GBI Tutorium Nr. 41

Foliensatz 13

Vincent Hahn – vincent.hahn@student.kit.edu | 31. Januar 2013

Outline/Gliederung

Vincent Hahn – vincent.hahn@student.kit.edu

1. Wiederholung
2. Unentscheidbare Probleme
3. Äquivalenzrelationen

Überblick

Vincent Hahn – vincent.hahn@student.kit.edu

1. Wiederholung
2. Unentscheidbare Probleme
3. Äquivalenzrelationen

Wiederholung

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung

Unentscheidbare Probleme

Äquivalenzrelationen

- Irgendwas zu Turingmaschinen
- Irgendwas zu Codierungen
- Irgendwas zu Relationen
- Reflexiv
- Transitiv
- Symmetrisch

Überblick

Vincent Hahn – vincent.hahn@student.kit.edu

1 Wiederholung

2 Unentscheidbare Probleme

3 Äquivalenzrelationen

Unentscheidbare Probleme

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung
Unentscheidbare Probleme
Äquivalenzrelationen

Es gibt Probleme, die lassen sich mit einer Turing-Maschine (oder äquivalent: einem Java-Programm) nicht lösen. (Auch nicht mit unendlich viel Zeit und Platz.)
Ein solches Problem ist nicht **entscheidbar**

Entscheidbarkeit

Für ein entscheidbares Problem gibt es eine Turingmaschine, die für jede Eingabe hält und das Eingabewort entweder akzeptiert oder nicht.

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung

Unentscheidbare Probleme

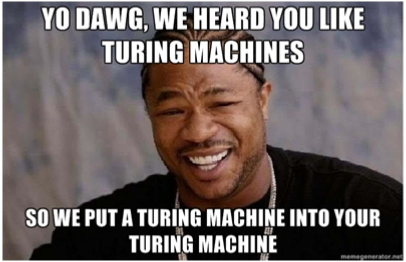
Äquivalenzrelationen

Codierung von Turingmaschinen

Bisher haben wir eine Turingmaschine formal so geschrieben $T = (Z, Z_0, X, f, g, m)$. Wir bauen uns eine Codierung, die die ganze Turingmaschine in ein Wort w_1 "packt".

Dieses Wort w_1 übergeben wir dann einer universellen Turingmaschine U, die

- übeprüft, ob w_1 eine Turingmaschine T codiert
- dann die Turingmaschine T "simuliert" und als Eingabe w_2 verwendet
- und schließlich das Ergebnis davon ausgibt



Halteproblem-Beweis 1

Vincent Hahn – vincent.hahn@student.kit.edu

Satz

Es ist nicht möglich, eine Turingmaschine U zu bauen, die für jede Turingmaschine T (codiert als w_1) und jede Eingabe w_2 entscheidet, ob T bei der Eingabe von w_2 hält.

Das lässt sich auch beweisen.

Beweis des Halteproblems

Vincent Hahn – vincent.hahn@student.kit.edu

Annahme: es gibt eine Super-Turingmaschine H. H bekommt als Eingabe:

- eine andere Turingmaschine T und
- ein Eingabewort w.

Die Super-Turingmaschine H

- simuliert die "normale" Turingmaschine T und
- benutzt als Eingabe für T das Wort w.

Die Super-Turingmaschine H gibt aus

- 1, wenn T mit w als Eingabe hält und
- 0, wenn T mit w als Eingabe nicht hält.

Beweis des Halteproblems

Vincent Hahn – vincent.hahn@student.kit.edu

Annahme: es gibt eine Super-Turingmaschine H. H bekommt als Eingabe:

- eine andere Turingmaschine T und
- ein Eingabewort w.

Die Super-Turingmaschine H

- simuliert die “normale” Turingmaschine T und
- benutzt als Eingabe für T das Wort w.

Die Super-Turingmaschine H gibt aus

- 1, wenn T mit w als Eingabe hält und
- 0, wenn T mit w als Eingabe nicht hält.

