GOETHE UNIVERSITÄT FRANKFURT AM MAIN

www.uni-frankfurt.de

Logikbasierte Systeme der Wissensverarbeitung

Allens Zeitintervalllogik

Prof. Dr. M. Schmidt-Schauß

SoSe 2024

Stand der Folien: 13. Juni 2024

Schließen über Zeit

- Darstellung und Inferenzen für zeitliche Zusammenhänge
- Es gibt verschiedene Zeit-Logiken:
  z.B. **Modallogiken**: eher Logik-Aspekte.
  Temporallogiken.
  Diese sprechen über Ereignisse in der Zukunft / Vergangenheit und haben Existenzquantoren.
  Temporallogiken erlauben teilweise exakte Zeitdauern.

Wir betrachten beispielhaft als einfache Variante die

Allensche Intervall-Logik

Beispiel: Käse-Sahne-Kuchen backen; Aktionen

Zutaten besorgen	Teig zubereiten	Teig ruhen lassen	Belag zubereiten
Sahne schlagen	Backform einfetten	Teig in Backform	Belag in Backform
Ofen heizt	Kuchen im Ofen	Kuchen kühlt aus	Kuchen entnehmen

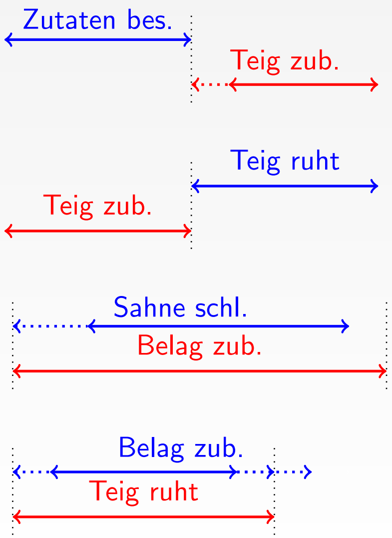
Zeitliche Zusammenhänge

Zutaten besorgen	Teig zubereiten	Teig ruhen lassen	Belag zubereiten	Sahne schlagen	Backform einfetten
Teig in Backform	Belag in Backform	Ofen heizt	Kuchen im Ofen	Kuchen kühlt aus	Kuchen entnehmen

- Aktionen entsprechen (nicht-leeren) **Zeitintervallen**
- Ausdrückbar:
  Aussagenlogische Formeln kombiniert
  mit Aussagen zur **relativen** Lage der Intervalle
  sonst nichts!: keine Zeit-Dauern
- Wie kann man dieses Wissen **repräsentieren**?
- Und wie daraus **Schlüsse ziehen?**
  Welche Inormationen kann man berechnen / erhalten?

Beispiele

Zutaten besorgen	vor	Teig zubereiten
Teig ruhen lassen	direkt nach	Teig zubereiten
Sahne schlagen	während, aber vorher endend	Belag zubereiten
Belag zubereiten	beginnt während	Teig ruhen lassen



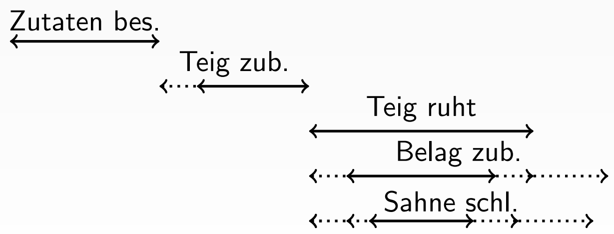
Welche Schlüsse lassen sich daraus ziehen?

- Neue Beziehungen zwischen Aktionen

  Darf der Belag vor dem Teig in die Form?
- Modell: Anordnung der Intervalle, die alle Beziehungen erfüllt

  Wie gelingt der Kuchen?
- Konsistenz: Gibt es ein Modell?

  Kann man den Kuchen überhaupt backen?



Allensche Intervalllogik

James F. Allen:

Maintaining knowledge about temporal intervals

Communications ACM, 1983

Keine Benutzung von Zeitpunkten, sondern:

- Benutzung von Zeitintervallen
- ohne Absolutwerte (weder von wann bis wann noch wie lang)
- sondern: nur die relative Lage von Zeit-Intervallen
- Keine parallelen Aktionen;
- unvollständig spezifizierte relativen Lagen sind erlaubt

Formeln und Basisrelationen

Allensche Formeln:

$$F ::= (A \; \mathtt{rel} \; B) \mid \neg F \mid F_1 \vee F_2 \mid F_1 \wedge F_2$$

wobei

- A, B sind Intervallnamen
- `rel` ist eine der Allenschen Basisrelationen

Basisrelationen: Gegeben zwei nichtleere reellwertige Intervalle:

- Wie können A und B zueinander liegen?
- Wieviele Möglichkeiten gibt es?

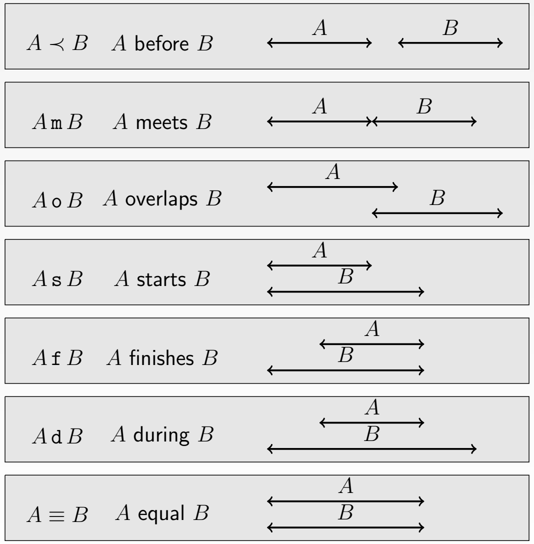
Allensche Basisrelationen (Notation wie im Original)

Für $[A_a, A_e]$ und $[B_a, B_e]$ und $A_e \leq B_e$.
Andere Fälle mit $A_e \geq B_e$: A, B tauschen.

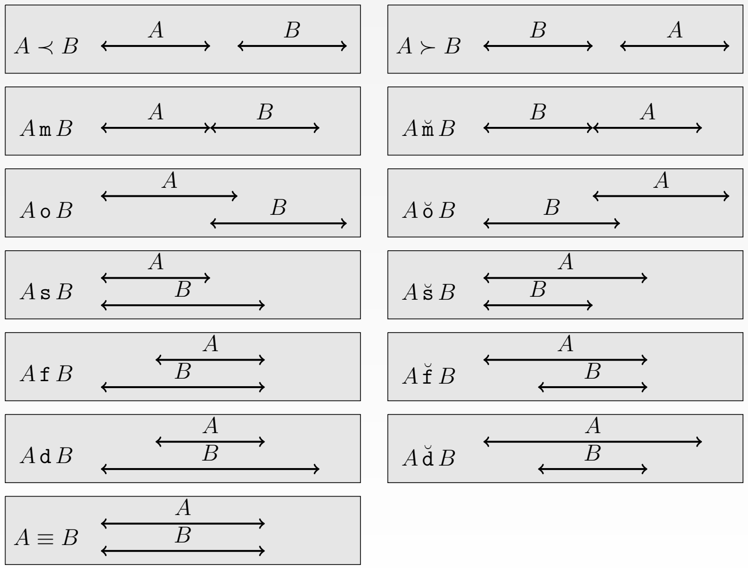
Bedingung	*Abkürzung*	*Bezeichnung*
$A_e < B_a$	$\prec$	A before B
$A_e = B_a$	$\mathtt{m}$	A meets B
$A_a < B_a < A_e < B_e$	$\mathtt{o}$	A overlaps B
$A_a = B_a < A_e < B_e$	$\mathtt{s}$	A starts B
$B_a < A_a < A_e = B_e$	$\mathtt{f}$	A finishes B
$B_a < A_a < A_e < B_e$	$\mathtt{d}$	A during B
$B_a = A_a, A_e = B_e$	$\equiv$	A equal B

- und inverse Relationen (ohne $\equiv$)
  (**invers**: rechts-links vertauscht bzw. Zeitumkehr)
- Inverse: $\breve{r}$ ist inverse Relation zu r
- Ausnahmen:
  - $\succ$ invers zu $\prec$
  - und $\equiv = \breve{\equiv}$

