Formulierung in einer Algebra von Relationen

Definiere Komposition $r_1 \circ r_2$ von *Basis*relationen r_1, r_2 :

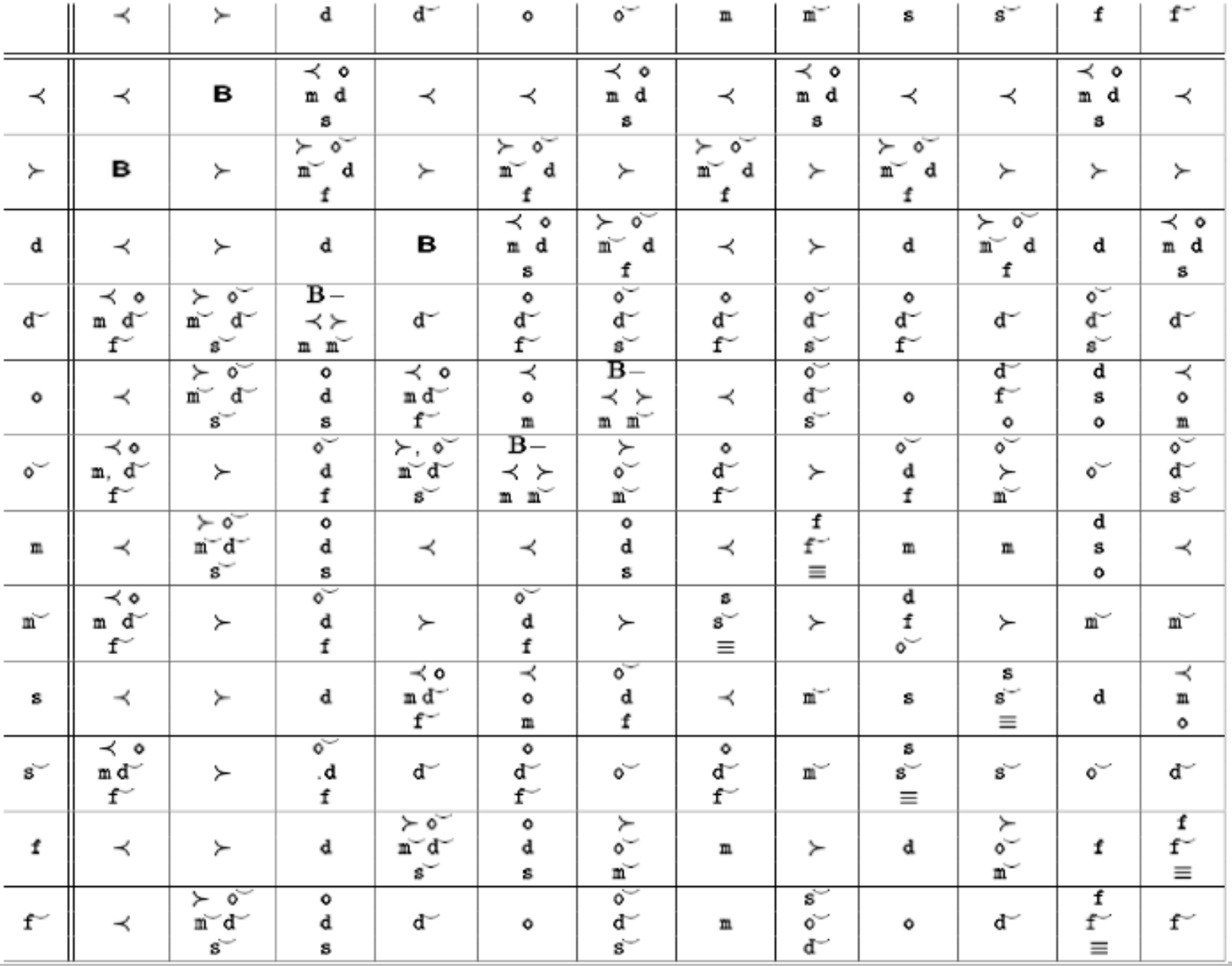
$I\ r_1 \circ r_2\ K$ ist die *Relation*, die sich ergibt, wenn es Intervall J gibt, sodass $I\ r_1\ J$ und $J\ r_2\ K$

