Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Zusammenfassung

INSTITUT FÜR THEORETISCHE INFORMATIK

Übersicht

- Chomsky Hierarchie
- Domino

Rückblick: Chomsky Typ 3

Äquivalente Sprachmodelle

- Chomsky-3 Grammatiken
  (= (links-)rechts-reguläre)
- Determ. Endliche Automaten
- Nichtdet. Endl. Automaten mit und ohne ε-Übergänge
- Reguläre Ausdrücke

Chomsky-3

Rückblick: Chomsky Typ 3

Äquivalente Sprachmodelle

- Chomsky-3 Grammatiken
  (= (links-)rechts-reguläre)
- Determ. Endliche
  Automaten
- Nichtdet. Endl. Automaten
  mit und ohne ε-Übergänge
- Reguläre Ausdrücke

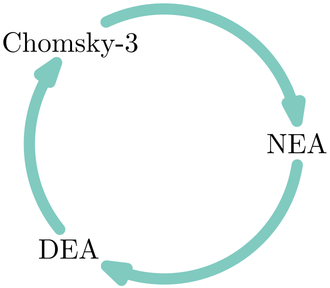
Chomsky-3

DEA

Rückblick: Chomsky Typ 3

Äquivalente Sprachmodelle

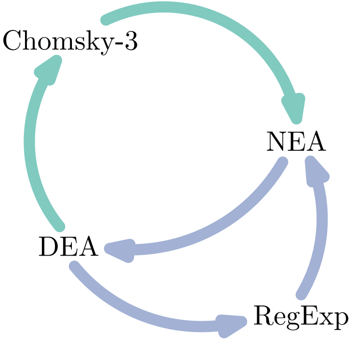
- Chomsky-3 Grammatiken (= (links-)rechts-reguläre)
- Determ. Endliche Automaten
- Nichtdet. Endl. Automaten mit und ohne ε-Übergänge
- Reguläre Ausdrücke



Rückblick: Chomsky Typ 3

Äquivalente Sprachmodelle

- Chomsky-3 Grammatiken (= (links-)rechts-reguläre)
- Determ. Endliche Automaten
- Nichtdet. Endl. Automaten mit und ohne ε-Übergänge
- Reguläre Ausdrücke



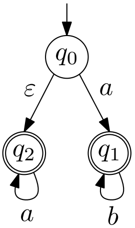
Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)

- **Grammatik**: Ableiten von links nach rechts, evtl. back tracking
- **DEA**: Ausführen und Überprüfen, ob Endzustand akzeptiert
- **NEA**: Wie DEA, evtl. mit back tracking

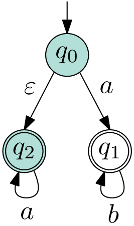
Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion
  (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma
  (für reguläre Sprachen)

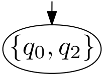


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion
  (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma
  (für reguläre Sprachen)

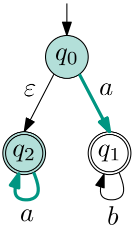


	a	b
$\{q_0, q_2\}$		

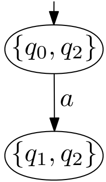


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion
  (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma
  (für reguläre Sprachen)

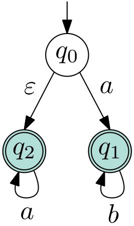


	a	b
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\emptyset$
$\{q_1, q_2\}$		

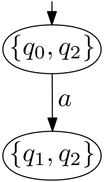


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)

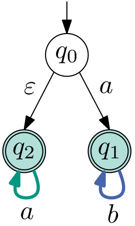


	a	b
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\emptyset$
$\{q_1, q_2\}$		

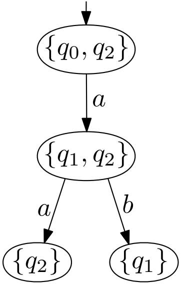


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)

