Pushdown Automata - III (part III)

Prof. Dan A. Simovici

UMB

Definition

A *deterministic pushdown automaton* (dpda) is a pda $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ that satisfies the following conditions:

1. For every $z \in Z$, $q \in Q$ and $a \in A \cup \{\lambda\}$ we have $|\delta(z, q, a)| \leq 1$.
2. If $\delta(z, q, \lambda) \neq \emptyset$, then $\delta(z, q, a) = \emptyset$ for every $a \in A$.

A language L is *deterministic context-free* (dcfl) if there is some dpda $\mathcal{M}$ such that $L = L(\mathcal{M})$.

The second condition means that if a null transition exists when the dpda $\mathcal{M}$ sees z at the top of the pushdown store and its internal state is q, then no symbol can be read from the input tape of the automaton.
Clearly, every deterministic context-free language is also a context-free language.

Example

$L = \{a^n b^n \mid n \in \mathbb{N}\}$ is a deterministic context-free language. Indeed, consider the dpda

$$\mathcal{M} = (\{a, b\}, \{z_0, a\}, \{q_0, q_1, q_2\}, \delta, q_0, z_0, \{q_2\}),$$

where δ is defined by the table shown next. For triples (z, q, a) that do not occur in that table we have $\delta(z, q, a) = \emptyset$.

Top	State	Input	$\delta(z, q, a)$
z_0	q_0	λ	(z_0, q_2)
z_0	q_2	a	$(z_0 a, q_1)$
a	q_1	a	(aa, q_1)
a	q_1	b	(λ, q_1)
z_0	q_1	λ	(z_0, q_2)

(Example cont'd)

For the word $aaabbb \in L$ we have the computation:

$$(z_0, q_0, aaabbb) \vdash (z_0, q_2, aaabbb) \vdash (z_0a, q_1, aabbb)$$
$$\vdash (z_0aa, q_1, abbb) \vdash (z_0aaa, q_1, bbb) \vdash (z_0aa, q_1, bb)$$
$$(z_0a, q_1, b) \vdash (z_0, q_1, \lambda) \vdash (z_0, q_2, \lambda),$$

which shows that $aaabbb \in L(\mathcal{M})$. Similarly, $\lambda \in L(\mathcal{M})$ because

$$(z_0, q_0, \lambda) \vdash (z_0, q_2, \lambda).$$

Note that $aab \notin L$. The computation that begins with the word bab on the input tape is

$$(z_0, q_0, aab) \vdash (z_0, q_2, aab) \vdash (z_0a, q_1, ab)$$
$$(z_0aa, q_1, b) \vdash (z_0a, q_1, \lambda),$$

and no further move is possible.

Therefore, the final state q_2 cannot be reached and we may conclude that $aab \notin L(\mathcal{M})$. For other input words, such as bab, the dpda is unable to read its input; indeed, in this case, we have a two-step computation

$$(z_0, q_0, bab) \vdash (z_0, q_2, bab),$$

and the impossibility of reading the input word implies $bab \notin L(\mathcal{M})$.
It is easy to prove that $L(\mathcal{M}) = \{a^n b^n \mid n \in \mathbb{N}\}$.

Example

The language $L = \{xcx^R \mid x \in \{a, b\}^*\}$ is deterministic. Consider the dpda $\mathcal{M} = (\{a, b, c\}, \{z_0, a, b\}, \{q_0, q_1, q_2\}, \delta, q_0, z_0, \{q_2\})$. Transition sets not mentioned in the table are empty.

Top	State	Input	$\delta(z, q, a)$
z_0	q_0	a	(z_0a, q_0)
z_0	q_0	b	(z_0b, q_0)
z_0	q_0	c	(z_0, q_1)
a	q_0	a	(aa, q_0)
a	q_0	b	(ab, q_0)
a	q_0	c	(a, q_1)
b	q_0	a	(ba, q_0)
b	q_0	b	(bb, q_0)
b	q_0	c	(b, q_1)
a	q_1	a	(λ, q_1)
b	q_1	b	(λ, q_1)
z_0	q_1	λ	(λ, q_2)

Definition

Let n be a positive natural number. The *parenthetic alphabet of order n* is the set

$$A_n = \{(_0, \ldots, (_{n-1},)_0, \ldots,)_{n-1}\}$$

The *parenthetic language* of order n is the context-free language PAR_n generated by the grammar $G_n = (\{S\}, A_n, S, P)$ whose set of productions is

$$P = \{S \to SS, S \to \lambda, S \to (_0 S)_0, \ldots, S \to (_{n-1} S)_{n-1}\}.$$

The next result lists some elementary properties of parenthetic languages.

Lemma

The following statements concerning the language PAR_n *hold:*

1. *if* $u, v \in \mathrm{PAR}_n$, *then* $uv \in \mathrm{PAR}_n$;
2. *if* $u \in \mathrm{PAR}_n$, *then* $({}_i u)_i \in \mathrm{PAR}_n$ *for* $0 \leqslant i \leqslant n-1$;
3. *if* $({}_i)_i u \in \mathrm{PAR}_n$ *for some* i, $0 \leqslant i \leqslant n-1$, *then* $u \in \mathrm{PAR}_n$;
4. *for every word* $w \in \mathrm{PAR}_n - \{\lambda\}$ *there are* $u, v \in \mathrm{PAR}_n$ *such that* $w = ({}_i u)_i v$ *for some* i, $0 \leqslant i \leqslant n-1$.

Example

Let $A_n = \{(_0, \ldots, (_{n-1},)_0, \ldots,)_{n-1}\}$. The parenthetic language PAR_n on the alphabet A_n is a deterministic context-free language.
The dpda

$$\mathcal{M} = (A_n, \{z_0\} \cup A_n, \{q_0, q_1\}, \delta, q_0, z_0, \{q_0\}),$$

whose transition function is given by the table contained in the next slide (where $0 \leq i, j \leq n-1$) accepts the language PAR_n.