Beispiel

- Es gelte $I < J$ und $J < K$. In welcher Relation stehen I und K?
  - ➥ Antwort: $I\ \{<\}\ K$
- Es gelte I o J und J d K . In welcher Relation stehen I und K?
  - ➥ Antwort: $I\ \{\mathtt{d}, \mathtt{o}, \mathtt{s}\}\ K$

Berechne/definiere **Kompositionstabelle** aller Basisrelationen (unter Auslassung des Einselements ≡) …

Kompositionstabelle



[Allen, 1983] nach B. Nebel

Komposition von Relationen

… ergibt sich als Vereinigung aller paarweisen Kompositionen der entsprechenden Basisrelationen.
Bestehende Relationen $I\ r\ K$ werden auf alle möglichen Weisen (alle Zwischenintervalle J) mit Kompositionen $I\ r_1 \circ r_2\ K$ verschärft:

$$I\ r\ K := I\ r\ K \cap I\ r_1 \circ r_2\ K$$

Mensabeispiel

Welche Relation gilt zwischen S und X?

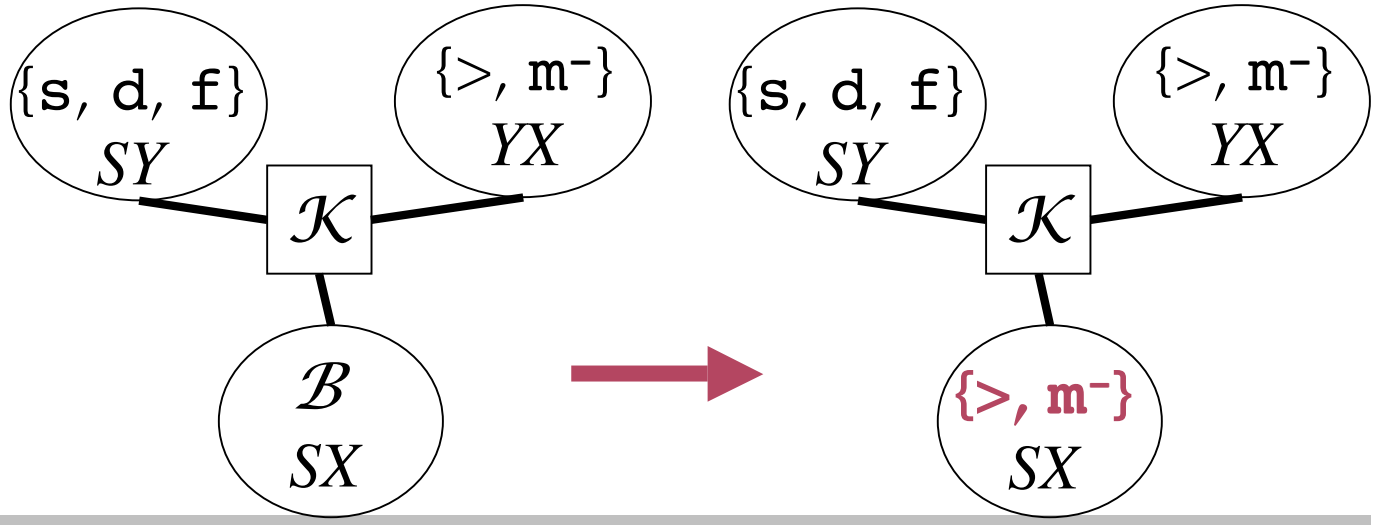
- Wir wissen bereits $S\ \mathcal{B}\ X$ und $S\ \{\mathtt{s}, \mathtt{d}, \mathtt{f}\}\ Y$ und $X\ \{<, \mathtt{m}\}\ Y$, also $Y\ \{>, \mathtt{m}^-\}\ X$.
- Durch Komposition $(S\ \mathcal{B}\ X \cap [S \cup(\mathtt{s}\circ>, \mathtt{s}\circ \mathtt{m}^-, \mathtt{d}\circ>, \ldots)\ X])$ ergibt sich: $S\ \{>, \mathtt{m}^-\}\ X$, also $X\ \{<, \mathtt{m}\}\ S$,
  also *Vorlesung X endet spätestens bei Mensaschluss*.