Allensche Basisrelationen, Teil 1



Alle Allensche Basisrelationen



Allensche Basisrelationen

Allensche Basisrelationen

Die 13 Allenschen Basis-Relationen sind:

$$\mathcal{R} := \{\equiv, \prec, \mathtt{m}, \mathtt{o}, \mathtt{s}, \mathtt{d}, \mathtt{f}, \succ, \breve{\mathtt{m}}, \breve{\mathtt{o}}, \breve{\mathtt{s}}, \breve{\mathtt{d}}, \breve{\mathtt{f}}\}.$$

Fakt

Die Allenschen Basis-Relationen sind paarweise disjunkt, d.h.

$$A\ r_1\ B \wedge A\ r_2\ B \implies r_1 = r_2.$$

Schreibweise zu Disjunktionen zu A, B

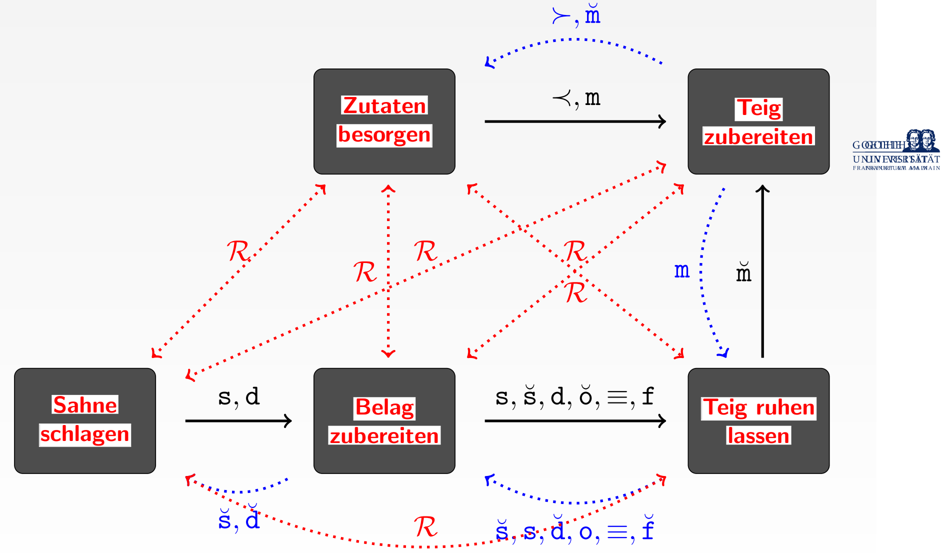
$$A\{r_1, \ldots, r_n\}B := (A\ r_1\ B) \vee (A\ r_2\ B) \ldots \vee (A\ r_n\ B)$$

$A\{r_1, \ldots, r_n\}B$ nennt man **atomares Allen-Constraint**.
Dh. $2^{13} = 8192$ verschiedene atomare Allen-Constraints zu A, B.

Beispiele

Zutaten besorgen	vor	Teig zubereiten	Zutaten $\{\prec, \mathtt{m}\}$ TeigZub
Teig ruhen lassen	direkt nach	Teig zubereiten	TeigR $\{\breve{\mathtt{m}}\}$ TeigZub
Sahne schlagen	während, aber vorher endend	Belag zubereiten	Sahne $\{\mathtt{s}, \mathtt{d}\}$ BelagZub
Belag zubereiten	beginnt während	Teig ruhen lassen	BelagZub $\{\mathtt{s}, \breve{\mathtt{s}}, \mathtt{d}, \breve{\mathtt{o}}, \equiv, \mathtt{f}\}$ TeigR

Beispiel als Constraintnetzwerk



Allensche Formeln

- Sind aussagenlogische Kombinationen der Basisrelation über Intervallnamen.
- Drücken Bedingungen aus, wie die Aktionen liegen müssen.
- Das ist i.a auch mehrdeutig ...
- Deswegen braucht man Schlussweisen und Analysen der Logik
- Ergebnisse:
  - Mögliche tatsächliche Lagen der Aktionen
  - oder Widersprüchlickeit: D.h. Anforderungen sind nicht erfüllbar.
  - ...

Allensche Formeln: Semantik

Interpretation I:
bildet Intervallnamen auf nichtleere Intervalle $[a, b]$ ab, wobei $a, b \in \mathbb{R}$ und $a < b$.

Interpretation von **atomaren Aussagen** $A\ r\ B$:
Sei $I(A) = [A_a, A_e]$ und $I(B) = [B_a, B_e]$.

- $I(A \prec B) = 1$, gdw. $A_e < B_a$
- $I(A\ \mathtt{m}\ B) = 1$, gdw. $A_e = B_a$
- $I(A\ \mathtt{o}\ B) = 1$, gdw. $A_a < B_a$, $B_a < A_e$ und $A_e < B_e$
- $I(A\ \mathtt{s}\ B) = 1$, gdw. $A_a = B_a$ und $A_e < B_e$
- $I(A\ \mathtt{f}\ B) = 1$, gdw. $A_a > B_a$ und $A_e = B_e$
- $I(A\ \mathtt{d}\ B) = 1$, gdw. $A_a > B_a$ und $A_e < B_e$
- $I(A \equiv B) = 1$, gdw. $A_a = B_a$ und $A_e = B_e$
- $I(A\ \breve{r}_0\ B) = 1$, gdw. $I(B\ r_0\ A) = 1$
- $I(A \succ B) = 1$, gdw. $I(B \prec A) = 1$

Allensche Formeln: Semantik-Varianten

Interpretation I:
Für die Intervall-Grenzen reicht als Semantik auch:
$\mathbb{Q}$ oder $\mathbb{Z}$ oder $\mathbb{N}$.

Aber:
Lässt man Intervalle der Länge 0 zu,
ergeben sich durch Sonderfälle weniger gute Transitivitätsregeln.
D.h. die Allen-Matrix (s.u.) würde sich ändern:
Z.B. gilt dann nicht mehr:
dass $A \texttt{ m } B$ und $B \texttt{ m } C$ die Relation $A \prec C$ impliziert.
Denn A, B, C könnten Länge 0 haben.

ALSO: Intervalle haben Länge > 0.

Allensche Formeln: Semantik (2)

Interpretation von **Allenschen Formeln**:

$$\begin{array}{lll} I(F \wedge G) = 1 & \text{gdw.} & I(F) = 1 \text{ und } I(G) = 1 \\ I(F \vee G) = 1 & \text{gdw.} & I(F) = 1 \text{ oder } I(G) = 1. \\ I(\neg F) = 1 & \text{gdw.} & I(F) = 0 \\ I(F \iff G) = 1 & \text{gdw.} & I(F) = I(G) \\ I(F \Rightarrow G) = 1 & \text{gdw.} & I(F) = 0 \text{ oder } I(G) = 1 \end{array}$$

D.h.: wie üblich

Modelle und Erfüllbarkeit

Interpretation I ist ein **Modell** für F gdw. $I(F) = 1$ gilt.

Eine Allensche Formel F ist:

- **widersprüchlich** (inkonsistent), wenn es kein Modell für F gibt.
- **allgemeingültig**, wenn jede Interpretation ein Modell für F ist.
- **erfüllbar**, wenn es mindestens ein Modell für F gibt.

Zwei Formeln F und G sind **äquivalent** gdw. $\forall I : I(F) = I(G)$

Semantische Folgerung: $G \models F$ gdw. $\forall I : I(G) = 1 \Rightarrow I(F) = 1$

Disjunktionen von atomaren Formeln

- Zur Erinnerung: $A \ S \ B$ mit $S \subseteq \mathcal{R}$ nennen wir atomares Allen-Constraint
- Z.B.: Statt $A \prec B \vee A \texttt{ s } B \vee A \texttt{ f } B$ schreiben wir $A \ \{\prec, \texttt{s}, \texttt{f}\} \ B$
- Beachte: Es gibt 2^{13} solche Mengen S.
- Auch erlaubt: $A \ \emptyset B$, Semantik: $I(A \ \emptyset \ B) = 0$.
- $A \ \mathcal{R} \ B$ bedeutet: alles ist möglich, $I(A \ \mathcal{R} \ B) = 1$.

