COS210 - Theoretical Computer Science
Pushdown Automata (Part 3)

Pushdown Automata and Context-Free Grammars

The descriptive power of a **non-deterministic pushdown automaton** (NPDA) is **equivalent to** that of a **context-free grammar**:

- If there exits an NPDA M with language $L = L(M)$, then there exists a context-free grammar G with language $L = L(G) = L(M)$
- If there exits a context-free grammar G with language $L = L(G)$, then there exits an NPDA M with language $L = L(M) = L(G)$

Pushdown Automata and Context-Free Grammars

Theorem

Let A be a language over an alphabet Σ. Then A is context-free

$$\Longleftrightarrow$$

there exists a non-deterministic pushdown automaton that accepts A.

- In this course we will only prove the " $\Longrightarrow$ " part of the theorem
- We will show how to convert an arbitrary context-free grammar to a non-deterministic pushdown automaton

Pushdown Automata and Context-Free Grammars

Proof:

- Let $G = (V, \Sigma, R, \$)$ be a context-free grammar such that $L(G) = A$
- We will construct an NPDA M that accepts $L(M) = L(G) = A$
- We know from Theorem 3.4.2 that every context-free grammar can be converted to Chomsky Normal Form (CNF)
- Hence, we can assume that G is in CNF and each substitution rule in R is of one of the following forms
  - $A \to BC$, where A, B, and C are variables, $B \neq \$$, and $C \neq \$$
  - $A \to a$, where A is a variable and a is a terminal
  - $\$ \to \epsilon$

Pushdown Automata and Context-Free Grammars

The pushdown automaton M must have the following property

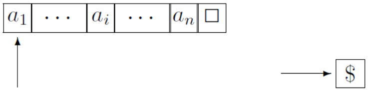
- For every string $w = a_1 a_2 \dots a_n$ over Σ

$$w \in L(G) \iff M \text{ accepts } w$$

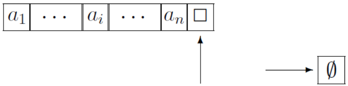
- Which means that

$$\$ \overset{*}{\Longrightarrow} a_1 a_2 \dots a_n$$

if and only if,
there exists a **run** of M that starts in the **initial configuration**



and ends in the **accepting configuration**



Pushdown Automata and Context-Free Grammars

- **Premise**: $\$ \stackrel{*}{\Longrightarrow} a_1 \dots a_n$ ($a_1 \dots a_n$ derivable from start variable \$)
- We can assume that in each step of this derivation, a **rule is applied to the leftmost variable** in the current string, e.g.

$$\$ \Longrightarrow AB \Longrightarrow aB \Longrightarrow ab$$

- Because the grammar is in CNF, at any time during the derivation the current string will have the form

$$a_1 a_2 \dots a_{i-1} A_k A_{k-1} \dots A_1$$

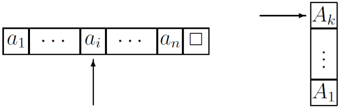
- At the **start** of the derivation, we have $i = 1$ and $k = 1$, and the current string is $A_k = \$$
- At the **end** of the derivation, we have $i = n + 1$ and $k = 0$, and the current string is $a_1 a_2 \dots a_n$

Pushdown Automata and Context-Free Grammars

- We will construct the pushdown automaton M such that the current string

$$a_1 a_2 \dots a_{i-1} A_k A_{k-1} \dots A_1$$

corresponds to the configuration



- i.e. the NPDA M has already read symbols $a_1 \dots, a_{i-1}$ in that order

Pushdown Automata and Context-Free Grammars

Given $G = (V, \Sigma, R, \$)$, we construct $M = (Q, \Sigma, \Gamma, \delta, q)$ as follows