Beweis des Halteproblems

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung

Unentscheidbare Probleme

Äquivalenzrelationen

Annahme: es gibt eine Super-Turingmaschine H. H bekommt als Eingabe:

- eine andere Turingmaschine T und
- ein Eingabewort w.

Die Super-Turingmaschine H

- simuliert die "normale" Turingmaschine T und
- benutzt als Eingabe für T das Wort w.

Die Super-Turingmaschine H gibt aus

- 1, wenn T mit w als Eingabe hält und
- 0, wenn T mit w als Eingabe nicht hält.

Vincent Hahn – vincent.hahn@student.kit.edu

Halteproblem-Beweis 2

Wir bauen uns eine unendlich große Tabelle, die

- nach rechts (in den Spalten) alle möglichen Worte w enthält und
- nach unten (in den Zeilen) die codierte Turingmaschine T_w zum Wort w enthält .

Unsere Super-Turingmaschine hat die Tabelle ausgefüllt mit

- 1, wenn $T_w(w)$ hält und
- 0, wenn $T_w(w)$ nicht hält.

	w_0	w_1	w_2	...
T_{w_0}	1	0	1	
T_{w_1}	0	0	0	
T_{w_2}	0	0	1	
...				

Vincent Hahn – vincent.hahn@student.kit.edu

Halteproblem-Beweis 2

Wir bauen uns eine unendlich große Tabelle, die

- nach rechts (in den Spalten) alle möglichen Worte w enthält und
- nach unten (in den Zeilen) die codierte Turingmaschine T_w zum Wort w enthält .

Unsere Super-Turingmaschine hat die Tabelle ausgefüllt mit

- 1, wenn $T_w(w)$ hält und
- 0, wenn $T_w(w)$ nicht hält.

	w_0	w_1	w_2	...
T_{w_0}	1	0	1	
T_{w_1}	0	0	0	
T_{w_2}	0	0	1	
...				

Halteproblem-Beweis 2

Wir bauen uns eine unendlich große Tabelle, die

- nach rechts (in den Spalten) alle möglichen Worte w enthält und
- nach unten (in den Zeilen) die codierte Turingmaschine T_w zum Wort w enthält .

Unsere Super-Turingmaschine hat die Tabelle ausgefüllt mit

- 1, wenn $T_w(w)$ hält und
- 0, wenn $T_w(w)$ nicht hält.

	w_0	w_1	w_2	...
T_{w_0}	1	0	1	
T_{w_1}	0	0	0	
T_{w_2}	0	0	1	
...				

Vincent Hahn – vincent.hahn@student.kit.edu

Halteproblem-Beweis 3

Nun nehmen wir die Diagonale und schreiben sie auch in die Tabelle (hier blau). Außerdem schreiben wir darunter das “Komplementär” der Diagonale (1 wird zu 0 und umgekehrt, hier in rot).

	w_0	w_1	w_2	...
T_{w_0}	1	0	1	...
T_{w_1}	0	0	0	...
T_{w_2}	0	0	1	...
...	...	...	...	...
T_d	1	0	1	... (← die Zeile war schon irgendwo)
$T_{\bar{d}}$	0	1	0	... (← die Zeile war sicher noch nirgends)

Halteproblem-Beweis 4

Vincent Hahn – vincent.hahn@student.kit.edu

	w_0	w_1	w_2	...
T_{w_0}	1	0	1	...
T_{w_1}	0	0	0	...
T_{w_2}	0	0	1	...
...	...	...	...	...
T_d	1	0	1	... ($\leftarrow$ die Zeile war schon irgendwo)
$T_{\bar{d}}$	0	1	0	... ($\leftarrow$ die Zeile war sicher noch nirgends)

Obwohl $T_{\bar{d}}$ sicher nirgends vorkam, könnten wir sie bauen:

- Wir wissen, dass T_d hält (sagt uns die Super-Turingmaschine), also gehen wir mit $T_{\bar{d}}$ in eine Endlosschleife.
- Wir wissen, dass T_d nicht hält (sagt uns die Super-Turingmaschine), also halten wir mit $T_{\bar{d}}$.