Top	State	Input	Transition Function
z_0	q_0	$($	$(z_0($, $q_1)$
$($	q_1	$($	$(($, $q_1)$
$($	q_1	$)$	(λ, q_1)
z_0	q_1	λ	(z_0, q_0)

For a dpda, we may write $\delta(z, q, a) = (w, p)$ instead of $\delta(z, q, a) = \{(w, p)\}$ in order to simplify the notation.
A dpda does not always have a next move. This may happen because

- for some pair $(z, q) \in Z \times Q$, such that $\delta(z, q, a) = \emptyset$ for every $a \in A \cup \{\lambda\}$, or
- because the pushdown store is empty.

However, it is possible to ensure that for each dpda there is an equivalent dpda that always has a next move.
To do this:

- Create $\mathcal{M}'$ by adding a marker symbol z_0', at the bottom of the pushdown store and a "sink" state $\bar{q}$, one that can never be left.
- Any computation of $\mathcal{M}$ that would have emptied the stack results in $\mathcal{M}'$ having z_0' at the top of the stack. This forces $\mathcal{M}'$ into state $\bar{q}$.
- Also, any pair (z, q) that could block a computation is augmented to allow a transition to $\bar{q}$ for each input symbol such that $\delta(z, q, a) = \emptyset$.

Theorem

For every dpda $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ *there exists a dpda* $\mathcal{M}' = (A, Z', Q', \delta', q_0', z_0', F')$ *that is equivalent to* $\mathcal{M}$ *and satisfies the following conditions:*

1. *For every* $z' \in Z'$, $q' \in Q'$ *exactly one of the following cases may occur:*
   1. $|\delta'(z', q', a)| = 1$ *for every* $a \in A$ *and* $\delta'(z', q', \lambda) = \emptyset$, *or*
   2. $|\delta'(z', q', \lambda)| = 1$, *and* $\delta'(z', q', a) = \emptyset$ *for every* $a \in A$.
2. *For every* $q', q'' \in Q$, *if* $\delta'(z_0', q', a) = (w, q'')$ *for some* $a \in A \cup \{\lambda\}$, *then* $w = z_0' u$ *for some* $u \in Z^*$.

Proof

Let $Z' = Z \cup \{z_0'\}$, $Q' = Q \cup \{q_0', \bar{q}\}$, where $z_0' \notin Z$ and $q_0', \bar{q} \notin Q$, $F' = F$, and let δ' be defined as follows.

- $\delta'(z_0', q_0', \lambda) = (z_0' z_0, q_0)$;
- $\delta(z_0', q, a) = (z_0', \bar{q})$ for $q \in Q$ and $a \in A$;
- if $\delta(z, q, a) \neq \emptyset$, then $\delta'(z, q, a) = \delta(z, q, a)$ for $z \in Z$, $q \in Q$, and $a \in A \cup \{\lambda\}$;
- if $\delta(z, q, \lambda) = \emptyset$ and there is $a \in A$ such that $\delta(z, q, a) = \emptyset$, then $\delta'(z, q, a) = (z, \bar{q})$;
- $\delta'(z, \bar{q}, a) = (z, \bar{q})$ for every $z \in Z$ and $a \in A$.

Proof (cont'd)

It is easy to see, by inspecting the definition of δ', that the conditions of the theorem are satisfied. Furthermore, we have $(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, q, \lambda)$ if and only if

$$(z_0', q_0', x) \underset{\mathcal{M}'}{\vdash} (z_0' z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (z_0' w, q, \lambda),$$

which implies $L(\mathcal{M}) = L(\mathcal{M}')$.
Note that the dpda $\mathcal{M}'$ defined above never empties its stack (because its initial stack symbol z_0' always remains at the bottom of the stack); also, $\mathcal{M}'$ always has a next move, so we may assume that for every dcfl L there exists a dpda $\mathcal{M}$ such that $\mathcal{M}$ always has a next move and $L = L(\mathcal{M})$.

Example

The dpda $\mathcal{M}'$ obtained from the dpda introduced in Example on Slide 4 is

$$\mathcal{M}' = (\{a, b\}, \{z_0', z_0, a\}, \{q_0, q_1, q_2, q_0', \bar{q}\}, \delta', q_0', z_0', \{q_2\}),$$

where the transition function is defined by the table shown next. Triples of the form (z, q, i) such that $i \in A \cup \{\lambda\}$ and $\delta(z, q, i) = \emptyset$ are omitted from this table.

Top	State	Input i	$\delta(z, q, i)$
z_0'	q_0'	λ	$(z_0' z_0, q_0)$
z_0'	$q \in \{q_0, q_1, q_2\}$	$i \in \{a, b\}$	$(z_0', \bar{q})$
z_0	q_0	λ	(z_0, q_2)
z_0	q_2	a	$(z_0 a, q_1)$
a	q_1	a	(aa, q_1)
a	q_1	b	(λ, q_1)
z_0	q_1	λ	(z_0, q_2)
$z \in \{a, z_0'\}$	q_0	$i \in \{a, b\}$	$(z, \bar{q})$
z_0'	q_2	$i \in \{a, b\}$	$(z, \bar{q})$
z_0	q_2	b	$(z_0, \bar{q})$
z_0'	q_1	$i \in \{a, b\}$	$(z, \bar{q})$
$z \in \{z_0, a, z_0'\}$	$\bar{q}$	$i \in \{a, b\}$	$(z, \bar{q})$

The dpda $\mathcal{M}'$ can read any input word. For example, we have the computation:

$$(z_0', q_0', bab) \vdash (z_0' z_0, q_0, bab) \vdash (z_0' z_0, q_2, bab)$$
$$\vdash (z_0' z_0, \bar{q}, ab) \vdash (z_0' z_0, \bar{q}, b) \vdash (z_0' z_0, \bar{q}, \lambda).$$

