Fakultät für
Mathematik und Informatik

Vervollständigung von partiellen Wissensänderungs-Operatoren

Abschlussarbeit im Studiengang M. Sc. Praktische Informatik

Marco Stock, 13.11.2020

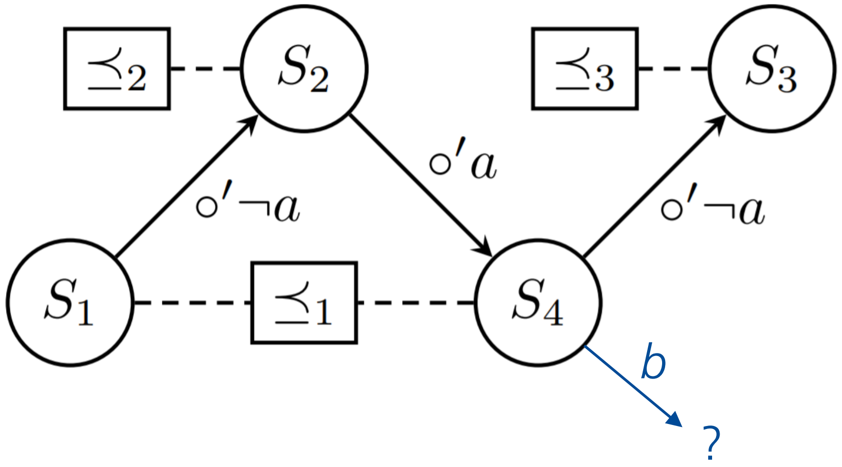
Fakultät für **Mathematik und Informatik**
Lehrgebiet Wissensbasierte Systeme
Betreuer: Prof. Dr. Christoph Beierle

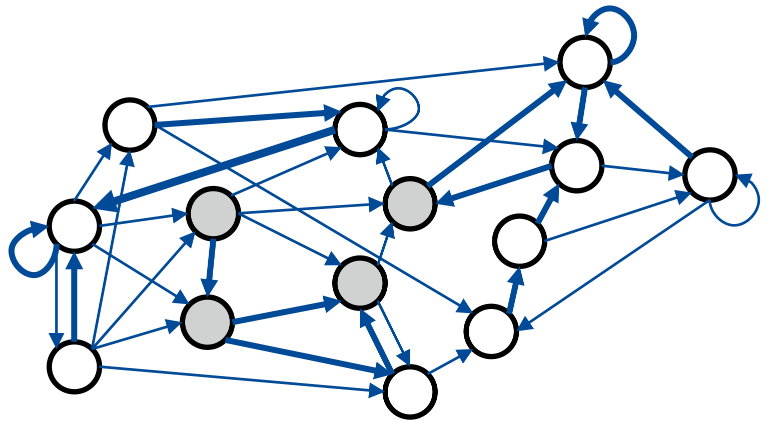
FernUniversität in Hagen

Summary

FOL conditions
for local belief changes

Partially specified change operator for $|\Omega| = 3$ worlds:
Aravanis et al, *Observations on darwiche and pearl's approach for iterated belief revision 2019*



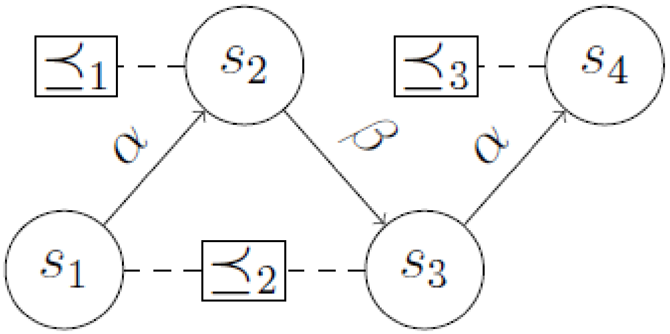


Amount of nodes $|N| \geq \sum_{k=0}^{|\Omega|} k!\ S(|\Omega|, k)$
Amount of edges $|E| = (2^{|\Omega|} - 1) \cdot |N|$

Änderungsräume

Definition 18 (Änderungsraum) *Sei Ω eine endliche Menge von Welten. Ein* Änderungsraum *über Ω ist ein Tupel $\mathbb{C} = (S, C, \tau, \ell)$, so dass*