Vereinfachungsregeln für Allensche Formeln

- Ein atomare Aussage der Form $A\ r\ A$ kann man immer vereinfachen zu $0, 1$:
  - $A\ r\ A \ \rightarrow\ 0$, wenn $r \neq \equiv$ und
  - $A\ \equiv\ A\ \rightarrow\ 1$.
- Negationszeichen kann man nach innen schieben.
- Eine Formel $\neg(A\ R\ B)$ kann man zu $A\ (\mathcal{R} \setminus R)\ B$ umformen.
- Unterformeln der Form $A\ R_1\ B \wedge A\ R_2\ B$ kann man durch $A\ (R_1 \cap R_2)\ B$ ersetzen.
- Unterformeln der Form $A\ R_1\ B \vee A\ R_2\ B$ kann man durch $A\ (R_1 \cup R_2)\ B$ ersetzen.
- atomare Formeln der Form $A\ \emptyset\ B$ kann man durch 0 ersetzen.
- atomare Formeln der Form $A\ \mathcal{R} B$ kann man durch 1 ersetzen.
- Alle aussagenlogischen Umformungen sind erlaubt.

Vereinfachungen (2)

Theorem

Jede Vereinfachungsregel für Allensche Formeln erhält die Äquivalenz, d.h. wenn $F \rightarrow F'$, dann sind F und F' äquivalente Formeln.

Beweis: Verwende die Semantik

Allensche Constraints

Mit den Vereinfachungen kann jede Allensche Formel umgeformt werden in ein

Disjunktives Allen-Constraint

1. (konjunktives) Allen-Constraint:
   Eine Konjunktion von atomaren Allen-Constraints:
   $A_1\ S_1\ B_1 \wedge \ldots \wedge A_n\ S_n\ B_n$
2. Disjunktives Allen-Constraint:
   Disjunktion von (konjunktiven) Allen-Constraints

Weniger geht nicht: Z.B. nicht vereinfachbar: $A \preceq B \vee C \preceq D$

Allensche Formeln: Anmerkungen zur Vereinfachung

Gelernt in der Aussagenlogik:

1. Sei F aussagenlogische Formel mit Allen-Basis-Formeln als atomare Aussagen.
2. Dann ist die Umformung in ein konjunktives Allen Constraint
   - worst case exponentiell. Aber Resultat äquivalent.
   - Polynomiell mit Tseitin-Transformation (schnelle CNF-Erzeugung);
     aber nur unter Erhaltung der Widersprüchlichkeit;
     (Äquivalenz bleibt nicht erhalten.)

Allenscher Kalkül

- Eingabe: Allen-Constraint
- Ausgabe: Weitere Beziehungen die daraus folgen, bzw. 0 (Widerspruch).
- Wir beschränken uns auf: Konjunktive Allen-Constraints
- Bei disjunktiven Allen-Constraints:
  bearbeite die konjunktiven Allen-Constraints unabhängig und füge dann zusammen.

Allenscher Kalkül (2)

Wesentliche Regel: „Transitivitätsregel“

- Aus $A \prec B \land B \prec C$ kann man $A \prec C$ folgern.
- Aus $A \prec B \land C \prec B$ kann man nichts Neues über die Beziehung zwischen A und C folgern (alles ist möglich)

Allenscher Kalkül (3)

Wie folgert man genau?

- Basisrelationen r_1, r_2: $A\ r_1\ B \land B\ r_2\ C$.
  Man braucht die Komposition $(r_1 \circ r_2)$, als kleinste Menge mit: $A\ r_1\ B \land B\ r_2\ C \models A(r_1 \circ r_2)C$.
  Beachte:$(r_1 \circ r_2)$ ist nicht unbedingt eine Basisrelation
- $R_1, R_2 \subseteq \mathcal{R}$: $A\ R_1\ B \land B\ R_2\ C$.
  Komposition der Mengen: Sei $R_1 \circ R_2$ gerade die (kleinste) Menge mit: $AR_1B \land BR_2C \models A(R_1 \circ R_2)C$.

Kompositionsmatrix

	$\prec$	$\succ$	d	$\breve{d}$	o	$\breve{o}$	m	$\breve{m}$	s	$\breve{s}$	f	$\breve{f}$
$\prec$	$\prec$	$\mathcal{R}$	$\prec\ o\ m\ d\ s$	$\prec$	$\prec$	$\prec\ o\ m\ d\ s$	$\prec$	$\prec\ o\ m\ d\ s$	$\prec$	$\prec$	$\prec\ o\ m\ d\ s$	$\prec$
$\succ$	$\mathcal{R}$	$\succ$	$\succ\ \breve{o}\ \breve{m}\ d\ f$	$\succ$	$\succ\ \breve{o}\ \breve{m}\ d\ f$	$\succ$	$\succ\ \breve{o}\ \breve{m}\ d\ f$	$\succ$	$\succ\ \breve{o}\ \breve{m}\ d\ f$	$\succ$	$\succ$	$\succ$
d	$\prec$	$\succ$	d	$\mathcal{R}$	$\prec\ o\ m\ d\ s$	$\succ\ \breve{o}\ \breve{m}\ d\ f$	$\prec$	$\succ$	d	$\succ\ \breve{o}\ \breve{m}\ d\ f$	d	$\prec\ o\ m\ d\ s$
$\breve{d}$	$\prec\ o\ m\ \breve{d}\ \breve{f}$	$\succ\ \breve{o}\ \breve{m}\ \breve{d}\ \breve{s}$	$\mathcal{R} \setminus \prec\ \succ\ m\ \breve{m}$	$\breve{d}$	$o\ \breve{d}\ \breve{f}$	$\breve{o}\ \breve{d}\ \breve{s}$	$o\ \breve{d}\ \breve{f}$	$\breve{o}\ \breve{d}\ \breve{s}$	$o\ \breve{d}\ \breve{f}$	$\breve{d}$	$\breve{o}\ \breve{d}\ \breve{s}$	$\breve{d}$
o	$\prec$	$\succ\ \breve{o}\ \breve{m}\ \breve{d}\ \breve{s}$	$o\ d\ s$	$\prec\ o\ m\ \breve{d}\ \breve{f}$	$\prec\ o\ m$	$\mathcal{R} \setminus \prec\ \succ\ m\ \breve{m}$	$\prec$	$\breve{o}\ \breve{d}\ \breve{s}$	o	$\breve{d}\ \breve{f}\ o$	$d\ s\ o$	$\prec\ o\ m$
$\breve{o}$	$\prec\ o\ m\ \breve{d}\ \breve{f}$	$\succ$	$\breve{o}\ d\ f$	$\succ\ \breve{o}\ \breve{m}\ \breve{d}\ \breve{s}$	$\mathcal{R} \setminus \prec\ \succ\ m\ \breve{m}$	$\succ\ \breve{o}\ \breve{m}$	$o\ \breve{d}\ \breve{f}$	$\succ$	$\breve{o}\ d\ f$	$\succ\ \breve{o}\ \breve{m}$	$\breve{o}$	$\breve{o}\ \breve{d}\ \breve{s}$
m	$\prec$	$\succ\ \breve{o}\ \breve{m}\ \breve{d}\ \breve{s}$	$o\ d\ s$	$\prec$	$\prec$	$o\ d\ s$	$\prec$	$\equiv\ f\ \breve{f}$	m	m	$d\ s\ o$	$\prec$
$\breve{m}$	$\prec\ o\ m\ \breve{d}\ \breve{f}$	$\succ$	$\breve{o}\ d\ f$	$\succ$	$\breve{o}\ d\ f$	$\succ$	$\equiv\ s\ \breve{s}$	$\succ$	$d\ f\ \breve{o}$	$\succ$	$\breve{m}$	$\breve{m}$
s	$\prec$	$\succ$	d	$\prec\ o\ m\ \breve{d}\ \breve{f}$	$\prec\ o\ m$	$\breve{o}\ d\ f$	$\prec$	$\breve{m}$	s	$\equiv\ s\ \breve{s}$	d	$\prec\ o\ m$
$\breve{s}$	$\prec\ o\ m\ \breve{d}\ \breve{f}$	$\succ$	$\breve{o}\ d\ f$	$\breve{d}$	$o\ \breve{d}\ \breve{f}$	$\breve{o}$	$o\ \breve{d}\ \breve{f}$	$\breve{m}$	$\equiv\ s\ \breve{s}$	$\breve{s}$	$\breve{o}$	$\breve{d}$
f	$\prec$	$\succ$	d	$\succ\ \breve{o}\ \breve{m}\ \breve{d}\ \breve{s}$	$o\ d\ s$	$\succ\ \breve{o}\ \breve{m}$	m	$\succ$	d	$\succ\ \breve{o}\ \breve{m}$	f	$\equiv\ f\ \breve{f}$
$\breve{f}$	$\prec$	$\succ\ \breve{o}\ \breve{m}\ \breve{d}\ \breve{s}$	$o\ d\ s$	$\breve{d}$	o	$\breve{o}\ \breve{d}\ \breve{s}$	m	$\breve{o}\ \breve{d}\ \breve{s}$	o	$\breve{d}$	$\equiv\ f\ \breve{f}$	$\breve{f}$