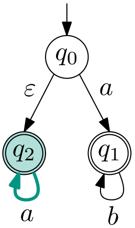


	a	b
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\emptyset$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1\}$
$\{q_2\}$		
$\{q_1\}$		

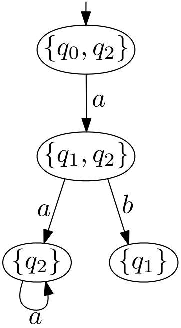


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)

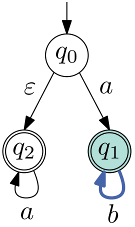


	a	b
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\emptyset$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1\}$
$\{q_2\}$	$\{q_2\}$	$\emptyset$
$\{q_1\}$		

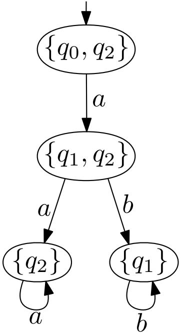


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)

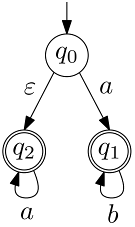


	a	b
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\emptyset$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1\}$
$\{q_2\}$	$\{q_2\}$	$\emptyset$
$\{q_1\}$	$\emptyset$	$\{q_1\}$

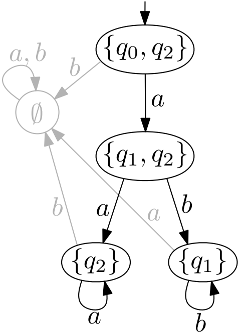


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)

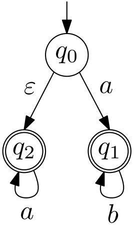


	a	b
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\emptyset$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1\}$
$\{q_2\}$	$\{q_2\}$	$\emptyset$
$\{q_1\}$	$\emptyset$	$\{q_1\}$
$\emptyset$	$\emptyset$	$\emptyset$

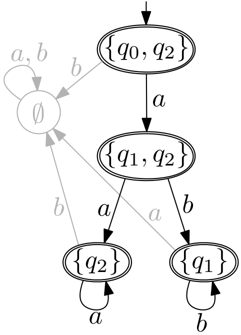


Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)

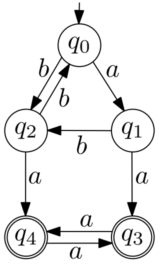


	a	b
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\emptyset$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1\}$
$\{q_2\}$	$\{q_2\}$	$\emptyset$
$\{q_1\}$	$\emptyset$	$\{q_1\}$
$\emptyset$	$\emptyset$	$\emptyset$



Rückblick: Chomsky-3 Verfahren

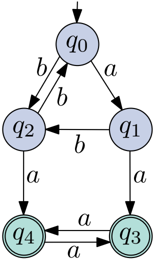
- Wortproblem
- Potenzmengenkonstruktion
  (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma
  (für reguläre Sprachen)



	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					
q_2					
q_3					
q_4					

Rückblick: Chomsky-3 Verfahren

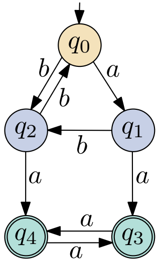
- Wortproblem
- Potenzmengenkonstruktion
  (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma
  (für reguläre Sprachen)



	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					
q_2					
q_3	ε	ε	ε		
q_4	ε	ε	ε		

