CS 4510

Chomsky Hierarchy Part 1: Regular Grammars

Name: Abrahim Ladha

1.1 Properties

We said every regular language is context free. Any context free language can be decided by a pushdown automata. You can convert any finite automata to a pushdown automata by simply ignoring the stack. The context free languages are also characterized by context free grammars, so it follows that there should be a class of grammars which enumerate regular languages.

1.2 Regular Grammars

Definition 1.1 *A right-regular grammar is a tuple (V, Σ, P, S) such that:*

- *V is a set of nonterminals, sometimes called variables. These are always denoted by upper case letters*
- *Σ is the finite alphabet, denoted by lower case letters.*
- *P is a set of production rules, only of the form:*
  - $A \to aB$
  - $A \to a$
  - $A \to \epsilon$
- *$S \in V$ is the starting symbol.*

Theorem 1.2 *A language is regular if and only if there exists a regular grammar to generate it.*

First we show for every right-regular[1] grammar, there exists an equivalent NFA for the same language.

Given a regular grammar $G = (V, \Sigma, P, S)$, we construct an NFA $N = (\Sigma, Q, q_0, \delta, F)$

- Σ is the same
- Q is a set of states, one for each non-terminal, plus an additional state. $Q = V \cup \{A\}$
- Recall that for an NFA, $\delta(\cdot, \cdot)$ is a set. For each nonterminal B and character a:
  - For each nonterminal C, if $B \to aC$, then $C \in \delta(B, a)$
  - If $B \to a$, then $A \in \delta(B, a)$
  - For each $a \in \Sigma$, define $\delta(A, a) = \emptyset$

[1] left-regular grammars correspond to the reverse of the languages. It has rules of the form $A \to Aa$. This is not something we care about so much. The expressive power of regular languages, and their corresponding machines are invariant to this choice. We choose right-regular because of pedagogy.

  - If $S \to \epsilon$, $A \in \delta(S, \epsilon)$

- q_0, the start state is associated with the starting nonterminal, S.
- $F = \{A\}$.

We show $L(N) = L(G)$ by set containment

($L(G) \subseteq L(N)$) Let $x = a_1 a_2 ... a_n \in L(G)$. Then there exists a production of x of the form

$$S \implies a_1 A_1 \implies a_1 a_2 A_2 \implies ... \implies a_1 a_2 ... a_{n-1} A_{n-1} \implies x \tag{1.1}$$

As defined, it is clear $A_1 \in \delta(S, a_1)$, and $A_i \in \delta(A_{i-1}, a_i)$. It follows then that $A \in \delta(S, x)$, and $A \in F \implies x \in L(N)$.

($L(N) \subseteq L(G)$) Let $x \in L(N)$. If $x = \varepsilon$, then we must have that $A \in \delta(S, \varepsilon)$, but by construction this implies that $S \to \varepsilon$ is a rule of G. So suppose $x \neq \varepsilon$. Then there exists a sequence of states, $S, A_1, ..., A_{n-1}, A$ such that $A_1 \in \delta(S, a_1)$, and $A_i \in \delta(A_{i-1}, a_i)$. This implies that we must have production rules of the form

$$S \implies a_1 A_1 \implies a_1 a_2 A_2 \implies ... \implies a_1 a_2 ... a_{n-1} A_{n-1} \implies x \tag{1.2}$$

So G derives x and $x \in L(G)$.

Now we show for every NFA, there exists an equivalent right-regular grammar. Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA[2] We construct a grammar $G = (V, \Sigma, P, S)$

- Let V correspond to the states of Q
- Σ is the same
- For each rule of the form $\delta(B, a) = C$, construct a production rule of the form $B \to aC$. If $\delta(B, a) = C$, and $C \in F$, then $B \to a$. If $q_0 \in F$, then add the production rule $S \to \varepsilon$
- S corresponds to the start state, q_0.

The proof is left as exercise.

1.3 Problems

1. It is true, that a regular grammar will always generate a regular language, and for each regular language, there exists a regular grammar to produce it. But can a regular language be generated by a context free grammar which is not a regular grammar? If yes, give an example. If no, prove it.
2. We constructed a grammar G given an NFA M. Prove that $S \overset{*}{\implies} x \iff \delta(q_0, x) \in F$. Just follow the reverse of the proof we showed.
3. Consider the regular language $\{x \in \Sigma^* \mid x \text{ has an even number of 0s}\}$. Give the minimal DFA for this language, and construct the right-regular grammar for that DFA (following the construction in the proof).

[2] Not NFA, simply because the proof is easier. It doesn't matter, because you know DFAs and NFAs have the same expressive power.

4. For the language $\{0^i1^j \mid i, j \geq 0\}$ give the minimal[3] right-regular grammar for this language, then construct the DFA associated with that grammar.

5. Let A_1, A_2 be any non terminals and let $a_1, ..., a_n$ be any terminals. Prove that a production rule of the form $A_1 \to a_1...a_nA_2$ can be converted to a set of production rules of a regular grammar.

Further Reading

[1] Noam Chomsky and George A Miller. "Finite state languages". In: *Information and control* 1.2 (1958), pp. 91–112.

[2] John E Hopcroft and Jeffrey D Ullman. *Formal languages and their relation to automata.* Addison-Wesley Longman Publishing Co., Inc., 1969.

[3] We haven't talked about what a minimal regular grammar is, so just give the smallest (with respect to number of nonterminals) one you can come up with