Temporales Schließen als CSP

Ermittlung der minimalen (bzgl. Zahl der Basisrelationen) Relationen zwischen Intervallen ist modellierbar als CSP („ICSP“):

- CSP-Variable sind Intervall-Paare (z.B. *SY*, *YX*, …)
- deren Wertebereiche sind Relationen (Teilmengen von $\mathcal{B}$)
- CSP-Relation zwischen allen passenden Variablentripeln ist jeweils die Kompositionstabelle $\mathcal{K}$

Mensabeispiel (Ausschnitt aus dem CSP)



Direkte Implementierung lokaler Konsistenz

Lokale Konsistenz in Kap.2 nur für binäre Constraints (AC-3)
Naive direkte Implementierung für ICSPs ist offensichtlich:

procedure NAIVE-LOCAL-ICSP-CONSISTENCY()

global variable: array $Table[i,j]$ holds relation between intervals i, j

repeat $old \leftarrow Table$

for each pair (i,j) where $1 \leq i,j \leq n$ **do**

for each k where $1 \leq k \leq n$ **do**

$Table[i,j] \leftarrow Table[i,j] \cap (Table[i,k] \bigcirc Table[k,j])$ **end end**

until $old = Table$

Terminiert, aber ineffizient (Doppelarbeit $Table[i,j]$ und $Table[j,i]$)
Lokale Konsistenz in $O(n^3)$ erzielbar

Von qualitativen zu quantitativen Intervallen

Rein qualitative Relationen zwischen Intervallen sind oft ausreichend/nützlich (Sprachverarbeitung, Reihenfolgeplanung,…)

Oft braucht man aber quantitative Information, z.B. über Dauern, Latenzzeiten, frühesten/spätesten Start einer Aktivität (Terminplanung/Scheduling)

Weit verbreiteter Ansatz zur Modellierung:

Temporal Constraint Satisfaction Problems (TCSPs)

Hier Konzentration auf ***Simple Temporal Problems*** (STPs)

R. Dechter, I. Meiri, J. Pearl: Temporal Constraint Networks. J. Art.Intell. 49:61-95 (1991)
Präsentation z.T. inspiriert von B. Williams

TCSPs

- Variablen $X_1,\ldots, X_n \in \Re^+$ (Zeitpunkte)
- Constraints sind jeweils Mengen von Paaren („Intervallen“) $[a,b]$ reeller Zahlen mit der Interpretation:
  - *unäre* Constraints $T_i = \{I_1, \ldots, I_k\} = \{[a_1,b_1], \ldots, [a_k,b_k]\}$ mit der Interpretation $(a_1 \leq X \leq b_1) \vee \ldots \vee (a_k \leq X \leq b_k)$ (Notation: $\langle a,b \rangle$)
  - *binäre* Constraints $T_{ij} = \{I_1, \ldots, I_l\} = \{[a_1,b_1], \ldots, [a_l,b_l]\}$ mit der Interpretation $(a_1 \leq X_j – X_i \leq b_1) \vee \ldots \vee (a_l \leq X_j – X_i \leq b_l)$
- Ein (binäres) TCSP ist eine Menge von Variablen und unären und binären Constraints auf diesen
- Ein STP ist ein TCSP, bei dem alle Constraints genau 1 Intervall enthalten, in dem also Disjunktion von Intervallen verboten ist
- $T_{ij} = \{[a_k,b_k],\ldots\}$ impliziert $T_{ji} = \{[-b_k,-a_k],\ldots\}$; ist ein T_{ij} nicht angegeben, interpretiere es als $T_{ij} = \{[-\infty,+\infty]\}$

Beispiel, sprachliche Formulierung

- John goes to work either by car (30-40 min) or by bus (at least 60 min)
- Fred goes to work either by car (20-30 min) or in a carpool (40-50 min)
- Today, John left home between 7:10 and 7:20
- Today, Fred arrived at work between 8:00 and 8:10
- Today, John arrived 10-20 min after Fred left home

