Informatik I WS 07/08

Tutorium 24

13.12.07

Bastian Molkenthin

E-Mail: infotut@sunshine2k.de

Web: http://infotut.sunshine2k.de

Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Organisatorisches / Review

- Bis zum **21.12** müssen alle Praxisaufgaben einschließlich Übungsblatt 5 vorgeführt werden
- Keine Vorlesungen von Sa, **22.12.07** bis So, **6.1.08**
- Nicht von letztjähriger Musterlösung abschreiben! -> Punktabzug!

„Die Schleife terminiert da j streng monoton wächst.“

⇒ Nicht ausreichend da nicht jede streng monoton wachsende Funktion jeden beliebigen Grenzwert überschreitet.

⇒ Richtig: „j wächst linear“ oder „j wächst in jedem Durchlauf um eins“

Graphen

Ein Graph wird durch zwei Elemente gebildet:

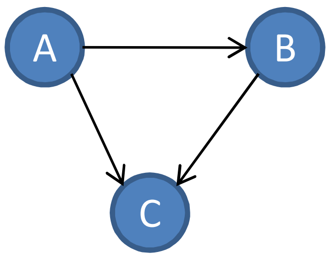
- Ecken bzw. Knoten
- Kanten

Was ist ein „gerichteter“ und ein „ungerichteter“ Graph?

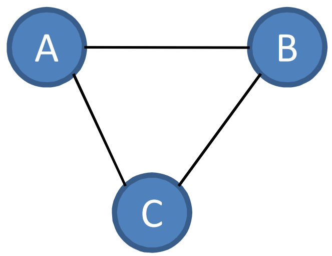
- Wenn mit den Kanten eine Richtung vorgegeben ist (z. B. von Knoten A nach Knoten B), handelt es sich um einen gerichteten Graphen.
  Kanten werden als Pfeile dargestellt.

- Bei ungerichteten Graphen kann entlang einer Kante in beide Richtungen gegangen werden

Gerichteter Graph



Ungerichteter Graph



Positionierung der Ecken/Kanten beliebig!

Graphen (2)

Was ist ein…

- *Weg ?*

  Eine oder mehr Kanten, welche zwei Knoten im Graph verbinden, bilden einen Weg.

- *Zyklus oder Kreis ?*
  - Existiert ein Weg mit gleichem Anfangs- und Endknoten, so spricht man von einem Zyklus bzw. Kreis.
  - Bei einem **einfachen** Zyklus ist jeder Knoten im Graph maximal einmal enthalten.

- *Eulerscher Weg ?*
  - Ein Weg, der alle Kanten im Graph genau einmal enthält.

- *Eulerscher Zyklus ?*
  - Ein Zyklus, der alle Kanten im Graph genau einmal enthält.

- *Hamiltonscher Kreis ?*

  Ein Zyklus, der jeden Knoten des Graphen genau einmal enthält.

Graphen (3)

