Wissensrepräsentation

–

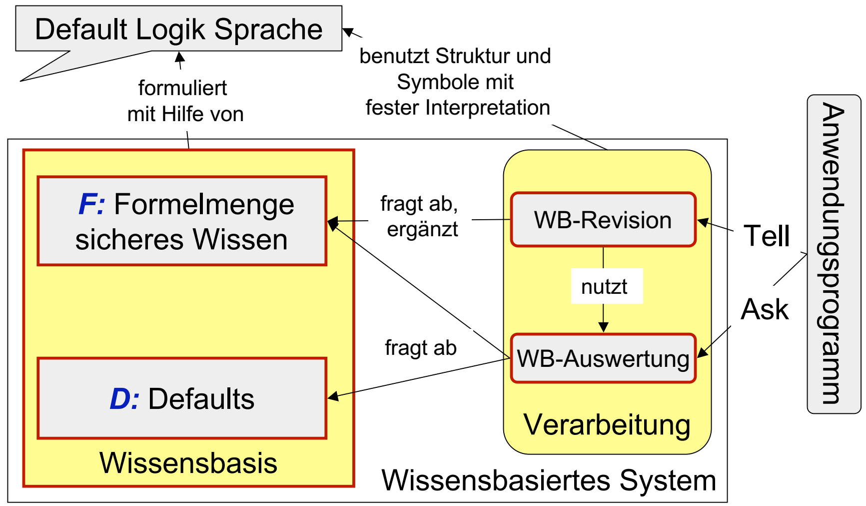
Christopher Habel, Hedda Schmidtke

Sommersemester 2004

Sitzung 19: Default Logik

- Beweis- / Ableitungsverfahren für Default-Logik: Verfahren zur Extensionsberechnung.
- Weiterentwicklungen der Default-Logik: Priorisierungen

Wissensbasiertes System mit Default Logik



Schließen mit Defaults: Gerechtfertigte Konsequenzen

Gegeben eine Default Theorie $\Delta = \langle F, D\rangle$.

Vernünftige Menge von impliziten, d.h. ableitbaren, Wissensentitäten, sollte enthalten:

1. **Das sichere Wissen aus *F*.** $[\alpha \in F]$
2. **Das aus *F* mit klassischen logischen Verfahren ableitbare Wissen.** $[F \vdash \alpha]$
3. **Das in Bezug auf *F* unter Verwendung von Defaults aus *D* plausible Wissen.** $[F \vdash_{\Delta L} \alpha]$
4. **Aber keine Formeln, die sich nicht aus *F* zusammen mit den Konsequenzen ‚anwendbarer' Defaults herleiten lassen.**

Extensionen einer Default Theorie: Reiters „Quasi-induktive" Charakterisierung

Gegeben eine Default Theorie $\Delta = \langle F, D\rangle$.

Sei $S_0, S_1, \ldots$ eine Sequenz von Formelmengen, für die gilt:

1. $S_0 = F$
2. $S = \cup_i S_i$
3. $S_{i+1} = S_i \cup \{ w(c) \mid$ $\dfrac{\alpha(c) : \beta_1(c),\ldots,\beta_m(c)}{w(c)}$ ist eine Instanz eines Defaults aus *D*, $S_i \vdash \alpha(c)$. $\beta_j(c)$ ist konsistent zu S, für alle j, $1 \le j \le m \}$

- Die Menge der logischen Konsequenzen aus **S** ist eine Extension von Δ.

Wissensverarbeitung mit Default-Logik

Δ-Ableitbarkeit (erste Charakterisierung)

Sei $\Delta = \langle F, D \rangle$ eine Default Theorie und α eine Formel.
$F \vdash_{\Delta} \alpha$, wenn α in allen Extensionen zu Δ enthalten ist.

- Defaults sind domänenspezifische Inferenzregeln.
  - im Gegensatz zu Inferenzregeln der klassischen Logik, die generell (domänenunabhängig) sind.
  - Die Wissensbasis von default-basierten WBS umfasst also Fakten und echte Inferenzregeln.
- Eine Definition von Δ-Ableitbarkeit setzt ein Ableitungsverfahren voraus.

Literatur

Zur Beweistheorie für die **Default Logik**

Antoniou, Grigoris (1997). *Nonmonotonic reasoning*. Cambridge, MA: MIT Press.