Definition

Let $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ be a dpda. An instantaneous description (w, q, x) of $\mathcal{M}$ is *looping* if for every $n \in \mathbb{N}$, $n > 0$, there exists an instantaneous description (w_n, q_n, x) such that $|w_n| \geq |w|$ for $n > 0$, and

$$(w, q, x) \underset{\mathcal{M}}{\vdash} (w_1, q_1, x) \underset{\mathcal{M}}{\vdash} \cdots \underset{\mathcal{M}}{\vdash} (w_n, q_n, x) \underset{\mathcal{M}}{\vdash} \cdots$$

The set of looping instantaneous descriptions of the dpda $\mathcal{M}$ will be denoted by $\mathsf{LOOP}(\mathcal{M})$. □

- If $c = (w, q, x) \in \mathrm{LOOP}(\mathcal{M})$, then the dpda can make an arbitrarily large number of moves starting from c without ever decreasing the length of the content of the pushdown store and without reading any input symbol.
- If $\mathcal{M}$ enters a looping instantaneous description while processing an input word, then the symbols that remain to be read will never be processed.
- An instantaneous description (w, q, x) is looping if and only if (z, q, λ) is a looping instantaneous description, where $w = w'z$ for some $w' \in Z^*$.

Let $n(\mathcal{M})$ be the number defined by

$$n(\mathcal{M}) = \begin{cases} \frac{|Q|(|Z|^{|Q||Z|\ell(\mathcal{M})+1} - |Z|)}{|Z|-1} & \text{if } |Z| > 1 \\ |Q| & \text{if } |Z| = 1, \end{cases}$$

where

$$\ell(\mathcal{M}) = \max\{|v| \mid \delta(z, q, a) = (v, q') \text{ for some } z \in Z, q \in Q, a \in A \cup \{\lambda\}\}$$

is the length of the longest word written on the pushdown store in one step.

Theorem

Let $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ *be a dpda. We have* $(z, q, \lambda) \in \mathrm{LOOP}(\mathcal{M})$ *if and only if there is an instantaneous description* (w, p, λ) *such that* $(z, q, \lambda) \overset{n(\mathcal{M})}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$.

Proof

If $(z, q, \lambda) \in \text{LOOP}(\mathcal{M})$, then it is clear that $(z, q, \lambda) \overset{n(\mathcal{M})}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$.

Conversely, suppose that $(z, q, \lambda) \overset{n(\mathcal{M})}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$. We need to consider two cases:

- `Case I:` There is $u \in Z^*$ such that $|u| > |Q||Z|\ell(\mathcal{M})$ and

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (u, r, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$$

  for some $r \in Q$, and
- `Case II:` Suppose now that for every $u \in Z^*$ such that

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (u, r, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda) \tag{1}$$

  we have $|u| \leq |Q||Z|\ell(\mathcal{M})$.

Proof (cont'd)

`Case I:` If there is $u \in Z^*$ such that $|u| > |Q||Z|\ell(\mathcal{M})$ and

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (u, r, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$$

for some $r \in Q$, then, since there are be more than $|Q||Z|$ steps in the computation $(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (u, r, \lambda)$, there are two instantaneous descriptions that have the same state and the same symbol at the top of the pushdown store. In other words, we have

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (vz', q', \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (vtz', q', \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (u, r, \lambda),$$

for some $u, v \in Z^*$. Because $(vz', q', \lambda) \overset{n}{\underset{\mathcal{M}}{\vdash}} (vtz', q', \lambda)$ we can write

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (vz', q', \lambda) \overset{nj}{\underset{\mathcal{M}}{\vdash}} (vt^j z', q', \lambda)$$

for any $j \in \mathbb{N}$, so $(z, q, \lambda) \in \mathrm{LOOP}(\mathcal{M})$.

Proof (cont'd)

`Case II:`

Suppose now that for every $u \in Z^*$ such that

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (u, r, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda) \tag{2}$$

we have $|u| \leq |Q||Z|\ell(\mathcal{M})$. Note that there are $\frac{|Z|^{|Q||Z|\ell(\mathcal{M})+1} - |Z|}{|Z|-1}$ nonnull words of length less or equal to $|Q||Z|\ell(\mathcal{M})$ on the alphabet Z, and, therefore, there are no more than $n(\mathcal{M})$ distinct instantaneous descriptions (u, r, λ) of the dpda $\mathcal{M}$ that can be reached from (z, q, λ). Consequently, in the computation (2) there are some repeated instantaneous descriptions and this implies $(z, q, \lambda) \in \mathrm{LOOP}(\mathcal{M})$.

Theorem

Let $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ *be a dpda. Then,*

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda),$$

where p *is a final state if and only if there is a computation*

$$(z, q, \lambda) \overset{m}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda),$$

where $m \leqslant n(\mathcal{M})$.

Proof

Suppose that $(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$ is the shortest computation that leads to an instantaneous description (w, p, λ) where p is a final state. By the argument presented previously, it follows that for every instantaneous description (u, r, λ) such that

$$(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (u, r, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda),$$

we have $|u| \leq |Q||Z|\ell(\mathcal{M})$.

Proof (cont'd)

Since there are $n(\mathcal{M})$ distinct instantaneous descriptions of the form (u, r, λ) it follows that the length of the computation $(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$ may not exceed $n(\mathcal{M})$ to avoid the presence of repeating instantaneous descriptions (which would contradict the definition of the computation $(z, q, \lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$).

Thus, we obtain $(z, q, \lambda) \overset{m}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$, where $m \leqslant n(\mathcal{M})$. The converse implication is obvious.

Consider the sets

$$
\begin{aligned}
I(\mathcal{M}) &= \{(z,q) \in Z \times Q \mid (z,q,\lambda) \in \mathrm{LOOP}(\mathcal{M}) \\
&\qquad \text{and } (z,q,\lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w,p,\lambda) \text{ for some } p \in F \text{ and } w \in Z^*\} \\
I'(\mathcal{M}) &= \{(z,q) \in Z \times Q \mid (z,q,\lambda) \in \mathrm{LOOP}(\mathcal{M}) \text{ and there is} \\
&\qquad \text{no } p \in F \text{ such that } (z,q,\lambda) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w,p,\lambda)\} \text{ for any } w \in Z^*.
\end{aligned}
$$

The previous theorem suggests the following algorithms for computing the sets $I(\mathcal{M})$ and $I(\mathcal{M}')$.

