Introduction to Computer Science

Géza Horváth

Eighth Lecture

Equivalence of acceptance by empty stack and acceptance by final states

The language class accepted by pushdown automata by final states and the language class accepted by pushdown automata by empty stack are the same. To prove this, we use two lemmas. First, we prove that for each PDA we can give PDA_e such that $L(PDA_e) = L(PDA)$, second we show the reverse case.

Equivalence of PDA_e and PDA

Lemma

For each $PDA = (Q, T, Z, q_0, z_0, \delta, F)$ *we can give* $PDA_e = (Q', T, Z, q_0, z_0, \delta')$ *such that* $L(PDA_e) = L(PDA)$.

Proof. We are going to define a pushdown automaton PDA_e, which works the same way as the pushdown automaton PDA does, but each time when the original automaton goes into a final state, the new automaton goes into the state q_f, as well. Then, PDA_e clears out the stack, when it is in the state q_f. Formally, let $Q' = Q \cup \{q_f\}$ where $\{q_f\} \cap Q = \emptyset$, and the transition function is the following:

1. Let $(q_2, r) \in \delta'(q_1, a, z)$ if $(q_2, r) \in \delta(q_1, a, z)$, for each $q_1, q_2 \in Q, a \in T \cup \{\lambda\}, z \in Z, r \in Z^*$,
2. let $(q_f, \lambda) \in \delta'(q_1, a, z)$ if $(q_2, r) \in \delta(q_1, a, z)$, for each $q_1 \in Q, q_2 \in F, a \in T \cup \{\lambda\}, z \in Z, r \in Z^*$, and
3. let $\delta'(q_f, \lambda, z) = \{(q_f, \lambda)\}$ for each $z \in Z$.

QED.

Equivalence of PDA_e and PDA

Lemma

For each $PDA_e = (Q, T, Z, q_0, z_0, \delta)$ *we can give* $PDA = (Q', T, Z', q_0', z_0', \delta', F)$ *such that* $L(PDA) = L(PDA_e)$.

Proof. Again, we have a constructive proof. The automaton PDA first puts the initial stack symbol of the automaton PDA$_e$ over the new initial stack symbol. Then it simulates the original PDA$_e$ automaton, but each time when the original automaton clears the stack completely, the new automaton goes into the new final state q_f. The automaton PDA defined below accepts the same language with final states which is accepted by the original automaton PDA$_e$ with empty stack. Let $Q' = Q \cup \{q_0', q_f\}$, where $\{q_0'\} \cap Q = \{q_f\} \cap Q = \emptyset$, let $Z' = Z \cup \{z_0'\}$, where $\{z_0'\} \cap Z = \emptyset$, and let $F = \{q_f\}$, so $\{q_f\}$ is the only final state, q_0' is the new initial state, and z_0' is the new initial stack symbol. The transition function is the following:

1. Let $\delta'(q_0', \lambda, z_0') = \{(q_0, z_0 z_0')\}$,
2. let $(q_2, r) \in \delta'(q_1, a, z)$ if $(q_2, r) \in \delta(q_1, a, z)$, for each $q_1, q_2 \in Q, a \in T \cup \{\lambda\}, z \in Z, r \in Z^*$, and
3. let $\delta'(q, \lambda, z_0') = \{(q_f, \lambda)\}$ for each $q \in Q$.

QED.

Equivalence of PDA_e and PDA

Theorem

The language class accepted by pushdown automata with final states is the same as the language class accepted by pushdown automata with empty stack.

Proof. This theorem is a direct consequence of the previous two lemmas.
QED.

Equivalence of context-free grammars and pushdown automata

Now we are going to prove that the languages accepted by pushdown automata are the context-free languages. Again, we are going to give constructive proofs. First, we demonstrate that for each pushdown automaton we can give a context-free grammar generating the same language as accepted by the PDA_e with empty stack, then we show how to construct a PDA_e which accepts the language generated by a context-free grammar.

Equivalence of context-free grammars and PDA

Lemma

For each $PDA_e = (Q, T, Z, q_0, z_0, \delta)$ *we can give a context-free grammar* $G = (N, T, S, P)$ *such that* $L(G) = L(PDA_e)$*.*

The proof is difficult.

Equivalence of context-free grammars and PDA

Lemma

For each context-free grammar $G = (N, T, S, P)$ we can give a pushdown automaton $PDA_e = (Q, T, Z, q_0, z_0, \delta)$ such that $L(PDA_e) = L(G)$.

Proof. The set of input letters of the PDA_e and the set of terminal symbols of grammar G are the same. Let $Q = \{q_0\}$, $Z = N \cup T$, and $z_0 = S$. The production rules are very simple.

1. Let $(q_0, r) \in \delta(q_0, \lambda, A)$, if $A \to r \in P$, and
2. let $(q_0, \lambda) \in \delta(q_0, a, a)$ for each $a \in T$.

During the computation of the PDA_e, we use λ-steps to simulate the work of grammar G. The current word is always in the stack memory. We can remove the letters one by one, reading them from the input and clearing them at the same time from the top of the stack. The process is finished, when each letter is read and the stack is empty.
QED.

Equivalence of context-free grammars and PDA

Theorem

A language is context-free, if and only if it is accepted by some pushdown automaton.

Proof. This theorem is a direct consequence of the previous two lemmas.
QED.

Finally, we have to note that for each context-free language we can give a pushdown automaton, which has only one state and accepts the context-free language by empty stack. This statement is a direct consequence of the proof of the latest lemma.

Deterministic pushdown automata – definition

Definition

The pushdown automaton $PDA_d = (Q, T, Z, q_0, z_0, \delta, F)$ *is deterministic, if*

1. $\delta(q, a, z)$ *has at most one element for each triple* $q \in Q$, $a \in T \cup \{\lambda\}$, *and* $z \in Z$, *and*
2. *if* $\delta(q, \lambda, z)$, $q \in Q$, $z \in Z$ *has an element, then* $\delta(q, a, z) = \emptyset$ *for each* $a \in T$.

The language class accepted by deterministic pushdown automata with final states is a proper subset of the language class accepted by pushdown automata.

Definition

The class of languages accepted by deterministic pushdown automata is called the class of deterministic context-free languages.

Deterministic pushdown automata – properties

We have already proven that the language class accepted by pushdown automata by final states and the language class accepted by pushdown automata by empty stack are the same. However, it is different for the deterministic case. The language class accepted by deterministic pushdown automata with empty stack is a proper subset of the language class accepted by deterministic pushdown automata with final states. Let us mark the deterministic pushdown automata accepting by empty stack with PDA_{de}. We can summarize these properties in the following expression:

$$L(PDA_{de}) \subset L(PDA_d) \subset L(PDA_e) = L(PDA).$$

References

This work was supported by the construction EFOP-3.4.3-16-2016-00021. The project was supported by the European Union, co-financed by the European Social Fund.