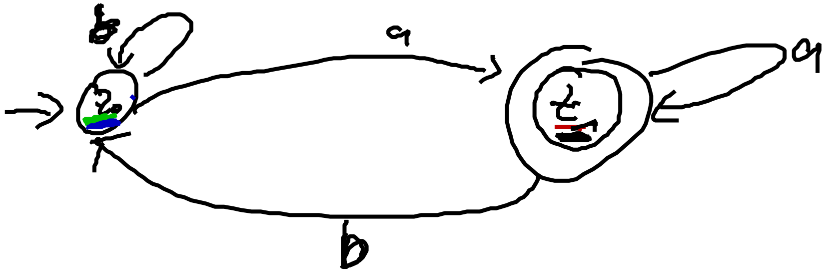
Kreuzproduktautomat

$M_1 = (Z_1, \Sigma, \delta_1, z_{01}, E_1)$ $\quad M_2 = (Z_2, \Sigma, \delta_2, z_{02}, E_2)$

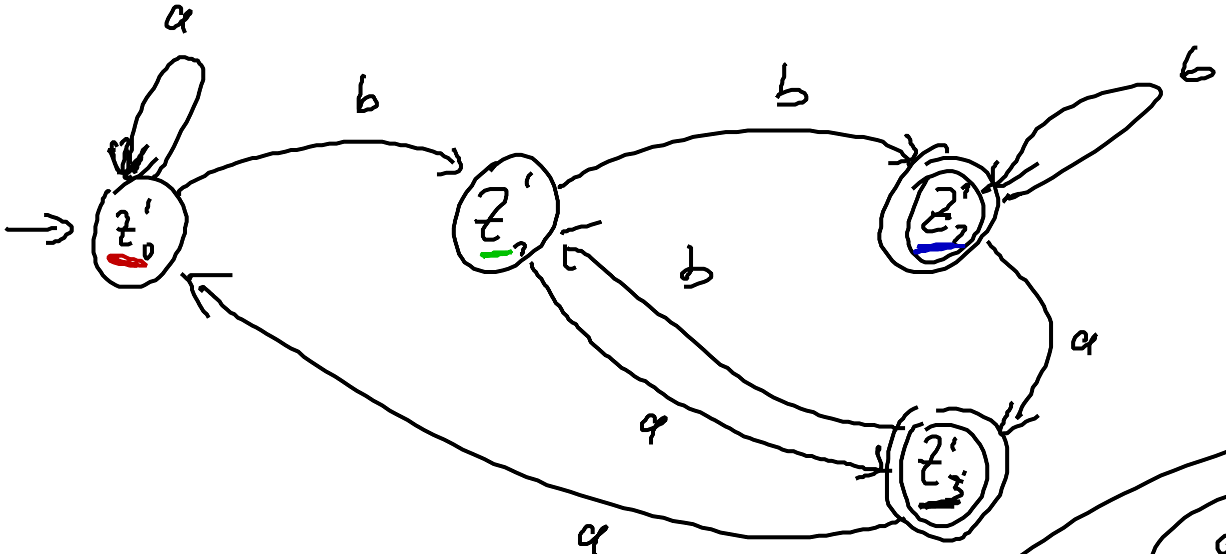
Konstr M mit $T(M) = T(M_1) \cap T(M_2)$:

