Theorem

Das Leerheitsproblem für kontextsensitive Sprachen ist nicht entscheidbar.

Beweis.

$$I := \left\{ \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}, \ldots, \begin{bmatrix} u_n \\ v_n \end{bmatrix} \right\}$$

$$S \to u_1 M v_1^R \mid \cdots \mid u_n M v_n^R$$
$$M \to u_1 M v_1^R \mid \cdots \mid u_n M v_n^R \mid D$$
$$\vdots$$

(Rest der Grammatik: Übungsaufgabe)

Sprache ist leer gdw. I keine Lösung hat. □

Theorem

Das Wortproblem für kontextsensitive Sprachen ist entscheidbar.

Beweis.

Es sei G eine kontextsensitive Grammatik und w ein Wort.

Wende Regeln rekursiv rückwärts auf alle möglichen Arten an.

Wir erhalten so alle α mit $\alpha \overset{*}{\Rightarrow} w$.

Ist das Startsymbol darunter?

Terminierung: $|\alpha| \leq |w|$ falls $\alpha \overset{*}{\Rightarrow} w$. □

Theorem

Das Wortproblem für Chomsky-0-Sprachen ist nicht entscheidbar.

Beweis.

$$I := \left\{ \boxed{\begin{matrix} u_1 \\ \\ v_1 \end{matrix}}, \boxed{\begin{matrix} u_2 \\ \\ v_2 \end{matrix}}, \ldots, \boxed{\begin{matrix} u_n \\ \\ v_n \end{matrix}} \right\}$$

$$\begin{aligned} S &\to u_1 M v_1^R \mid \cdots \mid u_n M v_n^R \\ M &\to u_1 M v_1^R \mid \cdots \mid u_n M v_n^R \mid D \\ 0D0 &\to D \\ 1D1 &\to D \\ D &\to \epsilon \end{aligned}$$

□

Modellierung nebenläufiger Systeme

Systeme, in welchen Komponenten parallel arbeiten:

1. Technische Systeme
2. Softwaresysteme
3. ...

Komponenten arbeiten teilweise

- unabhängig
- abhängig

voneinander.

Beispiel

Einfaches Programm:

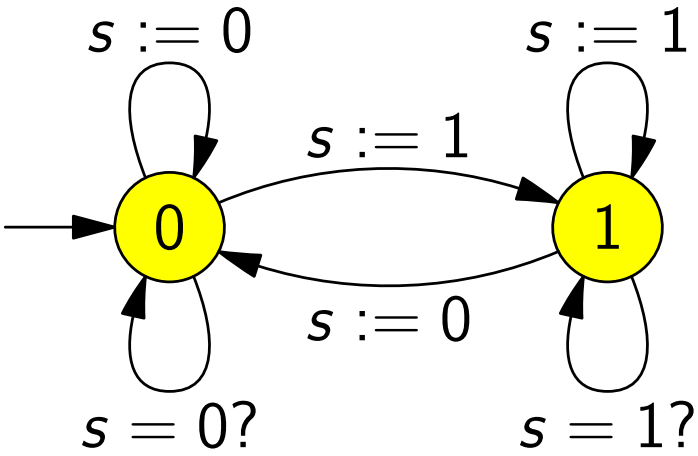
```
s := 0;
s := 1;
if (s=0) print;
else stop;
```

Zerlegen in Komponenten:

- Kontrollstruktur (welche Anweisung wird ausgeführt?)
- Inhalt der Variablen

Interaktion, falls Folgeanweisung vom Variableninhalt abhängt.

Modellierung einer boolschen Variable:

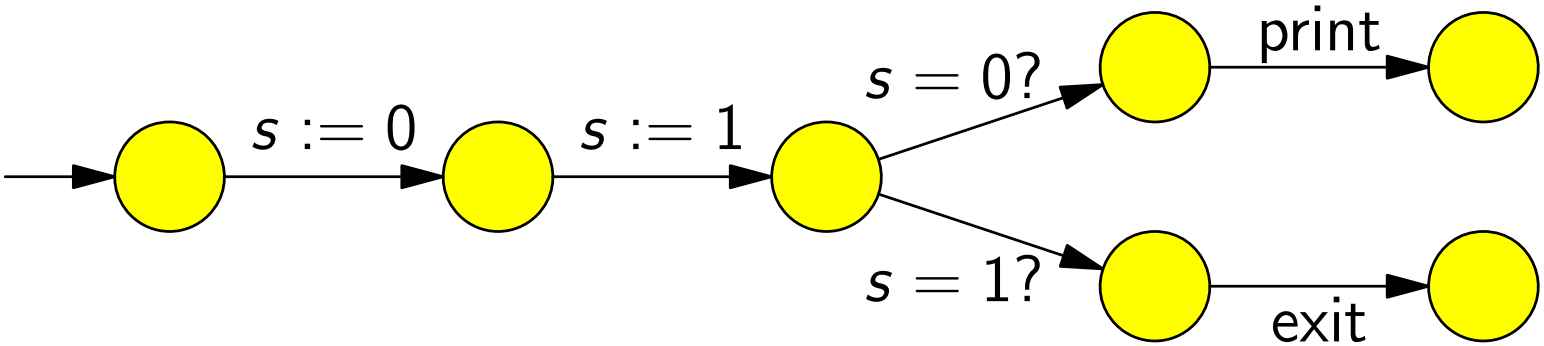


Mögliche Aktionen:

$s := 0$	Variable auf 0 setzen
$s := 1$	Variable auf 1 setzen
$s = 0?$	Variable auf 0 testen
$s = 1?$	Variable auf 1 testen

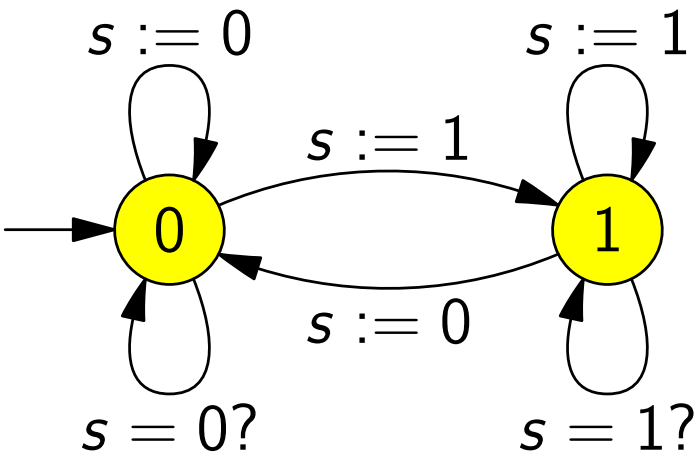
```
s := 0;
s := 1;
if (s=0) print;
else stop;
```

Modellierung der Kontrollstruktur:

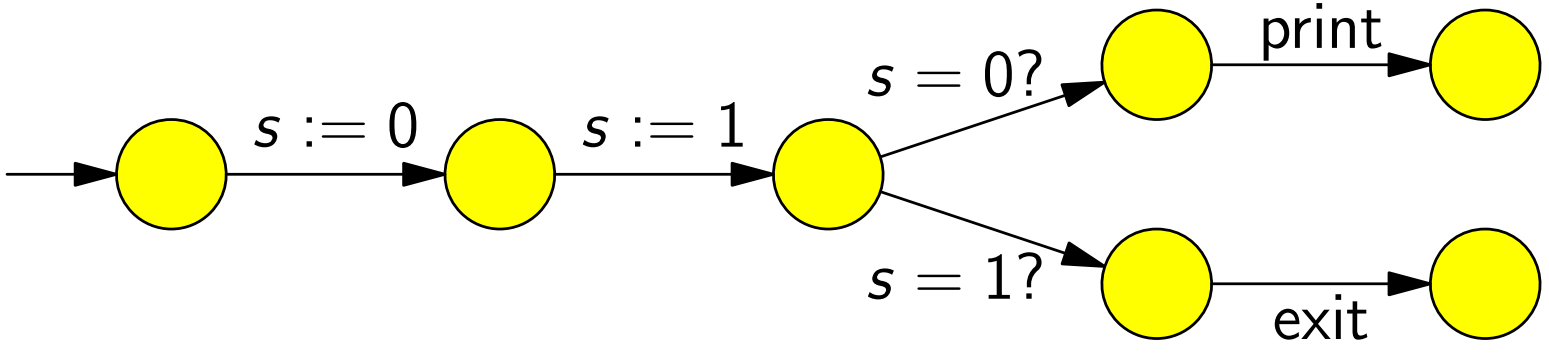


Synchronisiertes Produkt

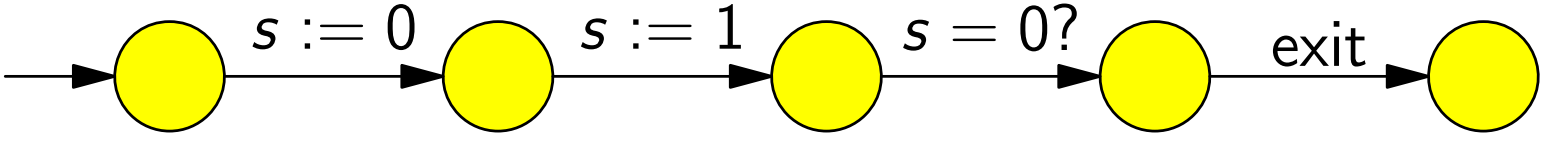
B:



K:



$B \circ K$:



Synchronisiertes Produkt

Definition

Es seien $M' = (\Sigma', Q', \delta', q_0', F')$ und $M'' = (\Sigma'', Q'', \delta'', q_0'', F'')$ zwei NFA's.

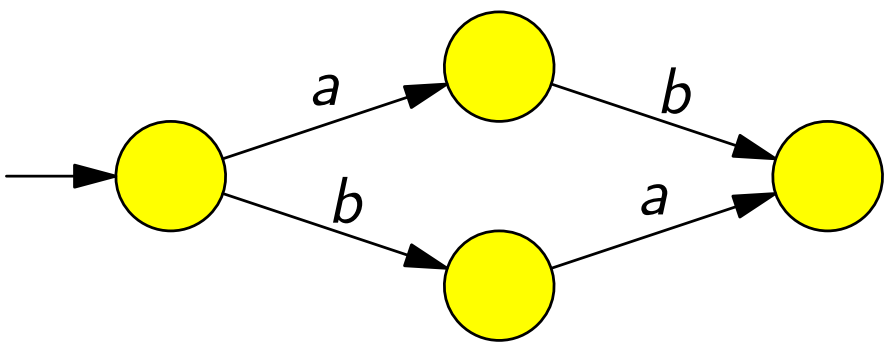
Wir definieren das *synchronisierte Produkt* $M = M' \circ M''$:

$M = (\Sigma, Q' \times Q'', \delta, (q_0', q_0''), F' \times F'')$ mit $\Sigma = \Sigma' \cup \Sigma''$ und

- $(q', p') \in \delta((q, p), a)$ falls $a \in \Sigma' \cap \Sigma''$ und $q' \in \delta'(q, a)$, $p' \in \delta''(p, a)$.
- $(q', p) \in \delta((q, p), a)$ falls $a \in \Sigma' \setminus \Sigma''$ und $q' \in \delta'(q, a)$.
- $(q, p') \in \delta((q, p), a)$ falls $a \in \Sigma'' \setminus \Sigma'$ und $p' \in \delta''(p, a)$.

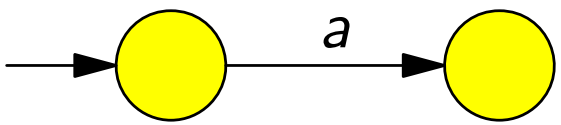
Beispiel

Was ist das synchronisierte Produkt?

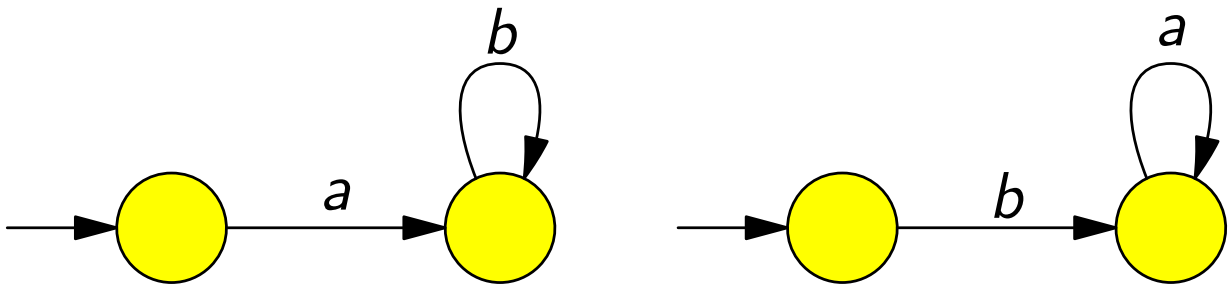


Beispiel

Was ist das synchronisierte Produkt?



Beispiel



Was ist das synchronisierte Produkt?