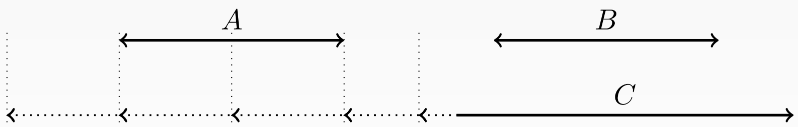
12×12-Matrix reicht, da:
$(r \circ \equiv) \quad = \quad r \quad = \quad (\equiv \circ r)$

Kompositionsmatrix

Die Einträge kann man per Hand ausrechnen.

Oder einmalige automatische Berechnung.
Beispiel: $\prec\ \circ\ \mathtt{d}$

Betrachte alle möglichen Lagen für $A \prec B \wedge B\ \mathtt{d}\ C$



Möglichkeiten: $A\ \{\prec, \mathtt{o}, \mathtt{m}, \mathtt{s}, \mathtt{d}\}\ C$.

Komposition der Mengen

Beispiel: Aus $A\ \{\mathtt{m},\mathtt{d}\}\ B \wedge B\ \{\mathtt{f},\mathtt{d}\}\ C$ kann man schließen

$$\begin{array}{rl} & A\ (\mathtt{m}\circ\mathtt{f} \cup \mathtt{m}\circ\mathtt{d} \cup \mathtt{d}\circ\mathtt{f} \cup \mathtt{d}\circ\mathtt{d})\ C \\ = & A\ \{\mathtt{d},\mathtt{s},\mathtt{o}\} \cup \{\mathtt{d},\mathtt{s},\mathtt{o}\} \cup \{\mathtt{d}\} \cup \{\mathtt{d}\}\ C \\ = & A\ \{\mathtt{d},\mathtt{s},\mathtt{o}\}\ C \end{array}$$

Allgemein gilt:

Lemma

Seien $r_1,\ldots,r_k,r'_1,\ldots,r'_k$ Allensche Basisrelationen. Dann gilt

$$\{r_1,\ldots,r_k\}\circ\{r'_1,\ldots,r'_k\} = \bigcup\{r_i\circ r'_j \mid i=1,\ldots,k, j=1,\ldots,k'\}$$

Inverse für Mengen

Inversion für Mengen von Basisrelationen

Sei $S = \{r_1,\ldots,r_k\} \subseteq \mathcal{R}$ und $\breve{S} = \{\breve{r}_1,\ldots,\breve{r}_k\}$.

Beachte. Es gilt: $\breve{\breve{r}} = r$

Damit gilt:

Lemma

Für $S \subseteq \mathcal{R}$ gilt: $A\ S\ B$ und $B\ \breve{S}\ A$ sind äquivalente Allensche Formeln.

Lemma

$\breve{}(r_1\circ r_2) = \breve{r}_2\circ\breve{r}_1$

$\overbrace{(r_1\circ r_2)}^{\breve{}} = \breve{r}_2\circ\breve{r}_1$

Allenscher Abschluss für Konjunktive Allen-Constraints

Eingabe: Konjunktives Allen-Constraint

Ausgabe: Allenscher Abschluss

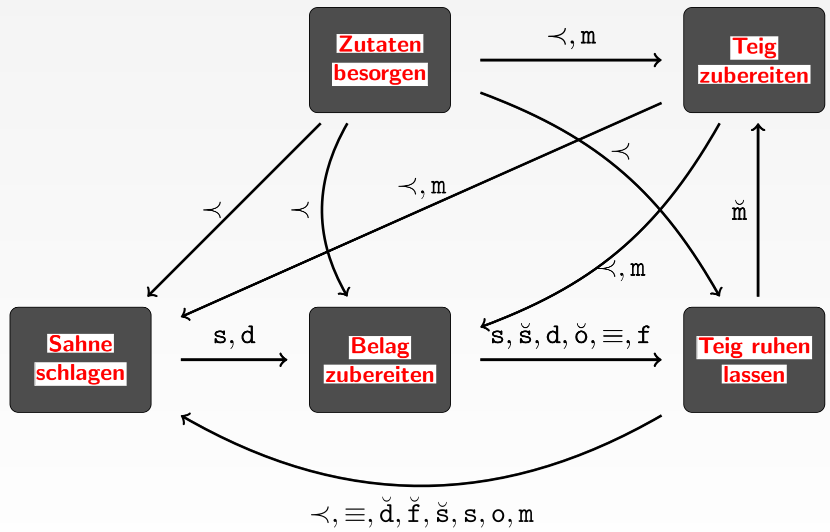
Verfahren: Berechne Fixpunkt bezüglich der Regeln (auf Subformeln):

- Vereinfachungen: ($\to$ bedeutet „ersetze“)
  - $A\ R_1\ B \wedge A\ R_2\ B \to A\ (R_1 \cap R_2)\ B$
  - $A\ \emptyset\ B \to \mathtt{False}$
  - $A\ \mathcal{R}\ B \to 1$
  - $A\ R\ A \to 0$, wenn $\equiv \notin R$.
  - $A\ R\ A \to 1$, wenn $\equiv \in R$.
- Folgerungen: ($\rightsquigarrow$ bedeutet „füge hinzu“)
  - $A\ R\ B \rightsquigarrow B\ \breve{R}\ A$, wobei $\breve{R} := \{\breve{r}_1,\ldots,\breve{r}_n\}$ für $R = \{r_1,\ldots,r_n\}$
  - $A\ R_1\ B \wedge B\ R_2\ C \rightsquigarrow A\ (R_1\circ R_2)\ C$.
- und übliche aussagenlogische Umformungen

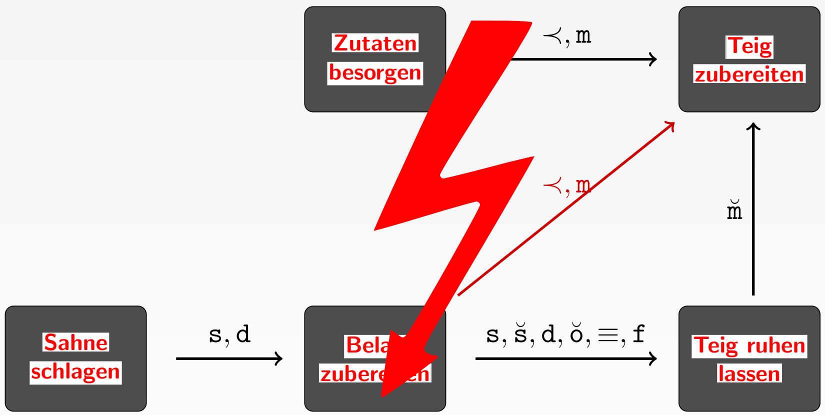
Allenscher Abschluss für alle Allen-Constraints

- Für konjunktive Allensche Constraints: Wende die Regeln des Allenschen Kalküls solange an, bis sich keine neuen Beziehungen mehr herleiten lassen (Fixpunkt)
- Disjunktive Constraints (Disjunktion von konjunktiven Allensche Constraints):
  Wende Fixpunktiteration auf jede Komponente an, und vereinfache anschließend
  - Komponente $= 1$: Disjunktiver Constraint ist äquivalent zu 1
  - Komponente $= 0$: Kann gestrichen werden
  - Alle Komponenten $= 0$: Disjunktiver Constraint widersprüchlich (Inkonsistenz)

Beispiel



Beispiel (2)



Schlägt fehl, da der Schnitt an der roten Kante eine leere Kantenbeschriftung erzeugt.

