4b
Eigenschaften von regulären Sprachen

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 6. Mai 2024
Basiert auf Folien von PD Dr. David Sabel

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Vereinigung, Produkt und Kleeneschem Abschluss. D.h. wenn L_1, L_2 regulär sind, dann sind $L_1 \cup L_2$, $L_1 L_2$ und ${L_1}^*$ regulär.

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Vereinigung, Produkt und Kleeneschem Abschluss. D.h. wenn L_1, L_2 regulär sind, dann sind $L_1 \cup L_2$, $L_1 L_2$ und ${L_1}^*$ regulär.

Beweis Vereinigung, Produkt und Kleenescher Abschluss werden durch reguläre Ausdrücke erzeugt und sind daher reguläre Sprachen.

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Vereinigung, Produkt und Kleeneschem Abschluss. D.h. wenn L_1, L_2 regulär sind, dann sind $L_1 \cup L_2$, $L_1 L_2$ und ${L_1}^*$ regulär.

Beweis Vereinigung, Produkt und Kleenescher Abschluss werden durch reguläre Ausdrücke erzeugt und sind daher reguläre Sprachen.

Seien reguläre Ausdrücke α_1, α_2 mit $L(\alpha_i) = L_i$ für $i \in \{1, 2\}$.

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Vereinigung, Produkt und Kleeneschem Abschluss. D.h. wenn L_1, L_2 regulär sind, dann sind $L_1 \cup L_2$, $L_1 L_2$ und ${L_1}^*$ regulär.

Beweis Vereinigung, Produkt und Kleenescher Abschluss werden durch reguläre Ausdrücke erzeugt und sind daher reguläre Sprachen.

Seien reguläre Ausdrücke α_1, α_2 mit $L(\alpha_i) = L_i$ für $i \in \{1, 2\}$.

- $(\alpha_1 | \alpha_2)$ erzeugt $L(\alpha_1 | \alpha_2) = L(\alpha_1) \cup L(\alpha_2) = L_1 \cup L_2$.
- $\alpha_1 \alpha_2$ erzeugt $L(\alpha_1 \alpha_2) = L(\alpha_1) L(\alpha_2) = L_1 L_2$.
- $(\alpha_1)^*$ erzeugt $L(\alpha_1^*) = L(\alpha_1)^* = {L_1}^*$. $\square$

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Komplementbildung. D.h. wenn L regulär ist, dann ist das Komplement $\overline{L}$ regulär.

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Komplementbildung. D.h. wenn L regulär ist, dann ist das Komplement $\overline{L}$ regulär.

Beweis Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA, der L akzeptiert.

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Komplementbildung. D.h. wenn L regulär ist, dann ist das Komplement $\overline{L}$ regulär.

Beweis Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA, der L akzeptiert.

Dann akzeptiert $\overline{M} = (Z, \Sigma, \delta, z_0, Z \setminus E)$ die Sprache $\overline{L}$:
Offensichtlich gilt $\widetilde{\delta}(z_0, w) \in Z \setminus E$ g.d.w. $\widetilde{\delta}(z_0, w) \notin E$.

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Komplementbildung. D.h. wenn L regulär ist, dann ist das Komplement $\overline{L}$ regulär.

Beweis Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA, der L akzeptiert.

Dann akzeptiert $\overline{M} = (Z, \Sigma, \delta, z_0, Z \setminus E)$ die Sprache $\overline{L}$:
Offensichtlich gilt $\widetilde{\delta}(z_0, w) \in Z \setminus E$ g.d.w. $\widetilde{\delta}(z_0, w) \notin E$.

Daher ist $\overline{L}$ regulär. $\square$

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Schnitt. D.h. wenn L_1, L_2 regulär sind, dann ist $L_1 \cap L_2$ regulär.

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Schnitt. D.h. wenn L_1, L_2 regulär sind, dann ist $L_1 \cap L_2$ regulär.

Beweis Dies folgt aus $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ und da reguläre Sprachen abgeschlossen bezüglich Vereinigung und Komplementbildung sind. $\square$

Abschlusseigenschaften

Satz

Die regulären Sprachen sind abgeschlossen bezüglich Schnitt. D.h. wenn L_1, L_2 regulär sind, dann ist $L_1 \cap L_2$ regulär.