Computation of $I(\mathcal{M})$ and $I'(\mathcal{M})$:
Input Data: A dpda $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$;
Output Data: The sets $I(\mathcal{M})$ and $I'(\mathcal{M})$;
begin
if (For an instantaneous description (z, q, λ) determine there exists an instantaneous description (w, p, λ) such that $(z, q, \lambda) \overset{n(\mathcal{M})}{\underset{\mathcal{M}}{\vdash}} (w, p, \lambda)$)
then
begin (z, q, λ) is a looping instantaneous description, so $(z, q) \in I(\mathcal{M}) \cup I'(\mathcal{M})$
end
if (for (z, q, λ) determine whether it is possible to reach an instantaneous description (w, p, λ), where p is a final state in no more than $n(\mathcal{M})$ steps)
then (place (z, q) in $I(\mathcal{M})$)
else (place (z, q) in $I(\mathcal{M}')$)
return $I(\mathcal{M})$ and $I'(\mathcal{M})$;

A major result that we prove in this section is that the class of deterministic context-free languages is closed with respect to the complement operation.
Note that:

- A dpda may not read its input tape in its entirety if it enters a looping instantaneous description. Thus, it could happen that there is an input word x such that x is neither in $L(\mathcal{M})$ nor in $L(\mathcal{M}')$.
- Even if no looping instantaneous descriptions exist, after reading its input and entering a final instantaneous descriptions a dpda may continue its computation using null transitions and go through some states that are final and through some states that are non-final; correspondingly, $\mathcal{M}'$ will reach non-final and final states. This implies the existence of input words that belong to both $L(\mathcal{M})$ and $L(\mathcal{M}')$.

To overcome these difficulties, we need the following result.

Theorem

Let $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ *be a dpda. There is a dpda* $\mathcal{M}_1$ *such that* $L(\mathcal{M}) = L(\mathcal{M}_1)$*, and for every* $x \in A^*$ *there is* $q \in Q$ *such that*

$$(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, q, \lambda),$$

for some $w \in Z^*$*.*

Proof

Starting from the dpda $\mathcal{M}$ define the dpda

$$\mathcal{M}_1 = (A, Z, Q \cup \{p, r\}, \delta_1, q_0, z_0, F \cup \{p\}),$$

where p, r are two new states. The transition function δ_1 is defined as follows.

- for $z \in Z$, $q \in Q$ and $a \in A$, $\delta_1(z, q, a) = \delta(z, q, a)$;
- if $(z, q\lambda)$ is not a looping instantaneous description in $\mathcal{M}$, that is, $(z, q) \notin I(\mathcal{M}) \cup I(\mathcal{M}')$, define $\delta_1(z, q, \lambda) = \delta(z, q, \lambda)$;
- if $(z, q) \in I(\mathcal{M})$, then $\delta_1(z, q, \lambda) = (z, p)$;
- if $(z, q) \in I'(\mathcal{M})$, then $\delta_1(z, q, \lambda) = (z, r)$;
- $\delta_1(z, p, a) = (z, r)$ and $\delta_1(z, r, a) = (z, r)$ for every $a \in A$ and $z \in Z$.

Proof (cont'd)

The dpda $\mathcal{M}_1$ always reads its input in its entirety. Indeed, $\mathcal{M}_1$ simulates the activity of $\mathcal{M}$. If no looping instantaneous description is encountered in the computation of $\mathcal{M}$, then $\mathcal{M}$ will read its entire input, and $\mathcal{M}_1$ will do the same. Otherwise, when $\mathcal{M}$ enters a looping instantaneous description, then $\mathcal{M}_1$ enters either the state p or the state r and the entire input word is read.
The definition of $\mathcal{M}_1$ implies that $L(\mathcal{M}) \subseteq L(\mathcal{M}_1)$ since every accepting computation in $\mathcal{M}$ can be simulated in $\mathcal{M}_1$.

Proof (cont'd)

Suppose now that $x \in L(\mathcal{M}_1)$. There exists a computation in $\mathcal{M}_1$ that ends in a state in F or in the state p.
In the first case it is clear that $x \in L(\mathcal{M})$; in the second, $\mathcal{M}$ reaches a looping instantaneous description (tz, q, y), that is, $(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (tz, q, y)$ and $(z, q) \in I(\mathcal{M})$. The definition of $I(\mathcal{M})$ implies that there exists a final state q' such that $(tz, q, y) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, q', y)$, so $x \in L(\mathcal{M})$.

Theorem

Let A be an alphabet and let $L \subseteq A^*$ be a dcfl. Then, the language $\bar{L} = A^* - L$ is a dcfl.

Proof

Since L is a dcfl there exists a dpda $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ such that $L = L(\mathcal{M})$. By Theorem 11 we can assume that $\mathcal{M}$ reads every input word completely. Define the pushdown automaton $\mathcal{M}' = (A, Z, Q', \delta', q', z_0, F')$, where $Q' = Q \times \{0, 1, 2\}$. The transition function δ' is defined by:

Proof (cont'd)

- For $z \in Z$, $q \in Q$, and $a \in A$, if $\delta(z, q, a) = (w, p)$, then

$$\delta'(z, (q, 0), a) = \delta'(z, (q, 2), a) = \begin{cases} (w, (p, 0)) & \text{if } p \in F \\ (w, (p, 1)) & \text{if } p \notin F. \end{cases}$$

  and $\delta'(z, (q, 1), \lambda) = (z, (q, 2))$.
- For $z \in Z$ and $q \in Q$, if $\delta(z, q, \lambda) = (w, p)$, then $\delta'(z, (q, 0), \lambda) = (w, (p, 0))$, and

$$\delta'(z, (q, 1), \lambda) = \begin{cases} (w, (p, 0)) & \text{if } p \in F \\ (w, (p, 1)) & \text{if } p \notin F. \end{cases}$$

The initial state q_0' is

$$q_0' = \begin{cases} (q_0, 0) & \text{if } q_0 \in F \\ (q_0, 1) & \text{if } q_0 \notin F, \end{cases}$$

and the set of final states is $F' = \{(q, 2) \mid q \in Q\}$. Note that once $\mathcal{M}'$ enters a final state $(q, 2)$, then no null transition is possible. Also, $\mathcal{M}'$ can reach a final state $(q, 2)$ only by using a null transition from a state $(q, 1)$.

