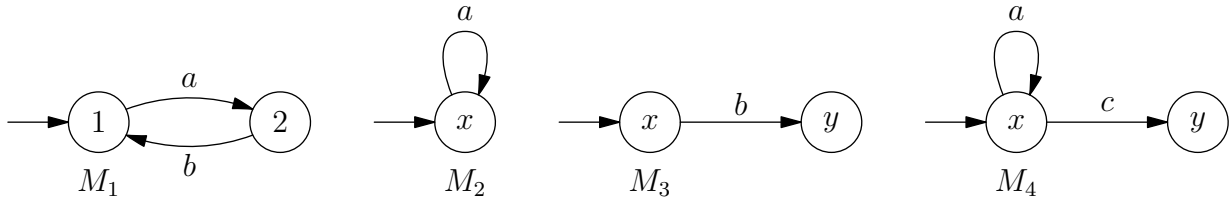


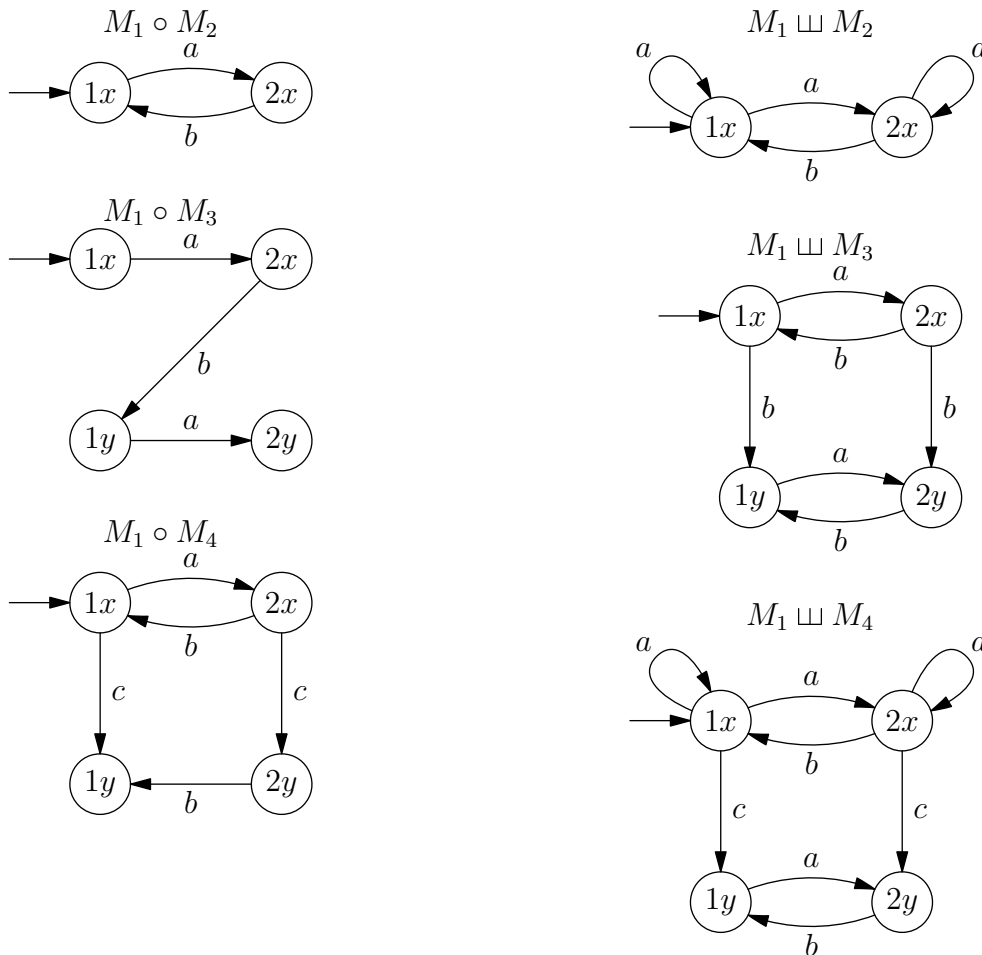
## Übungsblatt mit Lösungen 11

### Aufgabe T30

Bilden Sie  $M_1 \circ M_i$  und  $M_1 \sqcup M_i$ , für  $i \in \{2, 3, 4\}$ . Das Alphabet von  $M_1$  ist  $\{a, b\}$ , von  $M_2$   $\{a\}$ , von  $M_3$   $\{b\}$  und von  $M_4$   $\{a, c\}$ .