Alternativer Beweis

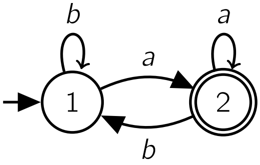
Seien $M_1 = (Z_1, \Sigma, \delta_1, z_{01}, E_1)$ und $M_2 = (Z_2, \Sigma, \delta_2, z_{02}, E_2)$ DFAs, die $L_1 = L(M_1)$ und $L_2 = L(M_2)$ akzeptieren.
Der Produktautomat von M_1 und M_2 ist der DFA
$M_1 \times M_2 = (Z_1 \times Z_2, \Sigma, \delta, (z_{01}, z_{02}), E_1 \times E_2)$ mit

$$\delta((z_1, z_2), a) := (\delta_1(z_1, a), \delta_2(z_2, a)) \quad \text{für alle } a \in \Sigma \text{ und } (z_1, z_2) \in Z_1 \times Z_2$$

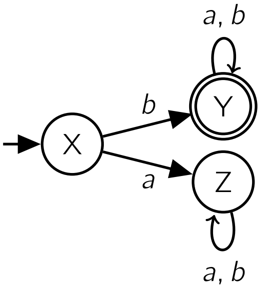
M akzeptiert $L_1 \cap L_2$, denn es gilt:
$\widetilde{\delta}((z_{01}, z_{02}), w) \in E_1 \times E_2$ g.d.w. $\widetilde{\delta_1}(z_{01}, w) \in E_1$ und $\widetilde{\delta_2}(z_{02}, w) \in E_2$. $\square$

Beispiel für den Produktautomaten

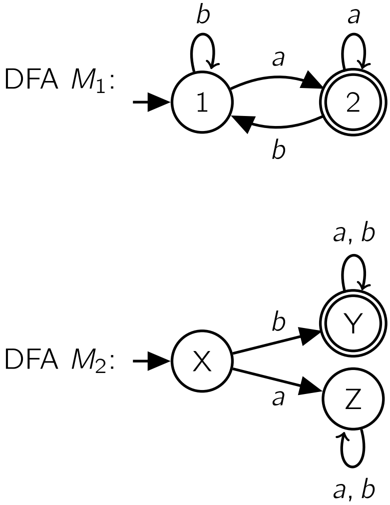
DFA M_1:



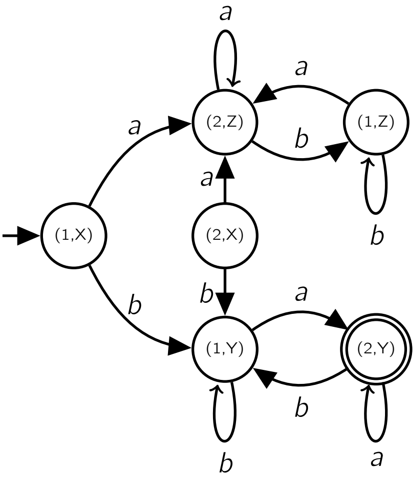
DFA M_2:



Beispiel für den Produktautomaten

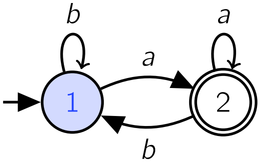


Produktautomat $M_1 \times M_2$:

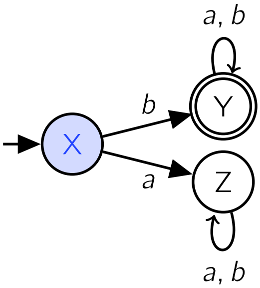


Beispiel für den Produktautomaten

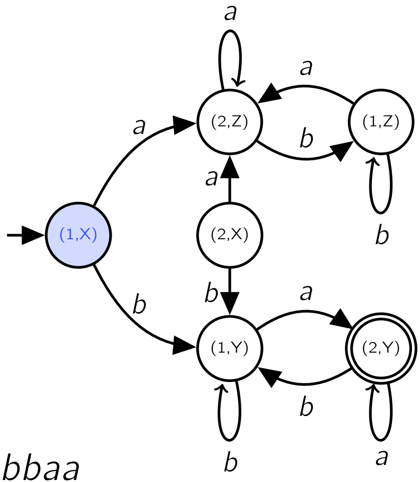
DFA M_1:



