

CIS 262 Fall 2009: Solutions to Homework 3

Problem 1

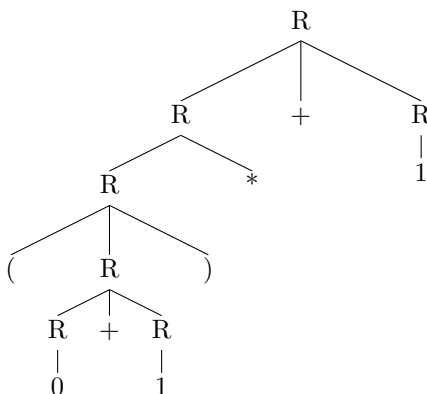
Consider the grammar G with a single variable R , which is also the start variable, over the alphabet $\{0, 1, +, *, (,)\}$:

$$R \rightarrow 0 \mid 1 \mid R* \mid R + R \mid (R)$$

1. Show a left-most derivation of the word $0 + (1 * + 0)$.

$$\begin{aligned} R &\Rightarrow_{\text{lm}} R + R \Rightarrow_{\text{lm}} 0 + R \Rightarrow_{\text{lm}} 0 + (R) \\ &\Rightarrow_{\text{lm}} 0 + (R + R) \Rightarrow_{\text{lm}} 0 + (R * + R) \Rightarrow_{\text{lm}} 0 + (1 * + R) \\ &\Rightarrow_{\text{lm}} 0 + (1 * + 0) \end{aligned}$$

2. Show a parse tree for the word $(0 + 1) * + 1$.



3. Describe in words the language $L(G)$.

$L(G)$ contains strings of regular expressions over $\{0, 1\}$ without concatenation.

Problem 2

Consider the language $L_1 = \{0^i 1^j \mid i \leq j \leq 2i\}$. Give a context-free grammar G for L . Prove that your construction is correct. That is, prove: a word w belongs to L if and only if it belongs to $L(G)$.

A grammar for L is $G = (\{S\}, \{0, 1\}, P, S)$ where the production rule P is

$$S \rightarrow \epsilon \mid 0S1 \mid 0S11.$$

We claim that $w \in L$ if and only if $w \in L(G)$ ($S \xrightarrow{*} w$) and thus $L(G) = L$.

Proof. In the forward direction (if), we show that if $w = 0^i 1^j$, $i \leq j \leq 2i$ then $w \in L(G)$ by strong induction on $|w|$. Note strong induction is necessary because in the induction step we reduce the size of w by more than one. We take our claim as our inductive hypothesis.

Basis In all cases we only consider $w \in L$. Otherwise, the claim holds vacuously.

- $|w| = \epsilon$: $S \rightarrow \epsilon$ is a production so $S \xrightarrow{*} w$.
- $|w| = 1$: w must be either $0^1 1^0$ or $1^0 0^1$ but neither strings are in L (both violate $i \leq j \leq 2i$).

- $|w| = 2$: w can only be $0^i 1^j$. The productions $S \rightarrow \epsilon$ and $S \rightarrow 0S1$ allow us to show $S \Rightarrow 0S1 \Rightarrow 01$.

Induction Consider $w = 0^i 1^j$ with $|w| \geq 3$. Because $w \in L$, w has the form $w = 0^i 1^j$, $i \leq j \leq 2i$. Either $i = j$ or $i < j$. In the first case, consider $w = 0x1$ where $x = 0^{i-1} 1^{j-1}$. Because $i-1 \leq j-1 \leq 2(i-1)$ our inductive hypothesis holds for x and gives us $S \xRightarrow{*} x$. We can apply $S \rightarrow 0S1$ to conclude $S \Rightarrow 0S1 \xRightarrow{*} w$. In the second case, consider $w = 0x11$ where $x = 0^{i-1} 1^{j-2}$. Because $i < j$ it follows that $i-1 \leq j-2 \leq 2(i-1)$ and thus our inductive hypothesis gives us $S \xRightarrow{*} x$. We can apply $S \rightarrow 0S11$ to conclude $S \Rightarrow 0S11 \xRightarrow{*} w$.

