Lecture 25

Simulating NPDAs by CFGs

We have shown that every CFL is accepted by some NPDA. Now we show conversely that NPDAs accept only CFLs. Thus NPDAs and CFGs are equivalent in expressive power. We will do this in two steps by showing that

(i) every NPDA can be simulated by an NPDA with one state; and

(ii) every NPDA with one state has an equivalent CFG.

Actually, step (ii) is the easier of the two, since we have already done all the work in Lecture 24. In that lecture, given a grammar in which all productions were of the form

$$A \to cB_1B_2 \cdots B_k$$

for some $k \geq 0$ and $c \in \Sigma \cup \{\epsilon\}$, we constructed an equivalent NPDA with one state. That construction is invertible. Suppose we have an NPDA with one state

$$M = (\{q\},\ \Sigma,\ \Gamma,\ \delta,\ q,\ \bot,\ \varnothing)$$

that accepts by empty stack. Define the grammar

$$G = (\Gamma,\ \Sigma,\ P,\ \bot),$$

D. C. Kozen, *Automata and Computability*

where P contains a production

$$A \to cB_1B_2 \cdots B_k$$

for every transition

$$((q,c,A),\ (q,B_1B_2 \cdots B_k)) \in \delta,$$

where $c \in \Sigma \cup \{\epsilon\}$. Then Lemma 24.1 and Theorem 24.2 apply verbatim, thus $L(G) = L(M)$.

It remains to show how to simulate an arbitrary NPDA by an NPDA with one state. Essentially, we will maintain all state information on the stack. By the construction of Supplementary Lecture E, we can assume without loss of generality that M is of the form

$$M = (Q,\ \Sigma,\ \Gamma,\ \delta,\ s,\ \bot,\ \{t\});$$

that is, M has a single final state t, and M can empty its stack after it enters state t.

Let

$$\Gamma' \stackrel{\text{def}}{=} Q \times \Gamma \times Q.$$

Elements of Γ' are written $\langle p\ A\ q\rangle$, where $p,q \in Q$ and $A \in \Gamma$. We will construct a new NPDA

$$M' = (\{*\},\ \Sigma,\ \Gamma',\ \delta',\ *,\ \langle s \perp t\rangle,\ \varnothing)$$

with one state $*$ that accepts by empty stack. The new machine M' will be able to scan a string x starting with only $\langle p\ A\ q\rangle$ on its stack and end up with an empty stack iff M can scan x starting in state p with only A on its stack and end up in state q with an empty stack.

The transition relation δ' of M' is defined as follows: for each transition

$$((p,c,A),\ (q_0,B_1B_2 \cdots B_k)) \in \delta,$$

where $c \in \Sigma \cup \{\epsilon\}$, include in δ' the transitions

$$((*,c,\langle p\ A\ q_k\rangle),\ (*,\langle q_0\ B_1\ q_1\rangle\langle q_1\ B_2\ q_2\rangle \cdots \langle q_{k-1}\ B_k\ q_k\rangle))$$

for all possible choices of $q_1, q_2, \ldots, q_k$. For $k = 0$, this reduces to: if

$$((p,c,A),\ (q_0,\epsilon)) \in \delta,$$

include in δ' the transition

$$((*,c,\langle p\ A\ q_0\rangle),\ (*,\epsilon)).$$

Intuitively, M' simulates M, guessing nondeterministically what states M will be in at certain future points in the computation, saving those guesses on the stack, and then verifying later that those guesses were correct.

The following lemma formalizes the intuitive relationship between computations of M and M'.

Lemma 25.1 *Let M' be the NPDA constructed from M as above. Then*

$$(p, x, B_1 B_2 \cdots B_k) \xrightarrow[M]{n} (q, \epsilon, \epsilon)$$

if and only if there exist $q_0, q_1, \ldots, q_k$ such that $p = q_0$, $q = q_k$, and

$$(*, x, \langle q_0\, B_1\, q_1\rangle\langle q_1\, B_2\, q_2\rangle \cdots \langle q_{k-1}\, B_k\, q_k\rangle) \xrightarrow[M']{n} (*, \epsilon, \epsilon).$$

In particular,

$$(p, x, B) \xrightarrow[M]{n} (q, \epsilon, \epsilon) \iff (*, x, \langle p\, B\, q\rangle) \xrightarrow[M']{n} (*, \epsilon, \epsilon).$$

Proof. By induction on n (What else?). For $n = 0$, both sides are equivalent to the assertion that $p = q$, $x = \epsilon$, and $k = 0$.