DFA M_2:



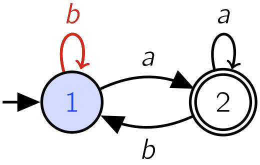
Produktautomat $M_1 \times M_2$:



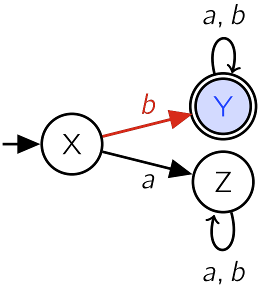
Wort: $bbaa$

Beispiel für den Produktautomaten

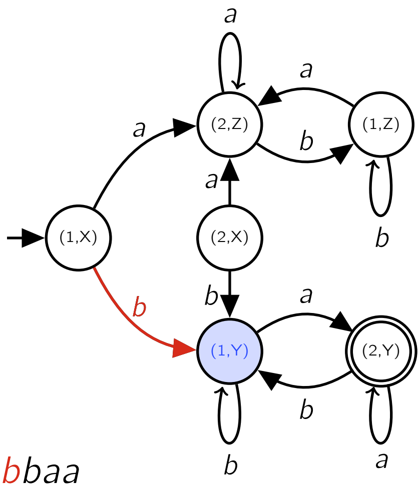
DFA M_1:



DFA M_2:



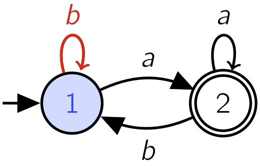
Produktautomat $M_1 \times M_2$:



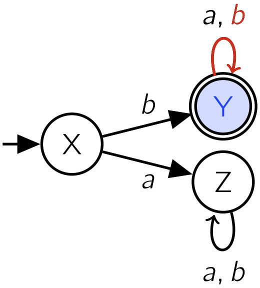
Wort: *bbaa*

Beispiel für den Produktautomaten

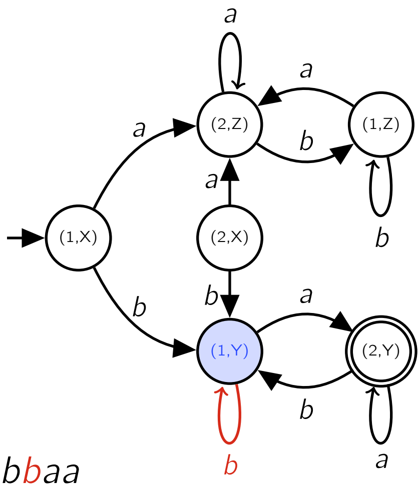
DFA M_1:



DFA M_2:



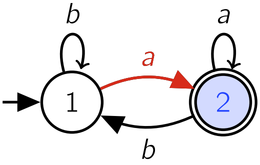
Produktautomat $M_1 \times M_2$:



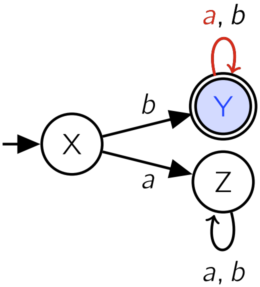
Wort: *bbaa*

Beispiel für den Produktautomaten

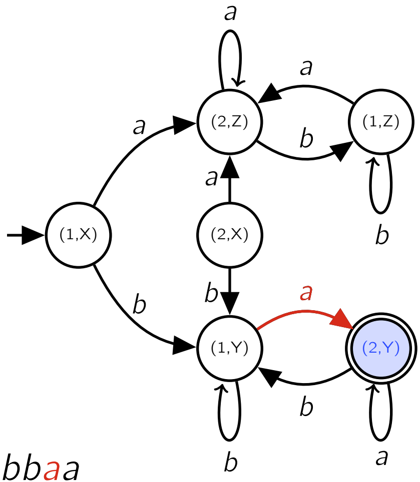
DFA M_1:



DFA M_2:



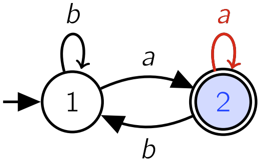
Produktautomat $M_1 \times M_2$:



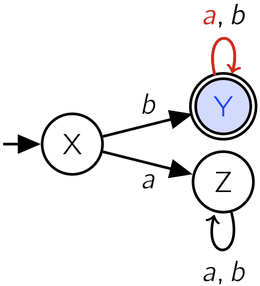
Wort: $bbaa$

Beispiel für den Produktautomaten

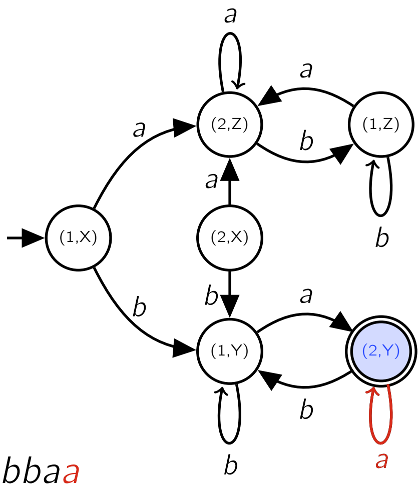
DFA M_1:



DFA M_2:



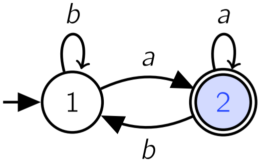
Produktautomat $M_1 \times M_2$:



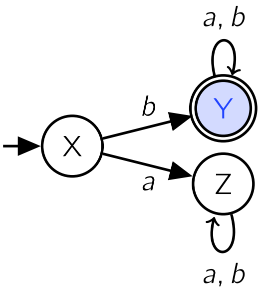
Wort: *bbaa*

Beispiel für den Produktautomaten

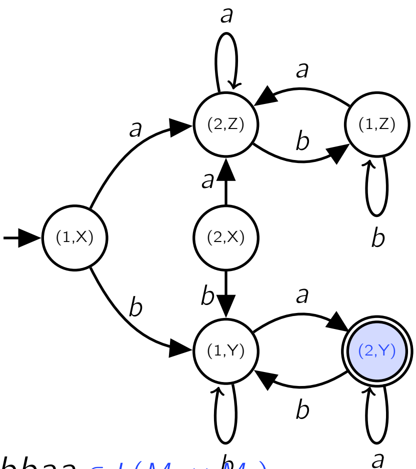
DFA M_1:



DFA M_2:



Produktautomat $M_1 \times M_2$:



Wort: $bbaa \in L(M_1 \times M_2)$

Abschlusseigenschaften zusammengefasst

Theorem (Abschlusseigenschaften der regulären Sprachen)

Die regulären Sprachen sind abgeschlossen bezüglich Vereinigung, Schnitt, Komplementbildung, Produkt und Kleeneschem Abschluss.

Nichtregularität zeigen mit Abschlusseigenschaften

Anleitung zum Widerlegen der Regularität von L mit Abschlusseigenschaften:

1. Nehme an, dass L regulär ist.
2. Operiere auf L unter Erhaltung der Regularität:
   vereinige, schneide, komplementiere, multipliziere L mit bekannt regulärer Sprache, bilde Kleeneschen Abschluss.
3. Kommt dabei eine bekannt nicht reguläre Sprache heraus,
   dann hat man einen Widerspruch und die Annahme war falsch.
   Daher ist L dann nicht regulär.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{a^n \mid n \text{ ist keine Primzahl}\}$ ist nicht regulär.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{a^n \mid n \text{ ist keine Primzahl}\}$ ist nicht regulär.

Beweis Durch Widerspruch. Wir nehmen an, L ist regulär.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{a^n \mid n \text{ ist keine Primzahl}\}$ ist nicht regulär.

Beweis Durch Widerspruch. Wir nehmen an, L ist regulär.

Dann ist $\overline{L}$ auch regulär.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{a^n \mid n \text{ ist keine Primzahl}\}$ ist nicht regulär.

Beweis Durch Widerspruch. Wir nehmen an, L ist regulär.

Dann ist $\overline{L}$ auch regulär.

$\overline{L} = \{a\}^* \setminus L = \{a^n \mid n \text{ ist eine Primzahl}\}$ ist aber nicht regulär (bereits gezeigt).
Widerspruch. □

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ ist nicht regulär.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ ist nicht regulär.