In the backwards direction (only if), we show that if $w \in L(G)$ then $w = 0^i 1^j$, $i \leq j \leq 2i$ by induction on the number of steps to derive $S \xRightarrow{*} w$. We again take our claim as our inductive hypothesis.

Basis In a single step we can only employ the production $S \rightarrow \epsilon$ which means $S \xRightarrow{*} \epsilon \in L$.

Induction Consider a $(n+1)$ -step derivation $S \xRightarrow{*} w$. Since the derivation consists of at least one step it can take the form $S \Rightarrow 0S1 \xRightarrow{*} 0x1 = w$ or $S \Rightarrow 0S11 \xRightarrow{*} 0x11 = w$. In the first case, $S \xRightarrow{*} x$ takes n steps so the inductive hypothesis applies and gives us $x = 0^i 1^j$ with $i \leq j \leq 2i$. The first step of the derivation adds one 0 to the front and one 1 to x so $w = 0^{i+1} 1^{j+1}$ and $i+1 \leq j+1 \leq 2(i+1)$ thus $w \in L$.

The second case follows similarly except we conclude $w = 0^{i+1} 1^{j+2}$ and $i+1 \leq j+2 \leq 2(i+1) = 2i+2$ thus $w \in L$.

□

Problem 3

Prove that every regular language can be captured by a context-free grammar. For this, consider a DFA $A = (Q, q_0, F, \delta)$ (over Σ), define a grammar G that accepts the same language as A . For each state q of A , G will have a corresponding nonterminal. Figure out the start symbol and productions of G , and prove that $L(G) = L(A)$.

We define the grammar $G = (V, T, P, S)$ as

$$\begin{aligned} V &= \{Q_i \mid q_i \in Q\} \\ T &= \Sigma \\ P &= \{Q_i \rightarrow aQ_j \mid \delta(q_i, a) = q_j\} \cup \{Q_i \rightarrow \epsilon \mid Q_i \in F\} \\ S &= Q_0 \end{aligned}$$

where the states of A correspond to variables in G (with q_i as a state of A and Q_i as the corresponding variable in G), the transitions correspond to the production rules, and the accepting states correspond to ϵ -productions.

We claim that $w \in L(A)$ if and only if $w \in L(G)$ and thus $L(A) = L(G)$.

Proof. In the forward direction (if) we show that if $w \in L(A)$ ($\hat{\delta}(q_0, w) \in F$) then $w \in L(G)$ ($Q_0 \xRightarrow{*} w$). We proceed by induction on $|w|$ but we must first generalize our inductive hypothesis to account for strings that are partially processed by A (and thus may or may not be in $L(A)$).

Our inductive hypothesis states that if $\hat{\delta}(q_i, w) = q_j$ then $Q_i \xRightarrow{*} wQ_j$.

Basis $|w| = 0$ and thus $w = \epsilon$. $\hat{\delta}(q_i, \epsilon) = q_i$ by the definition of $\hat{\delta}$ and $Q_i \xRightarrow{*} Q_i$ by the definition of $\xRightarrow{*}$ (as $\xRightarrow{*}$ is a reflexive relation).

Induction $w = ax$. $\hat{\delta}(q_i, w) = \hat{\delta}(\delta(q_i, a), x) = \hat{\delta}(q_j, x) = q_k$ where $\delta(q_i, a) = q_j$. x is smaller than w by one character so our inductive hypothesis applies to x and gives us $Q_j \xRightarrow{*} xQ_k$. Since $\delta(q_i, a) = q_j$ the production $Q_i \rightarrow aQ_j$ must exist. Therefore, we can derive

$$Q_i \Rightarrow aQ_j \xRightarrow{*} axQ_k.$$

To prove the original claim, we note that our inductive hypothesis allows us to conclude that $\hat{\delta}(q_0, w) = q_f$ implies $Q_0 \xRightarrow{*} wQ_f$. But by definition $q_f \in F$ so $Q_f \rightarrow \epsilon$ is a production and thus $Q_0 \xRightarrow{*} wQ_f \Rightarrow w$.