$$M = (Z_1 \times Z_2, \Sigma, \delta, (z_{01}, z_{02}), E_1 \times E_2)$$

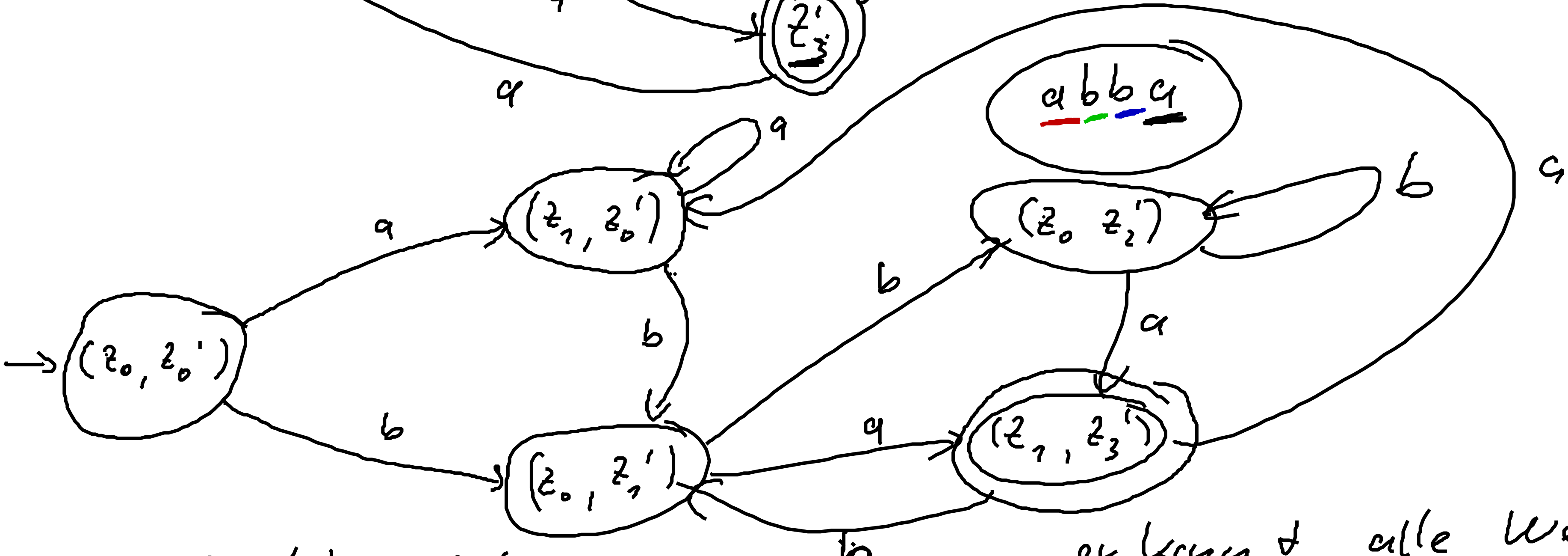
$$\delta((z, z'), a) = (\delta_1(z, a), \delta_2(z', a))$$



alle Worte, die auf a enden



erkennt alle Worte die an vorletzter Stelle ein b haben



erkennt alle Worte, die auf ba enden.

<u>Bew</u>: Konstruktion funktioniert auch für NFAs.

1.6.7 Entscheidbarkeit

Satz Das Wortproblem für reguläre Sprachen ist entscheidbar. (Für DFAs sogar in linearer Zeit)..

Satz Das Leerheitsproblem ($L = \emptyset$?) ist für reg. Spr. entscheidbar

Bew: Teste, ob es in dem DFA einen Pfad vom Anfangszustand zu einem Endzustand gibt.
Wenn ja, dann $L \neq \emptyset$ sonst $L = \emptyset$. □

Satz Das Endlichkeitsproblem ($|L| < \infty$) ist für reg. Spr. entscheidbar.

Bew: Für DFA: Entferne alle Zustände, die nicht von einem Anfangszustand aus erreichbar sind, und die, die von denen keine Endzustände erreichbar sind. Enthält der Automat dann noch Zyklen, dann gilt $|L| = \infty$, sonst $|L| < \infty$. □

Satz Das Disjunktheitsproblem ($L_1 \cap L_2 = \emptyset$?) ist für reg. Spr. entscheidbar.

Bew: Kreuzproduktautomat konstr. und auf Leerheit prüfen.

Satz. Das Äquivalenzproblem ($L_1 = L_2$?) ist für reg. Spr. entscheidbar.

Bew: Minimalautomaten konstruieren und auf Äquivalenz prüfen.

Mengentheoretisch: $L_1 = L_2$ gdw. $L_1 \cap \overline{L_2} = \emptyset$ und $\overline{L_1} \cap L_2 = \emptyset$

gdw. $(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \emptyset$

Bem: Effizienz hängt von Darstellung der Sprache ab.

Bsp: Äquivalenz ist einfach für DFAs ist effizient entscheidbar, für NFAs ist es schwierig (NP-hart).

1.7 kontextfreie Sprachen

Bsp $L = \{1^n 0^n \mid n \geq 1\}$ ist nicht regulär, aber kontextfrei $S \Rightarrow 10 \mid 1S0$

1.7.1 Normalformen

Def Eine kontextfreie Gram. G mit $\varepsilon \notin L(G)$ ist in Chomsky-Normalform (CNF), falls alle Regeln die folgende Form haben

$$A \to BC \text{ oder } A \to a,$$

wobei $A, B, C \in V$ und $a \in \Sigma$.

Satz Zu jeder Typ 2 Gram. G mit $\varepsilon \notin L(G)$ ex eine Gram G' in CNF mit $L(G) = L(G')$.