### Lösungsvorschlag



### Aufgabe T31

Gegeben sind die folgenden Programme  $P_1$  und  $P_2$ .

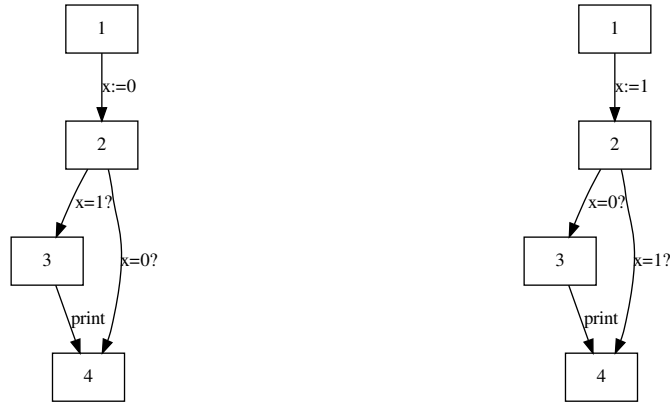
```
x := 1;
if(x=0) print;
```

```
x := 0;
if(x=1) print;
```

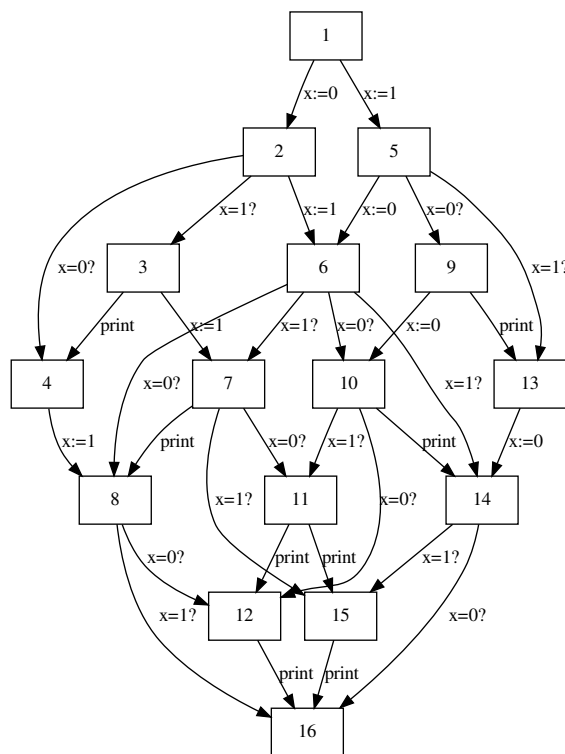
Kann `print` zweimal ausgeführt werden, falls  $P_1$  und  $P_2$  gleichzeitig laufen? Modellieren Sie das Problem mit Hilfe von synchronisierten und unsynchronisierten Produkten wie in der Vorlesung. Begründen Sie ihre Antwort mit Hilfe des entstandenen Automaten.

**Lösungsvorschlag**

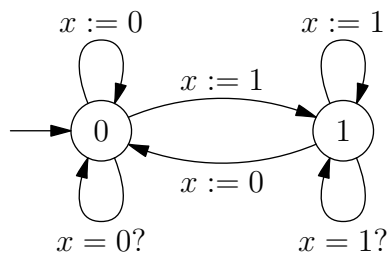
Wir beginnen mit der Modellierung der beiden Ablaufstrukturen  $P_1$  und  $P_2$  der Programme und erhalten:



