Automata & Process Theory LJK 27/5/2010

A note discussing the constructions of Theorems 4.52 and 4.53.

Recall <u>Theorem 4.52</u>:
Let L be a context-free language. Then there is a push-down automaton M with $\mathcal{L}(M) = L$.

The proof of this theorem gives a construction with which one obtains such an automaton given ~~E~~. a <u>recursive specification</u> over SA.

<u>Construction</u>

Let L be a context free language over alphabet A. This means there is a recursive specification over SA with initial variable S and $\mathcal{L}(S) = L$.

<u>step 1</u> Transform this recursive specification into Greibach normal form using variables $\mathcal{N}$.

<u>step 2</u> Define push-down automaton $M = (S, A, D, \rightarrow, \uparrow, \downarrow)$ as follows:

1) $S = \{\uparrow, \downarrow\}$ (so: two states)

2) $D = \mathcal{N}$ (one stack symbol for each variable)

3) $\uparrow \xrightarrow{\varepsilon, \tau, S} \downarrow$ (initially stack initial variable)

4) For each summand $a.X$ in the right hand side of a variable P, add a step $\downarrow \xrightarrow{P, a, X} \downarrow$ (for $a \in A$, $X \in \mathcal{N}^*$).

5) For each summand 1 in the right hand side of variable P, add a step $\downarrow \xrightarrow{P, \tau, \varepsilon} \downarrow$;

6) $\downarrow$ is a final state

Intuitively, this transformation stacks the variables of which the right-hand side still needs to be processed.

<u>Example</u>

Consider the recursive specification

$S = aSc + T$

$T = bTc + 1$

(which describes the language $L = \{a^n b^m c^k \mid k = n+m\}$ from exercise 4.2.10b).

First we transform this specification to Greibach normal form.

- Remove single variable summand T.

$S \approx a.S.c + b.T.c + 1$

$T \approx b.T.c + 1$

- Introduce equation ~~C~~ $C = c.1$.

$S \approx a.S.C + b.T.C + 1$

$T \approx b.T.C + 1$

$C \approx c.1$

which is in GNF.

Applying the construction, we get ~~states~~

1) $\mathcal{S} = \{\uparrow, \downarrow\}$ 2) $\mathcal{D} = \{S, T, C\}$

3) $\uparrow \xrightarrow{\varepsilon, J, S} \downarrow$

4) $\downarrow \xrightarrow{S, a, SC} \downarrow$

$\downarrow \xrightarrow{S, b, TC} \downarrow$

$\downarrow \xrightarrow{T, b, TC} \downarrow$

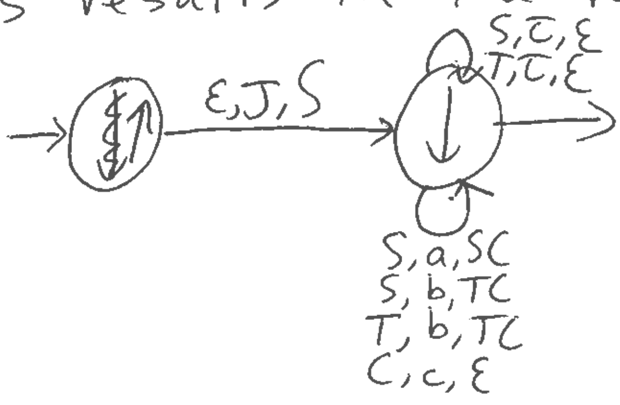
$\downarrow \xrightarrow{C, c, \varepsilon} \downarrow$

5) $\downarrow \xrightarrow{S, \tau, \varepsilon} \downarrow$

$\downarrow \xrightarrow{T, \tau, \varepsilon} \downarrow$

6) $\downarrow$ is a final state.

This results in the following automaton:



There also is a construction from push-down automaton to a recursive specification over SA. This construction is given in Theorem 4.

However, the construction of Theorem 4.53 imposes two restrictions on the input automaton

1. M has exactly one final state $\downarrow$, and this state is only entered when the stack content is ε.
2. M has only push and pop transitions, i.e. transitions $s \xrightarrow{z,a,dz} t$ ($a \in \mathcal{A} \cup \{\tau\}$, $z \in \mathcal{D} \cup \{\varepsilon\}$, $d \in \mathcal{D}$)

   or $s \xrightarrow{z,a,\varepsilon} t$.