Rückblick: Chomsky-3 Verfahren

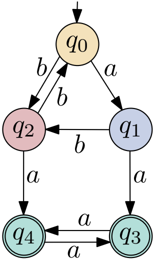
- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)



	q_0	q_1	q_2	q_3	q_4
q_0					
q_1	a				
q_2	a				
q_3	ε	ε	ε		
q_4	ε	ε	ε		

Rückblick: Chomsky-3 Verfahren

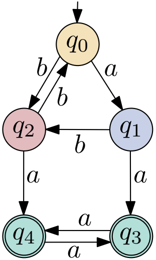
- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)



	q_0	q_1	q_2	q_3	q_4
q_0					
q_1	a				
q_2	a	ba			
q_3	ε	ε	ε		
q_4	ε	ε	ε		

Rückblick: Chomsky-3 Verfahren

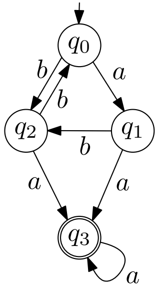
- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)



	q_0	q_1	q_2	q_3	q_4
q_0					
q_1	a				
q_2	a	ba			
q_3	ε	ε	ε		
q_4	ε	ε	ε	X	

Rückblick: Chomsky-3 Verfahren

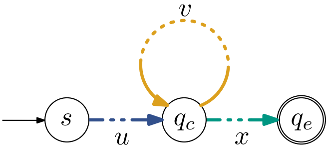
- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)



	q_0	q_1	q_2	q_3	q_4
q_0					
q_1	a				
q_2	a	ba			
q_3	ε	ε	ε		
q_4	ε	ε	ε	×	

Rückblick: Chomsky-3 Verfahren

- Wortproblem
- Potenzmengenkonstruktion (NEA $\longrightarrow$ DEA)
- Automatenminimierung
- Pumping Lemma (für reguläre Sprachen)



Rückblick: Chomsky-2

- Chomsky-2 Grammatiken (= kontextfrei)
- Nichtdeterministische Kellerautomaten (NPDAs)
- $\neq$ Deterministische Kellerautomaten (DPDAs)
- Greibach Normalform und Chomsky Normalform

Chomsky-2

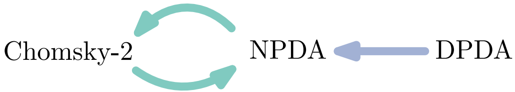
Rückblick: Chomsky-2

- Chomsky-2 Grammatiken ($=$ kontextfrei)
- Nichtdeterministische Kellerautomaten (NPDAs)
- $\neq$ Deterministische Kellerautomaten (DPDAs)
- Greibach Normalform und Chomsky Normalform

$$\begin{tikzcd}
\text{Chomsky-2} \arrow[r, bend right] & \text{NPDA} \arrow[l, bend right]
\end{tikzcd}$$

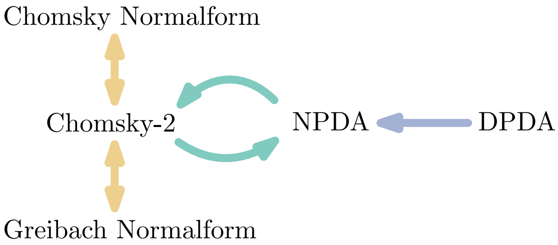
Rückblick: Chomsky-2

- Chomsky-2 Grammatiken (= kontextfrei)
- Nichtdeterministische Kellerautomaten (NPDAs)
- $\neq$ Deterministische Kellerautomaten (DPDAs)
- Greibach Normalform und Chomsky Normalform



Rückblick: Chomsky-2

- Chomsky-2 Grammatiken (= kontextfrei)
- Nichtdeterministische Kellerautomaten (NPDAs)
- $\neq$ Deterministische Kellerautomaten (DPDAs)
- Greibach Normalform und Chomsky Normalform



Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)

1. ε-Übergänge
2. Zyklische Einheitsableitungen
3. Nichtzyklische Einheitsableitungen
4. Zu lange + gemischte rechte Seiten