- the **set of states** Q consists of the initial state q only
- the **tape alphabet** Σ is equal to the **set of terminals**
- the **stack alphabet** is the **set of variables**: $\Gamma = V$
- the **transition function** δ is obtained from the **rules** in R as follows
  - For each rule of the form $A \to BC$ we introduce an instruction
  $$qaA \to qNCB, \text{ for all } a \in \Sigma$$
  - For each rule of the form $A \to a$ we introduce an instruction
  $$qaA \to qR\epsilon$$
  - If there is the rule $\$ \to \epsilon$, then we introduce the instruction
  $$q\Box\$ \to qN\epsilon \quad \text{(accepting termination)}$$

Pushdown Automata and Context-Free Grammars

We still need to show that the language of M is the language of G, i.e.

$$\$ \stackrel{*}{\Longrightarrow} a_1 a_2 \ldots a_n \quad \Longleftrightarrow \quad M \text{ accepts } a_1 a_2 \ldots a_n$$

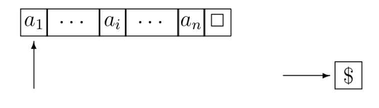
We first prove the following:

Claim

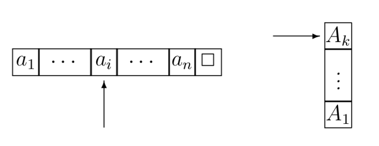
$$\$ \stackrel{*}{\Longrightarrow} a_1 a_2 \ldots a_{i-1} A_k A_{k-1} \ldots A_1$$

$$\Longleftrightarrow$$

there exists a run in M
from the initial configuration



to the configuration



Proof of Claim:

Induction over the structure of strings w

Base Case:

- The claim holds for the string $w = \$$ (we have $\$ \overset{*}{\Longrightarrow} \$$)

Inductive Step:

- Assume that the claim holds for $w = a_1 a_2 \dots a_{i-1} A_k A_{k-1} \dots A_1$
- Need to show that claim still holds after applying any rule in R to A_k
- Such a rule must be in one of the following forms:
  - $A_k \rightarrow BC$ or
  - $A_k \rightarrow a_i$
- In both cases the claim will still hold for the resulting string $\square$

Pushdown Automata and Context-Free Grammars

Now we can show

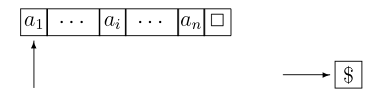
$$\$ \overset{*}{\Longrightarrow} a_1 a_2 \dots a_n \quad \iff \quad M \text{ accepts } a_1 a_2 \dots a_n$$

by applying the claim with $i = n + 1$ and $k = 0$

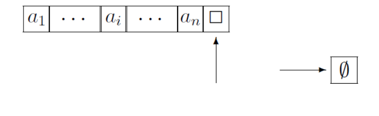
$$\$ \overset{*}{\Longrightarrow} a_1 a_2 \dots a_n$$

$$\iff$$

there exists a run in M from the initial configuration



to the configuration



We can conclude that $L(M) = L(G)$ □

Pushdown Automata and Context-Free Grammars

We can even strengthen the theorem as follows:

Theorem

Let A be a language over an alphabet Σ. Then A is context-free

$$\iff$$

there exists a non-deterministic pushdown automaton that accepts A **and the automaton has only one state.**

Proof:

- Since A is context-free, there exists a grammar G_0 with $L(G_0) = A$
- There exists a grammar G in Chomsky Normal Form with $L(G) = L(G_0)$
- The grammar G can be coverted to an NPDA M with $L(M) = L(G)$ that has only one state

Context-Free Grammar to Pushdown Automaton: Example

Assume the following grammar in Chomsky Normal Form
$G = (V, \Sigma, R, S = \$)$ where $\Sigma = \{0\}$, $V = \{S, A, B, A_1, A_2\}$ and R:

$$\begin{aligned}
S &\rightarrow BB|AB|BA|A_2A_2|BA_1|\epsilon \\
A &\rightarrow BB|AB|BA|A_2A_2|BA_1 \\
B &\rightarrow A_2A_2 \\
A_1 &\rightarrow AB \\
A_2 &\rightarrow 0
\end{aligned}$$