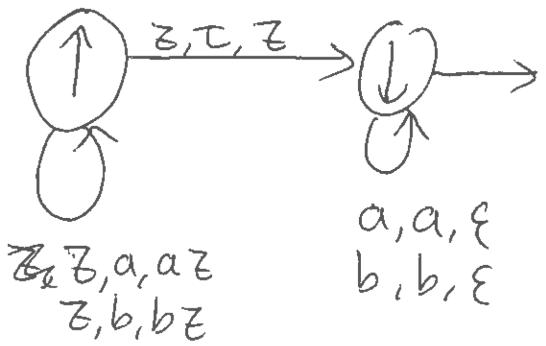
This causes a blow-up in the transformation. Therefore we lift these restrictions: Restriction 1 is dropped completely, and we relax restriction 2 to also allow transitions $s \xrightarrow{z,a,z} t$ ($z \in \mathcal{D} \cup \{\varepsilon\}$).

Now let $M = (\mathcal{S}, \mathcal{A}, \mathcal{D}, \rightarrow, \uparrow, \downarrow)$ be such an automaton. The recursive specification is as follows:

1. $\mathcal{N} = \{V_{s\varepsilon} \mid s \in \mathcal{S}\} \cup \{V_{sdt} \mid s, t \in \mathcal{S}, d \in \mathcal{D}\}$
2. $V_{s\varepsilon}$ has summands $\{a.V_{tdu} \mid s \xrightarrow{\varepsilon,a,d} t, u \in \mathcal{S}\}$
3. V_{sdt} has summands $\{a.1 \mid s \xrightarrow{d,a,\varepsilon} t\}$
4. V_{sdu} has summands $\{a.V_{tev} \cdot V_{vdu} \mid s \xrightarrow{d,a,ed} t, v \in \mathcal{S}\}$
5. $V_{s\varepsilon}$ has summand 1 if $s \in \downarrow$
6. V_{sdu} has summands $\{a.V_{tdu} \mid s \xrightarrow{d,a,d} t\}$
7. $V_{s\varepsilon}$ has summands $\{a.V_{t\varepsilon} \mid s \xrightarrow{\varepsilon,a,\varepsilon} t\}$

We apply this construction to the automaton in Figure 4.5.

Recall the automaton:



$z \in D \cup \{\varepsilon\}$

~~We obt.~~

Observe this automaton adheres to our format, and we obtain variables:

$\mathcal{X} = \{V_{\uparrow\varepsilon}, V_{\downarrow\varepsilon}, V_{\uparrow a\uparrow}, V_{\uparrow a\downarrow}, V_{\uparrow b\uparrow}, V_{\uparrow b\downarrow}, V_{\downarrow a\downarrow}, V_{\downarrow b\downarrow}, V_{\downarrow a\uparrow}, V_{\downarrow a\uparrow}\}$

With the following equations; note that each summand shows in brackets the rule according to which it is introduced.

$$V_{\uparrow\varepsilon} = a.V_{\uparrow a\uparrow}^{(2)} + a.V_{\uparrow a\downarrow}^{(2)} + b.V_{\uparrow b\downarrow}^{(2)} + b.V_{\uparrow b\uparrow}^{(2)} + \tau.V_{\downarrow\varepsilon}^{(7)}$$

$$V_{\downarrow\varepsilon} = 1\ (5)$$

$$V_{\uparrow a\uparrow} = a.V_{\uparrow a\uparrow}\cdot V_{\uparrow a\uparrow}^{(4)} + a.V_{\uparrow a\downarrow}\cdot V_{\downarrow a\uparrow}^{(4)} + b.V_{\uparrow b\uparrow}\cdot V_{\uparrow a\uparrow}^{(4)} + b.V_{\uparrow b\downarrow}\cdot V_{\downarrow a\uparrow}^{(4)} + \tau.V_{\downarrow a\uparrow}^{(7)}$$

$$V_{\uparrow b\uparrow} = a.V_{\uparrow a\uparrow}\cdot V_{\uparrow b\uparrow} + a.V_{\uparrow b\downarrow}\cdot V_{\downarrow a\uparrow} + b.V_{\uparrow b\uparrow}\cdot V_{\uparrow b\uparrow} + b.V_{\uparrow b\downarrow}\cdot V_{\downarrow b\uparrow} + \tau.V_{\downarrow a\uparrow}.$$

$$V_{\uparrow a\downarrow} = a.V_{\uparrow a\uparrow}\cdot V_{\uparrow a\downarrow} + a.V_{\uparrow a\downarrow}\cdot V_{\downarrow a\downarrow} + b.V_{\uparrow b\uparrow}\cdot V_{\uparrow a\downarrow} + b.V_{\uparrow b\downarrow}\cdot V_{\downarrow a\downarrow} + \tau.V_{\downarrow a\downarrow}.$$