INSTITUT FÜR THEORETISCHE INFORMATIK

Rückblick: Chomsky-2 Verfahren

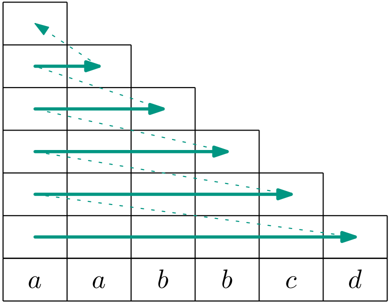
Karlsruher Institut für Technologie

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)

a	a	b	b	c	d

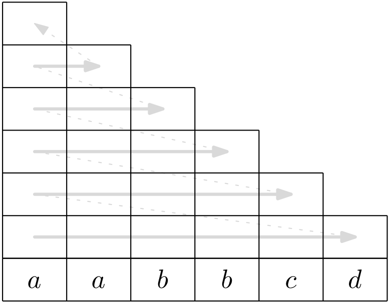
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



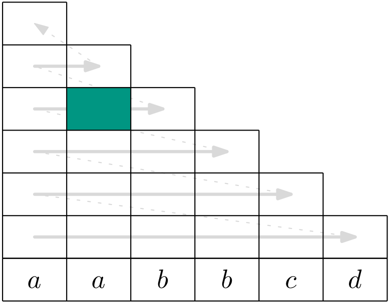
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



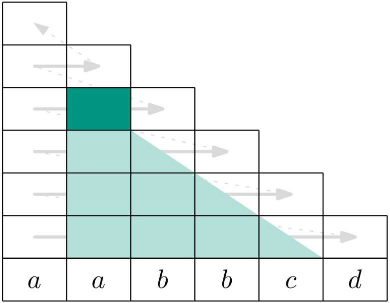
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



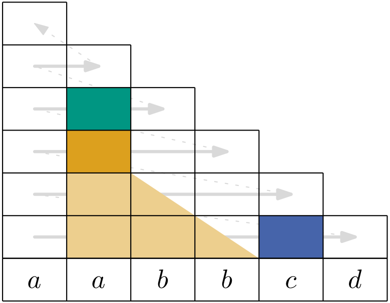
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



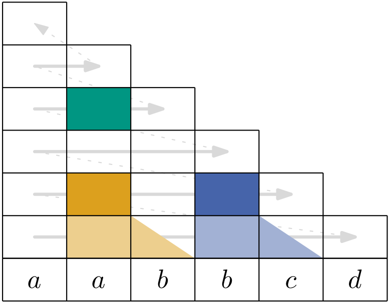
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



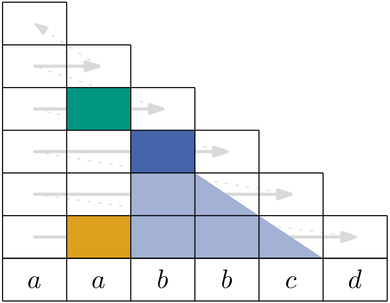
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



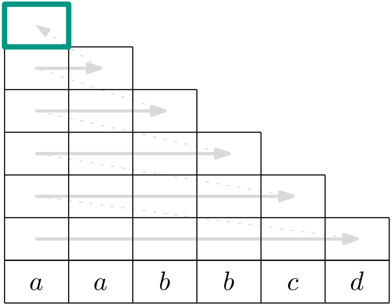
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



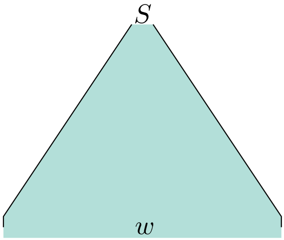
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



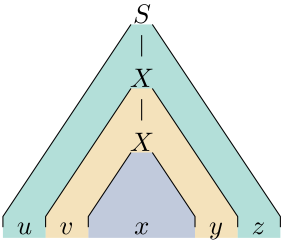
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



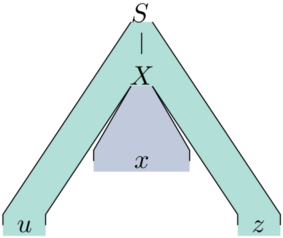
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



