CSL253 - Theory of Computation

Tutorial 7

Team Members

1. Maharshi Soni - 12341350
2. Avinash Patel - 12340400
3. Ayush Khelwal - 12340430

Question 7

Let $INFINITE_{PDA} = \{\langle M \rangle \mid M \text{ is a PDA and } L(M) \text{ is an infinite language}\}$. Show that $INFINITE_{PDA}$ is decidable.

Solution

Overview of the Approach

The decision procedure involves two main steps:

1. **Conversion**: Convert the PDA M into an equivalent context-free grammar (CFG) G such that $L(G) = L(M)$.
2. **Infiniteness Test**: Decide if G generates an infinite language by checking for a recursive nonterminal using its dependency graph.

Detailed Algorithm

Step 1. PDA to CFG Conversion

Given a PDA

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F),$$

where:

- Q is the finite set of states.
- Σ is the input alphabet.

- Γ is the stack alphabet.
- δ is the transition function.
- $q_0 \in Q$ is the start state.
- $Z_0 \in \Gamma$ is the initial stack symbol.
- $F \subseteq Q$ is the set of accepting states.

We construct an equivalent CFG $G = (V, \Sigma, R, S)$ with:

Nonterminals: For every pair $p, q \in Q$, include a nonterminal A_{pq} which generates the strings that take the PDA from state p with an empty stack to state q with an empty stack.

Start Symbol: Choose

$$S = A_{q_0 f}, \quad \text{for some } f \in F,$$

which represents all strings that lead M from the start state to an accepting state.

Productions: Define rules in two parts:

a. *Epsilon Productions:* For every $p \in Q$, include

$$A_{pp} \to \epsilon.$$

b. *Transition Productions:* For each PDA transition

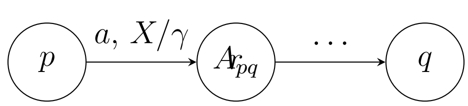
$$\delta(p, a, X) \ni (r, \gamma),$$

where $a \in \Sigma \cup \{\epsilon\}$, $X \in \Gamma$ is the popped symbol, and $\gamma = Y_1 Y_2 \cdots Y_k$ (with $k \geq 0$) is the string pushed, add productions that simulate intermediate moves:

$$A_{pq} \to a\, A_{r_1 r_2}\, A_{r_2 r_3} \cdots A_{r_k q},$$

choosing states $r_1, r_2, \ldots, r_k$ so that the effect of pushing γ is correctly simulated. When $\gamma = \epsilon$ (i.e., $k = 0$), the production captures a simple pop.

TikZ Diagram: The diagram below illustrates how a PDA transition is mapped into the CFG.



Correctness: Every derivation in G simulates a valid computation of M from q_0 to an accepting state f, ensuring

$$L(G) = L(M).$$

Step 2. Remove Useless Variables

Remove any nonterminals in G that do not derive any terminal string. This standard procedure simplifies the grammar before analyzing infiniteness.

Step 3. Decide Infiniteness of the CFG

A CFG G has an infinite language if and only if there exists a nonterminal A such that

$$A \Rightarrow^* uAv,$$

for some strings u, v. We:

a. Construct the dependency graph of G with nodes as nonterminals and an edge from A to B if a production $A \to \alpha B \beta$ exists.

b. Perform a graph search from the start symbol S. If a cycle is found and it is productive (i.e., can generate terminal strings), then G (and M) generates an infinite language.

Step 4. Construct the Decision Procedure

The overall decider D works as follows:

1. **Input:** A description $\langle M \rangle$ of a PDA.
2. **Convert:** Build an equivalent CFG G from M.
3. **Clean-up:** Remove useless productions from G.
4. **Analyze:** Check the dependency graph of G for a productive cycle.
5. **Output:** Accept if such a cycle exists (i.e., $L(M)$ is infinite); otherwise, reject.

Since all these steps are decidable, the language $INFINITE_{PDA}$ is decidable.

Example

Consider the PDA M recognizing the language

$$L = \{a^n b^n \mid n \geq 0\}.$$

A standard equivalent CFG is:

$$S \to aSb \mid \epsilon.$$

- **Observation:** The production $S \to aSb$ is recursive.
- **Cycle Detection:** The dependency graph contains a self-loop on S, showing that L is infinite.
- **Decision:** Hence, the decider accepts $\langle M \rangle$.

Conclusion

We have provided a decision procedure for $INFINITE_{PDA}$ by converting the PDA into an equivalent CFG, cleaning up the grammar, and checking its dependency graph for productive cycles. Since each step is decidable, so is $INFINITE_{PDA}$.