Pushdown automata

Arno Pauly

February 25, 2021

Typical examples of context-free non-regular languages

1. $\{a^n b^n \mid n \in \mathbb{N}\}$ is context-free, but not regular.
2. $\{w \mid w = w^{\mathrm{R}}\}$ is context-free, but not regular.

"Problem": Finite automata cannot remember anything except via states.

A stack

Reminder: A stack can be implemented by a list, where we can only

1. look at the first element of the list (including testing whether there is none),
2. remove the first element of the list,
3. and append a new element to the front of the list.

Pushdown automata - Informal

- Basic idea: A pushdown automaton is a finite automaton equipped with one single stack.
- When choosing an outgoing transition, we can take both the input symbol and the top of the stack into account.
- Upon completing a transition, we can either push a symbol onto the stack or pop it.

Formal definition

Definition

A pushdown automaton over an alphabet Σ is given by a tuple (Q, Γ, δ, F) where

1. Q is the set of states, there is a special start state $q_0 \in Q$, and $F \subseteq Q$ is the set of final states.
2. Γ is the stack alphabet, with a special symbol $\perp \in \Gamma$ indicating that the stack is empty.
3. $\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma$ is the transition relation.

Configurations

Definition

1. A configuration of a pushdown automaton is an element of $Q \times \Gamma^*$.
2. The initial configuration is $(q_0, \perp)$.
3. If the current configuration is $(q, \alpha_0\alpha)$, the current input symbol is a and $(q, a, \alpha_0, q', \beta) \in \delta$ for $\beta \neq \perp$, then a valid subsequent configuration is $(q', \beta\alpha_0\alpha)$.
4. If the current configuration is $(q, \alpha_0\alpha)$, the current input symbol is a and $(q, a, \alpha_0, q', \perp) \in \delta$, then a valid subsequent configuration is (q', α).

Configurations, continued

Definition

1. If the current configuration is $(q, \alpha_0\alpha)$ and $(q, \varepsilon, \alpha_0, q', \beta) \in \delta$ for $\beta \neq \perp$, then a valid subsequent configuration is $(q', \beta\alpha_0\alpha)$ which we can reach without progressing on the input.
2. If the current configuration is $(q, \alpha_0\alpha)$ and $(q, \varepsilon, \alpha_0, q', \perp) \in \delta$, then a valid subsequent configuration is (q', α) which we can reach without progressing on the input.
3. The accepting configurations are $(q, \perp)$ for $q \in F$.

Definition

A word is accepted by a pushdown automaton if we can reach an accepting configuration upon reading it.

Some facts on pushdown automata

- The requirement that the stack needs to be emptied is not necessary.
- We could push multiple symbols on the stack at the same time.
- If we would use a queue instead of a stack, we get something more powerful (witness $\{ww \mid w \in \Sigma^*\}$).
- If we had two stacks, we could recognize all computably enumerable languages.