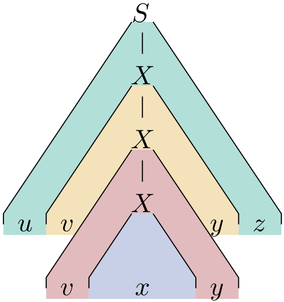
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



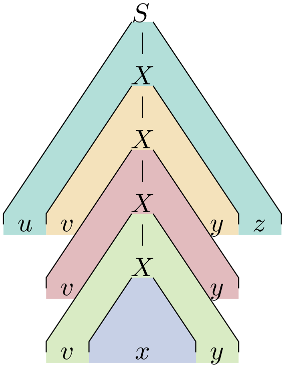
Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



Rückblick: Chomsky-2 Verfahren

- Chomsky Normalform
- Wortproblem
  Cocke Younger Kasami (CYK)
- Pumping Lemma
  (für kontextfreie Sprachen)



Rückblick: Chomsky-0/1

- Turingmaschinen
  - Chomsky-1: Linear beschränkter Platz, Nichtdeterministisch
  - Chomsky-0: Deterministisch oder Nichtdeterministisch

- Wortproblem
  - Lazy Ableitbarkeitsgraph
  - Chomsky-1: Machbar, aber langsam (im Allg. NP-schwer und schwerer)
  - Chomsky-0: Nicht machbar (im Allg. nur semientscheidbar)

Rückblick: Chomsky-0/1

- Turingmaschinen
  - Chomsky-1: Linear beschränkter Platz, Nichtdeterministisch
  - Chomsky-0: Deterministisch oder Nichtdeterministisch

- Wortproblem
  - Lazy Ableitbarkeitsgraph
  - Chomsky-1: Machbar, aber langsam (im Allg. NP-schwer und schwerer)
  - Chomsky-0: Nicht machbar (im Allg. nur semientscheidbar)

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

Karlsruher Institut für Technologie

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

Rückblick: Entscheidbarkeit

- Turingvollständigkeit, weitere Maschinenmodelle
  - Mehrband-, Mehrkopf-, mehrdimensionale TM
  - RAM, Registermaschine
  - Kellerautomaten mit 2 Stacks
- Halteproblem und Diagonalsprache, PKP
- Gödelnummer, Universelle Turingmaschine, Satz von Rice
- Entscheidbarkeit vs. Semientscheidbarkeit vs. Unentscheidbarkeit
  - Es gibt auch Funktionen, die weder entscheidbar noch semientscheidbar sind
- Fast alle interessanten Fragen über TMs sind unentscheidbar

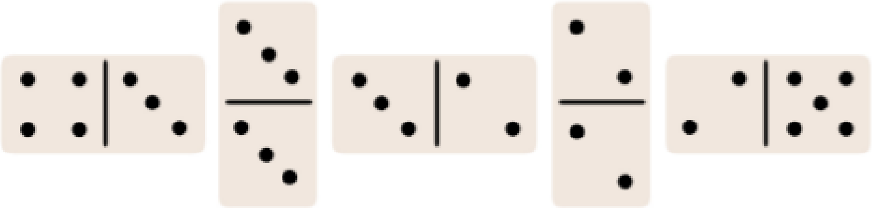
Übersicht

- Chomsky Hierarchie
- Domino

Domino

Verallgemeinertes Domino:

- Steine: $\{a, b\} \in \mathcal{P}(S)$ mit $S = \{0, 1, 2, \ldots, c\}$
- Spieler i bekommt Multiset T_i von Steinen
- Aussetzen verboten
- Perfect Information (Steine offen)
- Spieler legen eine Dominokette



Domino: Probleme

COOPERATIVE DOMINOES: $n \geq 1$ Spieler, Steine liegen offen. Können die Spieler kooperieren, sodass Spieler 1 alle seine Steine ablegen kann?

COMPETITIVE DOMINOES: $n \geq 2$ Spieler, Steine liegen offen. Kann Spieler 1 gewinnen, unabhängig davon was die anderen Spieler tun?