Verrückt: Wenn es die Super-Turingmaschine gibt, dann könnten wir die Turing-Maschine $T_{\bar{d}}$ bauen, die es eigentlich nicht gibt. Das ist ein Widerspruch, also kann es die Super-Turingmaschine nicht geben.

Halteproblem-Beweis 4

Vincent Hahn – vincent.hahn@student.kit.edu

	w_0	w_1	w_2	...
T_{w_0}	1	0	1	...
T_{w_1}	0	0	0	...
T_{w_2}	0	0	1	...
...	...	...	...	...
T_d	1	0	1	... (← die Zeile war schon irgendwo)
$T_{\bar{d}}$	0	1	0	... (← die Zeile war sicher noch nirgends)

Obwohl $T_{\bar{d}}$ sicher nirgends vorkam, könnten wir sie bauen:

- Wir wissen, dass T_d hält (sagt uns die Super-Turingmaschine), also gehen wir mit $T_{\bar{d}}$ in eine Endlosschleife.
- Wir wissen, dass T_d nicht hält (sagt uns die Super-Turingmaschine), also halten wir mit $T_{\bar{d}}$.

Verrückt: Wenn es die Super-Turingmaschine gibt, dann könnten wir die Turing-Maschine $T_{\bar{d}}$ bauen, die es eigentlich nicht gibt. Das ist ein Widerspruch, also kann es die Super-Turingmaschine nicht geben.

Überblick

Vincent Hahn – vincent.hahn@student.kit.edu

1. Wiederholung
2. Unentscheidbare Probleme
3. Äquivalenzrelationen

Äquivalenzrelation

Definition: Äquivalenzrelation

Eine Relation R ist genau dann eine **Äquivalenzrelation**, wenn sie

- symmetrisch,
- reflexiv und
- transitiv

ist.

Eigenschaften von Relationen

Vincent Hahn – vincent.hahn@student.kit.edu

Welche Eigenschaften haben diese Relationen (stets auf ganze Zahlen)?

- $\leq$
- $>$
- $=$

Wie sieht das in einem Graphen aus? (Tafel)

Eigenschaften von Relationen

Vincent Hahn – vincent.hahn@student.kit.edu

Welche Eigenschaften haben diese Relationen (stets auf ganze Zahlen)?

- $\leq$ reflexiv, transitiv
- $>$
- $=$

Wie sieht das in einem Graphen aus? (Tafel)

Eigenschaften von Relationen

Vincent Hahn – vincent.hahn@student.kit.edu

Welche Eigenschaften haben diese Relationen (stets auf ganze Zahlen)?

- $\leq$ reflexiv, transitiv
- $>$ transitiv
- $=$

Wie sieht das in einem Graphen aus? (Tafel)

Eigenschaften von Relationen

Vincent Hahn – vincent.hahn@student.kit.edu

Welche Eigenschaften haben diese Relationen (stets auf ganze Zahlen)?

- $\leq$ reflexiv, transitiv
- $>$ transitiv
- $=$ reflexiv, transitiv, symmetrisch

Wie sieht das in einem Graphen aus? (Tafel)

Äquivalenzklasse

Vincent Hahn – vincent.hahn@student.kit.edu

Definition: Äquivalenzklasse

Sind zwei Elemente $(x, y) \in R$, so schreibt man auch xRy (Infixschreibweise). Alle Elemente, die miteinander in Relation stehen, befinden sich in der selben **Äquivalenzklasse**:

$$[x]_{\mathrm{R}} = \{y | yRx\}$$

Quiz

Vincent Hahn – vincent.hahn@student.kit.edu

- Stimmt es, dass auch xRy folgt: $[x]_{\mathrm{R}} = [y]_{\mathrm{R}}$?
- Existiert ein $z \in [x]_{\mathrm{R}}$ und $z \in [y]_{\mathrm{R}}$, so ist $[x]_{\mathrm{R}} = [y]_{\mathrm{R}}$.
- Wieviele Äquivalenzklassen gibt es zu $R = \quad \bmod 6$?