The role of the second component of a state of $\mathcal{M}'$ is to record whether $\mathcal{M}$ has entered a final state after its last non-null transition.

Suppose that $x \in L(\mathcal{M})$, that is, $(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, q, \lambda)$, where $q \in F$. If (q_0, i) is the initial state of $\mathcal{M}'$, the previous computation yields

$$(z_0, (q_0, i), x) \overset{*}{\underset{\mathcal{M}'}{\vdash}} (w, (q, 0), \lambda),$$

and $\mathcal{M}'$ cannot reach a final state. Thus, $x \notin L(\mathcal{M}')$. This shows that $L(\mathcal{M}') \subseteq A^* - L(\mathcal{M})$.

Proof (cont'd)

Conversely, suppose that $x \in A^* - L(\mathcal{M})$. Since $\mathcal{M}$ reads its entire input, we have $(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (wz, q, \lambda)$, where $q \notin F$ and q is the state entered by $\mathcal{M}$ after reading the last symbol of x. This implies the existence of the computation $(z_0, (q_0, i), x) \overset{*}{\underset{\mathcal{M}'}{\vdash}} (wz, (q, 1), \lambda)$. The definition of δ' implies $\delta'((q, 1), \lambda) = (z, (q, 2))$, so

$$(z_0, (q_0, i), x) \overset{*}{\underset{\mathcal{M}'}{\vdash}} (wz, (q, 1), \lambda) \overset{*}{\underset{\mathcal{M}'}{\vdash}} (wz, (q, 2), \lambda),$$

which shows that $x \in L(\mathcal{M}')$. Thus, we conclude that $L(\mathcal{M}') = A^* - L(\mathcal{M})$, so $A^* - L(\mathcal{M})$ is indeed a dcfl.

Example

The details of the construction presented in the proof of the previous theorem can be followed in this example.
Consider the dpda $\mathcal{M}'$ that accepts the language $\{a^n b^n \mid n \in \mathbb{N}\}$ and is capable of reading any input word $x \in \{a, b\}^*$. The dpda

$$\mathcal{M}'' = (\{a, b\}, \{z_0', z_0, a\}, Q' \times \{0, 1, 2\}\}, \delta'', (q_0', 1), z_0', Q' \times \{2\}),$$

constructed according to the algorithm accepts the language $\{a, b\}^* - \{a^n b^n \mid n \in \mathbb{N}\}$.
Here $Q' = \{q_0, q_1, q_2, q_0', \bar{q}\}$. Since q_0 is not a final state of $\mathcal{M}'$, the initial state of $\mathcal{M}''$ is $(q_0', 1)$. The word bab is accepted by $\mathcal{M}''$ using the following computation:

Example (cont'd)

Transition in $\mathcal{M}'$	Instantaneous description of $\mathcal{M}''$ and transition in $\mathcal{M}''$
	$(z_0', (q_0', 1), bab)$ (initial i.d.)
$\delta'(z_0', q_0', \lambda) = (z_0' z_0, q_0)$	$\delta''(z_0', (q_0', 1), \lambda) = (z_0' z_0, (q_0, 1))$
	$(z_0' z_0, (q_0, 1), bab)$
$\delta'(z_0, q_0, \lambda) = (z_0, q_2)$	$\delta''(z_0, (q_0, 1), \lambda) = (z_0, (q_2, 0))$ (since $q_2 \in F$)
	$(z_0' z_0, (q_2, 0), bab)$
$\delta'(z_0, q_2, b) = (z_0, \bar{q})$	$\delta''(z_0, (q_2, 0), b) = (z_0, (\bar{q}, 1))$ (since $\bar{q} \notin F$)
	$(z_0' z_0, (\bar{q}, 1), ab)$
$\delta'(z_0, \bar{q}, a) = (z_0, \bar{q})$	$\delta''(z_0, (\bar{q}, 1), \lambda) = (z_0, (\bar{q}, 2))$ $\delta''(z_0, (\bar{q}, 2), a) = (z_0, (\bar{q}, 1))$ (since $\bar{q} \notin F$)
	$(z_0' z_0, (\bar{q}, 2), ab)$ $(z_0' z_0, (\bar{q}, 1), b)$
$\delta'(z_0, \bar{q}, b) = (z_0, \bar{q})$	$\delta''(z_0, (\bar{q}, 1), \lambda) = (z_0, (\bar{q}, 2))$ $\delta''(z_0, (\bar{q}, 2), b) = (z_0, (\bar{q}, 1))$ (since $\bar{q} \notin F$)
	$(z_0' z_0, (\bar{q}, 2), b)$ $(z_0' z_0, (\bar{q}, 1), \lambda)$
	$\delta''(z_0, (\bar{q}, 1), \lambda) = (z_0, (\bar{q}, 2))$
	$(z_0' z_0, (\bar{q}, 2), \lambda)$

Corollary

For every dcfl L there exists a dpda that accepts L and has no null transitions from its final states.