2-PLAYER COOPERATIVE DOMINOES

Satz:
2-COOP-DOMINOES ist NP-vollständig.

Beweis:

- 2-COOP-DOMINOES $\in$ NP: Zugfolge sodass S1 gewinnt ist Zeuge
- Reduktion von HAMILTON PATH:
  - Für Graph $G = (V, E)$ verwende Steine $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.
  - $\{\star, \star\}$ passt nirgends $\Rightarrow$ S2 kann nicht gewinnen weil S1 anfängt
  - Steine von S1 passen nie aneinander $\Rightarrow$ Einzige Möglichkeit: Alternierend Steine von S1 und S2
  - Reduktion ist in $\mathcal{O}(|V| + |E|)$ möglich (polynomiell)

Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

2-PLAYER COOPERATIVE DOMINOES

Satz:
2-COOP-DOMINOES ist NP-vollständig.

Beweis:

- 2-COOP-DOMINOES $\in$ NP: Zugfolge sodass S1 gewinnt ist Zeuge
- Reduktion von HAMILTON PATH:
  - Für Graph $G = (V, E)$ verwende Steine $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.
  - $\{\star, \star\}$ passt nirgends $\Rightarrow$ S2 kann nicht gewinnen weil S1 anfängt
  - Steine von S1 passen nie aneinander $\Rightarrow$ Einzige Möglichkeit: Alternierend Steine von S1 und S2
  - Reduktion ist in $\mathcal{O}(|V| + |E|)$ möglich (polynomiell)

Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

2-PLAYER COOPERATIVE DOMINOES

Satz:
$\mathrm{2\text{-}COOP\text{-}DOMINOES}$ ist NP-vollständig.

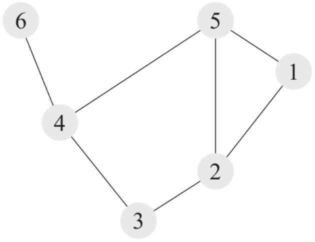
Beweis:

- $\mathrm{2\text{-}COOP\text{-}DOMINOES} \in$ NP: Zugfolge sodass S1 gewinnt ist Zeuge
- Reduktion von $\mathrm{HAMILTON\ PATH}$:
  - Für Graph $G = (V, E)$ verwende Steine $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.
  - $\{\star, \star\}$ passt nirgends $\Rightarrow$ S2 kann nicht gewinnen weil S1 anfängt
  - Steine von S1 passen nie aneinander $\Rightarrow$ Einzige Möglichkeit: Alternierend Steine von S1 und S2
  - Reduktion ist in $\mathcal{O}(|V| + |E|)$ möglich (polynomiell)

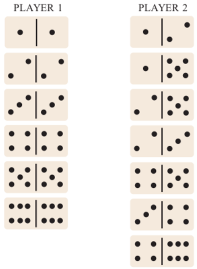
Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

2-PLAYER COOPERATIVE DOMINOES

Karlsruher Institut für Technologie



(a) An instance of Hamiltonian path



(b) Corresponding instance of 2-player cooperative dominoes

Fig. 1. Reduction from Hamiltonian path to 2-player cooperative dominoes

Fig. 2. Hamiltonian path represented as domino chain

Quelle: E. Demaine, F. Ma & E. Waingarten,
Playing Dominoes Is Hard, Except by Yourself
In: Fun with Algorithms, 2014

2-PLAYER COOPERATIVE DOMINOES

Karlsruher Institut für Technologie

- HAMILTON-PATH-Instanz $G = (V, E)$
- Dominos $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.

Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

($\Rightarrow$) S1 hat genau einen Stein für jeden Knoten von G und S2 genau einen Stein für jede Kante $\Rightarrow$ Wenn S1 alle Steine gelegt hat muss die Kette einen Hamiltonpfad beschreiben

($\Leftarrow$) Die Spieler können ihre Steine so legen, dass diese den Hamiltonpfad beschreiben und S1 gewinnt

Da HAMILTON PATH NP-schwer ist und 2-PLAYER COOPERATIVE DOMINOES in NP liegt, ist es also NP-vollständig. □

Erinnerung für die Klausur: **Antwortsätze sind wichtig!**

2-PLAYER COOPERATIVE DOMINOES

- $\mathrm{HAMILTON\text{-}PATH}$-Instanz $G = (V, E)$
- Dominos $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.

Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

($\Rightarrow$) S1 hat genau einen Stein für jeden Knoten von G und S2 genau einen Stein für jede Kante $\Rightarrow$ Wenn S1 alle Steine gelegt hat muss die Kette einen Hamiltonpfad beschreiben

($\Leftarrow$) Die Spieler können ihre Steine so legen, dass diese den Hamiltonpfad beschreiben und S1 gewinnt

Da $\mathrm{HAMILTON\ PATH}$ NP-schwer ist und $\mathrm{2\text{-}PLAYER\ COOPERATIVE\ DOMINOES}$ in NP liegt, ist es also NP-vollständig. $\square$

Erinnerung für die Klausur: **Antwortsätze sind wichtig!**

2-PLAYER COOPERATIVE DOMINOES

- HAMILTON-PATH-Instanz $G = (V, E)$
- Dominos $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.

Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

($\Rightarrow$) S1 hat genau einen Stein für jeden Knoten von G und S2 genau einen Stein für jede Kante $\Rightarrow$ Wenn S1 alle Steine gelegt hat muss die Kette einen Hamiltonpfad beschreiben

($\Leftarrow$) Die Spieler können ihre Steine so legen, dass diese den Hamiltonpfad beschreiben und S1 gewinnt

Da HAMILTON PATH NP-schwer ist und 2-PLAYER COOPERATIVE DOMINOES in NP liegt, ist es also NP-vollständig. $\square$

Erinnerung für die Klausur: **Antwortsätze sind wichtig!**

2-PLAYER COOPERATIVE DOMINOES

Karlsruher Institut für Technologie

- HAMILTON-PATH-Instanz $G = (V, E)$
- Dominos $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.

Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

($\Rightarrow$) S1 hat genau einen Stein für jeden Knoten von G und S2 genau einen Stein für jede Kante $\Rightarrow$ Wenn S1 alle Steine gelegt hat muss die Kette einen Hamiltonpfad beschreiben

($\Leftarrow$) Die Spieler können ihre Steine so legen, dass diese den Hamiltonpfad beschreiben und S1 gewinnt

Da HAMILTON PATH NP-schwer ist und 2-PLAYER COOPERATIVE DOMINOES in NP liegt, ist es also NP-vollständig. $\square$

Erinnerung für die Klausur: **Antwortsätze sind wichtig!**

2-PLAYER COOPERATIVE DOMINOES

Karlsruher Institut für Technologie

- $\textsc{Hamilton-Path}$-Instanz $G = (V, E)$
- Dominos $T_1 = \{\{i, i\} \mid i \in V\}$ und $T_2 = \{\{i, j\} \mid (i, j) \in E\} \cup \{\{\star, \star\}\}$.

Behauptung
S1 gewinnt $\Leftrightarrow$ Dominokette beschreibt Hamiltonpfad in G

($\Rightarrow$) S1 hat genau einen Stein für jeden Knoten von G und S2 genau einen Stein für jede Kante $\Rightarrow$ Wenn S1 alle Steine gelegt hat muss die Kette einen Hamiltonpfad beschreiben

($\Leftarrow$) Die Spieler können ihre Steine so legen, dass diese den Hamiltonpfad beschreiben und S1 gewinnt

Da HAMILTON PATH NP-schwer ist und 2-PLAYER COOPERATIVE DOMINOES in NP liegt, ist es also NP-vollständig. $\square$

Erinnerung für die Klausur: **Antwortsätze sind wichtig!**