We construct an equivalent pushdown automaton $M = (Q, \Sigma, \Gamma, \delta, q)$ with

- $Q = \{q\}$
- $\Sigma = \{0\}$
- $\Gamma = \{S, A, B, A_1, A_2\}$
- $\delta \ldots$

Context-Free Grammar to Pushdown Automaton: Example

For each rule in R of the form $A \rightarrow BC$ we add the instructions

$$qaA \rightarrow qNCB, \text{ for all } a \in \Sigma$$

Rules:

$$S \rightarrow BB|AB|BA|A_2A_2|BA_1|\epsilon$$
$$A \rightarrow BB|AB|BA|A_2A_2|BA_1$$
$$B \rightarrow A_2A_2$$
$$A_1 \rightarrow AB$$
$$A_2 \rightarrow 0$$

Added instructions:

$$q0A \rightarrow qNBB$$
$$q0A \rightarrow qNAB$$
$$q0A \rightarrow qNBA$$
$$q0A \rightarrow qNA_2A_2$$
$$q0A \rightarrow qNA_1B$$

Context-Free Grammar to Pushdown Automaton: Example

For each rule in R of the form $A \rightarrow BC$ we add the instructions

$$qaA \rightarrow qNCB, \text{ for all } a \in \Sigma$$

Rules:

$$S \rightarrow BB|AB|BA|A_2A_2|BA_1|\epsilon$$
$$A \rightarrow BB|AB|BA|A_2A_2|BA_1$$
$$B \rightarrow A_2A_2$$
$$A_1 \rightarrow AB$$
$$A_2 \rightarrow 0$$

Added instructions:

$$q0S \rightarrow qNBB$$
$$q0S \rightarrow qNAB$$
$$q0S \rightarrow qNBA$$
$$q0S \rightarrow qNA_2A_2$$
$$q0S \rightarrow qNA_1B$$

Context-Free Grammar to Pushdown Automaton: Example

For each rule in R of the form $A \rightarrow BC$ we add the instructions

$$qaA \rightarrow qNCB, \text{ for all } a \in \Sigma$$

Rules:

$$S \rightarrow BB|AB|BA|A_2A_2|BA_1|\epsilon$$
$$A \rightarrow BB|AB|BA|A_2A_2|BA_1$$
$$B \rightarrow A_2A_2$$
$$A_1 \rightarrow AB$$
$$A_2 \rightarrow 0$$

Added instructions:

$$q0B \rightarrow qNA_2A_2$$
$$q0A_1 \rightarrow qNBA$$

Context-Free Grammar to Pushdown Automaton: Example

For each rule in R of the form $A \to a$ we add the instruction

$$qaA \to qR\epsilon$$

Rule:

$$A_2 \to 0$$

Added instruction:

$$q0A_2 \to qR\epsilon$$

If R contains the rule $\$ \to \epsilon$, then we add the instruction

$$q\square\$ \to qN\epsilon$$

Rule:

$$S \to \epsilon$$

Added instruction:

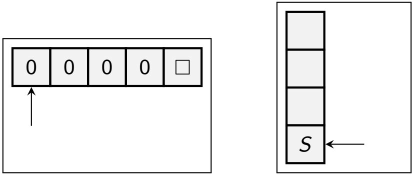
$$q\square S \to qN\epsilon$$

Exercise

Given the following derivation of the grammar G:

$$S \Rightarrow BB \Rightarrow A_2A_2B \Rightarrow 0A_2B \Rightarrow 00B \Rightarrow 00A_2A_2 \Rightarrow 000A_2 \Rightarrow 0000$$

and the initial configuration of the pushdown automaton M:



- Write down the sequence of instructions corresponding to the derivation above
- Update the configuration of M for each instruction step