$V_{\uparrow b\downarrow}$ = analogous to $V_{\uparrow a\downarrow}$

$$V_{\downarrow a \downarrow} = a \cdot 1 \quad (3)$$

$$V_{\downarrow b \downarrow} = b \cdot 1 \quad (3).$$

$$V_{\downarrow a \uparrow} = 0$$

$$V_{\downarrow b \uparrow} = 0$$

Observe that $v_{\downarrow a \uparrow}$ and $v_{\downarrow b \uparrow}$ are non-productive, and can be removed. As a result, we can also remove 0's in the other equations, resulting in the following specification.

$$V_{\uparrow \varepsilon} = a \cdot V_{\uparrow a \uparrow} + a \cdot V_{\uparrow a \downarrow} + b \cdot V_{\uparrow b \downarrow} + b \cdot V_{\uparrow b \uparrow} + \tau \cdot V_{\downarrow \varepsilon}$$

$$V_{\downarrow \varepsilon} = 1$$

$$V_{\uparrow a \uparrow} = a \cdot V_{\uparrow a \uparrow} \cdot V_{\uparrow a \uparrow} + b \cdot V_{\uparrow b \uparrow} \cdot V_{\uparrow a \uparrow}$$

$$V_{\uparrow b \uparrow} = a \cdot V_{\uparrow a \uparrow} \cdot V_{\uparrow b \uparrow} + b \cdot V_{\uparrow b \uparrow} \cdot V_{\uparrow b \uparrow}$$

$$\begin{aligned} V_{\uparrow a \downarrow} = {} & a \cdot V_{\uparrow a \uparrow} \cdot V_{\uparrow a \downarrow} + a \cdot V_{\uparrow a \downarrow} \cdot V_{\downarrow a \downarrow} \\ & + b \cdot V_{\uparrow b \uparrow} \cdot V_{\uparrow a \downarrow} + b \cdot V_{\uparrow b \downarrow} \cdot V_{\downarrow a \downarrow} \\ & + \tau \cdot V_{\downarrow a \downarrow} \end{aligned}$$

$$\begin{aligned} V_{\uparrow b \downarrow} = {} & a \cdot V_{\uparrow a \uparrow} \cdot V_{\uparrow b \downarrow} + a \cdot V_{\uparrow a \downarrow} \cdot V_{\downarrow b \downarrow} \\ & + b \cdot V_{\uparrow b \uparrow} \cdot V_{\uparrow b \downarrow} + b \cdot V_{\uparrow b \downarrow} \cdot V_{\downarrow b \downarrow} \\ & + \tau \cdot V_{\downarrow b \downarrow} \end{aligned}$$

$$V_{\downarrow a \downarrow} = a \cdot 1$$

$$V_{\downarrow b \downarrow} = b \cdot 1.$$

Now, using the observation that $V_{\uparrow a \uparrow}$ and $V_{\uparrow b \uparrow}$ are non-productive, we can further simplify to:

$$V_{\uparrow \varepsilon} = a \cdot V_{\uparrow a \downarrow} + b \cdot V_{\uparrow b \downarrow} + \tau \cdot V_{\downarrow \varepsilon}$$

$$V_{\downarrow \varepsilon} = 1$$

$$V_{\uparrow a \downarrow} = a \cdot V_{\uparrow a \downarrow} \cdot V_{\downarrow a \downarrow} + b \cdot V_{\uparrow b \downarrow} \cdot V_{\downarrow a \downarrow} + \tau \cdot V_{\downarrow a \downarrow}$$

$$V_{\uparrow b \downarrow} = a \cdot V_{\uparrow a \downarrow} \cdot V_{\downarrow b \downarrow} + b \cdot V_{\uparrow b \downarrow} \cdot V_{\downarrow b \downarrow} + \tau \cdot V_{\downarrow b \downarrow}.$$

$$V_{\downarrow a \downarrow} = a \cdot 1$$

$$V_{\downarrow b \downarrow} = b \cdot 1$$

The intuition behind this translation is as follows. We start in $V_{\uparrow \varepsilon}$, encoding the initial state with an empty stack (which is indeed the initial state!). Variable V_{sdt} encodes that from state s, there is a desire to reach state t ~~with~~ such that symbol d is on the top of the stack if you reach t. In push transitions this is done by going through an intermediate state; in pop transitions you have reached some desired state, and you remove a recursion variable.

In effect, the stack is encoded by a sequence of recursion variables, the number of ~~state~~ variables to be processed changes only in push and pop transitions.