Example

We give an example of a context-free language that is not deterministic.

$$L = \{a^n b^m c^p \mid m, n, p \in \mathbb{N} \text{ and } n = m \text{ or } m = p\}$$

is a context-free language that is inherently ambiguous. If L were a deterministic context-free language, the language $\{a, b, c\}^* - L$ would be a deterministic context-free language, and therefore, the language

$$\begin{aligned} K &= (\{a, b, c\}^* - L) \cap \{a\}^*\{b\}^*\{c\}^* \\ &= \{a^n b^m c^p \mid n, m, p \in \mathbb{N}, n \neq m \text{ and } m \neq p\} \end{aligned}$$

would be a context-free language.

Example (cont'd)

However, we prove that K is not context-free. Indeed, suppose that K were a context-free language generated by a context-free grammar G. Let n_G be the number that corresponds to G by Ogden's Lemma. Then, we have $w = a^{n_G+n_G!}b^{n_G}c^{n_G+n_G!} \in K$. Suppose that the set of marked positions are the n_G positions in w occupied by the bs. Then, we can write $w = xyzut$ such that at least one of y or u contains a marked position, yzu no more than n_G marked positions, and $xy^nzu^nt \in L(G)$ for all $n \in \mathbb{N}$. In other words, at least one of y or u must contain a b.

Observe that neither y nor u may contain two distinct symbols. For example, if y were to contain both as and bs then, by pumping, b could precede as and this would conflict with the definition of K. Let $|y| = k$ and $|u| = \ell$. We need to consider the following cases:

1. $y \in \{b\}^*$ and $u \in \{b\}^*$;
2. $y \in \{a\}^*$ and $u \in \{b\}^*$;
3. $y \in \{b\}^*$ and $u \in \{c\}^*$.

In the first case, note that $k + \ell \leqslant n_G$ and that there are $n_G - (k + \ell)$ symbols b that do not occur in y or u. The effect of the pumping of y and u is to increase the number of bs to $n_G - (k + \ell) + (k + \ell)i = n_G + (k + \ell)(i - 1)$. Note that there is an $i \in \mathbb{N}$ such that

$$n_G + n_G! = n_G + (k + \ell)(i - 1)$$

because $k + \ell \leqslant n_G$. By pumping y and u a number of $i = n_G!/(k + \ell) + 1$ times we obtain a word that has the same number of as and bs and cs; this contradicts the definition of K.

Example (cont'd)

In the second case, $\ell \leqslant n_G$. The effect of the pumping is to increase the number of as and bs while the number of cs remains constant. The word $w_i = xy^i zu^i t$ will contain $ki + n_G + n_G! - k$ as, and $\ell i + n_G - \ell = n_G + \ell(i-1)$ bs. By choosing $i = n_G!/\ell + 1$, the number of bs equals the number of cs, which contradicts the definition of K. The third case can be treated in a similar manner.

To prove other closure properties of deterministic context-free languages we introduce a normal form for dpdas.

Lemma

For every dpda $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ *there exists an equivalent dpda* $\mathcal{M}_1 = (A, Z, Q_1, \delta_1, q_0, z_0, F)$ *such that if* $\delta_1(z, q_1, a) = \{(w, p_1)\}$ *for some* $q_1, p_1 \in Q_1$, *then* $|w| \leqslant 2$.

Proof

For each transition $\delta(z, q, a) = \{(u, p)\}$ such that $u = z_0 \cdots z_{k-1}$ and $k > 2$ we introduce $k - 2$ new states $r_0, \ldots, r_{k-3}$ and define

$$\begin{aligned}
\delta_1(z, q, a) &= \{(z_0 z_1, r_0)\} \\
\delta_1(z_1, r_0, \lambda) &= \{(z_1 z_2, r_1)\} \\
&\vdots \\
\delta_1(z_{k-3}, r_{k-4}, \lambda) &= \{(z_{k-3} z_{k-2}, r_{k-3})\} \\
\delta_1(z_{k-2}, r_{k-3}, \lambda) &= \{(z_{k-2} z_{k-1}, p)\}.
\end{aligned}$$

For every other transitions $\delta_1(z, q, a) = \delta(z, q, a)$.

Proof (cont'd)

If

$$(uz, q, ax) \underset{\mathcal{M}}{\vdash} (uz_0 \cdots z_{k-1}, p, x),$$

then we have the computation

$$\begin{aligned}
(uz, q, ax) &\underset{\mathcal{M}_1}{\vdash} (uz_0 z_1, r_0, x) \\
&\underset{\mathcal{M}_1}{\vdash} (uz_0 z_1 z_1, r_1, x) \\
&\vdots \\
&\underset{\mathcal{M}_1}{\vdash} (uz_0 z_1 z_1 \cdots z_{k-3}, r_{k-4}, x) \\
&\underset{\mathcal{M}_1}{\vdash} (uz_0 z_1 z_1 \cdots z_{k-3} z_{k-2}, r_{k-3}, x) \\
&\underset{\mathcal{M}_1}{\vdash} (uz_0 z_1 z_1 \cdots z_{k-3} z_{k-2} z_{k-1}, p, x).
\end{aligned}$$

Thus, the dpda are easily seen to be equivalent and $\mathcal{M}_1$ is a dpda that satisfies the condition of the lemma.

Theorem

(Normal Form Theorem for Pushdown Automata) *For every dpda* $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ *there exists an equivalent dpda* $\mathcal{M}' = (A, Z, Q', \delta, q_0', z_0', F')$ *such that if* $\delta'(z, q, a) = \{(w, p)\}$, *then*

$$w \in \{\lambda, z\} \cup \{zz' \mid z' \in Z\}.$$

Proof

We can assume without loss of generality that if $\delta(z, q, a) = (w, p)$, then $|w| \leqslant 2$. Define $Q' = Z \times Q$, $q'_0 = (z_0, q_0)$, $Z' = Z \cup \{z'_0\}$, and $F' = Z' \times F$, where z'_0 is a new symbol, $z_0 \notin Z$. The transition function δ' is given by:

1. if $\delta(z, q, a) = (\lambda, p)$, then $\delta'(z', (z, q), a) = (\lambda, (z', p))$;
2. if $\delta(z, q, a) = (y, p)$, then $\delta'(z', (z, q), a) = (z', (y, p))$;
3. if $\delta(z, q, a) = (yy_1, p)$, then $\delta'(z', (z, q), a) = (z'y, (y_1, p))$,

for every $z, z' \in Z$, $q \in Q$ and $a \in A \cup \{\lambda\}$.
In addition, $\delta'(z'_0, (z_0, q_0), \lambda) = (z'_0 z_0, (z_0, q_0))$.