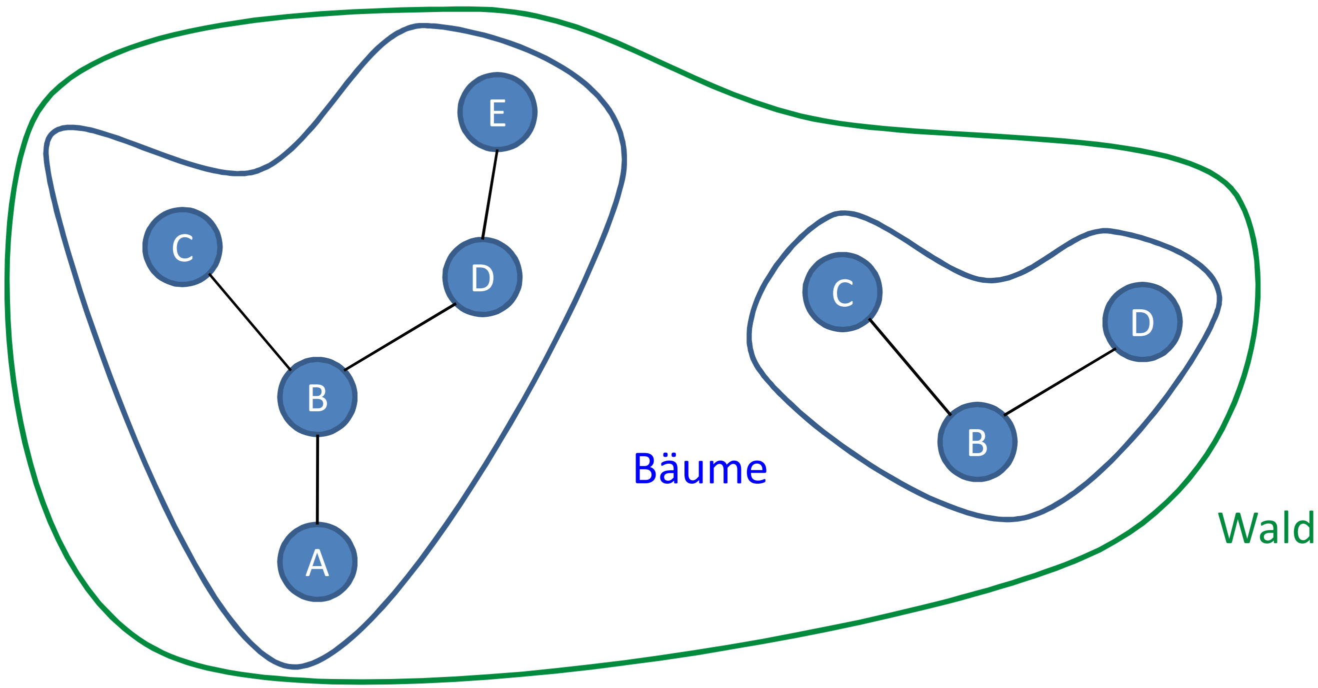
Was ist ein…

- *Baum ?*

  Ein Graph ohne Zyklen.

- *Wald ?*

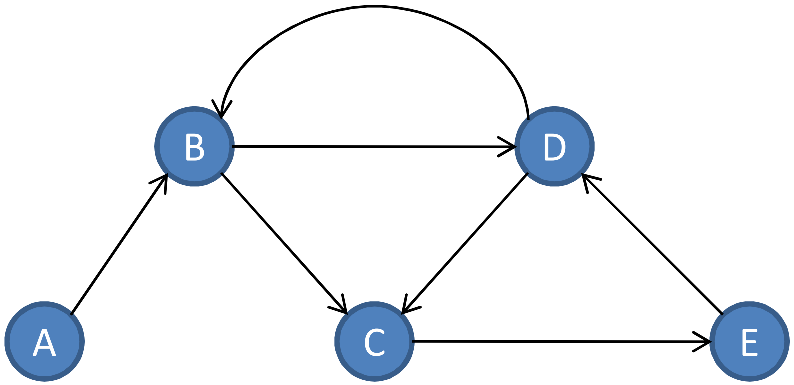
  - Mehrere Bäume.
  - Zwei Wälder lassen sich durch eine zusätzliche Kante zu einem Baum zusammenführen.



Graph (Beispiel)

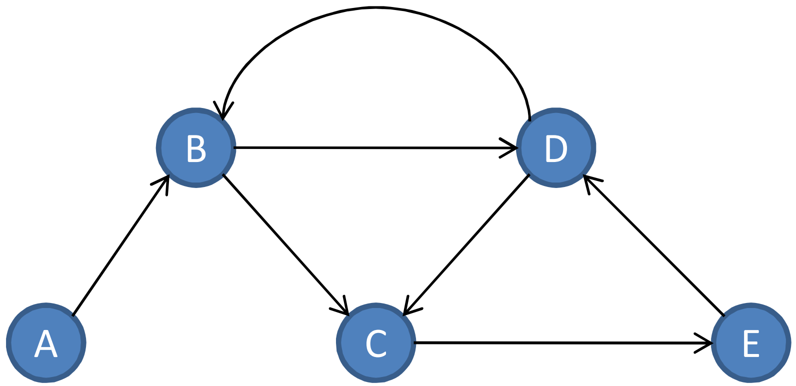
Hat der folgende Graph …

- einen (einfachen) Zyklus ? Ja
- einen Hamiltonschen Kreis ? Nein
- einen Eulerschen Weg ? Ja
- einen Eulerschen Zyklus ? Nein



Graph (Beispiel 2)

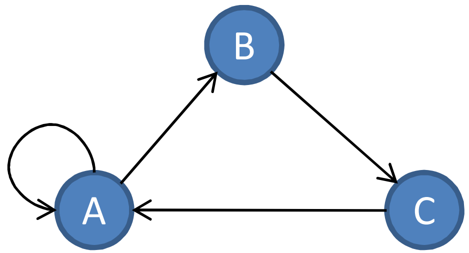
Bestimmt den Eingangs- und Ausgangsgrad der Knoten des Graphen G !



