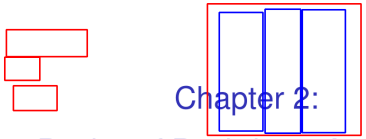
Script **generated by TTT**

Title: Petter: Compiler Construction (07.05.2020) - 11: Pushdown Automata

Date: Mon Apr 27 16:33:53 CEST 2020

Duration: 22:19 min

Pages: 7



Chapter 2:
Basics of Pushdown Automata

Example:

States: $0, 1, 2$
Start state: 0
Final states: $0, 2$

0	a	11
1	a	11
11	b	2
12	b	2

Conventions:

- We do not differentiate between pushdown symbols and states
- The rightmost / upper pushdown symbol represents the state
- Every transition consumes / modifies the upper part of the pushdown

Definition: Pushdown Automaton


Friedrich Bauer


Klaus Samelson

A pushdown automaton (PDA) is a tuple $M = (Q, T, \delta, q_0, F)$ with:

- Q a finite set of states;
- T an input alphabet;
- $q_0 \in Q$ the start state;
- $F \subseteq Q$ the set of final states and
- $\delta \subseteq Q^+ \times (T \cup \{\epsilon\}) \times Q^*$ a finite set of transitions

... for example:

States: $0, 1, 2$
Start state: 0
Final states: $0, 2$

0	a	11
1	a	11
11	b	2
12	b	2

Definition: Deterministic Pushdown Automaton

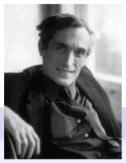
The pushdown automaton M is deterministic, if every configuration has maximally one successor configuration.

This is exactly the case if for distinct transitions (γ_1, x, γ_2) , $(\gamma_1', x', \gamma_2') \in \delta$ we can assume:
Is γ_1 a suffix of γ_1', then $x \neq x' \wedge x \neq \epsilon \neq x'$ is valid.

Pushdown Automata

Theorem:

For each context free grammar $G = (N, T, P, S)$
a pushdown automaton M with $\mathcal{L}(G) = \mathcal{L}(M)$ can be built.


M. Schützenberger


A. Öttinger

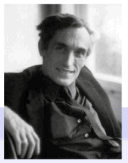
The theorem is so important for us, that we take a look at two constructions for automata, motivated by both of the special derivations:

- M_G^L to build Leftmost derivations
- M_G^R to build reverse Rightmost derivations

Pushdown Automata

Theorem:

For each context free grammar $G = (N, T, P, S)$
a pushdown automaton M with $\mathcal{L}(G) = \mathcal{L}(M)$ can be built.


M. Schützenberger


A. Öttinger

The theorem is so important for us, that we take a look at two constructions for automata, motivated by both of the special derivations:

- M_G^L to build Leftmost derivations
- M_G^R to build reverse Rightmost derivations