- *(S, C) ein (gerichteter) Graph mit Knotenmenge S und Kantenmenge C ist, wobei die Elemente von S als Zustände bezeichnet werden und die Elemente von C als Änderungen,*
- *τ ordnet jedem Zustand $s \in S$ eine totale Quasiordnung $\tau(s) \subseteq \Omega \times \Omega$ zu, und*
- *ℓ ordnet jeder Änderung $(s_1, s_2) \in C$ eine Teilmenge $\ell(s_1, s_2) \subseteq \Omega$ zu.*



Aravanis et al, *Observations on darwiche and pearl's approach for iterated belief revision 2019*

Änderungsräume

- Ein Änderungsraum heißt *endlich*, falls S eine endliche Menge ist.
- Ein Änderungsraum wird als *deterministisch* bezeichnet, falls es für jedes $s_1 \in S$ und jede Menge $\alpha \subseteq \Omega$ höchstens ein $s_2 \in S$ gibt mit $\ell(s_1, s_2) = \alpha$.
- Ein Änderungsraum $\mathbb{C}^* = (S^*, C^*, \tau^*, \ell^*)$ *enthält einen Änderungsraum* $\mathbb{C} = (S, C, \tau, \ell)$ *komplett*, kurz $\mathbb{C} \sqsubseteq \mathbb{C}^*$, wenn $S \subseteq S^*$ und $C \subseteq C^*$, und $\tau(s_1) = \tau^*(s_2)$ und $\ell^*(s_1, s_2) = \ell(s_1, s_2)$ für alle $s_1, s_2 \in S$.
- Des Weiteren wird ein Änderungsraum (S, C, τ, ℓ) bezüglich Ω als *vollständig* bezeichnet, falls folgende zwei Bedingungen zutreffen:
  - Für jede totale Quasiordnung $\preceq \subseteq \Omega \times \Omega$ gibt es einen Zustand $s \in S$, so dass $\tau(s) = \preceq$ gilt.
  - Zu jedem Zustand $s \in S$ und jeder Teilmenge $\Omega' \subseteq \Omega$ gibt es einen Zustand s^*, so dass $(s, s^*) \in C$ und $\ell(s, s^*) = \Omega'$ gilt.

FO-BC Signatur

$$Pred = \{\preceq_1 /2,\ \preceq_2 /2, Mod_\alpha/1\}$$
$$Func = \{c/0\}$$

Definition 20 (Erfüllungsrelation auf Änderungsräumen) *Sei* $\mathbb{C} = (S, C, \tau, \ell)$ *ein Änderungsraum und* φ *eine Formel über einer FO-BC-Signatur.*
Für jedes $(s_1, s_2) \in C$ *wird die Interpretation*

$$\mathcal{A}^{\mathbb{C}}_{s_1,s_2} = (\Omega, \{c^{\mathcal{A}_{s_1,s_2}}\}, \{\tau(s_1), \tau(s_2), \ell(s_1, s_2)\})$$

definiert, wobei $c^{\mathcal{A}_{s_1,s_2}}$ *ein beliebiges Objekt aus* Ω *ist.*
Es sei $\mathbb{C} \models \varphi$*, falls* $\mathcal{A}^{\mathbb{C}}_{s_1,s_2} \models \varphi$ *für alle* $(s_1, s_2) \in C$.

Algorithmus

Algorithmus 1 : Computing the completion for a change space

Funktion `completion` **mit**

Input : Deterministischer endlicher Änderungsraum $\mathbb{C}$ über Ω und eine Formel φ über einer FO-BC-Signatur, so dass $\mathbb{C} \models \varphi$ gilt.

Output : Ein vollständiger deterministischer endlicher Änderungsraum $\mathbb{C}^*$ über Ω, der $\mathbb{C}$ komplett enthält und für den $\mathbb{C}^* \models \varphi$ gilt. Fehlermeldung, falls ein solcher Änderungsraum nicht existiert.