Knoten	Eingangsgrad |•e|	Ausgangsgrad |e•|
A	0	1
B	2	2
C	2	1
D	2	2
E	1	1
⇒ Grad des Graphen: grad(G) = 7		

Repräsentation von Graphen

- Graphisch:



- Adjazenzliste:

A: [A, B]
B: [C]
C: [A]

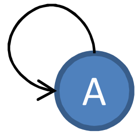
- Adjazenzmatrix:

	A	B	C
A	1	1	0
B	0	0	1
C	1	0	0

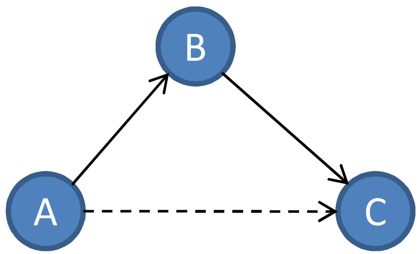
"Von hier" "nach hier"

Graphentheorie

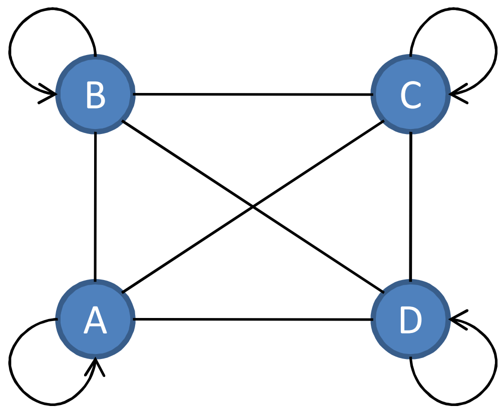
- Reflexivität:
  - Jeder Knoten steht in Relation zu sich selbst
  - d.h. als Relation ausgedrückt: A ⟶ A



- Transitivität:
  - Wenn gilt: A ⟶ B und B ⟶ C
  - Dann folgt: A ⟶ C

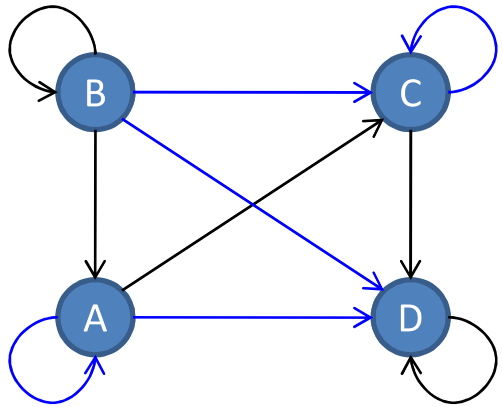


- Vollständigkeit:
  - Für jeweils zwei beliebige Knoten A, B gilt:
    A ⟶ B (Jeder Knoten ist von Jedem erreichbar)



Reflexive transitive Hülle

- Für die reflexive transitive Hülle wird der Graph um reflexive und transitive Kanten ergänzt
- Für bestehende "indirekte" Verbindungen wird eine "direkte" Verbindung hinzu-gefügt, sodass man sofort erkennen kann, welche Knoten von einem Ausgangsknoten erreichbar sind
  - Reflexive Kanten: Jeder noch nicht mit einer Schlinge versehender Knoten wird um eine solche ergänzt
  - Transitive Kanten: Wenn für beliebige Knoten $e_1, e_2, \ldots, e_n$ gilt:
  - $e_1 \rightarrow e_2 \rightarrow \ldots \rightarrow e_n$, so wird die Kante, welche die Relation $e_1 \rightarrow e_n$ repräsentiert, ergänzt