Korrektheit, Vollständigkeit

Wir sagen, der Allen-Kalkül ist

- **korrekt**, wenn bei $F \to F'$ stets gilt: F und F' sind äquivalente Formeln
- **herleitungs-vollständig**, wenn er für jedes konjunktive Constraint alle semantisch folgerbaren Einzel-Relationen herleiten kann.
- **widerspruchs-vollständig**, wenn er für jedes unerfüllbare konjunktive Constraint herausfinden kann, dass es widersprüchlich ist (Herleitung der 0)

Fragestellungen

- Wie aufwändig ist die Berechnung des Abschlusses der Allenschen Relationen?
- Ist der Allen-Kalkül korrekt?
- Ist die Berechnung herleitungs- bzw- widerspruchs-vollständig?
- Was ist die Komplexität der Logik und der Herleitungsbeziehung, evtl. für eingeschränkte Eingabeformeln?
- Wie kann man den Allenschen Kalkül für aussagenlogische Kombinationen von Intervallformeln verwenden?
- Was kann man über andere Zeitkalküle sagen?

Implementierung der Allen-Vervollständigung

- Wesentliche Regel: Transitivitätsregel
  $A\ R_1\ B \wedge B\ R_2\ C \to A\ R_1 \circ R_2\ C$.
- Konjunktive Allen-Constraints haben die Form(schon zusammengefasst)

$$\bigwedge_{i,j \in \{1,\dots,n\}} A_i\ R_{i,j}\ A_j$$

  Nicht vorhandene Relationen werden auf $\mathcal{R}$ gesetzt.
- Abschluss kann mit einer $n \times n$-Tabelle gemacht werden
- Sobald $\emptyset$ irgendwo auftaucht, kann man abbrechen
- Ähnlich zum Warshall-Algorithmus
- Bei disjunktiven Allen-Constraints: bearbeite die Allen-Constraints separat und fasse dann zusammen.

Beispiel für das Eingabearray

$$\text{Zutaten } \{\prec, \mathtt{m}\} \text{ TeigZub}$$
$$\wedge$$
$$\text{TeigRuht } \{\breve{\mathtt{m}}\} \text{ TeigZub}$$
$$\wedge$$
$$\text{Sahne } \{\mathtt{s}, \mathtt{d}\} \text{ BelagZub}$$
$$\wedge$$
$$\text{BelagZub } \{\equiv, \mathtt{s}, \breve{\mathtt{s}}, \mathtt{d}, \mathtt{f}, \breve{\mathtt{o}}\} \text{ TeigRuht}$$

$R_{i,j}$	(1) Zutaten	(2) TeigZub	(3) TeigRuht	(4) Sahne	(5) Belagzub
(1) Zutaten	$\{\equiv\}$	$\{\prec, \mathtt{m}\}$	$\mathcal{R}$	$\mathcal{R}$	$\mathcal{R}$
(2) TeigZub	$\{\succ, \breve{\mathtt{m}}\}$	$\{\equiv\}$	$\{\mathtt{m}\}$	$\mathcal{R}$	$\mathcal{R}$
(3) TeigRuht	$\mathcal{R}$	$\{\breve{\mathtt{m}}\}$	$\{\equiv\}$	$\mathcal{R}$	$\{\equiv, \breve{\mathtt{d}}, \breve{\mathtt{f}}, \mathtt{s}, \breve{\mathtt{s}}, \mathtt{o}\}$
(4) Sahne	$\mathcal{R}$	$\mathcal{R}$	$\mathcal{R}$	$\{\equiv\}$	$\{\mathtt{d}, \mathtt{s}\}$
(5) Belagzub	$\mathcal{R}$	$\mathcal{R}$	$\{\equiv, \mathtt{s}, \breve{\mathtt{s}}, \mathtt{d}, \mathtt{f}, \breve{\mathtt{o}}\}$	$\{\breve{\mathtt{d}}, \breve{\mathtt{s}}\}$	$\{\equiv\}$

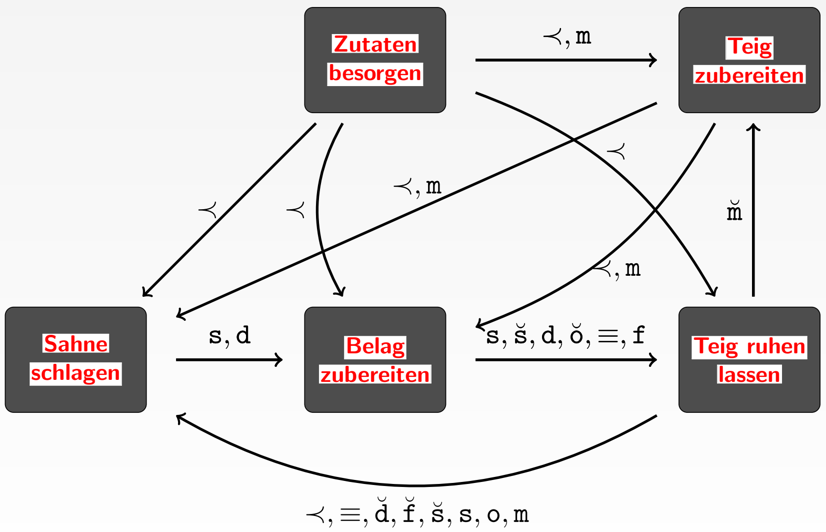
Beispiel für die Vervollständigung

$$\text{Zutaten } \{\prec, \mathtt{m}\} \text{ TeigZub}$$
$$\wedge$$
$$\text{TeigRuht } \{\breve{\mathtt{m}}\} \text{ TeigZub}$$
$$\wedge$$
$$\text{Sahne } \{\mathtt{s}, \mathtt{d}\} \text{ BelagZub}$$
$$\wedge$$
$$\text{Belagzub } \{\equiv, \mathtt{s}, \breve{\mathtt{s}}, \mathtt{d}, \mathtt{f}, \breve{\mathtt{o}}\} \text{ TeigRuht}$$

Vervollständigung:

$R_{i,j}$	Zutaten	Teigzub	TeigRuht	Sahne	Belagzub
(1) Zutaten	$\{\equiv\}$	$\{\prec, \mathtt{m}\}$	$\prec$	$\prec$	$\prec$
(2) TeigZub	$\{\succ, \breve{\mathtt{m}}\}$	$\{\equiv\}$	$\{\mathtt{m}\}$	$\{\succ, m\}$	$\{\succ, m\}$
(3) TeigRuht	$\succ$	$\{\breve{\mathtt{m}}\}$	$\{\equiv\}$	$\{\succ, \equiv, \breve{\mathtt{d}}, \breve{\mathtt{f}}, \breve{\mathtt{s}}, \mathtt{m}, \mathtt{o}, \mathtt{s}\}$	$\{\equiv, \breve{\mathtt{d}}, \breve{\mathtt{f}}, \breve{\mathtt{s}}, \mathtt{o}, \mathtt{s}\}$
(4) Sahne	$\succ$	$\{\succ, \breve{\mathtt{m}}\}$	$\{\equiv, \succ, \breve{\mathtt{m}}, \breve{\mathtt{o}}, \breve{\mathtt{s}}, \mathtt{d}, \mathtt{f}, \mathtt{s}\}$	$\{\equiv\}$	$\{\mathtt{d}, \mathtt{s}\}$
(5) BelagZub	$\succ$	$\{\succ, \breve{\mathtt{m}}\}$	$\{\equiv, \breve{\mathtt{o}}, \breve{\mathtt{s}}, \mathtt{d}, \mathtt{f}, \mathtt{s}\}$	$\{\breve{\mathtt{d}}, \breve{\mathtt{s}}\}$	$\{\equiv\}$

Die Vervollständigung als Graphik:



Algorithmus 1

Algorithmus **Allenscher Abschluss, Variante 1**

Eingabe: $(n \times n)$-Array R, mit Einträgen $R_{i,j} \subseteq \mathcal{R}$

Algorithmus:

```
repeat
  change := False;
  for i := 1 to n do
    for j := 1 to n do
      for k := 1 to n do
        R' := R_{i,j} ∩ (R_{i,k} ∘ R_{k,j});
        if R_{i,j} ≠ R' then
          R_{i,j} := R';
          change := True;
        endif
      endfor
    endfor
  endfor
until change=False
```


Erläuterung

$R' := R_{i,j} \cap (R_{i,k} \circ R_{k,j})$

entspricht gerade

$$\begin{array}{ll} & A_i\ R_{i,k}\ A_k \wedge A_k\ R_{k,j}\ A_j\ \wedge\ A_i\ R_{i,j}\ A_j \\ \to & A_i\ R_{i,k}\ A_k \wedge A_k\ R_{k,j}\ A_j \wedge A_i\ R_{i,k} \circ R_{k,j}\ A_j \wedge A_i\ R_{i,j}\ A_j \\ \to & A_i\ R_{i,k}\ A_k \wedge A_k\ R_{k,j}\ A_j \wedge A_i\ (R_{i,k} \circ R_{k,j}) \cap R_{i,j}\ A_j \end{array}$$

Eigenschaften Algorithmus 1

- Ähnlich zu Warshall-Algorithmus, aber iteriert (notwendig!)
- solange bis Fixpunkt erreicht ist
- Korrekt: Offensichtlich

Laufzeit: Im worst-case $O(n^5)$

Begründung:

- 3 geschachtelte `for`-Schleifen: $O(n^3)$
- `repeat`-Schleife: Im schlechtesten Fall wird nur ein $R_{i,j}$ um eins verkleinert
- pro $R_{i,j}$ maximal 13 Verkleinerungen
- Es gibt n^2 Mengen $R_{i,j}$
- Daher: `repeat`-Schleife wird maximal $O(n^2)$ mal durchlaufen
- **ergibt:** $O(n^5)$

Algorithmus 2

Algorithmus **Allenscher Abschluss, Variante 2**

Eingabe: $(n \times n)$-Array R, mit Einträgen $R_{i,j} \subseteq \mathcal{R}$

Algorithmus:

queue := $\{(i,k,j) \mid 1 \leq i \leq n, 1 \leq k \leq n, 1 \leq j \leq n\}$;

while queue $\neq \emptyset$ **do**

  Wähle und entferne Tripel (i,k,j) aus queue;

  $R' := R_{i,j} \cap (R_{i,k} \circ R_{k,j})$;

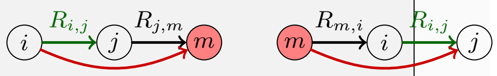
  if $R_{i,j} \neq R'$ **then**

    $R_{i,j} := R$;

    queue := queue ++ $\{(i,j,m) \mid 1 \leq m \leq n\}$ ++ $\{(m,i,j) \mid 1 \leq m \leq n\}$

  endif

endwhile



Eigenschaften Algorithmus 2

Korrektheit: Bei Änderung von $R_{i,j}$ werden alle Nachbarn, die evtl. neu berechnet werden müssen, in eine queue eingefügt

Laufzeit:

- Am Anfang: queue enthält n^3 Tripel
- `while`-Schleife entfernt pro Durchlauf ein Element aus queue
- Einfügen in queue in der Summe:
  - $R_{i,j}$ kann höchstens 13 mal verändert werden.
  - D.h. höchstens $n^2 * 13$ mal wird eingefügt
  - Einmal einfügen: $2 * n$ Tripel werden hinzugefügt

  Insgesamt: Es werden höchstens $13 * 2 * n * n^2$ Tripel zu queue hinzugefügt
- Ergibt $O(n^3)$ Durchläufe der while-Schleife

  (von denen maximal $O(n^2)$ Durchläufe $O(n)$ Laufzeit verbrauchen und die restlichen $O(n)$ in konstanter Laufzeit laufen)

Algorithmus 2 hat worst-case-Laufzeit $O(n^3)$

Allenscher Kalkül: Korrektheit

Korrektheit

Der Allensche Kalkül ist korrekt, d.h. wenn $F \to F'$, dann sind F und F' äquivalente Formeln

Beweis (Skizze): Verwende die Semantik

- Aussagenlogische Umformungen: klar
- $A\ R_1\ B \wedge A\ R_2\ B$ ist äquivalent zu $A\ (R_1 \cap R_2)\ B$:

  Sei $R_1 = \{r_1, \ldots, r_k\}$, $R_2 = \{r'_1, \ldots, r'_{k'}\}$.

$$
\begin{array}{ll}
 & A\ R_1\ B \wedge A\ R_2\ B \\
= & (\bigvee A r_i B)\ \wedge\ (\bigvee A\ r'_{i'}\ B) \\
\sim & \bigvee\{(A\ r_i\ B) \wedge (A\ r'_{i'}\ B) \mid 1 \leq i \leq k, 1 \leq i' \leq k'\} \text{ (ausmultiplizieren)} \\
\sim & \bigvee\{(A\ r_i\ B) \wedge (A\ r'_{i'}\ B) \mid 1 \leq i \leq k, 1 \leq i' \leq k', r_i = r'_{i'}\} \text{ (Basisrelationen disjunkt)} \\
= & A\ (R_1 \cap R_2)\ B
\end{array}
$$

- $A\ \emptyset\ B \sim 0$ und $A\ \mathcal{R}\ B \sim 1$ (klar)

Allenscher Kalkül: Korrektheit (2)

Beweis (Fortsetzung)

- $A\ R\ A$ ist äquivalent zu 0, wenn $\equiv \notin R$ und
  $A\ R\ A$ ist äquivalent zu 1, wenn $\equiv \in R$:
  Jede Interpretation bildet $I(A)$ eindeutig auf ein Intervall ab.

-Transitivitätsregel:
  Basisrelationen: Man muss die Korrektheit der Matrix prüfen.
  Für mehrelementige Mengen:

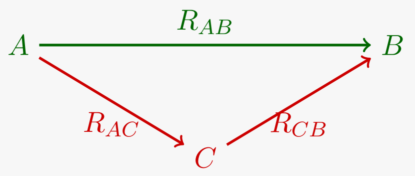
$$
\begin{array}{ll}
 & A\ \{r_1, \ldots, r_k\}\ B \wedge B\ \{r'_1, \ldots, r'_{k'}\}\ C \\
= & (A\ r_1\ B \vee \ldots \vee A\ r_k\ B) \wedge (B\ r'_1\ C \vee \ldots \vee B\ r'_{k'}\ C) \text{ (ausmultiplizieren)} \\
\sim & \bigvee\{(A\ r_i\ B \wedge B\ r'_{i'}\ C) \mid 1 \leq i \leq k, 1 \leq i' \leq k'\} \text{ (Basis)} \\
\sim & \bigvee\{(A\ r_i\ B \wedge B\ r'_{i'}\ C \wedge A\ r_i \circ r'_{i'}\ C) \mid 1 \leq i \leq k, 1 \leq i' \leq k'\} \\
\sim & A\ \{r_1, \ldots, r_k\}\ B \wedge B\ \{r'_1, \ldots, r'_{k'}\}\ C \wedge \bigvee\{(A\ r_i \circ r'_{i'}\ C) \mid 1 \leq i \leq k, 1 \leq i' \leq k'\} \\
= & A\ \{r_1, \ldots, r_k\}\ B \wedge B\ \{r'_1, \ldots, r'_{k'}\}\ C \wedge A\ \{r_1, \ldots, r_k\} \circ \{r'_1, \ldots, r'_{k'}\}\ C
\end{array}
$$

Partielle Vollständigkeit

Der Allensche Kalkül ist vollständig in eingeschränktem Sinn:

Satz (Pfadkonsistenz)

Der Allensche Abschluss ist 3-konsistent:



D.h.: Jede Belegung I der Intervalle A und B mit $I(A\ R_{AB}\ B) = \mathtt{True}$ kann auf das Intervall C erweitert werden, so dass $I(A\ R_{AC}\ C) = \mathtt{True} = I(C\ R_{CB}\ B)$.

