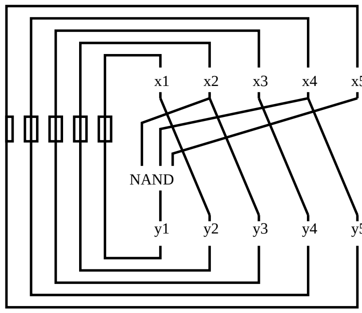
Übungsserie 4 zu Automaten und formalen Sprachen — Uni Leipzig
abzugeben am 16. 12. vor der Vorlesung (autotool: bis 15 Uhr) — Inst. f. Informatik

---

1. (schriftlich) Für Zahlen $1 \leq i_1 < \ldots < i_k$ betrachten wir Schaltnetze (Schieberegister) $S[i_1, \ldots, i_k]$ der folgenden Form:

- Eingänge $x_1, \ldots$, Ausgänge $y_1, \ldots$,
- $y_1 = \neg(x_{i_1} \wedge \ldots \wedge x_{i_k})$,
- für alle $1 \leq i$: $y_{i+1} = x_i$,
- für alle $1 \leq i$: eine Verzögerungsleitung (Rückführung) von y_i zu x_i
- Startzustand: für $1 \leq i$: $x_i = 1 =$ True



Das Schaltnetz $S[2, 4, 5]$

Jedes solche Netz definiert eine unendliche Folge $[y_1(0), y_1(1), \ldots y_1(t), \ldots]$ über $\Sigma = \{0, 1\}$ der Werte an y_1 zum jeweiligen Takt t.

Für das Beispiel $S[2, 4, 5]$ entsteht die Folge $[0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, \ldots]$.

Beweisen Sie, daß für alle $n > 0$ das Netz $S[1, 2n, 2n + 1]$ eine Folge mit Periodenlänge $4n$ erzeugt. 2 P.

Beweisen Sie, daß für *jedes* $S[i_1, \ldots, i_k]$ die Folge $y_1(\cdot)$ schließlich periodisch ist, d. h. es gibt Zahlen q (Vorperiode) und p (Periode), so daß $\forall t \geq q : y_1(t) = y_1(t + p)$.

Bestimmen Sie dazu eine allgemein gültige Schranke für p (und q) in Abhängigkeit von $[i_1, \ldots, i_k]$. Hinweis: wieviele verschiedene Zustände kann das Netz annehmen? 2 P.

2. (autotool: Shift) Finden Sie solche Parameter $i_1 < \ldots < i_k \leq 25$, für die die Periode von $S[i_1, \ldots, i_k]$ möglichst groß wird. Benutzen Sie das Format: High-score

```
import Shift.Type
student = Shift { pins = [1,10,19,20] , vorperiode = 0 , periode = 181 }
```

---

3. (autotool: Equiv) Minimieren Sie nach dem Verfahren aus Vorlesung/Skript einen (personalisierten) deterministischen endlichen Automaten. Konstruieren Sie dabei schrittweise die Äquivalenzrelationen $\sim_i$. Geben Sie dazu Paare von nicht äquivalenten Zuständen in diesem Format an: 1 P.

```
student = [[(1,2,'b'),(3,4,'a')],[(1,3,'a')]]
```

Das Tupel (p, q, c) soll genau dann in der i-ten Liste vorkommen, wenn

$$(p \sim_{i-1} q) \wedge (f(p, c) \not\sim_{i-1} f(q, c)).$$

Beispiel-Aufgabe und -Lösung: `http://theopc.informatik.uni-leipzig.de/~autotool/challenger-web/Equiv/1234567/`

(Aufgabe 4 auf der Rückseite)

4. Betrachten Sie die Grammatik $G = (M, A, R, X)$ mit $M = \{X, Y, Z\}, A = \{a, b, c\}$ und der Regelmenge

$$R = \{X \to abZ, X \to aXYZ, ZY \to YZ, bY \to bb, bZ \to bc, cZ \to cc\}$$

und die Sprache $D = \{a^n b^n c^n \mid n \geq 1\}$.
(autotool: ABL) Geben Sie für einige Wörter $w \in (M \cup A)^*$ Ableitungen $X \to_R^* w$ an. Benutzen Sie das Format 1 P.

```
student = [ ["X","abZ","abc"], ["X","aabbcc"] ]
```

(schriftlich) Zeigen Sie durch Induktion nach Länge der Ableitung:
Für jedes $w \in (M \cup A)^*$ mit $X \to_R^* w$ gilt:

- $|w|_{\{a\}} = |w|_{\{b,Y\}} = |w|_{\{c,Z\}}$ 1 P.
- $w \in a^* \ (\{X\} \cup b^*c^*) \ \{Y, Z\}^*$ 2 P.

Hierbei bedeutet $|w|_M$ die Anzahl der Positionen in w, auf denen Buchstaben aus M vorkommen.
Folgern Sie daraus $L(G) \subseteq D$. 1 P.
(Diese Grammatik ist entnommen aus: A. Salomaa: *Formal Languages*, New York 1973, dt.: Berlin 1978, Abschnitt I.2)