$S^* \leftarrow S$ **and** $C^* \leftarrow C$ **and** $\tau^* \leftarrow \tau$ **and** $\ell^* \leftarrow \ell$

foreach *total preorder* $\preceq$ *over* Ω **do**

    if *there is no* $s \in S$ *with* $\preceq_s = \preceq$ **then**

        $S^* \leftarrow S^* \cup \{s^*\}$ `//` `where` s^* `is a fresh state`

        $\tau^* \leftarrow \tau^* \cup \{s^* \mapsto \preceq\}$

foreach $s \in S^*$ **and** $\alpha \subseteq \Omega$ **do**

    if *there is no* $s^* \in S^*$ *with* $(s, s^*) \in C^*$ *and* $\ell^*(s, s^*) = \alpha$ **then**

        if *there is* $s^* \in S^*$ *with* $\mathcal{A}_{s,s^*}^{(S^*,C^*,\tau^*,\ell^*)} \models \varphi$ **then**

            $C^* \leftarrow C^* \cup \{(s, s^*)\}$

            $\ell^* \leftarrow \ell^* \cup \{(s, s^*) \mapsto \alpha\}$

        else

            return *no*

return (S, C, τ, ℓ) `// return change space` $\mathbb{C}^*$

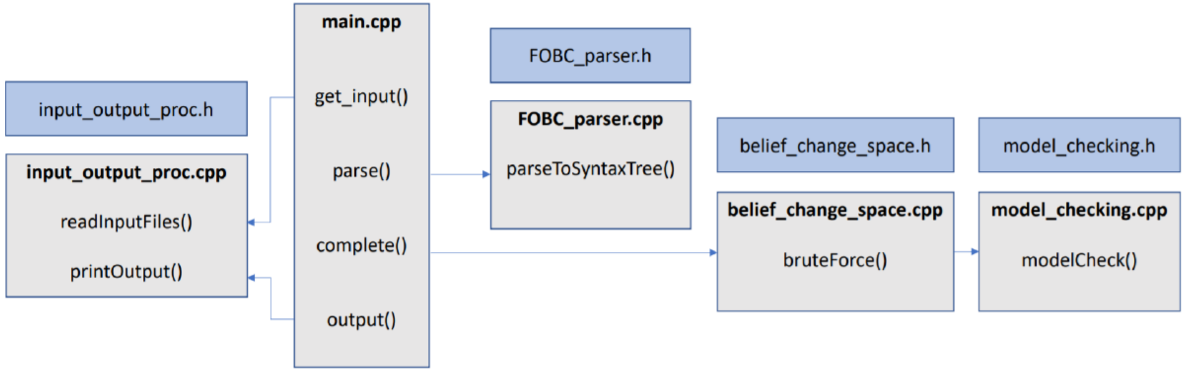


Abbildung 5: Softwarearchitektur des Programms (C++)

Algorithmus - Eingaben

Eingabe der Knoten
des Änderungsraums (.csv)

	A	B	C	D	E	F
1	*Predefined States s with corresponding total preorders t(s) over worlds w					
2	*Insert plausibility index as natural numbers (0 is most probable)					
3						
4	States/Worlds	w1	w2	w3		
5	s0	0	0	0		
6	s1	0	1	2		
7						
8						
9						

Eingabe der Kanten
des Änderungsraums (.csv)

	A	B	C	D	E	F	G	H
1	*Predefined Edges with corresponding worlds I(s1, s2)							
2	*Insert <true> for worlds, that are part of alpha for the corresponding edge, <false> for worlds, which are not							
3	*Make sure to at least have one <true> for each row, since edges without corresponding worlds are not accepted							
4	*make sure, that the states and worlds match the ones in the first input file							
5								
6	origin	destination/alpha	w1	w2	w3			
7	s0	s0	true	true	true			
8	s1	s0	false	false	true			
9								
10								

Algorithmus - Eingaben

Eingabe der FOBC Bedingungen (.txt)

```
* Variables are defined as characters in range [u, v,..., z]
* If more variables are needed two numbers can be added (e.g. u1, x15, w79)
*
* Brackets can be used. Default binding definition:
*       (strongest) ~, &, |, =>, <=>, A, E (weakest)
*
* The algorithm can get rid of spaces in the sentences.
* You can enter multiple sentences.
*
*
*----------------------------------------------------------------
*       Insert conditions below (one sentence per line)
*----------------------------------------------------------------

*CR1
(Mod(x) & Mod(y)) => (TPO1(x,y) <=> TPO2(x,y))

*CR2
(~Mod(x) & ~Mod(y)) => (TPO1(x,y) <=> TPO2(x,y))
.....
```