Es gilt **nicht** (globale Konsistenz):

Jede Belegung von k Knoten kann auf $k+1$ Knoten unter Erhaltung der Erfüllbarkeit erweitert werden

Unvollständigkeit des Allen-Kalküls

Leider gilt:

Theorem

Der Allensche Kalkül ist nicht herleitungs-vollständig.

Beweis: Gegenbeispiel: Für den Allenschen Constraint:

$$D\ \{\mathtt{o}\}\ B \land D\ \{\mathtt{s},\mathtt{m}\}\ C \land D\ \{\mathtt{s},\mathtt{m}\}\ A \land A\ \{\mathtt{d},\breve{\mathtt{d}}\}\ B \land C\ \{\mathtt{d},\breve{\mathtt{d}}\}\ B$$

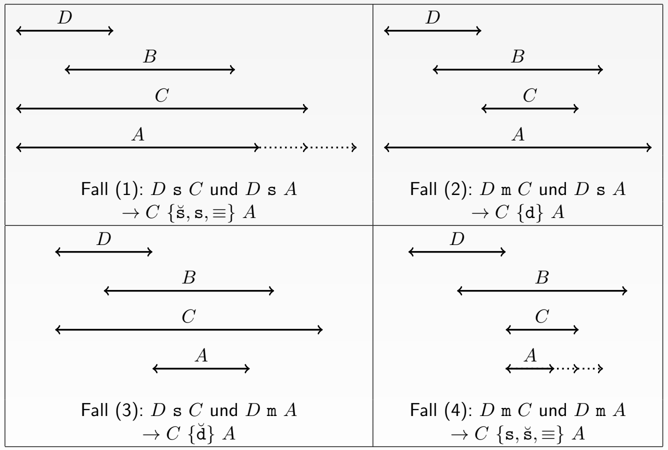
ist der Allensche Abschluss:

$$D\ \{\mathtt{o}\}\ B \land D\ \{\mathtt{s},\mathtt{m}\}\ C \land D\ \{\mathtt{s},\mathtt{m}\}\ A \land A\ \{\mathtt{d},\breve{\mathtt{d}}\}\ B \land C\ \{\mathtt{d},\breve{\mathtt{d}}\}\ B$$
$$\land C\ \{\mathtt{s},\breve{\mathtt{s}},\equiv,\mathtt{o},\breve{\mathtt{o}},\mathtt{d},\breve{\mathtt{d}},\mathtt{f},\breve{\mathtt{f}}\}\ A$$

Aber $C\ \{\mathtt{f},\breve{\mathtt{f}},\mathtt{o},\breve{\mathtt{o}}\}\ A$ ist nicht möglich (nächste Folie)
D.h. das Allen-Verfahren erkennt diese Unmöglichkeit nicht

Beweis (Fortsetzung)

- Die Lage von B zu D ist eindeutig.
- Möglichkeiten D zu A und D zu C: 4 Fälle $\{s,m\} \times \{s,m\}$



$C\ \{\mathtt{f},\breve{\mathtt{f}},\mathtt{o},\breve{\mathtt{o}}\}\ A$ nicht möglich!

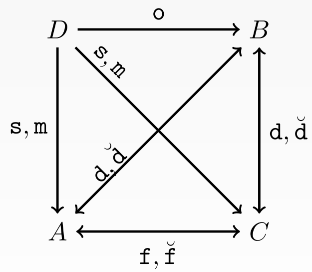
Unvollständigkeit des Allen-Kalküls (2)

Ebenso gilt:

Theorem

Der Allensche Kalkül ist nicht widerlegungsvollständig.

Beweis: Gegenbeispiel: Leichte Abwandlung des Beispiels davor
Füge $A\ \{f,\breve{f}\}\ C$ hinzu, d.h. man erhält das Constraintnetzwerk:



Allenscher Abschluss: Verändert das Netzwerk nicht, aber es ist widersprüchlich!

Konsequenzen der Unvollständigkeit

Frage: Ist Allen-Constraint F widersprüchlich?

- Abschluss $= 0$, dann JA
- Abschluss $= 1$, dann NEIN
- Abschluss weder 0 noch 1: man weiß nichts

Frage: Ist Allen-Constraint F erfüllbar?

- Abschluss $= 0$, dann NEIN
- Abschluss $= 1$, dann JA (Tautologie)
- Abschluss weder 0 noch 1: man weiß nichts

Eindeutige Allen-Constraints

Definition

Ein Allensches Constraint nennt man eindeutig, wenn für alle Paare A, B von Intervallkonstanten gilt: Das Constraint enthält genau eine Beziehung $A\ r\ B$, wobei r eine der dreizehn Basisrelationen ist.

Es gilt:

Satz

Der Allensche Abschluss eines eindeutigen Allenschen Constraints F ist entweder 0, oder wiederum F.

Beweis: Jede Transitivitätsregelanwendung leitet $\emptyset$ her, oder lässt Eintrag unverändert.

Eindeutige Allen-Constraints (2)

Satz (Valdés-Pérez, 1987)

Ein eindeutiges Allensches Constraint ist erfüllbar, gdw. der Allensche Kalkül bei Vervollständigung das Constraint nicht verändert, d.h. wenn es ein Fixpunkt ist.

Beweisidee: Zeige, wenn Allen-Kalkül keinen Widerspruch entdeckt, dann ist Constraint erfüllbar.
Es gibt dann eine totale Ordnung der Intervallenden

Korollar

Auf eindeutigen Allen-Constraints ist der Allen-Kalkül korrekt und vollständig

Eindeutige Allen-Constraints: Menge

Zu jedem Allenschen Constraint C kann man die **Menge aller zugehörigen eindeutigen Allenschen Constraints** D definieren, wobei gelten muss:
Wenn $A\ r\ B$ in D vorkommt und $A\ R\ B$ in C, dann gilt $r \in R$.

Lemma

Ein Allen-Constraint ist erfüllbar, gdw. es ein zugehöriges eindeutiges Constraint gibt, das erfüllbar ist.

Beweis: Klar

Vollständiges Verfahren

Algorithmus **Erfüllbarkeitstest für konjunktive Allensche Constraints**

Eingabe: $(n \times n)$-Array R, mit Einträgen $R_{i,j} \subseteq \mathcal{R}$
Ausgabe: `True` (Widerspruch) oder `False` (erfüllbar)

function AllenSAT(R):
  R' := AllenAbschluss(R);
  if $\exists R'_{i,j}$ mit $R'_{i,j} = \emptyset$ **then return** `True` **endif**;// Widerspruch
  if $\forall R'_{i,j}$ gilt: $|R'_{i,j}| = 1$ **then return** `False` // eindeutig und erfüllbar
  else
    wähle $R'_{i,j}$ mit $R'_{i,j} = \{r_1, r_2, \ldots\}$;
    $R^l := R'$; $R^l_{i,j} := \{r_1\}$;// kopiere R' und setze (i,j) auf r_1
    $R^r := R'$; $R^r_{i,j} := R'_{i,j} \setminus \{r_1\}$;// kopiere R' und setze (i,j) auf $r_2, \ldots$
    return (ASAT(R^l) $\wedge$ ASAT(R^r));
  endif

Der Algorithmus ist korrekt und vollständig. Die Laufzeit ist im worst-case **exponentiell**. Mittlere Verzweigungsrate: 6,5

Komplexität des Problems

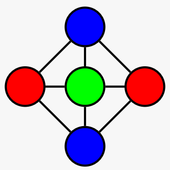
Satz
Das Erfüllbarkeitsproblem für konjunktive Allenschen Constraints ist $\mathcal{NP}$**-vollständig**.

Beweis:
Problem ist in $\mathcal{NP}$:
- Rate lineare Reihenfolge der Intervallanfänge und -enden
- D.h. Ordnung auf allen X_a, X_e für alle Intervalle X
- Verifiziere ob Reihenfolge das Constraint erfüllt
- Verifikation geht in Polynomialzeit

Beweis (Fortsetzung)

$\mathcal{NP}$-Härte:
Reduktion von 3-Färbbarkeit auf Erfüllbarkeit von Allen-Constraints

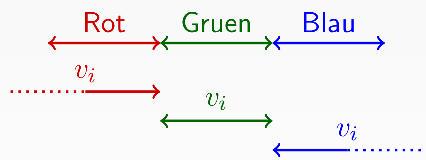
3-Färbbarkeit:
Kann man die Knoten eines ungerichteten Graphen mit drei Farben färben, so dass benachbarte Knoten stets verschiedene Farben haben?