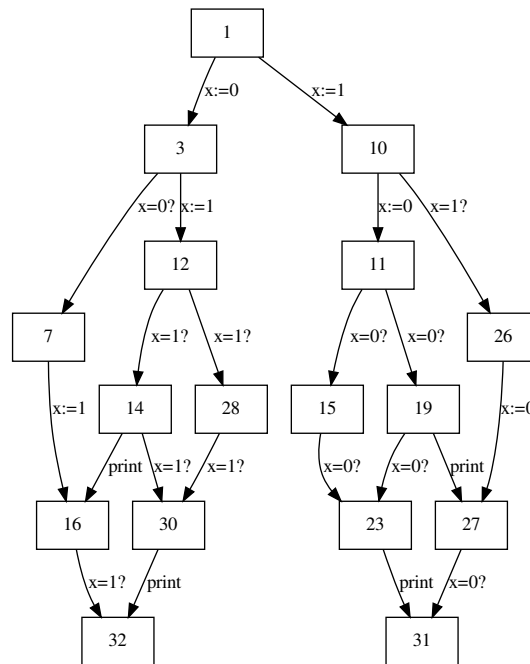
Das unsynchronisierte Produkt  $P_{12}$  von ihnen sieht dann so aus:



Den Automaten  $P_x$  für eine Variable  $x$  haben wir schon in der Vorlesung gesehen:



Schließlich berechnen wir  $P_x \circ P_{12}$ , welches ein Automat ist, dessen Sprache gerade alle möglichen Abläufe im Gesamtsystem enthält:



Wie zu erwarten fallen viele Läufe, welche in  $P_{12}$  noch möglich waren, jetzt weg. Insbesondere ist jetzt leicht zu erkennen, dass die Aktion `print` nicht zweimal ausgeführt werden kann.

### Aufgabe T32

Modellieren Sie folgende Prozesse:

```
while(u=0){
u := 1;
}
print;
```

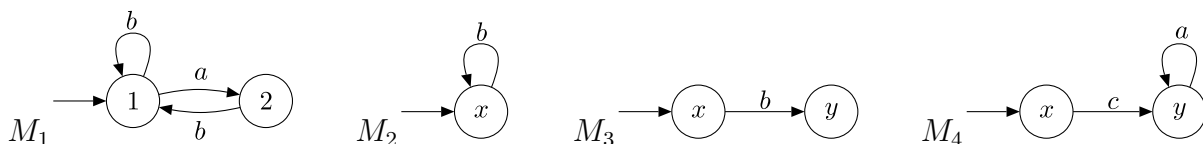
```
while(u=1){
u := 0;
}
print;
```

### Lösungsvorschlag

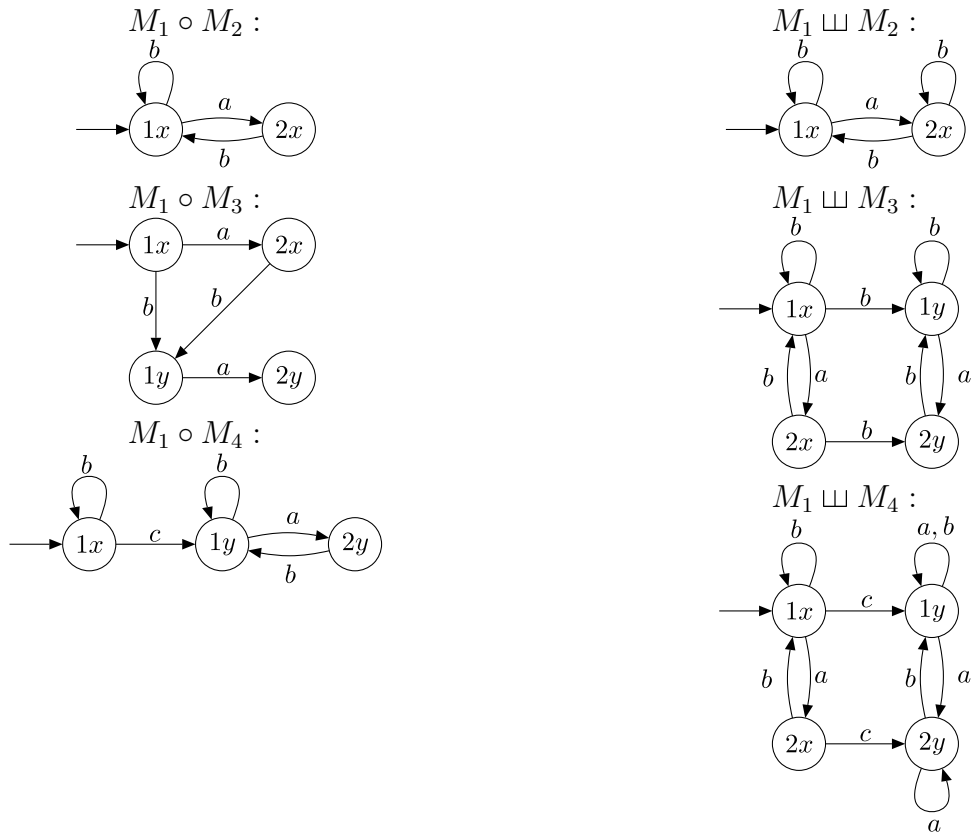
Die Prozesse sind in der Lösung von Aufgabe H37 zu finden.