Bsp:

$\cancel{S}\ X \to 10$

$\cancel{S}\ X \to A\,\cancel{T}\,X\,0$

$\cancel{T \to S}$

$\cancel{S \to T}$

$A \to B$

$B \to 1$

S,T ersetzen durch X

$X \to \cancel{10}\ E\,N$

$X \to A\,X\,\cancel{0}\ N$

$\cancel{A \to B}$ $\qquad A \to 1$

$B \to 1$

$E \to 1$

$N \to 0$

$X \to E\,N$

$\cancel{X \to A\,X\,N}$

$A \to 1$

$\cancel{B \to 1}$

$E \to 1$

$N \to 0$

$X \to A\,C_1$

$C_1 \to X\,N$

Bew

1) Eliminierung der Regeln der Art $A \to B$

1a) Identifiziere Regeln der Art

$$B_1 \to B_2, \quad B_2 \to B_3, \ldots, \quad B_k \to B_1$$

Ersetze die B_i's durch ein neues Symbol B und lösche die Regeln.

1b) Nummeriere die Var. so, dass für Regeln $A_i \to A_j \in P$ gilt $i < j$.
Absteigend $k = n, \ldots, 1$ eliminiere Regeln der Art $A_l \to A_k$, und nimm folgendes hinzu:
Falls $A_k \to x_1, \; A_k \to x_2, \ldots, A_k \to x_u, \quad x_i \in (V \cup \Sigma)^*$,
dann nehme hinzu

$$A_l \to x_1, \; A_l \to x_2, \ldots, A_l \to x_k.$$

Alle Regeln haben die Form

$A \to a$, $A \in V$, $a \in \Sigma$

$A \to x$, $A \in V$, $x \in (V \cup \Sigma)^*$, $|x| \geq 2$

2a) Füge für jedes Terminalsymbol $b \in \Sigma$ eine Variable B hinzu und ergänze die Regeln um $B \to b$

2b) Ersetze alle Terminalsymbole auf der rechten Seite durch die entsprechende Variable (außer bei Regeln der Art $A \to a$)

Alle Regeln haben jetzt die Form

$A \to a$, $A \in V$, $a \in \Sigma$

$A \to x$, $A \in V$, $x \in V^*$, $|x| \geq 2$

3) Spalte Regeln der Art

$A \to B_1 B_2 \dots B_k$, $k > 2$, auf:

$A \to B_1 C_2$

$C_2 \to B_2 C_3$

$\vdots$

$C_{k-1} \to B_{k-1} B_k$ für neue Variablensymb. $C_1, \dots, C_{k-1}$

Jetzt ist die Gram. in der gewünschten Form, und offensichtlich wird auch die gleiche Sprache erzeugt. $\square$

Def Eine Typ 2-Gram. G mit $\varepsilon \notin L(G)$ ist in <u>Greibach-Normalform</u>, falls alle Regeln die folgende Gestalt haben:

$$A \to \underline{a} B_1 \dots B_k \quad (k \geq 0), \quad A, B_i \in V, \quad a \in \Sigma .$$

Satz Zu jeder Typ 2-Gram. G mit $\varepsilon \notin L(G)$, ist es eine Gram G' in Greibach-Normalform mit $L(G') = L(G)$.

Bew: <u>ohne</u>

1.7.2 Pumping-Lemma für kontextfreie Sprachen

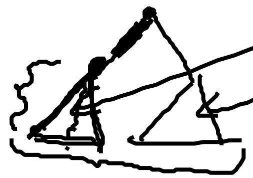
Lemma Sei B ein Binärbaum mit mindestens 2^k Blättern. Dann hat B mindestens einen Pfad der Länge k oder größer.

Bew: Induktion über k.

IA: Jeder Binärbaum mit $2^0 = 1$ Blättern hat einen Pfad der Länge 0.

IV: Die Beh gelte für k.

IS: Betrachte Binärbaum mit $\geq 2^{k+1}$ Blätter



Verlängerung bis zur Wurzel $\Rightarrow$ Es ex. ein Pfad der Länge $k+1$

Satz (Pumping-Lemma für kf. Sprachen, uvwxy-Theorem)

Sei L eine kontextfreie Sprache. Dann ex. eine Zahl $n \in \mathbb{N}$, so dass sich alle Wörter $z \in L$ mit $|z| \geq n$ zerlegen lassen in $z = uvwxy$ mit folgenden Eigenschaften:

(1) $|vx| \geq 1$

(2) $|vwx| \leq n$

(3) $u\underline{v}^i w \underline{x}^i y \in L, \quad \forall i \geq 0$

$z = uvw$

$|v| \geq 1$

$|uv| \leq n$

$uv^i w \in L$