Listing 1: Eingabedatei 3 (FO-BC Formeln)

Symbol	Vereinfachte Eingabe für Parser	Bedeutung
$x \preceq_1 y$	$TPO1(x,y)$	Totale Quasiordnung Ausgangs-Wissensbasis
$x \preceq_2 y$	$TPO2(x,y)$	Totale Quasiordnung aktualisierte Wissensbasis
$x \in Mod_\alpha$	$Mod(x)$	Enthalten in neuer Information der Wissensänderung
$\neg$	$\sim$	Negation
$\wedge$	$\&$	Konjunktion
$\vee$	$\vert$	Disjunktion
$\rightarrow$	$=>$	Materielle Implikation
$\leftrightarrow$	$<=>$	Äquivalenz
$\forall x$	$A(x)$	Allquantor
$\exists x$	$E(x)$	Existenzquantor
c	c	Konstantensymbol
$\{u, v, ..., z\}$	$\{u, v, ..., z)$, $(u1, u2..., u99)$, $(v1, v2, ..., v99)$, ..., $(z1, z2, ..., z99)\}$	Variablen

Tabelle 4: Vereinfachte Schreibweise für die Programmeingabe

Algorithmus - Ausgaben

GraphViz

Listing 2: Ausgabedatei (.dot Format)

```
digraph CompletedOperator{

rankdir=LR; concentrate=false;

subgraph Predef {
node[color=blue];
edge[color=blue];
's0' -> 's0' [label=<<font color='blue'>{w1, w2, w3}</font>>]
's1' -> 's0' [label=<<font color='blue'>{w3}</font>>]
node[color=blue shape=box style=dashed];
edge[color=blue style=dashed arrowhead=none];
'{w1, w2, w3}' -> 's0'
'{w1} < {w2} < {w3}' -> 's1'
}

node[style=filled];
edge[color=grey];
's0' -> 's8*' [label='{w1}']
's0' -> 's4*' [label='{w2}']
's0' -> 's10*' [label='{w1, w2}']
.....

}
```

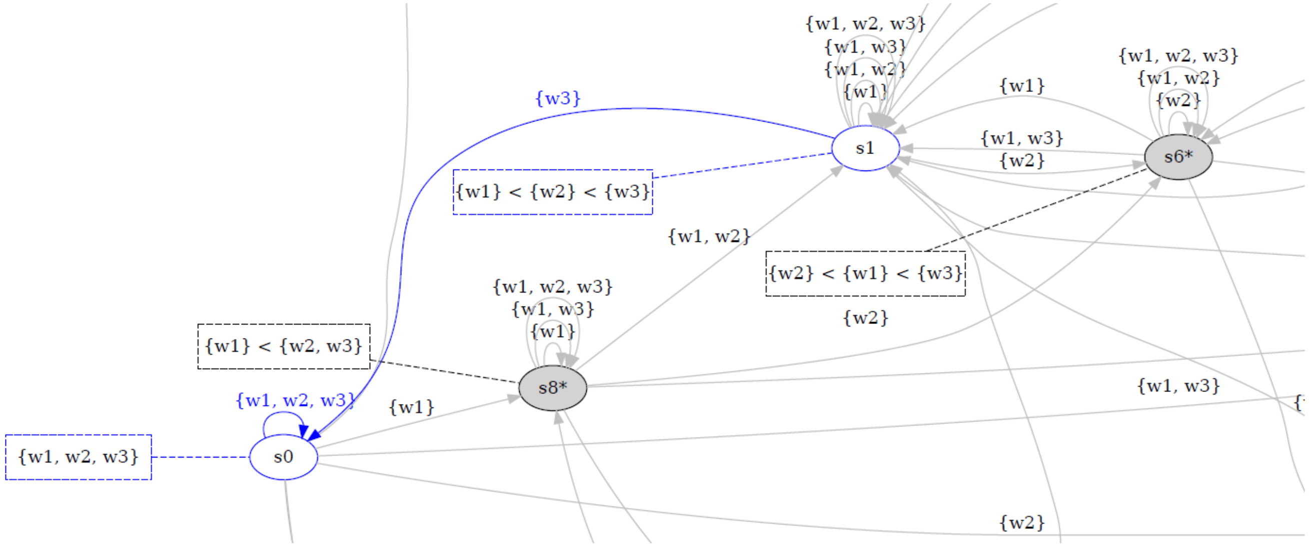


Abbildung 8: Interpretierte Ausgabedatei (.dot Format)