### Aufgabe H36 (10 Punkte)

Bilden Sie  $M_1 \circ M_i$  und  $M_1 \sqcup M_i$ , für  $i \in \{2, 3, 4\}$ . Das Alphabet von  $M_1$  ist  $\{a, b\}$ , von  $M_2$   $\{b\}$ , von  $M_3$   $\{b\}$  und von  $M_4$   $\{a, c\}$ .



### Lösungsvorschlag



### Aufgabe H37 (10 Punkte)

Gegeben sind die folgenden Programme  $P_1$  und  $P_2$ .

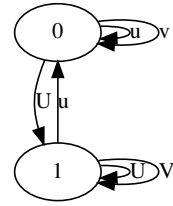
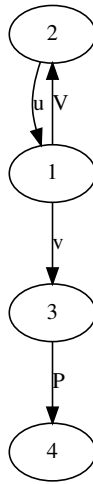
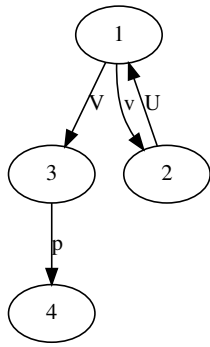
```
while(u=0){
u := 1;
}
print;
```

```
while(u=1){
u := 0;
}
print;
```

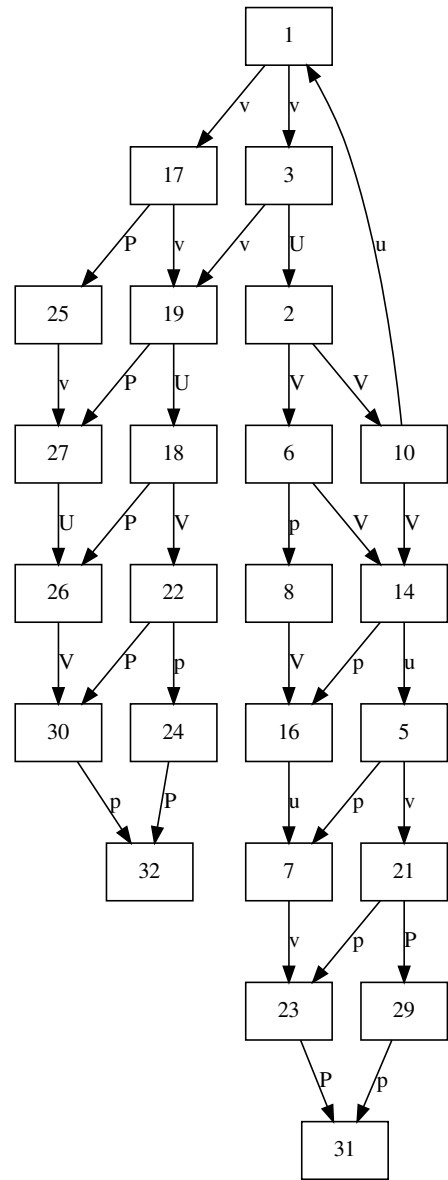
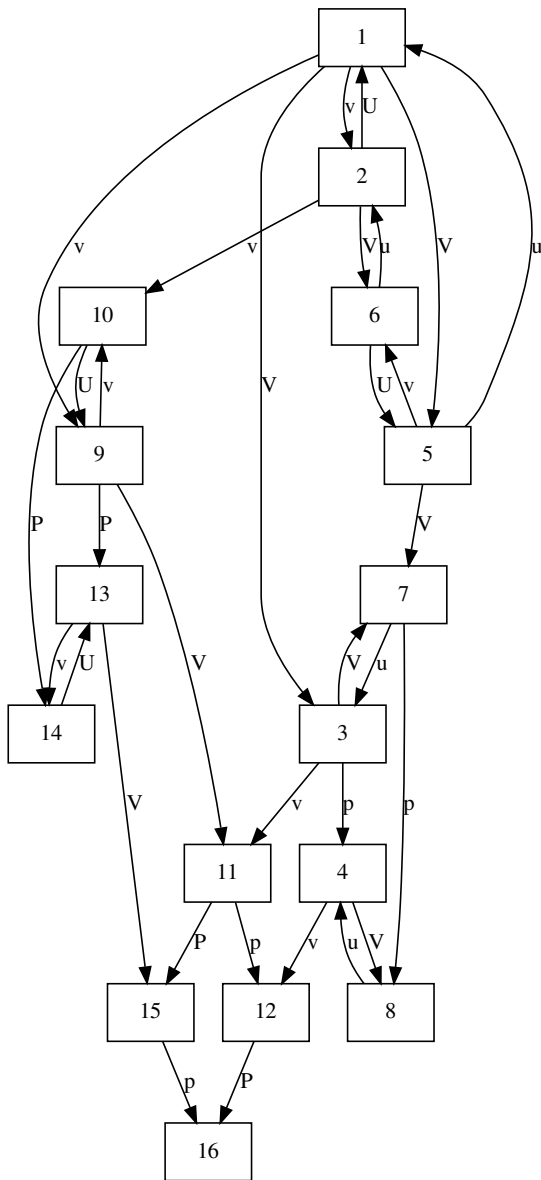
Wie oft kann `print` ausgeführt werden, falls  $P_1$  und  $P_2$  gleichzeitig laufen? Geben Sie alle Möglichkeiten an. Modellieren Sie das Problem mit Hilfe von synchronisierten und unsynchronisierten Produkten wie in der Vorlesung. Begründen Sie Ihre Antwort mit Hilfe des entstandenen Automaten.