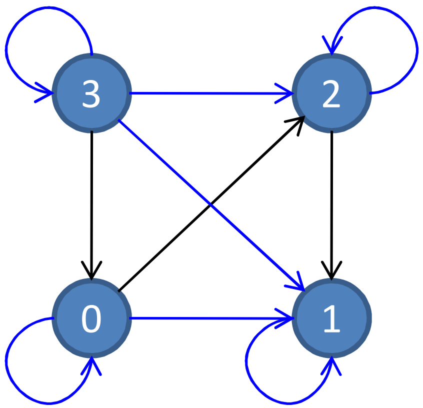
Warshall-Algorithmus

Algorithmus zur Bestimmung der reflexiven transitiven Hülle

Gegeben sei ein Graph mit n Knoten 0, 1, 2, …, n durch eine Adjazenzmatrix

Anschauliche Durchführung des Algorithmus:

- Ergänze sämtliche Knoten um eine Schlinge (Einsen auf der Diagonalen der Matrix)
- Betrachte k-te Zeile der Matrix. Alle weiteren Zeilen, welche in **Spalte** k ebenfalls eine 1 beinhalten können verändert werden:
  - Sämtliche Nullen in diesen Zeilen werden durch eine Eins ersetzt, sofern die Ausgangszeile k an dieser Stelle eine 1 enthält



$$\begin{matrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{matrix} \qquad \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix}$$

$$\begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix} \qquad \begin{matrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$

Boolsche Algebra

Signatur der booleschen Algebra $\mathcal{B} = \mathcal{B}(A,\perp,\top,☾,\vee,\wedge)$
$\perp$ ist kleinstes Element und $\top$ ist größtes Element

Gesetze der booleschen Algebra ($x, y, z \in A$):

V1 Assoziativität	$(x\wedge y)\wedge z = x\wedge(y\wedge z)$	$(x\vee y)\vee z = x\vee (y\vee z)$
V2 Kommutativität	$x\wedge y = y\wedge x$	$x\vee y = y\vee x$
V3 Idempotenz	$x\wedge x = x$	$x\vee x = x$
V4 Verschmelzung	$(x\vee y)\wedge x = x$	$(x\wedge y)\vee y = y$
V5 Distributivität	$x\vee (y\wedge z) = (x\vee y)\wedge(x\vee z)$	$x\wedge(y\vee z) = (x\wedge y)\vee(x\wedge z)$
V6 Modularität		
(falls $z \leq x$ gilt):	$x\wedge(y\vee z) = (x\wedge y)\vee z$	
V7 Neutrales Element	$x\wedge\perp = \perp$	$x\vee\perp = x$
	$x\wedge\top = x$	$x\vee\top = \top$
V8 Komplement	$x\wedge ☾x = \perp$	$x\vee ☾x = \top$
V9 Involution	$☾(☾x) = x$	
V10 DeMorgan	$☾(x\wedge y) = ☾x\vee ☾y$	$☾(x\vee y) = ☾x\wedge ☾y$

Boolsche Algebra (2)

- 0-stellige Operationen: $\perp$, $\top$
- 1-stellige Operationen : ☾
- 2-stellige Operationen: $\vee$, $\wedge$

- Negation bindet stärker als Konjunktion und Disjunktion
- In Info1 binden Konjunktion und Disjunktion gleich stark!

⇒ **Richtige Klammerung notwendig!**

Boolsche Algebra - Aufgabe

Vereinfache folgenden booleschen Ausdruck:

☾(☾a ∧ (a ∨ ☾b))

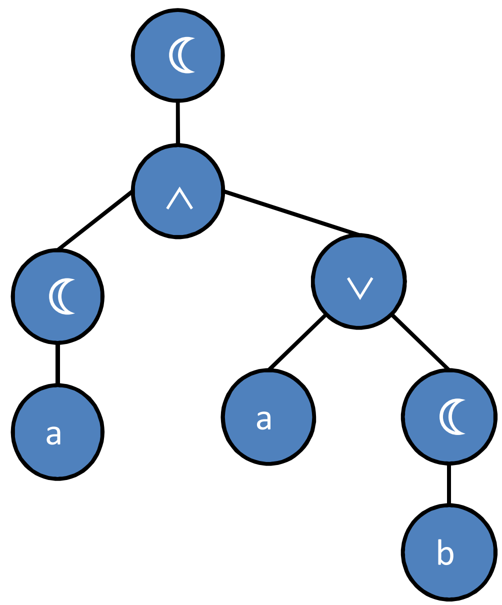
(Distributivität)	↔ ☾((☾a ∧ a) ∨ (☾a ∧ ☾b))
(Komplement)	↔ ☾(⊥ ∨ (☾a ∧ ☾b))
(neutr. Element)	↔ ☾(☾a ∧ ☾b)
(De Morgan)	↔ a ∨ b