Beweis Durch Widerspruch. Wir nehmen an, L ist regulär.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ ist nicht regulär.

Beweis Durch Widerspruch. Wir nehmen an, L ist regulär.

Die Sprache $L' = \{a^n b^m \mid n, m \in \mathbb{N}\}$ ist regulär, da der reguläre Ausdruck $a^* b^*$ sie erzeugt.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ ist nicht regulär.

Beweis Durch Widerspruch. Wir nehmen an, L ist regulär.

Die Sprache $L' = \{a^n b^m \mid n, m \in \mathbb{N}\}$ ist regulär, da der reguläre Ausdruck $a^* b^*$ sie erzeugt.

Da L und L' regulär sind, ist auch $L \cap L'$ regulär.

Nichtregularität zeigen mit Abschlusseigenschaften

Satz

Die Sprache $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ ist nicht regulär.

Beweis Durch Widerspruch. Wir nehmen an, L ist regulär.

Die Sprache $L' = \{a^n b^m \mid n, m \in \mathbb{N}\}$ ist regulär, da der reguläre Ausdruck $a^* b^*$ sie erzeugt.

Da L und L' regulär sind, ist auch $L \cap L'$ regulär.

$L \cap L' = \{a^j b^j \mid j \in \mathbb{N}\}$ ist aber nicht regulär (bereits gezeigt). Widerspruch. $\square$

Wortproblem für reguläre Grammatiken

Definition

Das Wortproblem für Typ i-Grammatiken ist die Frage, ob für eine gegebene Typ i-Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$ $w \in L(G)$ gilt oder nicht.

Wortproblem für reguläre Grammatiken

Definition

Das Wortproblem für Typ i-Grammatiken ist die Frage, ob für eine gegebene Typ i-Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$ $w \in L(G)$ gilt oder nicht.

Satz

Das Wortproblem für reguläre Grammatiken ist entscheidbar:
Es gibt einen Algorithmus, der bei Eingabe von regulärer Grammatik G und Wort w nach endlicher Zeit entscheidet, ob $w \in L(G)$ gilt oder nicht.

Wortproblem für reguläre Grammatiken

Definition

Das Wortproblem für Typ i-Grammatiken ist die Frage, ob für eine gegebene Typ i-Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$ $w \in L(G)$ gilt oder nicht.

Satz

Das Wortproblem für reguläre Grammatiken ist entscheidbar:
Es gibt einen Algorithmus, der bei Eingabe von regulärer Grammatik G und Wort w nach endlicher Zeit entscheidet, ob $w \in L(G)$ gilt oder nicht.

Beweis Sei G eine reguläre Grammatik und sei M ein DFA mit $L(M) = L(G)$.
Für M ist das Wortproblem in Linearzeit in der Länge des Wortes lösbar, denn die Berechnung von $\widetilde{\delta}(z_0, w)$ braucht für einen DFA nur $|w|$ Schritte. $\square$

Leerheitsproblem für reguläre Grammatiken

Satz

Das Leerheitsproblem für reguläre Grammatiken ist entscheidbar.

Leerheitsproblem für reguläre Grammatiken

Satz

Das Leerheitsproblem für reguläre Grammatiken ist entscheidbar.

Beweis Sei G eine reguläre Grammatik und sei M ein DFA mit $L(M) = L(G)$.

Leerheitsproblem für reguläre Grammatiken

Satz

Das Leerheitsproblem für reguläre Grammatiken ist entscheidbar.

Beweis Sei G eine reguläre Grammatik und sei M ein DFA mit $L(M) = L(G)$.

Dann gilt $L(M) = \emptyset$ g.d.w. es keinen Pfad vom Startzustand zu einem Endzustand in M gibt.

Leerheitsproblem für reguläre Grammatiken

Satz

Das Leerheitsproblem für reguläre Grammatiken ist entscheidbar.

Beweis Sei G eine reguläre Grammatik und sei M ein DFA mit $L(M) = L(G)$.

Dann gilt $L(M) = \emptyset$ g.d.w. es keinen Pfad vom Startzustand zu einem Endzustand in M gibt.