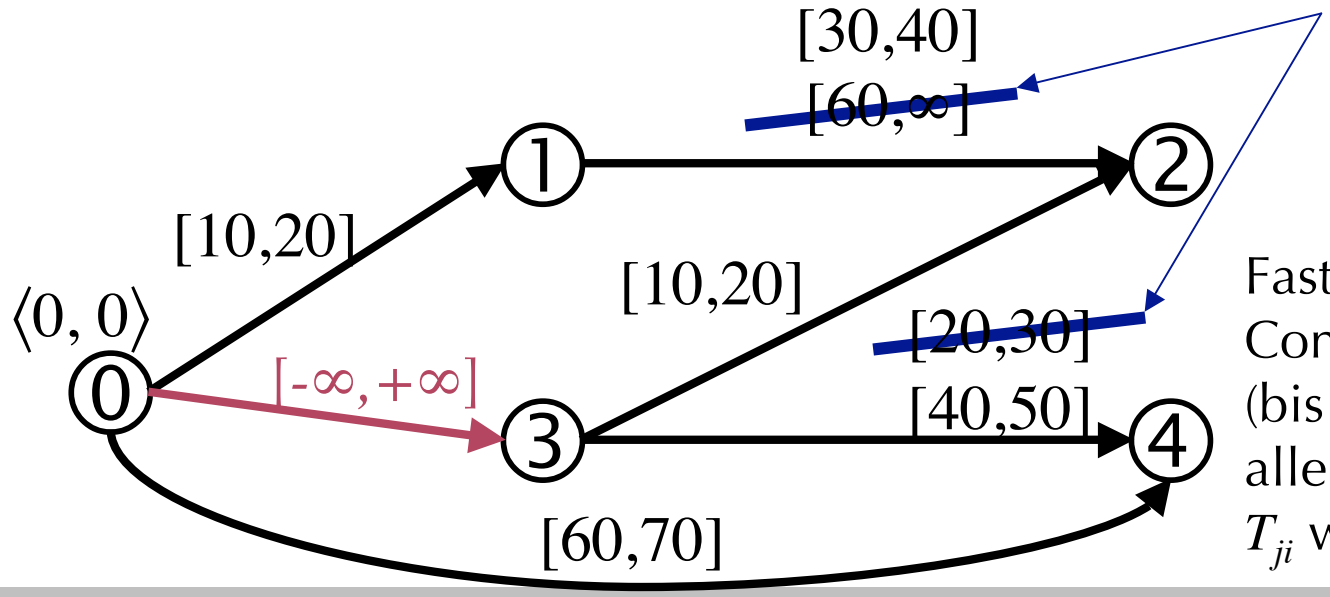
➥ Is this information consistent?

➥ Is it possible John took the car, Fred the carpool?

➥ At what time could Fred have left home?

Beispiel, Modellierung als binäres TCSP

- X_0: Referenzzeitpunkt, hier: 7:00 Uhr; Zeiteinheit: Minuten
- X_1: John verlässt das Haus
- X_2: John kommt bei der Arbeit an
- X_3: Fred verlässt das Haus
- X_4: Fred kommt bei der Arbeit an