Now suppose that $(p, x, B_1 B_2 \cdots B_k) \xrightarrow[M]{n+1} (q, \epsilon, \epsilon)$. Let

$$((p, c, B_1),\ (r, C_1 C_2 \cdots C_m))$$

be the first transition applied, where $c \in \Sigma \cup \{\epsilon\}$ and $m \geq 0$. Then $x = cy$ and

$$\begin{aligned}(p, x, B_1 B_2 \cdots B_k) &\xrightarrow[M]{1} (r, y, C_1 C_2 \cdots C_m B_2 \cdots B_k)\\ &\xrightarrow[M]{n} (q, \epsilon, \epsilon).\end{aligned}$$

By the induction hypothesis, there exist $r_0, r_1, \ldots, r_{m-1}, q_1, \ldots, q_{k-1}, q_k$ such that $r = r_0$, $q = q_k$, and

$$\begin{aligned}&(*, y, \langle r_0\, C_1\, r_1\rangle\langle r_1\, C_2\, r_2\rangle \cdots \langle r_{m-1}\, C_m\, q_1\rangle\langle q_1\, B_2\, q_2\rangle \cdots \langle q_{k-1}\, B_k\, q_k\rangle)\\ &\xrightarrow[M']{n} (*, \epsilon, \epsilon).\end{aligned}$$

Also, by construction of M',

$$((*, c, \langle p\, B_1\, q_1\rangle),\ (*, \langle r_0\, C_1\, r_1\rangle\langle r_1\, C_2\, r_2\rangle \cdots \langle r_{m-1}\, C_m\, q_1\rangle))$$

is a transition of M'. Combining these, we get

$$\begin{aligned}&(*, x, \langle p\, B_1\, q_1\rangle\langle q_1\, B_2\, q_2\rangle \cdots \langle q_{k-1}\, B_k\, q_k\rangle)\\ &\xrightarrow[M']{1} (*, y, \langle r_0\, C_1\, r_1\rangle\langle r_1\, C_2\, r_2\rangle \cdots \langle r_{m-1}\, C_m\, q_1\rangle\langle q_1\, B_2\, q_2\rangle \cdots \langle q_{k-1}\, B_k\, q_k\rangle)\\ &\xrightarrow[M']{n} (*, \epsilon, \epsilon).\end{aligned}$$

Conversely, suppose

$$(*, x, \langle q_0\, B_1\, q_1\rangle\langle q_1\, B_2\, q_2\rangle \cdots \langle q_{k-1}\, B_k\, q_k\rangle) \xrightarrow[M']{n} (*, \epsilon, \epsilon).$$

Let

$$((*, c, \langle q_0\, B_1\, q_1\rangle),\ (*, \langle r_0\, C_1\, r_1\rangle\langle r_1\, C_2\, r_2\rangle \cdots \langle r_{m-1}\, C_m\, q_1\rangle))$$

be the first transition applied, where $c \in \Sigma \cup \{\epsilon\}$ and $m \geq 0$. Then $x = cy$ and

$$\begin{aligned}&(*, x, \langle q_0\, B_1\, q_1\rangle\langle q_1\, B_2\, q_2\rangle \cdots \langle q_{k-1}\, B_k\, q_k\rangle)\\ &\xrightarrow[M']{1} (*, y, \langle r_0\, C_1\, r_1\rangle\langle r_1\, C_2\, r_2\rangle \cdots \langle r_{m-1}\, C_m\, q_1\rangle\langle q_1\, B_2\, q_2\rangle \cdots \langle q_{k-1}\, B_k\, q_k\rangle)\\ &\xrightarrow[M']{n} (*, \epsilon, \epsilon).\end{aligned}$$

By the induction hypothesis,

$$(r_0, y, C_1C_2 \cdots C_mB_2 \cdots B_k) \xrightarrow[M]{n} (q_k, \epsilon, \epsilon).$$

Also, by construction of M',

$$((q_0, c, B_1),\ (r_0, C_1C_2 \cdots C_m))$$

is a transition of M. Combining these, we get

$$\begin{aligned}(q_0, x, B_1B_2 \cdots B_k) &\xrightarrow[M]{1} (r_0, y, C_1C_2 \cdots C_mB_2 \cdots B_k)\\ &\xrightarrow[M]{n} (q_k, \epsilon, \epsilon).\end{aligned}$$

□

Theorem 25.2 $L(M') = L(M)$.

Proof. For all $x \in \Sigma^*$,

$$\begin{aligned}x \in L(M') &\iff (*, x, \langle s \perp t\rangle) \xrightarrow[M']{*} (*, \epsilon, \epsilon)\\ &\iff (s, x, \perp) \xrightarrow[M]{*} (t, \epsilon, \epsilon) \qquad \text{Lemma 25.1}\\ &\iff x \in L(M).\end{aligned}$$

□

Historical Notes

Pushdown automata were introduced by Oettinger [95]. The equivalence of PDAs and CFGs was established by Chomsky [19], Schützenberger [112], and Evey [36].