4CS4-6: Theory of Computation (Decision and Decidable Properties of CFL)

Lecture 17: April, 01, 2019

Prof. K.R. Chowdhary *: Professor of CS*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

17.1 Decidable Properties of CFLs

There are some properties of CFLs which are decidable, i.e., there exists algorithms to answer yes/no about those properties, give a corresponding context-free grammar. Following are some properties, for which we describe algorithms.

Membership Problem For a CFG G, find out if $w \in L(G)$ is called membership problem. The membership algorithm can be specified as: Let given string is $|w| = n$. Convert the grammar G into CNF, and generate all the possible derivations of height (of parse-tree) $\log n$. Since, this is a finite number, it requires an exhaustive search to find out w. If we are successful to generate w, then $w \in L(G)$, else not.

Empty language problem The decision problem is, if $L(G) = \phi$? Hence, if CFL is empty, then it has no useful symbols, including, i.e., no variable appears in derivation of any terminal string. This can be checked using dependency graph discussed earlier in removal of useless productions and rules.

Infinite language problem For a context-free grammar G, find out if $L(G)$ is infinite. The algorithm is as follows:

1. Remove useless symbols and productions,
2. Remove and unit productions,
3. Create dependency graph for variables,
4. If there is a loop in the dependency graph, then the language is infinite.

Other approach can be as follows: Let G is CNF, and V is set of non-terminals. Without repeating any variable, longest word can be 2^{V-1}. Try the words of length between $2^{|V|}$ and $2^{|V|+1}$. If any of them is in $L(G)$, then the language $L(G)$ is infinite, else it is finite.

Example 17.1 *Show that the language corresponding to the grammar* $\{S \to AB, A \to aCb \mid a, C \to cBS, B \to bB \mid bb\}$ *is infinite.*

The graph shown in Fig. 17.1 indicates that there is an infinite loop in the dependency graph. Thus, the language corresponding to above grammar is infinite.

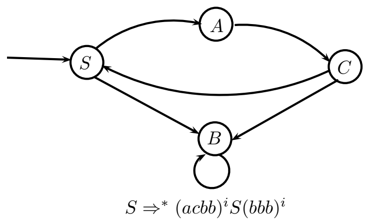

Figure 17.1: Infinite language dependency graph

17.2 Closure properties of of CFL

Theorem 17.2 *Prove that Intersection of CFLs and Regular Languages is CFLs.*

Proof: Let M_1 be a NPDA that accepts L_1 by final state, and let M_2 be the DFA that accepts the regular language L_2. Another NPDA M recognizing $L_1 \cap L_2$ can simulates M_1 and M_2 in parallel, and accepts, if both M_1 and M_2 accept the input.

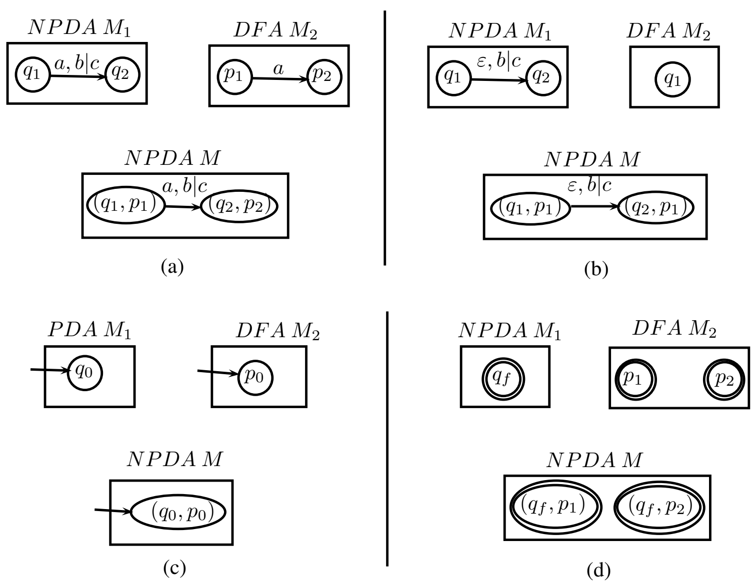

Figure 17.2: (a) Finite transitions in M_1, M_2, and M, (b) ε-transition in M_1, (c) Single-state M_1, M_2, M, (d) Final-state in M_1, M_2, M

The Fig. 17.2(a) shows the finite transitions of M_1, M_2, and M. Like in a "cross-product" construction of DFA, states of the new machine M has pairs of states, where one element of pair corresponds to the state of M_1 and other corresponds to the state of M_2.

The Fig. 17.2(b) shows the ε-transition. Whenever, an input symbol is read, both M_1 and M_2 are simulated; if M_1 needs to make a ε-move, the M_2 remains stationary, and M makes a ε move.

Fig. 17.2(c) shows start of state M_1, M_2, M and Fig. 17.2(d) shows the final states in three machines.

Since all the states in M, which is inter-section of an NPDA M_1 accepting on final state, and the DFA M_2, have been determined, this proves the theorem. ■