Raymond Reiter (1980). A Logic for Default Reasoning. *Artificial Intelligence, 13.* 81-132.

David Poole. Default Logic. In *D. Gabbay, C. Hogger, and J. Robinson (eds.), Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning,* Oxford University Press (1994), pp. 189 – 215.

Zur Erinnerung: Die Syntax von Defaults

Definition

Ein Default δ hat die Form, $\dfrac{\alpha : \beta_1,\ldots,\beta_m}{w}$

wobei α, β_i (m≥ 1) und w offene Formeln aus $\mathcal{PL}1$.

- α : Vorbedingung / precondition *prec*(δ)
- β_i : Rechtfertigungen / justifications (m≥ 1) *just*(δ)
- w : Konsequenz / consequence *cons*(δ)

➢ **Ein Default δ ist ein Schema, das durch Instanziierung zu geschlossenen Defaults führt.**

- Derartige Schemata werden häufig auch als „offene Defaults“ bezeichnet.
- Verschiedene Ansätze unterscheiden sich darin, ob sie von offenen oder geschlossenen Defaults ausgehen.

Antonious Verfahren zur Extensionsberechnung Übersicht

Default-Sequenzen

stellen eine Anordnung dar, in der Defaults angewendet werden können.

In- und Out-Mengen

repräsentieren das gegenwärtige Wissen nach Anwendung einer Sequenz von Defaults (In) bzw. die Gegebenheiten, die nicht zum Wissensbestand gehören dürfen (Out).

Ableitungen

sind Sequenzen, die gewisse Bedingungen für die erfolgreiche Anwendung von Defaults berücksichtigen.

Extensionen

sind die deduktive Hülle der In-Mengen erfolgreicher, abgeschlossener Ableitungen.

Antonious Verfahren zur Extensionsberechnung (1)
Default Sequenzen

Gegeben: Eine Default Theorie $\Delta = \langle F, D \rangle$.

- Sei $\Pi = (\delta_0, \delta_1, \ldots)$ eine endliche oder unendliche Folge von Instanzen von Defaults aus Δ (ohne mehrfaches Auftreten einer Default-Instanz).
  - $\Pi[k]$ bezeichnet das Anfangssegment der ersten k Defaults aus Π. (Mit der Konvention, dass ein Anfangssegment dieser Länge existiert.)

➢ Π stellt eine Anordnung dar, in der Defaults angewendet werden können.

➢ Intendierte Interpretation:
Default-Anwendungsprozesse für Δ

Antonious Verfahren zur Extensionsberechnung (2)
In- und Out-Mengen

Zu jeder Sequenz Π von $\Delta = \langle F, D \rangle$ existieren zwei Mengen von Formeln *In*(Π) und *Out*(Π).

- $In(\Pi) = Th_{\mathcal{PL}1}(M)$, wobei $M = F \cup \{cons(\delta) \mid \delta \text{ aus } \Pi\}$
  - $In(\Pi)$ sammelt die Formeln, die durch Anwendung von Defaults aus Δ entstehen, und repräsentiert das gegenwärtige Wissen nach Anwendung der Defaults aus Π.
  - $In(\Pi[0]) = Th_{\mathcal{PL}1}(F)$
- $Out(\Pi) = \bigcup_{\delta \in \Pi} Out(\delta)$, wobei
  $Out(\delta) = \{\neg\psi \mid \psi \in just(\delta)\}$
  - $just(\delta)$ bezeichnet die Menge der Rechtfertigungen von δ
  - $Out(\Pi)$ sammelt die Formeln, die weder gegenwärtig noch später zum Wissensbestand gehören dürfen.

Default-Anwendungssequenzen – Beispiel (1)
INs & OUTs

$$D \quad \delta_1 = \frac{green(x) : \neg likesCars(x)}{\neg likesCars(x)} \qquad \delta_2 = \frac{likesCars(x) : possessCar(x)}{possessCar(x)}$$

F = {*green(Pete)*}

1. $\Pi_1 = (\delta_1[Pete])$
   - $In(\Pi_1) = Th_{\mathcal{PL}1}(\{green(Pete), \neg likesCars(Pete)\})$
   - $Out(\Pi_1) = \{likesCars(Pete)\}$