Quiz

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung

Unentscheidbare Probleme

Äquivalenzrelationen

- Stimmt es, dass auch *xRy* folgt: $[x]_\mathrm{R} = [y]_\mathrm{R}$? Ja.
- Existiert ein $z \in [x]_\mathrm{R}$ und $z \in [y]_\mathrm{R}$, so ist $[x]_\mathrm{R} = [y]_\mathrm{R}$.
- Wieviele Äquivalenzklassen gibt es zu $R = \quad \bmod 6$?

Quiz

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung

Unentscheidbare Probleme

Äquivalenzrelationen

- Stimmt es, dass auch xRy folgt: $[x]_{\mathrm{R}} = [y]_{\mathrm{R}}$? Ja.
- Existiert ein $z \in [x]_{\mathrm{R}}$ und $z \in [y]_{\mathrm{R}}$, so ist $[x]_{\mathrm{R}} = [y]_{\mathrm{R}}$. Ja.
- Wieviele Äquivalenzklassen gibt es zu $R = \quad \bmod 6$?

Quiz

Vincent Hahn – vincent.hahn@student.kit.edu

- Stimmt es, dass auch xRy folgt: $[x]_\mathrm{R} = [y]_\mathrm{R}$? Ja.
- Existiert ein $z \in [x]_\mathrm{R}$ und $z \in [y]_\mathrm{R}$, so ist $[x]_\mathrm{R} = [y]_\mathrm{R}$. Ja.
- Wieviele Äquivalenzklassen gibt es zu $R =$ mod 6? 6

Nerode-Äquivalenzrelation

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung

Unentscheidbare Probleme

Äquivalenzrelationen

Definition: Nerode-Äquivalenzrelation

Sei $L \subseteq A^*$ eine formale Sprache. w_1 und w_2 seien Wörter $\in A^*$. Die Wörter heißen **Nerode-Äquivalent** ($\equiv_L$), falls gilt:

$$w_1 \equiv_L w_2 \leftrightarrow (\forall w \in A^* : w_1 w \in L \leftrightarrow w_2 w \in L)$$

Beispiel zur Nerode Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

- Alphabet $A = \{a, b\}$
- Sprache $L \subset A*$, L enthält alle Wörter ohne das Teilwort *ba*:
  $L = \langle a^* b^* \rangle$

Wie sieht der zugehörige Automat aus?

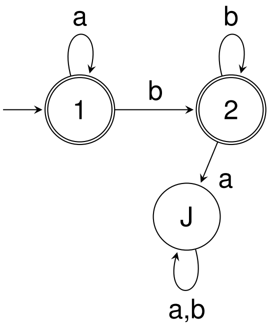
Nerode-Äquivalenzklassen:

Beispiel zur Nerode Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

- Alphabet $A = \{a, b\}$
- Sprache $L \subset A*$, L enthält alle Wörter ohne das Teilwort *ba*:
  $L = \langle a^* b^* \rangle$

Wie sieht der zugehörige Automat aus?

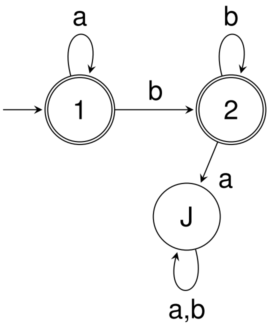


Nerode-Äquivalenzklassen:

Beispiel zur Nerode Äquivalenz

- Alphabet $A = \{a, b\}$
- Sprache $L \subset A*$, L enthält alle Wörter ohne das Teilwort *ba*:
  $L = \langle a^* b^* \rangle$

Wie sieht der zugehörige Automat aus?