Kantorowitsch-Bäume

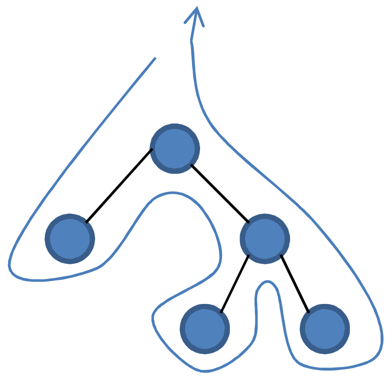
- Ein Term der boolschen Algebra lässt sich durch einen sogenannten Kantorowitsch-Baum darstellen
- Ziel: minimale Darstellung der Vorränge in Ausdrücke
- Aus diesem lässt sich leicht die Infix- Präfix- und Postfix-Schreibweise ablesen

Beispiel:
Stelle folgenden Term als Kantorowitsch-Baum dar!

☾(☾a ∧ (a ∨ ☾b))



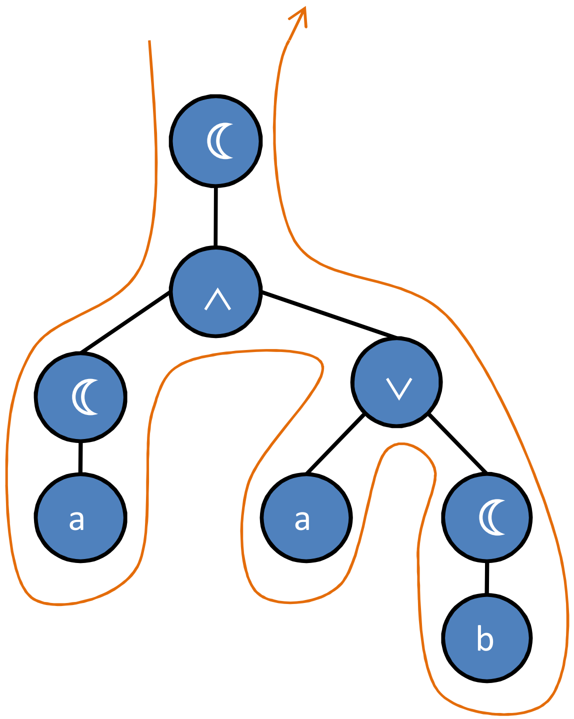
Kantorowitsch-Bäume (2)



Präfixschreibweise: Schreibe Knoten bei *1. Besuch* auf.

Postfixschreibweise: Schreibe Knoten beim *letzten Besuch* auf.

Gib den zum folgenden Kantorowitsch-Baum gehörigen Term in Präfix- und Postfix-Schreibweise an!



Präfixschreibweise: ☾∧☾a∨a☾b

Postfixschreibweise: a☾ab☾∨∧☾

Boolsche Algebra - Normalformen

Disjunktive Normalform :

- Disjunktion konjunktiv verknüpfter Teilterme
- z.B. $(\neg a \land b \land c) \lor (a \land \neg b \land c) \lor (a \land b \land \neg c)$

Konjunktive Normalform :

- Konjunktion disjunktiv verknüpfter Teilterme
- z.B. $(\neg a \lor b \lor c) \land (a \lor \neg b \lor c) \land (a \lor b \lor \neg c)$

In den Teiltermen müssen jeweils **alle** Variablen des gesamten Ausdrucks vorkommen!

Boolsche Algebra – Normalformen (2)

- Eine Möglichkeit, die DNF oder KNF eines booleschen Ausdrucks zu bestimmen ist das Ablesen der Normalform aus der Wahrheitstabelle.

- Für die DNF betrachtet man die Zeilen, an denen der Ausdruck 1 ergibt. Die Teilterme setzen sich nun so aus den links stehenden Werten zusammen, dass die Konjunktion der Variablen 1 ergibt.