2. $\Pi_2 = (\delta_2[Pete], \delta_1[Pete])$
   - $In(\Pi_2) = Th_{\mathcal{PL}1}(\{green(Pete), posCar(Pete), \neg likCars(Pete)\})$
   - $Out(\Pi_2) = \{\neg possessCar(Pete), likesCars(Pete)\}$
   - ➢ Dies besagt nichts über die Anwendbarkeit der Defaults in der Sequenz Π_2

Default-Anwendungssequenzen – Beispiel (2)
Default-Anwendung

$$D \quad \delta_1 = \frac{green(x) : \neg likesCars(x)}{\neg likesCars(x)} \qquad \delta_2 = \frac{likesCars(x) : possessCar(x)}{possessCar(x)}$$

F = {*green(Pete)*}

1. $\Pi_2 = (\delta_2[Pete], \delta_1[Pete])$ keine anwendbare D-Sequenz
   - $likCars(Pete) \notin In(\Pi_2[0]) = Th_{\mathcal{PL}1}(F) = Th_{\mathcal{PL}1}(\{green(Pete)\})$
   - $In(\Pi_2[0]) \approx$ Wissensbestand vor dem „Versuch“ $\delta_2[Pete]$ anzuwenden.

2. $\Pi_1 = (\delta_1[Pete])$
   - $green(Pete) \in In(\Pi_1[0]) = Th_{\mathcal{PL}1}(F) = Th_{\mathcal{PL}1}(\{green(Pete)\})$
   - Also: Π_1 ist eine anwendbare D-Sequenz

Antonious Verfahren zur Extensionsberechnung (3)
Default-Ableitungen

Definition

Sei $\Delta = \langle F, D \rangle$ eine Default Theorie.

Eine Defaultsequenz $\Pi = (\delta_0, \delta_1, \ldots)$ ist eine Ableitung zu Δ genau dann, wenn gilt:

Jedes δ_k (aus Π) ist in *In*($\Pi[k]$) anwendbar.

- „δ anwendbar in einer Formelmenge *M* “:
  - $prec(\delta) \in Th_{\mathcal{PL}1}(M)$
  - $Out(\delta) \cap Th_{\mathcal{PL}1}(M) = \varnothing$
- Die Indizierung der Sequenz Π ist so gewählt, dass vor der Anwendung von δ_k eine Anfangssequenz von k Defaults existiert.

Default-Anwendungssequenzen – Beispiel (3)
Default-Ableitungen

D $\delta_1 = \dfrac{green(x) : \neg likesCars(x)}{\neg likesCars(x)}$ $\delta_2 = \dfrac{likesCars(x) : possessCar(x)}{possessCar(x)}$

F = {*green(Pete)*}

Zeige: $\Pi = (\delta_1[Pete], \delta_2[Pete])$ ist keine Ableitung zu Δ.

0. $In(\Pi[0]) = Th_{\mathcal{PL}1}(F) = Th_{\mathcal{PL}1}(\{green(Pete)\})$
   - $Prec(\delta_1) = green(Pete) \in In(\Pi[0])$
   - $out(\delta_1) = \{ likesCars(Pete) \} \rightarrow out(\delta_1) \cap In(\Pi[0]) = \varnothing$
1. $In(\Pi[1]) = Th_{\mathcal{PL}1}(\{green(Pete), \neg likesCars(Pete)\})$
   - $Prec(\delta_2) = likesCars(Pete) \notin In(\Pi[1])$
   - Also: δ_2 nicht anwendbar in $\Pi[1]$
   - Also: Π ist keine Ableitung zu Δ.

Default-Ableitungen: Eigenschaften

Definition

Sei $\Pi = (\delta_0, \delta_1, \ldots)$ eine Ableitung zu $\Delta = \langle F, D \rangle$.

1. **Π ist erfolgreich, gdw. *In*(Π) $\cap$ *Out*(Π) = $\varnothing$.**
   Anderenfalls ist Π gescheitert / fehlgeschlagen.
   - **Erfolg** ≈ Es ist nichts schiefgegangen, d.h. in ***Out*(Π)** sind keine Formeln aufgenommen worden, die für ***In*(Π)** – den Wissensbestand – akzeptiert werden.
2. **Π ist abgeschlossen, gdw. jedes $\delta \in D$, das in *In*(Π) anwendbar ist, in Π vorkommt.**
   - **Abschluss** ≈ Nicht nur im Hinblick auf die deduktive Hülle der logischen Schlüsse, sondern auch bei den Defaults, wurden alle möglichen Schlüsse durchgeführt.