The actions of $\mathcal{M}$ and $\mathcal{M}'$ are described succinctly in the following table:

Action of $\mathcal{M}$	Action of $\mathcal{M}'$
$\mathcal{M}$ pops the stack.	$\mathcal{M}'$ pops the stack and stores the popped symbol in its state.
$\mathcal{M}$ alters the top of the stack to y.	$\mathcal{M}'$ leaves the top of the stack intact and stores y in its state.
$\mathcal{M}$ pushes yy_1 on the stack.	$\mathcal{M}'$ stores y_1 in its state and pushes y on the stack.

We claim that $(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (z_{i_0} \cdots z_{i_{m-1}}, q, \lambda)$ if and only if

$$(z_0', (z_0, q_0), x) \overset{*}{\underset{\mathcal{M}'}{\vdash}} (z_0' z_{i_0} \cdots z_{i_{m-2}}, (z_{i_{m-1}}, q), \lambda).$$

Suppose that

$$(z_0, q_0, x) \overset{n}{\underset{\mathcal{M}}{\vdash}} (z_{i_0} \cdots z_{i_{m-1}}, q, \lambda) \tag{3}$$

We prove by induction on n that

$$(z_0', (z_0, q_0), x) \overset{*}{\underset{\mathcal{M}'}{\vdash}} (z_0' z_{i_0} \cdots z_{i_{m-2}}, (z_{i_{m-1}}, q), \lambda).$$

In the base case, $n = 0$, we have $m = 1$ and $z_{i_0} = z_0$ and the implication follows immediately. Suppose that the statement holds for computations of length less than n and that we have the computation (3). We need to consider three cases, depending on the action of $\mathcal{M}$ during the last step of the computation.

`Case I:` If $\mathcal{M}$ popped the stack, then the computation (3) has the form:

$$\begin{aligned}(z_0, q_0, x) &\overset{n-1}{\underset{\mathcal{M}}{\vdash}} (z_{i_0} \cdots z_{i_{m-1}} z, q', a)\\ &\underset{\mathcal{M}}{\vdash} (z_{i_0} \cdots z_{i_{m-1}}, q, \lambda),\end{aligned}$$

where $\delta(z, q', a) = (\lambda, q)$, $x = x'a$, and $a \in A \cup \{\lambda\}$. Consequently, $\delta'(z_1, (z, q'), a) = (\lambda, (z_1, q))$ for every $z_1 \in Z$. By the inductive hypothesis, we have the computation:

$$\begin{aligned}(z_0', (z_0, q_0), x) &\overset{*}{\underset{\mathcal{M}'}{\vdash}} (z_0' z_{i_0} \cdots z_{i_{m-1}}, (z, q'), a)\\ &\underset{\mathcal{M}'}{\vdash} (z_0' z_{i_0} \cdots z_{i_{m-2}}, (z_{i_{m-1}}, q), \lambda),\end{aligned}$$

which completes the proof of the implication.

`Case II:` If $\mathcal{M}$ alters only the top of the stack by changing it to y, the computation (3) can be written as

$$\begin{aligned}(z_0, q_0, x) &\overset{n-1}{\underset{\mathcal{M}}{\vdash}} (z_{i_0} \cdots z_{i_{m-1}} z, q', a) \\ &\underset{\mathcal{M}}{\vdash} (z_{i_0} \cdots z_{i_{m-1}} y, q, \lambda)\end{aligned}$$

and we have $\delta(z, q', a) = (y, q)$. Thus, $\delta'(z_1, (z, q'), a) = (z_1, (y, q))$ for every $z_1 \in Z$. By the inductive hypothesis, we have the computation:

$$\begin{aligned}(z_0', (z_0, q_0), x) &\overset{*}{\underset{\mathcal{M}'}{\vdash}} (z_0' z_{i_0} \cdots z_{i_{m-1}}, (z, q'), a) \\ &\underset{\mathcal{M}'}{\vdash} (z_0' z_{i_0} \cdots z_{i_{m-1}}, (y, q), \lambda).\end{aligned}$$

`Case III:` Finally, suppose that $\mathcal{M}$ pushes yy_1 on the stack using the transition $\delta(z, q', a) = (yy_1, p)$. In this case, the computation (3) can be written as:

$$\begin{aligned}(z_0, q_0, x) &\overset{n-1}{\underset{\mathcal{M}}{\vdash}} (z_{i_0} \cdots z_{i_{m-1}} z, q', a) \\ &\underset{\mathcal{M}}{\vdash} (z_{i_0} \cdots z_{i_{m-1}} y y_1, p, \lambda),\end{aligned}$$

where $x = x'a$. We have $\delta'(z_1, (z, q'), a) = (z_1 y, (y_1, p))$ for every $z_1 \in Z$. Combining this fact with the inductive hypothesis yields the computation:

$$\begin{aligned}(z_0', (z_0, q_0), x) &\overset{*}{\underset{\mathcal{M}'}{\vdash}} (z_0' z_{i_0} \cdots z_{i_{m-1}}, (z, q'), a) \\ &\underset{\mathcal{M}'}{\vdash} (z_0' z_{i_0} \cdots z_{i_{m-1}} y, (y_1, p), \lambda).\end{aligned}$$

Conversely, we can prove by induction of n that

$$(z_0', (z_0, q_0), x) \overset{n}{\underset{\mathcal{M}'}{\vdash}} (z_0' z_{i_0} \cdots z_{i_{m-2}}, (z_{i_{m-1}}, q), \lambda).$$

implies

$$(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (z_{i_0} \cdots z_{i_{m-1}}, q, \lambda).$$

The argument is similar to the one presented above and is omitted.