- Für die KNF betrachtet man die Zeilen, an denen der Ausdruck 0 ergibt. Die Teilterme setzen sich nun so aus den links stehenden Werten zusammen, dass die Disjunktion der Variablen 0 ergibt.

a	b	$a \vee (\neg a \wedge b)$
0	0	0
0	1	1
1	0	1
1	1	1

DNF:
$(\neg a \wedge b) \vee (a \wedge \neg b) \vee (a \wedge b)$

KNF:
$a \vee b$

Boolsche Algebra – Normalformen (3)

- Weitere Möglichkeit zur Bestimmung der Normalform ist das Anwenden der booleschen Rechenregeln.

Beispiel : Überführe folgenden Ausdruck in disjunktive Normalform!

$$(a \Leftrightarrow b) \vee c$$

$\leftrightarrow (a \wedge b) \vee (\neg a \wedge \neg b) \vee c$

$\leftrightarrow (a \wedge b \wedge \top) \vee (\neg a \wedge \neg b \wedge \top) \vee (c \wedge \top)$

$\leftrightarrow (a \wedge b \wedge (c \vee \neg c)) \vee (\neg a \wedge \neg b \wedge (c \vee \neg c)) \vee (c \wedge (a \vee \neg a))$

$\leftrightarrow (a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge \neg b \wedge \neg c) \vee (c \wedge a) \vee (c \wedge \neg a)$

$\leftrightarrow (a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge \neg b \wedge \neg c) \vee (c \wedge a \wedge \top) \vee (c \wedge \neg a \wedge \top)$

$\leftrightarrow$ … *Fehlt hier nur aus Platzgründen! Alle Schritte angeben!*

$\leftrightarrow (a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge c)$

ÜBlatt : in jeder Zeile angewendete Regel angeben!

Semi-Thue-Systeme

- Ein Semi-Thue-System besteht aus zwei Mengen:
    - Zeichenvorrat Σ
    - Menge von Regeln T
- Eine Regel aus T besteht aus einer linken und einer rechten Seite, getrennt durch einen einfachen Pfeil ($\rightarrow$)
- Existiert der linke Teil im eingegebenen Wort, so kann dieser durch den rechten Teil ersetzt werden
- Die Reihenfolge der Regeln oder die Position der zu ersetzenden Zeichen im Wort spielt keine Rolle – es kann jede Regel zu jeder Zeit angewendet werden **(!)**
- Der Übergang zwischen zwei Worten wird durch einen Doppfelpfeil ($\Rightarrow$) gekennzeichnet

  $\Rightarrow^{+}$ Bedeutet, dass 1 oder mehr Regeln angewendet wurden

  $\Rightarrow^{*}$ Bedeutet, dass 0 oder mehr Regeln angewendet wurden

Semi-Thue-Systeme (2)

„Pseudcode“:

```
Solange anwendbare Regel vorhanden
{
        wähle beliebige anwendbare Regel
        wähle beliebige Anwendungsstelle
        wende Regel an
}
```

Beispiel:

Gegeben ist das Semi-Thue-System mit

$\Sigma = \{I\}$
$T = \{IIII \rightarrow I, III \rightarrow I\}$

Geben Sie alle Ableitungen des Wortes IIIIII an:

- 1. Möglichkeit: $IIIIII \Rightarrow III \Rightarrow I$ (Regel 1, 2)
- 2. Möglichkeit: $IIIIII \Rightarrow IIII \Rightarrow I$ (Regel 2, 1)
- 3. Möglichkeit: $IIIIII \Rightarrow IIII \Rightarrow II$ (Regel 2, 2)

Semi-Thue-Systeme (3)

Terminieren folgende Semi-Thue-Systeme? Begründung!

$\Sigma = \{I\}$ und $T = \{II \rightarrow I, \varepsilon \rightarrow I\}$

Terminiert nie, da Semi-Thue-Systeme mit ε-Regeln nicht terminieren

$\Sigma = \{a, b\}$ und $T = \{aaa \rightarrow a, aa \rightarrow b, a \rightarrow aa\}$

Terminiert nicht immer. Wenn z.B. immer nur die erste und letzte Regel angewendet werden, terminiert das Semi-Thue-System nicht:
$aaa \Rightarrow a \Rightarrow aa \Rightarrow aaa \Rightarrow a \Rightarrow \ldots$

$\Sigma = \{I\}$ und $T = \{III \rightarrow I, IIII \rightarrow II\}$

Terminiert immer, da die Wortlänge bei jeder Regelanwendung echt kleiner wird

Fragen ???

Viel Spaß mit dem Übungsblatt!