Algorithmus - Ausgaben

Listing 3: Konsolenausgabe

```
.....
SyntaxTree 5/5:
L-Branch of <=>
  L-Branch of &
    Mod with Arguments x and
  R-Branch of &
    A with Argument y
      L-Branch of =>
        Mod with Arguments y and
      R-Branch of =>
        TPO1 with Arguments x and y
R-Branch of <=>
  A with Argument z
    TPO2 with Arguments x and z




outputMode = random
reducedDiskSpace = 0



----------------------SUCESSFULLY FINISHED----------------------

Total States: 13
Anzahl gefundene einzelne Graphen: 22* 10^9
Execution time: 0s
```

Laufzeitbetrachtungen

– Initiale Aufrufe rekursiver Erfüllungscheck mit

$$= \left[(2^n - 1)\left(\sum_{k=0}^{n} k! \left\{ n \atop k \right\} + d\right) - E_{in}\right] \left(\sum_{m=1}^{S} n^{V_m} + \underbrace{\overline{i_{Fail}}(\overline{i_{Success}} - 1)}_{P}\right)$$

Anzahl Prädikatensymbole	Anzahl Welten n	minimale Aufrufe	maximale Aufrufe
1	2	9	54
2	4	$1,1 * 10^3$	$3,4 * 10^5$
3	8	$1,4 * 10^8$	$6,1 * 10^{14}$
4	16	$3,5 * 10^{20}$	$3,0 * 10^{37}$

Tabelle 6: Initiale Aufrufe des rekursiven Erfüllungschecks

Laufzeitbetrachtungen

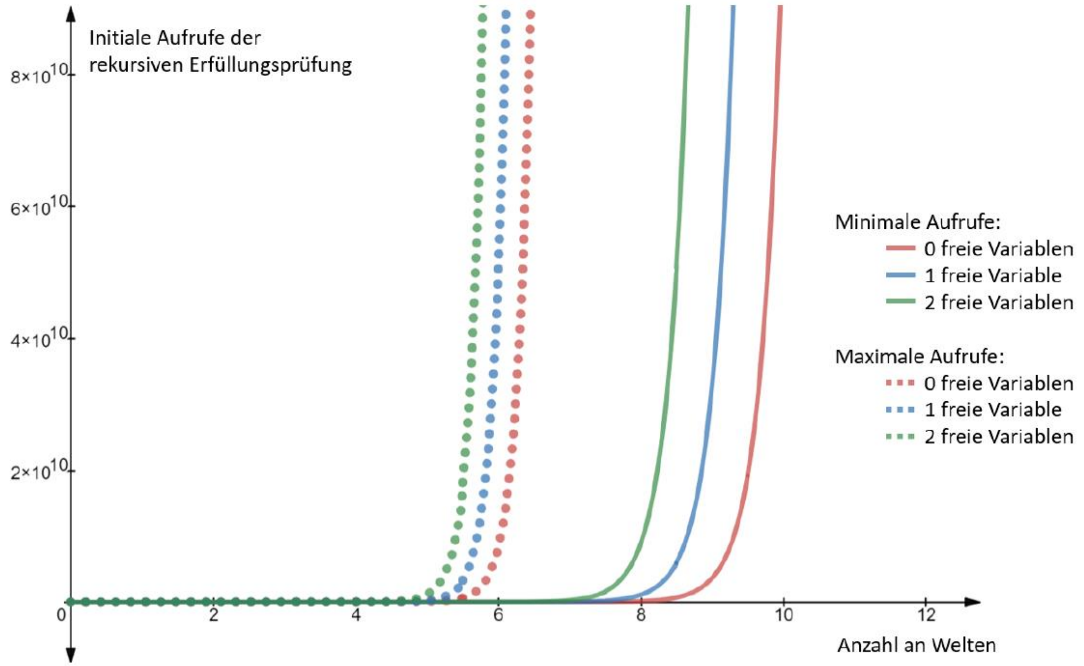


Abbildung 9: Zusammenhang von Laufzeit zu Welten und freien Variablen

Beispiel - Revision

AGM Revision

(Mod(x) & (A(y) (Mod(y) => TPO1(x, y)))) <=> (A(z) TPO2(x, z))