Dies kann man leicht mit einer Tiefensuche (Depth-First-Search) auf dem Zustandsgraph von M prüfen. $\square$

Endlichkeitsproblem für reguläre Grammatiken

Satz

Das Endlichkeitsproblem für reguläre Grammatiken ist entscheidbar.

Endlichkeitsproblem für reguläre Grammatiken

Satz

Das Endlichkeitsproblem für reguläre Grammatiken ist entscheidbar.

Beweis Sei G eine reguläre Grammatik und sei M ein DFA mit $L(M) = L(G)$.

Endlichkeitsproblem für reguläre Grammatiken

Satz

Das Endlichkeitsproblem für reguläre Grammatiken ist entscheidbar.

Beweis Sei G eine reguläre Grammatik und sei M ein DFA mit $L(M) = L(G)$.

Es gilt $|L(M)| < \infty$ g.d.w. es keinen Pfad vom Startzustand zu einem Endzustand gibt, der eine Schleife enthält.

Endlichkeitsproblem für reguläre Grammatiken

Satz

Das Endlichkeitsproblem für reguläre Grammatiken ist entscheidbar.

Beweis Sei G eine reguläre Grammatik und sei M ein DFA mit $L(M) = L(G)$.

Es gilt $|L(M)| < \infty$ g.d.w. es keinen Pfad vom Startzustand zu einem Endzustand gibt, der eine Schleife enthält.

Prüfe dies mit einer Tiefensuche auf dem Zustandsgraph von M. $\square$

Schnittproblem für reguläre Grammatiken

Satz

Das Schnittproblem für reguläre Grammatiken ist entscheidbar.

Schnittproblem für reguläre Grammatiken

Satz

Das Schnittproblem für reguläre Grammatiken ist entscheidbar.

Beweis Seien G_1, G_2 reguläre Grammatiken und seien M_1, M_2 DFAs mit $L(M_i) = L(G_i)$ für $i \in \{1, 2\}$.

Schnittproblem für reguläre Grammatiken

Satz

Das Schnittproblem für reguläre Grammatiken ist entscheidbar.

Beweis Seien G_1, G_2 reguläre Grammatiken und seien M_1, M_2 DFAs mit $L(M_i) = L(G_i)$ für $i \in \{1, 2\}$.

Konstruiere den Produktautomaten $M_1 \times M_2$ mit $L(M_1 \times M_2) = L(M_1) \cap L(M_2)$.

Schnittproblem für reguläre Grammatiken

Satz

Das Schnittproblem für reguläre Grammatiken ist entscheidbar.

Beweis Seien G_1, G_2 reguläre Grammatiken und seien M_1, M_2 DFAs mit $L(M_i) = L(G_i)$ für $i \in \{1, 2\}$.

Konstruiere den Produktautomaten $M_1 \times M_2$ mit $L(M_1 \times M_2) = L(M_1) \cap L(M_2)$.

Prüfe das Leerheitsproblem für $L(M_1 \times M_2)$. $\square$

Äquivalenzproblem für reguläre Grammatiken

Satz

Das Äquivalenzproblem für reguläre Grammatiken ist entscheidbar.

Äquivalenzproblem für reguläre Grammatiken

Satz

Das Äquivalenzproblem für reguläre Grammatiken ist entscheidbar.

Beweis Seien G_1, G_2 reguläre Grammatiken und seien M_1, M_2 DFAs mit $L(M_i) = L(G_i)$ für $i \in \{1, 2\}$.

Äquivalenzproblem für reguläre Grammatiken

Satz

Das Äquivalenzproblem für reguläre Grammatiken ist entscheidbar.

Beweis Seien G_1, G_2 reguläre Grammatiken und seien M_1, M_2 DFAs mit $L(M_i) = L(G_i)$ für $i \in \{1, 2\}$.

Minimiere M_1 und M_2.

Äquivalenzproblem für reguläre Grammatiken

Satz

Das Äquivalenzproblem für reguläre Grammatiken ist entscheidbar.

Beweis Seien G_1, G_2 reguläre Grammatiken und seien M_1, M_2 DFAs mit $L(M_i) = L(G_i)$ für $i \in \{1, 2\}$.

Minimiere M_1 und M_2.

Prüfe die minimalen DFAs auf Gleichheit bis auf Umbenennung. $\square$