### Lösungsvorschlag

Wir modellieren die zwei Programme und die Variable  $u$  wie folgt. Wir verwenden hier  $p$  und  $P$  um zwischen den beiden Programmen unterscheiden zu können. Weiter steht  $u$  bzw.  $U$  für die Variable  $u$  auf 0 bzw. 1 zu setzen. Zudem steht  $v$  bzw.  $V$  für die Verifizierung, dass die Variable 0 bzw. 1 auf steht.



Dann ergibt das unsynchronisierte Produkt  $P_1 \sqcup P_2$  der Automat wie links zu sehen:



Synchronisiert mit Modellierung von  $u$  ergibt sich der rechte Automat. Nun können wir leicht

ablesen wie oft `print` aufgerufen wird: Die einzige Schleife in diesem Automaten besteht aus den Zuständen 1, 3, 2, 10 und ohne ein `print`. Folgt man stets dieser Schleife, so wird `print 0` mal aufgerufen. Wenn nicht, enthält jeder Lauf genau zwei `print`.

**Aufgabe H38** (10 Punkte)

Konstruieren Sie einen deterministischen Kellerautomaten, der die folgende Sprache erkennt:

$$L := \{a, b\}^* \setminus \{a^n b^n \mid n \in \mathbf{N}\}$$

Beschreiben Sie die Idee hinter der Konstruktion. Ein formaler Beweis ist nicht notwendig.

**Lösungsvorschlag**