Example

Let $Z' = \{z_0, a, b, z_0'\}$, $Q' = \{z_0, a, b\} \times \{q_0, q_1, q_2\}$, and let $\mathcal{M}'$ be the dpda

$$\mathcal{M}' = (\{a, b, c\}, Z', Q', \delta', (z_0, q_0), z_0', Z' \times \{q_2\})$$

whose transition function δ' is shown in next is in normal form. The transitions shown in the figure are valid for every $z' \in Z'$; we also have $\delta'(z_0', (z_0, q_0), \lambda) = (z_0' z_0, (z_0, q_0))$.

z	q	a	$\delta(z, q, a)$	$\delta(z', (z, q), a)$
z_0	q_0	a	$(z_0 a, q_0)$	$(z' z_0, (a, q_0))$
z_0	q_0	b	$(z_0 b, q_0)$	$(z' z_0, (b, q_0))$
z_0	q_0	c	(z_0, q_1)	$(z', (z_0, q_1))$
a	q_0	a	(aa, q_0)	$(z' a, (a, q_0))$
a	q_0	b	(ab, q_0)	$(z' a, (b, q_0))$
a	q_0	c	(a, q_1)	$(z', (a, q_1))$
b	q_0	a	(ba, q_0)	$(z' b, (a, q_0))$
b	q_0	b	(bb, q_0)	$(z' b, (b, q_0))$
b	q_0	c	(b, q_1)	$(z', (b, q_1))$
a	q_1	a	(λ, q_1)	$(\lambda, (z', q_1))$
b	q_1	b	(λ, q_1)	$(\lambda, (z', q_1))$
z_0	q_1	λ	(λ, q_2)	$(\lambda, (z', q_2))$

The counterpart of the computation presented previously that leads to the acceptance of the word $x = abbcbba$ is:

$$
\begin{aligned}
(z_0', (z_0, q_0), abbcbba) &\vdash (z_0' z_0, (z_0, q_0), abbcbba) \\
&\vdash (z_0' z_0, (a, q_0), bbcbba) \\
&\vdash (z_0' z_0 a, (b, q_0), bcbba) \\
&\vdash (z_0' z_0 ab, (b, q_0), cbba) \\
&\vdash (z_0' z_0 ab, (b, q_1), bba) \\
&\vdash (z_0' z_0 a, (b, q_1), ba) \\
&\vdash (z_0' z_0, (a, q_1), a) \\
&\vdash (z_0', (z_0, q_1), \lambda) \\
&\vdash (\lambda, (z_0', q_2)).
\end{aligned}
$$

Theorem

If L is a deterministic context-free language and R is a regular language, then $L \cap R$ is a deterministic context-free language.

Proof

Suppose that $L = L(\mathcal{M})$, where $\mathcal{M} = (A, Z, Q, \delta, q_0, z_0, F)$ is a dpda that reads its entire input, and $R = L(\mathcal{M}')$, where $\mathcal{M}' = (A, Q', \delta', q_0', F')$ is a dfa. Define the dpda $\mathcal{M}_1 = (A, Z, Q \times Q', \delta_1, (q_0, q_0'), z_0, F \times F')$ by

$$\delta_1(z, (q, q'), a) = \begin{cases} (v, (q_1, q')) & \text{if } a = \lambda \text{ and} \\ & \delta(z, q, \lambda) = (v, q_1) \\ (v, (q_1, \delta'(q', a))) & \text{if } a \in A \text{ and} \\ & \delta(z, q, a) = \{(q_1, v)\}, \end{cases}$$

for $z \in Z, (q, q') \in Q \times Q'$ and $a \in A \cup \{\lambda\}$.

Proof (cont'd)

We claim that $(z_0, (q_0, q_0'), x) \overset{*}{\underset{\mathcal{M}_1}{\vdash}} (w, (q, q'), \lambda)$ if and only if

$$(z_0, q_0, x) \overset{*}{\underset{\mathcal{M}}{\vdash}} (w, q, \lambda) \text{ and } \delta'^*(q_0', x) = q'.$$

We prove, by induction on n, that if

$$(z_0, (q_0, q_0'), x) \overset{n}{\underset{\mathcal{M}_1}{\vdash}} (w, (q, q'), \lambda),$$

then $(z_0, q_0, x) \overset{n}{\underset{\mathcal{M}}{\vdash}} (w, q, \lambda)$ and $\delta'^*(q_0', x) = q'$. The base case, $n = 0$, is immediate since we have $q = q_0$, $q' = q_0'$, $w = z_0$, and $x = \lambda$.

Suppose that the implication holds for computations of length n and that $(z_0, (q_0, q_0'), x) \overset{n+1}{\underset{\mathcal{M}_1}{\vdash}} (w, (q, q'), \lambda)$. This computation can be written as

$$(z_0, (q_0, q_0'), x) \overset{n}{\underset{\mathcal{M}_1}{\vdash}} (w_1, (q_1, q_1'), a) \underset{\mathcal{M}_1}{\vdash} (w, (q, q'), \lambda),$$

where $x = ya$ and $a \in A \cup \{\lambda\}$. By the inductive hypothesis, we have

$$(z_0, q_0, y) \overset{n}{\underset{\mathcal{M}}{\vdash}} (w_1, q_1, \lambda) \text{ and } \delta'^*(q_0', y) = q_1'.$$

Also, $(w_1, q_1, a) \underset{\mathcal{M}}{\vdash} (w, q, \lambda)$ and $\delta'(q_1', a) = q'$. Therefore, we have

$$(z_0, q_0, x) = (z_0, q_0, ya) \overset{n}{\underset{\mathcal{M}}{\vdash}} (w_1, q_1, a) \underset{\mathcal{M}}{\vdash} (w, q, \lambda),$$

and

$$\delta'^*(q_0', x) = \delta'(\delta'^*(q_0', y), a) = \delta'(q_1', a) = q'.$$