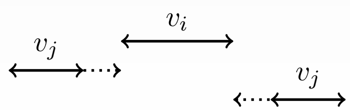
Beweis (Fortsetzung)

Für $G = (V, E)$ erzeuge:
- (Rot $\mathtt{m}$ Gruen) $\wedge$ (Gruen $\mathtt{m}$ Blau)

- Für die Knoten: $\forall v_i \in V : v_i \ \{\mathtt{m}, \equiv, \breve{\mathtt{m}}\}$ Gruen



- Für die Kanten: $\forall (v_i, v_j) \in E : \ v_i \ \{\mathtt{m}, \breve{\mathtt{m}}, \prec, \succ\} \ v_j$



Beweis (Fortsetzung)

Daher gilt: Der Graph ist dreifärbbar, gdw. die Allenschen Relationen erfüllbar sind. Die Zuordnung ist:

- v_i hat Farbe grün gdw. $v_i \equiv$ Gruen
- v_i hat Farbe rot gdw. v_i $\mathtt{m}$ Gruen
- v_i hat Farbe blau gdw. v_i $\breve{\mathtt{m}}$ Gruen

Übersetzung ist in Polynomialzeit durchführbar, daher Erfüllbarkeit $\mathcal{NP}$-Hart

Folgerungen

- Jeder vollständige Algorithmus braucht Exponentialzeit. (unter Annahme $\mathcal{NP} = EXPTIME$)
- Die polynomielle Allen-Vervollständigung ist **im allgemeinen unvollständig**

Varianten

Es gibt polynomielle, vollständige Verfahren für

Allensche Constraints mit **eingeschränkter Syntax**

Eine haben wir bereits gesehen:

- Eindeutige Allen-Constraints

Varianten (2)

Neue Variante:

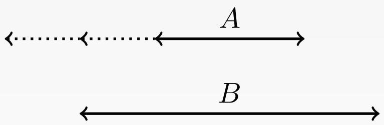
- Erlaube nur Allensche Relationen, so dass:
- Übersetzung in Bedingungen über die Endpunkte nur Konjunktionen von der Form $x < y$ oder $x = y$
- Dann gilt: Man braucht keine Fallunterscheidung

Passender Satz von Relationen:

- Alle Basisrelationen,
- $\{\mathtt{d}, \mathtt{o}, \mathtt{s}\}$, und $\{\breve{\mathtt{o}}, \mathtt{f}, \mathtt{d}\}$ und deren Konverse. d.h. $\{\breve{\mathtt{d}}, \breve{\mathtt{o}}, \breve{\mathtt{s}}\}$, und $\{\mathtt{o}, \breve{\mathtt{f}}, \breve{\mathtt{d}}\}$.

Varianten (3)

Z.B. $A\{\mathtt{d}, \mathtt{o}, \mathtt{s}\}B$ als Ungleichung über den Endpunkten:



Wenn $A = [A_a, A_e], B = [B_a, B_e]$, dann entspricht obige Relation gerade

$$A_a < A_e, B_a < B_e, A_e < B_e, B_a < A_e$$

Varianten (4)

Auf solchen Constraints kann man Erfüllbarkeit in Polynomialzeit testen

- Transitiver Abschluss der Endpunktbeziehungen
- anschließend lineare Reihenfolge mit topologischem Sortieren

Es gilt aber sogar

Satz (Nebel, Bürckert, 1995)

Auf den so eingeschränkten Allen-Constraints ist der Allensche Kalkül korrekt und vollständig.

Hintergrund

Diese spezielle Klasse lässt sich als Grund-Hornklauseln darstellen, d.h. Klauseln mit maximal einem positiven Literal.

Für Grund-Hornklauselmengen ist Erfüllbarkeit in polynomieller Zeit testbar.

Man hat Fakten in der Form $a < b$ und $c = d$, wobei a, b, c, d unbekannte Konstanten sind. Es gibt auch Hornklauseln, die von der Symmetrie und Transitivität stammen:

$$\begin{array}{lcl}
x < y \land y < z & \Rightarrow & x < z \\
x = y \land y = z & \Rightarrow & x = z \\
x = y & \Rightarrow & y = x \\
x < y \land y = z & \Rightarrow & x < z \\
x = y \land y < z & \Rightarrow & x < z
\end{array}$$

Hintergrund (2)

Man kann weitere Allensche Constraints zulassen, und behält die Vollständigkeit des Allen-Kalküls:

- Alle Constraints deren Übersetzung in Constraints über Endpunkten Hornklauseln ausschließlich mit Literalen $a \leq b$, $a = b$ und $\neg(a = b)$ erzeugt.
- Von den $2^{13} = 8192$ möglichen Beziehungen erfüllen 868 diese Eigenschaft

Man kann diese auch für die Fallunterscheidung des exponentiellen Verfahrens verwenden.

Vorteil: Kleinere mittlere Verzweigungsrate
(Statt 6,5 nur 2,533 (Nebel 1997))

Kuchen backen

Zutaten besorgen	Teig zubereiten	Teig ruhen lassen	Belag zubereiten	Sahne schlagen	Backform einfetten
Teig in Backform	Belag in Backform	Ofen heizt	Kuchen im Ofen	Kuchen kühlt aus	Kuchen entnehmen

```
Zutaten  {<,m}   TeigZub,  Zutaten  {<,m}         TeigRuht,    Zutaten  {<,m}     BelZub,
Zutaten  {<,m}   Sahne,    TeigRuht {m}           TeigForm,    TeigForm {<,m}     BelForm,
BelForm  {<,m}   Backen,   Backen   {f}           Heizen,      Sahne    {s,d}     BelZub,
Heizen   {S,o,>} TeigRuht, Heizen   {<,m}         Kuehlen,     Kuehlen  {<,m}     Entnehmen,
Zutaten  {<,m}   Einfett,  Zutaten  {<,m}         Heizen,      TeigZub  {<,m}     TeigForm,
BelZub   {<,m}   BelForm,  TeigZub  {<,>,m,M}     BelZub,      Einfett  {<,>,m,M} BelZub,
Einfett  {>,M}   TeigZub,  Einfett  {<,m,M,>}     TeigForm,    Einfett  {<,>,m,M} BelForm,
Einfett  {<,m}   TeigForm, TeigZub  {m}           TeigRuht,    Einfett  {>,M}     Zutaten,
Backen   {m}     Kuehlen,  BelZub   {s,S,d,O,=,f} TeigRuht,    Einfett  {>}       Sahne
```

Allen-Programm:

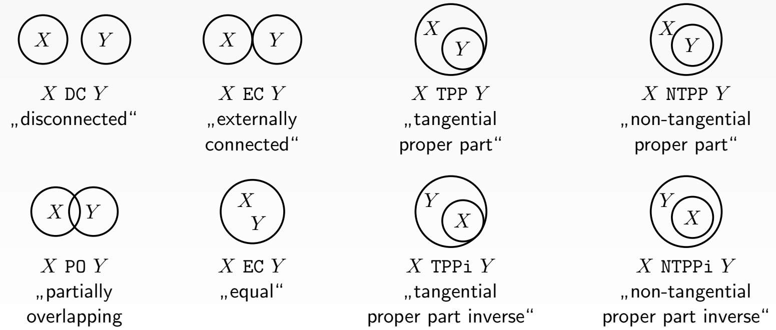
```
Max. #Modelle in der Eingabe       : 177.247.393.995.618.482.069.389.150.242.626.279.322.671.526.463.930.368
Max. #Modelle nach Allen-Abschluss: 5.898.240

Anzahl Modelle                     : 1.536
Allenscher Abschluss genau?        : True
```


Varianten

Qualitatives räumliches Schließen

- Eindimensional: Genau die Allensche Intervalllogik
- Zweidimensional: Region-Connection-Calculus (RCC8), (Randell, Cui & Cohn, 1992)



Varianten

- Exakte Zeitpunkte vermeiden: Nur vorher, überlappend