Extensionen einer Default-Theorie

Definition (Antoniou):

Eine Formelmenge *E* ist eine Extension einer Default-Theorie $\Delta = \langle F, D \rangle$ genau dann, wenn es eine abgeschlossene und erfolgreiche Ableitung Π zu Δ gibt, so dass *E* = *In*(Π).

➢ **Diese Charakterisierung**
- ist äquivalent zu der Definition von Reiter (Vorl. 18).
- ist eine Operationalisierung der „quasi-induktiven“ Charakterisierung von Reiter (Vorl. 18).

Extensionsberechnung mit Ableitungsbäumen

Ableitungsbaum zu
$\Delta = \langle F, D\rangle$

Kanten haben Defaults als Markierung.

Knoten sind markiert mit:

- In-set
- Out-set
- gegebenfalls Information über success & closure

Wurzel ist Markiert mit

- In = Th(F)
- Out = $\varnothing$

Aufbau des Ableitungsbaums

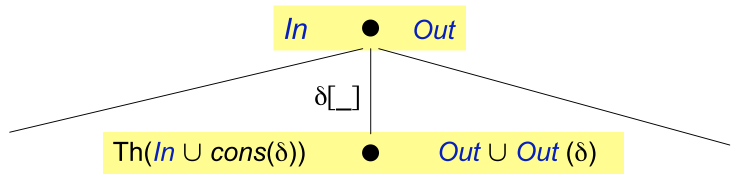
N ein Knoten mit *In* & *Out*

- N wird nur dann expandiert, wenn *In* $\cap$ *Out* = $\varnothing$; anderenfalls: failed
- Falls *In* $\cap$ *Out* = $\varnothing$ aber kein anwendbarer Default existiert: closed & successful
- Nachfolger N(δ) für alle δ, die bisher (auf dem Weg zu N) nicht auftreten, und die in *In* anwendbar sind.
- Nachfolger N(δ) ist markiert mit
  - *In*(N(δ))= Th(*In* $\cup$ {*cons*(δ)})
  - *Out*(N(δ))= *Out* $\cup$ *Out* (δ)

Aufbau des Ableitungsbaums

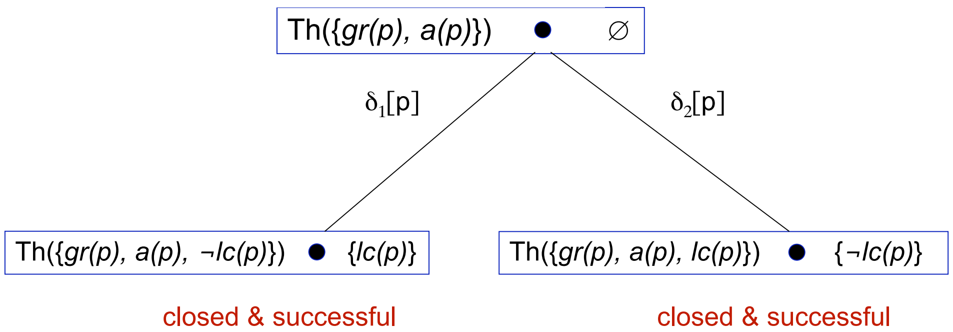
N ein Knoten mit *In* & *Out*

- N wird nur dann expandiert, wenn *In* $\cap$ *Out* = $\varnothing$; sonst: failed (Test der Bedingung für erfolgreiche Ableitung)
- Falls *In* $\cap$ *Out* = $\varnothing$ aber kein anwendbarer Default existiert: closed & successful (Test der Abschluss der Ableitung)
- Nachfolger N(δ) für alle δ, die bisher (auf dem Weg zu N) nicht auftreten, und die in *In* anwendbar sind.
- Nachfolger N(δ) ist markiert mit
  - *In*(N(δ))= Th(*In* $\cup$ {*cons*(δ)})
  - *Out*(N(δ))= *Out* $\cup$ *Out* (δ)



