $y = pq$, where each of p and q are off by one (two off-by-ones will make an off by 2). In this case we can generate x as follows: $S \to aB \to aaBB'$. We know inductovely that $B \to^* p$ and $B' \to^* q$.
- $x = aby$ for some balanced y. Then the derivation $S \to aB \to abS' \to^* aby$ will work, and we know inductively that y can be generated by S'.
- $x = bay$ for some balanced y. Then the derivation $S \to bA \to baS' \to^* bay$ will work, and we know inductively that y can be generated by S'.
- $x = bby$ for some y with 2 more a's than b's. This means that $y = pq$, where each of p and q are off by one. In this case, we can generate x as follows: $S \to bA \to bbAA'$. We know inductovely that $A \to^* p$ and $A' \to^* q$.

Problem 19.2

$$\begin{aligned} S &\to LR|L'R'|LA|L'A'|SS \\ A &\to SR \\ A' &\to SR' \\ L &\to (\\ L' &\to [\\ R &\to) \\ R' &\to] \end{aligned}$$

Let's consider an obviously equivalent grammar: $S \to () \,|\, [] \,|\, [S] \,|\, (S)$, which is derived by substituting from the non-terminals.

This is clearly the same as the previous grammar for PAREN_2, except this one will not produce ϵ. So the previous proof applies.

Problem 20.1

a) Suppose $B = \{b(n)\$b(n+1)|n \geq 1\}$ is a CFL. Then the Pumping Lemma applies, i.e. there exists k such that the string $z = 10^k0\$10^k1$ (which is in B) can be broken into 5 strings $z = uvwxy$, with $|vx| \geq 1$, $|vwx| \leq k$, and $uv^iwx^iy \in B \forall i \geq 0$. There are several possibilities for the string vwx:

- either of v or x contains the symbol '\$', in which case repeating them more than once gives a string not in B that is supposed to be;
- one of v or x contains only 0's (either from before or after the \$, but not both, because $|vwx| \leq k$), in which case repeating them more than once gives a string with too many 0's either to the left or right of the \$;
- one of v or x contains a 1 (one one or the other side of the \$), in which case repeating them more than once gives a string with too many 1's one one side.

b) Show $\{\textbf{rev } b(n)\$b(n+1)|n \geq 1\}$ is a CFL. We do this by creating a PDA for the language. Our stack consists of symbols from $\{0, 1\}$, plus a bottom-of-stack symbol $\perp$. We accept on empty stack, and start in state q_1. The symbol S below is being used to match whatever is on top of the stack, to save writing.

1. $((q_1, 0, S), (q_0, S1))$
2. $((q_1, 1, S), (q_1, S0))$
3. $((q_0, 0, S), (q_0, S0))$

4. $((q_0, 1, S), (q_0, S1))$
5. $((q_0, \$, S), (q_2, S))$
6. $((q_1, \$, S), (q_2, S1))$
7. $((q_2, 0, 0), (q_2, \epsilon))$
8. $((q_2, 1, 1), (q_2, \epsilon))$
9. $((q_2, \epsilon, \bot), (q_2, \epsilon))$

Basically, the state q_1 indicates a carry, and q_0 indicates no carry as we do binary arithmetic on the input. As we read in **rev** $b(n)$, we add 1 to it. Then when we read the \$, we begin checking that the next number is $b(n+1)$.