Als STN-Beispiel

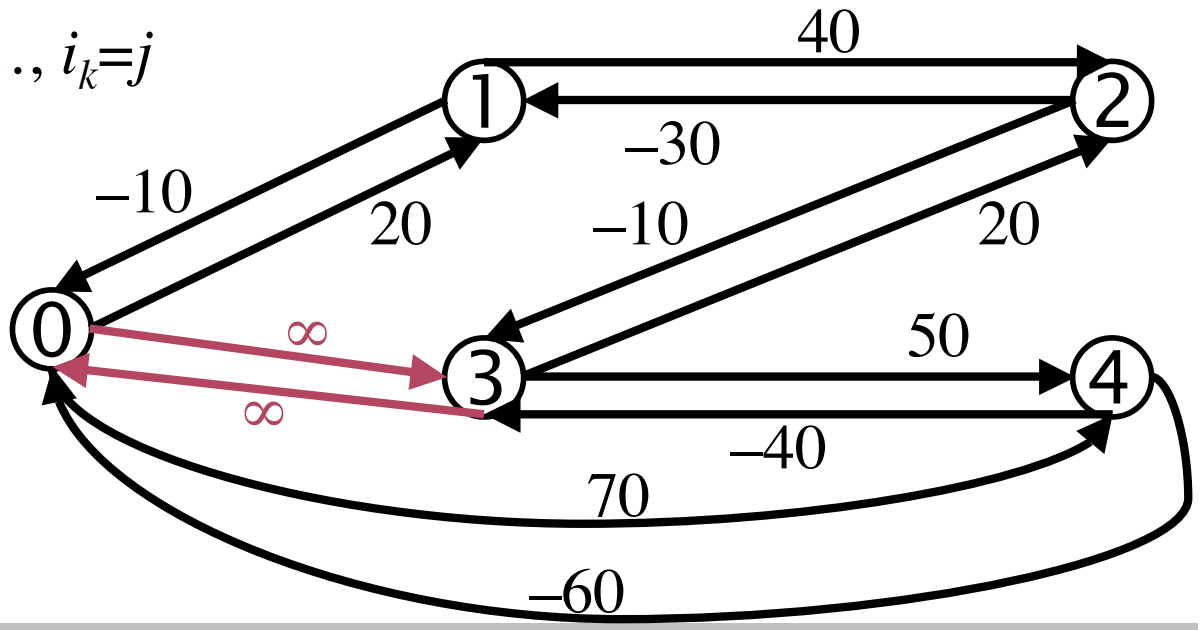
Fast alle impliziten Constraints weggelassen (bis auf T_{03}); alle konversen Constraints T_{ji} weggelassen

Distanzgraph eines STP

- Gleiche Knotenmenge wie CSP (STP-Variablen)
- Je zwischen zwei Knoten i,j gerichtete Kante $i \rightarrow j$ mit Gewicht d_{ij} (Zeit-Distanz)
- Für jeden binären Constraint $T_{ij} = \{[a,b]\}$ initialisiere $d_{ij} := b$ (und entsprechend konversem Constraint Kante $j \rightarrow i$ mit $-a$)
- Für alle Pfade $i=i_0, i_1, \ldots, i_k=j$ von i nach j:

$$d_{ij} \leq \sum_{l=1}^{k} d_{i_{l-1},i_l}$$

- gesucht sind d_{ij}^*, die minimalen d_{ij}



Berechnung der minimalen Zeitdistanzen

procedure FLOYD-WARSHALL(G) **returns** a minimal distance matrix
input: distance graph G with n nodes

for $i \leftarrow 1$ **to** n **do** $d_{ii} \leftarrow 0$ **end**
for $i,j \leftarrow 1$ **to** n **do** $d_{ij} \leftarrow b_{ij}$ **end**
for $k \leftarrow 1$ **to** n **do**
  for $i,j \leftarrow 1$ **to** n **do** $d_{ij} \leftarrow \min\{d_{ij}, d_{ik} + d_{kj}\}$ **end end**
return $[d_{ij}]$

- ☺ Speicherbedarf: $O(n^2)$
- ☺ Zeitbedarf: $O(n^3)$
- ☺ Terminiert sicher
- ☺ **Satz**: Ein STP ist konsistent gdw. sein minimaler Distanzgraph in der Diagonalen keine negativen Zahlen enthält

Lösungen des Beispielproblems

	X_0	X_1	X_2	X_3	X_4
X_0	0	20	∞	∞	70
X_1	-10	0	40	∞	∞
X_2	∞	-30	0	-10	∞
X_3	∞	∞	20	0	50
X_4	-60	∞	∞	-40	0

	X_0	X_1	X_2	X_3	X_4
X_0	0	20	50	30	70
X_1	-10	0	40	20	60
X_2	-40	-30	0	-10	30
X_3	-20	-10	20	0	50
X_4	-60	-50	-20	-40	0

Setzt man X_0: als Zeitpunkt 0, ergeben sich:

- Oberste Zeile: späteste Zeiten der Zeitpunkte X_i
- Linkeste Spalte: –1× früheste Zeiten der Zeitpunkte X_i

Entsprechend Rückübersetzung in STP mit minimalen W'bereichen

Die Antworten auf (fast) alle Fragen

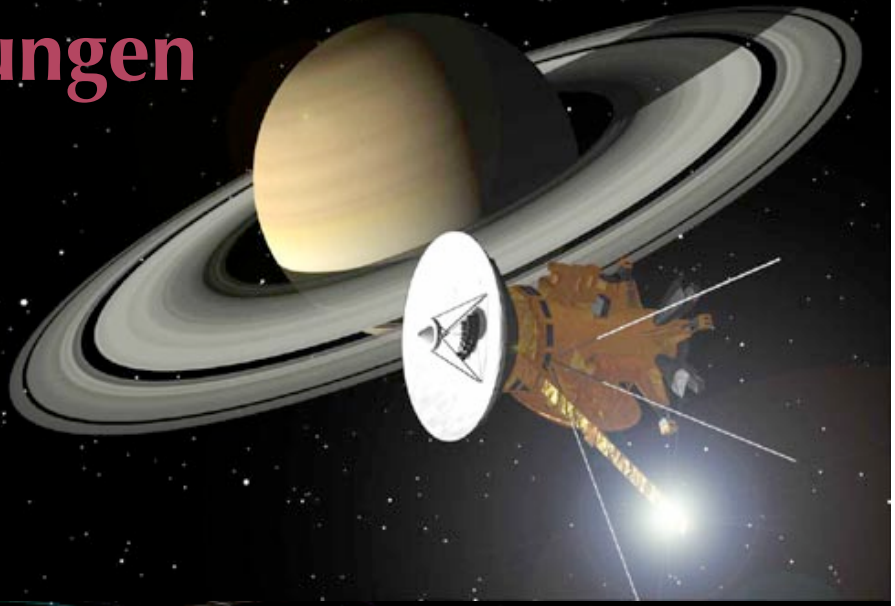
	X_0	X_1	X_2	X_3	X_4
X_0	0	20	50	30	70
X_1	-10	0	40	20	60
X_2	-40	-30	0	-10	30
X_3	-20	-10	20	0	50
X_4	-60	-50	-20	-40	0

- Is this information consistent? ✓
- Is it possible John took the car, Fred the carpool? ✓
- At what time could Fred have left home? (X_3) abs. [7:20, 7:30]

- Konsistenztest für volle TCSPs (mit Disjunktion) ist NP-hart
- Es gibt polynomielle Algorithmen für lokale Konsistenz

Anwendungen

Terminplanung
(Fertigung, Logistik, …)



… und Satellitensteuerung: z.B. CASSINI/Huygens Saturn- und Titanmission, Start 15.11.1997
Daten derzeit (1.12.2004):

- 4,138 mio km vom Saturn,
- 1249 mio km von der Erde,
- 6.649 km/h

http://saturn.jpl.nasa.gov/home/index.cfm

Modellierung von Wandel in Logik

Defizit der genannten Constraint-basierten Ansätze:

Die Intervalle sind „inhaltsfrei“!

Beispiel

Bei qualitativen wie bei TCSPs können straflos Intervalle überlappen, während derer Widersprüchliches gilt!

Modellierungen in Logik von Zeit/Wandel/Ereignis verbessern das

Beispiele

- modale Temporallogik
- Event-Kalkül
- Situationskalkül

Zur Ontologie des Situationskalküls

- Modelliere Veränderung als Übergang zwischen Situationen („Schnappschüssen der Welt“) in PL 1.Stufe
- Zeit kommt nur qualitativ als Folge von Situationen vor
- Gültigkeit mancher Prädikate bezieht sich auf Situationen; diese bekommen ein entsprechendes zusätzliches Argument: $P(x) \rightarrow P(x,s)$ (s Situations-Argument): propositionale Fluenten
- Ebenso können Funktionen zeitabhängig sein: $f(x) \rightarrow f(x,s)$: Fluenten unterschiedlichen Typs (numerische,…)
- Nicht situationsabhängige Prädikate und Funktionen bleiben wie sie waren