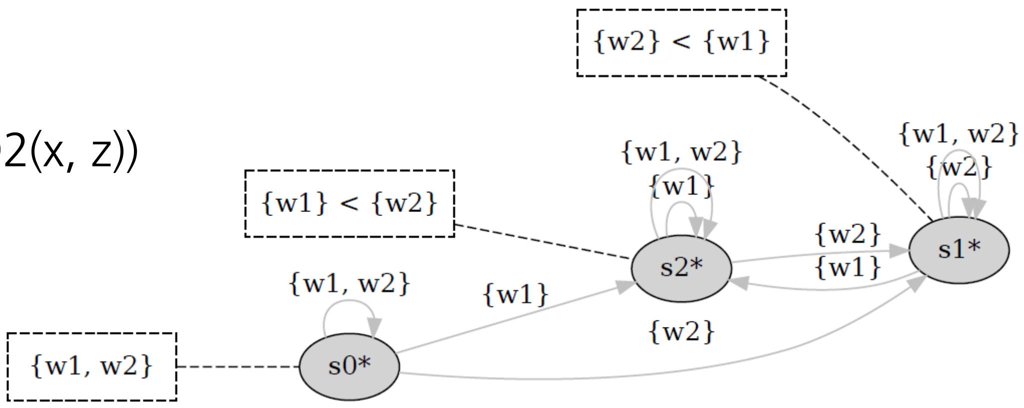


Abbildung 11: AGM-Revisionsoperator mit zwei Welten

DP Postulate (Iterated Belief Revision)

CR1 (Mod(x) & Mod(y)) => (TPO1(x, y) <=> TPO2(x, y))

CR2 (Mod(x) & Mod(y)) => (TPO1(x, y) <=> TPO2(x, y))

CR3 (Mod(x) & Mod(y)) => (TPO1(y, x) => TPO2(y, x))

CR4 (Mod(x) & Mod(y)) => (TPO1(x, y) => TPO2(x, y))

Beispiel - Kontraktion

AGM Kontraktion

((~Mod(x) & (A(y) (~Mod(y) => TPO1(x, y)))) | (A(y) TPO1(x, y))) <=> (A(z) TPO2(x, z))

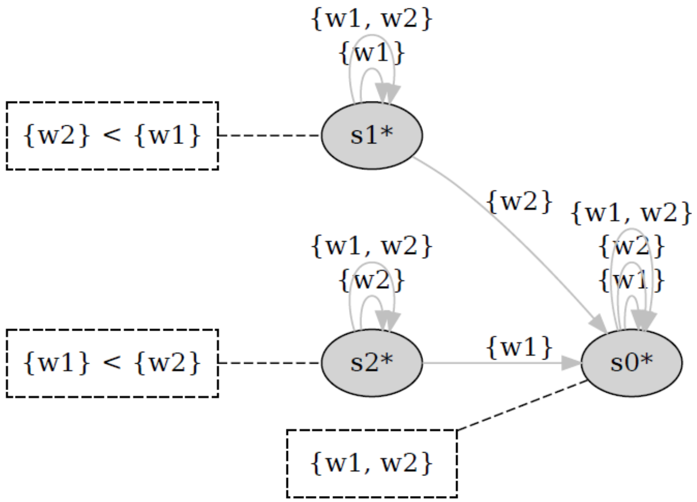


Abbildung 12: AGM-Kontraktionsoperator mit zwei Welten

Ausblick

- Erweiterung auf partielle Quasiordnungen
- Erweiterung der FO-BC Signatur
- Programm zur Anwendung des erzeugten Operators
- Vergleich von verschiedenen Auswahlkriterien für die einzufügenden Kanten