In the backwards direction (only if) we prove if $Q_0 \xRightarrow{*} w$ then $\hat{\delta}(q_0, w) \in F$. We again proceed by induction on steps of a derivation produced by G . Like the previous induction, we must generalize our induction hypothesis but in this case we do so to account for each of the possible variables Q_i in G .

Our inductive hypothesis states that if $Q_i \xRightarrow{*} wQ_j$ then $\hat{\delta}(q_i, w) = q_j$.

Basis In a derivation with one step we can derive $Q_i \Rightarrow aQ_j$ if $Q_i \rightarrow aQ_j$ is a production. But this means that $\delta(q_i, a) = q_j$ must be a transition and thus $\hat{\delta}(q_i, a) = q_j$.

Induction Consider a $(n+1)$ -step derivation of $w = ax$ of the form $Q_i \Rightarrow aQ_j \xRightarrow{*} axQ_k$. Clearly $Q_j \xRightarrow{*} xQ_k$ is a derivation of n steps so our inductive hypothesis gives us $\hat{\delta}(q_j, x) = q_k$. Since we employed $Q_i \rightarrow aQ_j$ in the first step of the derivation then $\delta(q_i, a) = q_j$. Thus $\hat{\delta}(q_i, ax) = \hat{\delta}(\delta(q_i, a), x) = q_k$.

To prove our original claim, we note that the only way we can derive $Q_0 \xRightarrow{*} w$ is if $Q_0 \xRightarrow{*} wQ_i$ and $Q_i \rightarrow \epsilon$. From our inductive hypothesis, we know that $Q_0 \xRightarrow{*} wQ_i$ implies $\hat{\delta}(q_0, w) = q_i$. By definition we know that the $Q_i \rightarrow \epsilon$ implies $q_i \in F$ so $\hat{\delta}(q_0, w) \in F$. \square

(Note: as many people have pointed out, the inductive hypotheses when considered together

$$(\hat{\delta}(q_i, a) = q_j) \Leftrightarrow (Q_i \xRightarrow{*} aQ_j)$$

seem to prove something trivially true since our construction was in terms of generic states — see our definition of P above. At the core this is true, but we are really proving a fundamental property of our construction that you'll see in other mathematical contexts:

We defined our translation from a DFA to a CFG based on the *single-step* functions δ and \Rightarrow respectively. Our inductive hypotheses prove that we can *lift* that translation to the *multi-step* case naturally, i.e., without further modification, via $\hat{\delta}$ and $\xRightarrow{*}$.)

Problem 4

Consider the grammar G over the alphabet $\{i, e\}$ with a single non-terminal S and productions given by:

$$S \rightarrow iS \mid iSeS \mid \epsilon$$

1. What is the language generated by G ?

Suppose for a word u , $d(u)$ is the difference between the number of i s in u and the number of e s in u . Then $L(G) = L = \{w \mid \text{for each prefix } u \text{ of } w, d(u) \geq 0\}$. That is, each e in u is matched by at least one i to the left of it.

More to the point, if we interpret i as *if* and e as *else*, the grammar describes the possible configurations of *if-else* statements in a typical programming language.

2. Show that G is ambiguous.

The following are two possible left-most derivations for $w = iie$.

$$\begin{aligned} S &\Rightarrow_{\text{lm}} iS \Rightarrow_{\text{lm}} iiSeS \Rightarrow_{\text{lm}} iieS \Rightarrow_{\text{lm}} iie \\ S &\Rightarrow_{\text{lm}} iSeS \Rightarrow_{\text{lm}} iieS \Rightarrow_{\text{lm}} iie \end{aligned}$$

3. Find a grammar G' such that $L(G) = L(G')$ and G' is unambiguous.

Define G' with variables S and A and productions:

$$\begin{aligned} S &\rightarrow \epsilon \mid iS \mid iAeS \\ A &\rightarrow \epsilon \mid iAeA \end{aligned}$$

Note that A is the same as an unambiguous grammar for balanced parentheses where i is a left bracket and e is a right bracket. Intuitively, G' tries to match e with the last unmatched i . If we interpret i as *if* and e as *else* then we can also view G' as associating *elses* only with the inner-most *if* that precedes it, a typical assumption that programming language parsers make.