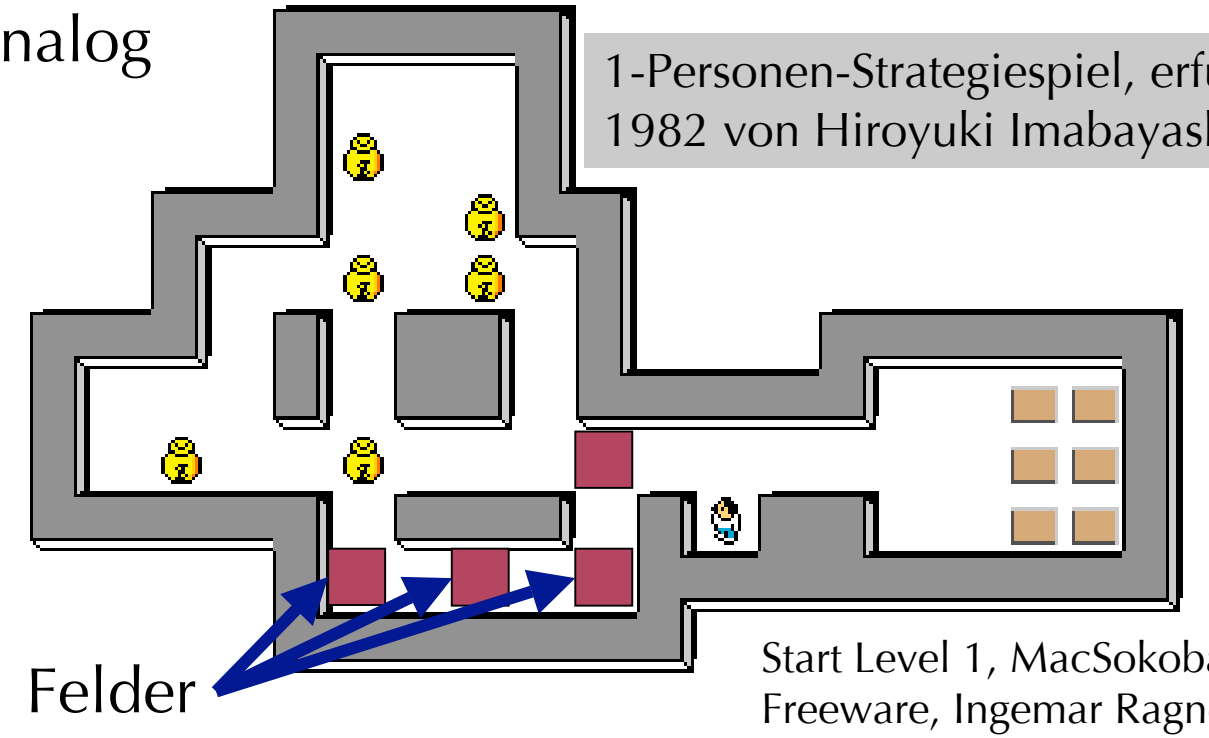
Beispieldomäne: Soko-Ban („Lagerarbeiter“)

- Bewege n Säcke in n Zielpositionen
- Säcke können nur *geschoben* werden, immer *nur 1 Sack*
- Spieler kann nur in benachbarte Felder *N*, *O*, *S*, *W*, wenn frei; Sack schieben analog