Ableitungsbaum für den Grünen-ADAC-Diamond

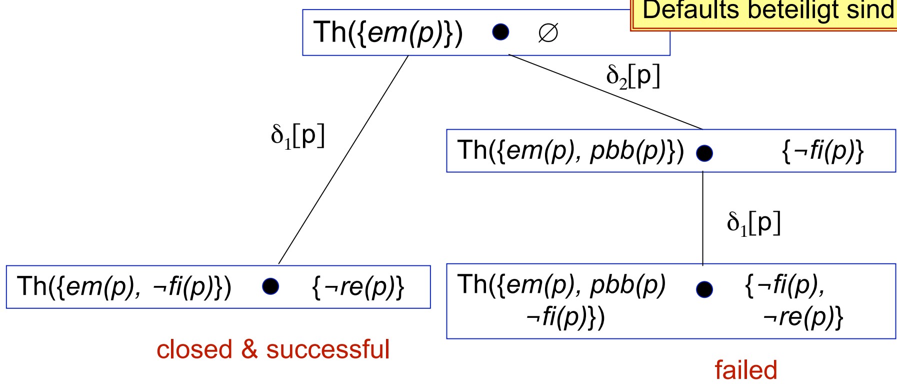
$$\boldsymbol{D} \quad \delta_1 = \frac{green(x) : \neg\, likesCars(x)}{\neg\, likesCars(x)} \quad \delta_2 = \frac{ADACmemb\,(x) : likesCars(x)}{likesCars(x)}$$



Spätere Defaultanwendung kann die Anwendbarkeit einer früheren Defaultanwendung zerstören

$$\boldsymbol{D}\delta_1 = \frac{: reliable(x)}{\neg\, fired(x)} \quad \delta_2 = \frac{: fired(x)}{pb\_bonus(x)}$$

dieses Verhalten ergibt sich nur, wenn nicht-normale Defaults beteiligt sind.



Maschinelle Beweisverfahren für Default-Theorien (1)

Aufbau von Ableitungsbäumen

- unproblematisch bei endlichen Default-Theorien
- bei nicht-endlichen Default-Theorien ist die Prüfung des Abschlusses gegebenenfalls problematisch
- Setzt immer einen klassischen Theorembeweiser voraus
  - für die Berechnung von *In*(N(δ))= Th(*In* ∪ {*cons*(δ)})
  - für die Prüfung von *In* ∩ *Out* = ∅

In ● *Out*

δ[_]

Th(*In* ∪ {*cons*(δ)}) ● *Out* ∪ *Out* (δ)

Prüfung auf *In* ∩ *Out* = ∅

Maschinelle Beweisverfahren für Default-Theorien (2) Abschließende Bemerkungen

Aufbau von Ableitungsbäumen

- Prototypisches Prolog-Programm bei Antoniou (Ch. 4.3)
  - Nur für Defaults mit einer Rechtfertigung
  - Setzt einen klassischen Beweiser voraus.

Ein wichtiges generelles Resultat

Reiter (1980) Sect. 4 (pp. 99ff)

- Extensionsberechnung ist nicht semi-entscheidbar, denn sie beinhaltet prädikatenlogische Erfüllbarkeitsprüfung, die nur semi-entscheidbar ist, für den Aufbau des Beweises.

  In Antonious Verfahren des Aufbaus von Ableitungsbäumen, ist diese Erfüllbarkeitsprüfung Bestandteil der Berechnung von *In*(N(δ))= Th(*In* ∪ *cons*(δ)) und der Prüfung von *In* ∩ *Out* = ∅.

Normale Default-Theorien

Ein Default ist normal, falls er die Form $\frac{\alpha : w}{w}$ hat, d.h. die Konsequenz die einzige Rechtfertigung ist .

Eine Default-Theorie ist normal, falls sie nur normale Defaults enthält.