Wie kann jeder Zustand erreicht werden?

Nerode-Äquivalenzklassen:

Beispiel zur Nerode Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

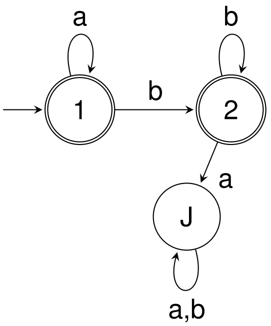
Wiederholung

Unentscheidbare Probleme

Äquivalenzrelationen

- Alphabet $A = \{a, b\}$
- Sprache $L \subset A*$, L enthält alle Wörter ohne das Teilwort *ba*:
  $L = \langle a^* b^* \rangle$

Wie sieht der zugehörige Automat aus?



Wie kann jeder Zustand erreicht werden?

- a^*
- $a^* bb^*$
- $a^* bb^* a \{a, b\}^*$

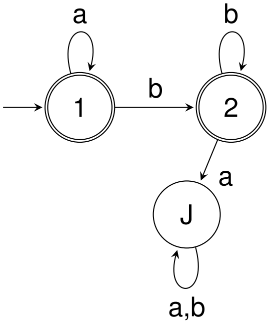
Nerode-Äquivalenzklassen:

Beispiel zur Nerode Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

- Alphabet $A = \{a, b\}$
- Sprache $L \subset A*$, L enthält alle Wörter ohne das Teilwort *ba*:
  $L = \langle a^* b^* \rangle$

Wie sieht der zugehörige Automat aus?



Wie kann jeder Zustand erreicht werden?

- a^*
- $a^* bb^*$
- $a^* bb^* a \{a, b\}^*$

Nerode-Äquivalenzklassen:
$[\epsilon]$, $[b]$, $[ba]$.

Faktormenge

Vincent Hahn – vincent.hahn@student.kit.edu

Definition: Faktormenge

Die Menge aller Äquivalenzklassen einer Menge zur Relation R bezeichnet man als **Faktormenge** und schreibt $M_{/\mathrm{R}}$.

Beispiel 2 zur Nerode-Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

Gegeben sei die Sprache $L = \{a^k b^k | k \in \mathbb{N}_0\}$.
Wie sieht hier ein endlicher Akzeptor aus?

Beispiel 2 zur Nerode-Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

Gegeben sei die Sprache $L = \{a^k b^k | k \in \mathbb{N}_0\}$.
Wie sieht hier ein endlicher Akzeptor aus? Es gibt keinen.

Beispiel 2 zur Nerode-Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

Wiederholung

Unentscheidbare Probleme

Äquivalenzrelationen

Gegeben sei die Sprache $L = \{a^k b^k | k \in \mathbb{N}_0\}$.
Wie sieht hier ein endlicher Akzeptor aus? Es gibt keinen.
Nennt einige Nerode-Äquivalenzklassen.

Beispiel 2 zur Nerode-Äquivalenz

Gegeben sei die Sprache $L = \{a^k b^k | k \in \mathbb{N}_0\}$.
Wie sieht hier ein endlicher Akzeptor aus? Es gibt keinen.
Nennt einige Nerode-Äquivalenzklassen.
Wieviele gibt es?

Beispiel 2 zur Nerode-Äquivalenz

Vincent Hahn – vincent.hahn@student.kit.edu

Gegeben sei die Sprache $L = \left\{a^k b^k | k \in \mathbb{N}_0\right\}$.
Wie sieht hier ein endlicher Akzeptor aus? Es gibt keinen.
Nennt einige Nerode-Äquivalenzklassen.
Wieviele gibt es?
Es gibt unendlich viele Nerode-Äquivalenzklassen. Die Faktormenge hat also unendlich viele Elemente.