Sack

Spieler

Zielposition

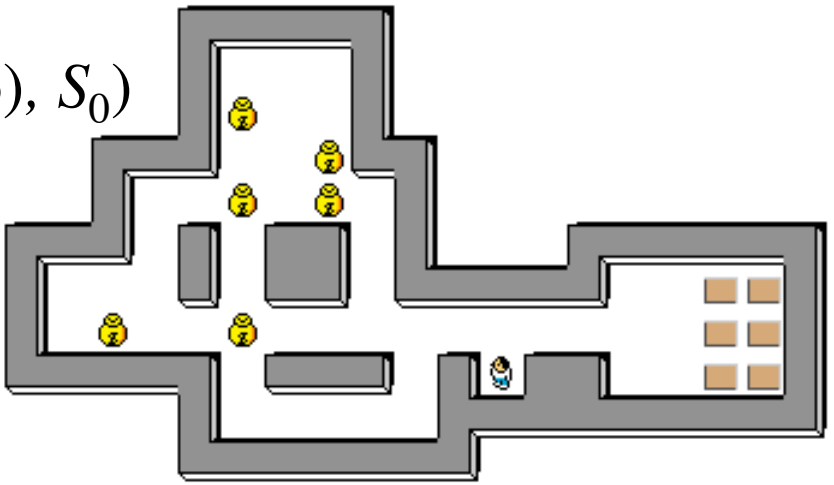


1-Personen-Strategiespiel, erfunden 1982 von Hiroyuki Imabayashi

Start Level 1, MacSokoban 2.1, Freeware, Ingemar Ragnemalm

Modellierung von Soko-Ban (I: Prädikate, F'en)

- Benenne Spalten mit Zahlen (beginnend links mit 1);
  benenne Zeilen mit Zahlen (beginnend oben mit 1);
  Spalten zuerst: Level 1 hat Felder von (1,1) bis (17,9)
- Funktion $F(.,.)$ ist Konstruktor für Felder: $F(1,6)$ bezeichnet Feld
- Prädikat $Ziel(.)$ für Ziel-Felder: $Ziel(F(16,6))$, …, $Ziel(F(17,8))$
- Startsituation heißt S_0
- propositionale Fluente:
  - $Frei(f,s)$: Feld f frei in s: $Frei(F(1,6), S_0)$
  - $Sack(f,s)$: Sack auf Feld f in s
    $Sack(F(2,7), S_0)$
  - $Ban(f,s)$: Spieler auf Feld f in s
    $Ban(F(11,8), S_0)$



Aktionen im Situationskalkül

Aktionen werden beschrieben durch Vorbedingungen (was muss gelten, damit Aktion anwendbar) und Nachbedingungen (welche Effekte erzeugt Aktion)

- Funktion (Situations-Fluent) *Res(a,s)* für Aktion *a* bezeichnet die Situation, die resultiert aus Anwendung von *a* in *s*
- Aktion *a* wird modelliert durch (1. Ansatz!!)
  - Vorbedingungs-Axiome der Form:
    Vorbedingung $\Rightarrow$ *Poss(a,s)*
  - Nachbedingungs-Axiome der Form
    Poss(a,s) $\Rightarrow$ Effekte der Aktion *a*

Hier und im Folgenden Allquantoren überall weggelassen!

Modellierung von Soko-Ban (II: Aktionen)

Aktion Geh

- Modelliert als 4 Aktionen *GehN, GehO, GehS, GehW*
  - Vorbedingungs-Axiome *GehN* (andere analog):
    $Ban(F(h,v),s) \wedge Frei(F(h,v-1),s) \Rightarrow Poss(GehN,s)$
  - Nachbedingungs-Axiome *GehN* (andere analog):
    $Poss(GehN,s) \Rightarrow Ban(F(h,v-1),Res(GehN,s)) \wedge \neg Ban(F(h,v),Res(GehN,s))$

Aktion Schieb

- Modelliert als 4 Aktionen *SchiebN, SchiebO, SchiebS, SchiebW*
  - Vorbedingungs-Axiome *SchiebN* (andere analog):
    $Ban(F(h,v),s) \wedge Sack(F(h,v-1),s) \wedge Frei(F(h,v-2),s) \Rightarrow Poss(SchiebN,s)$
  - Nachbedingungs-Axiome *SchiebN* (andere analog):
    $Poss(SchiebN,s) \Rightarrow Ban(F(h,v-1),Res(SchiebN,s)) \wedge \neg Ban(F(h,v),Res(SchiebN,s)) \wedge Sack(F(h,v-2),Res(SchiebN,s)) \wedge \neg Sack(F(h,v-1),Res(SchiebN,s)) \wedge Frei(F(h,v-1),Res(SchiebN,s)) \wedge \neg Frei(F(h,v-2),Res(SchiebN,s))$