Wichtige Eigenschaften:

1. Jede zulässige Default-Sequenz zu einer normalen Default-Theorie ist erfolgreich.
2. Normale Default-Theorien besitzen stets nicht-leere Extensionen. (Falls *F* nicht leer ist.)
3. Wenn E_1 und E_2 verschiedene Extensionen einer normalen Default-Theorie sind, dann ist $E_1 \cup E_2$ inkonsistent. [Orthogonalität der Extensionen]

Das Verhinderungsproblem (preemption problem): Beispiel (1)

D $\delta_1 = \frac{canadian(x) :\ speaks\_english(x)}{speaks\_english(x)}$ $\delta_2 = \frac{quebecian(x) : speaks\_french(x)}{speaks\_french(x)}$

F = {*quebecian(Pierre)*, ∀ x *quebecian (x)* ⇒ canadian*(x)*}

Th({*qu(p), ca(p)*}) ● ∅

δ_1[p] — δ_2[p]

Th({*qu(p), ca(p), se(p)*}) ● {¬*se(p)*}

Th({*qu(p), ca(p), sf(p)*}) ● {¬*sf(p)*}

δ_2[p] — δ_1[p]

Th({*qu(p), ca(p), se(p), sf(p)*}) ● {¬*se(p)*, ¬*sf(p)*}

Th({*qu(p), ca(p), sf(p), se(p)*}) ● {¬*sf(p)*, ¬*se(p)*}

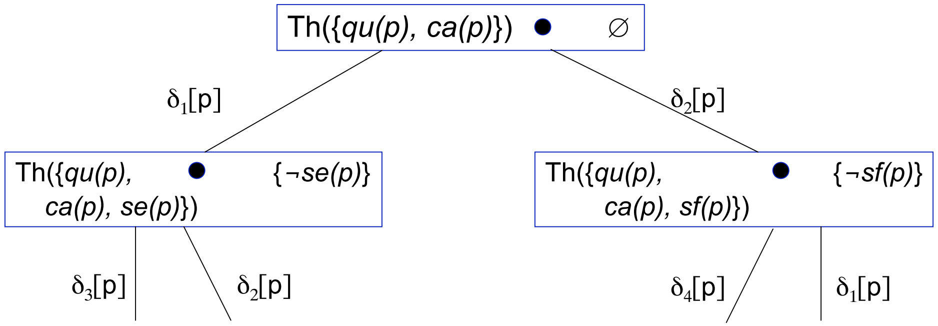
closed & successful — closed & successful

Das Verhinderungsproblem (preemption problem): Beispiel (2)

D $\delta_1 = \frac{\textit{canadian(x) : speaks\_english(x)}}{\textit{speaks\_english(x)}}$ $\delta_2 = \frac{\textit{quebecian(x) : speaks\_french(x)}}{\textit{speaks\_french(x)}}$

$\delta_3 = \frac{\textit{sp\_engl(x)} : \neg \textit{sp\_french(x)}}{\neg \textit{sp\_french(x)}}$ $\delta_4 = \frac{\textit{sp\_french(x)} : \neg \textit{sp\_engl(x)}}{\neg \textit{sp\_engl(x)}}$

F = {*quebecian(Pierre)*, ∀ x *quebecian (x)* ⇒ canadian*(x)*}

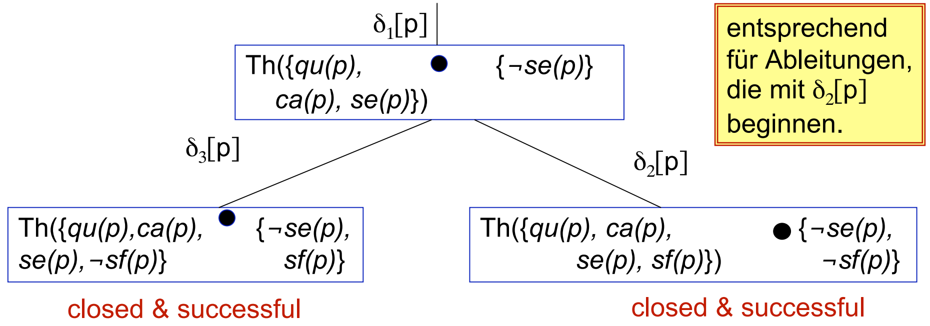


Das Verhinderungsproblem (preemption problem): Beispiel (2) – Fortsetzung

D $\delta_1 = \frac{\textit{canadian(x) : speaks\_english(x)}}{\textit{speaks\_english(x)}}$ $\delta_2 = \frac{\textit{quebecian(x) : speaks\_french(x)}}{\textit{speaks\_french(x)}}$

$\delta_3 = \frac{\textit{sp\_engl(x)} : \neg \textit{sp\_french(x)}}{\neg \textit{sp\_french(x)}}$ $\delta_4 = \frac{\textit{sp\_french(x)} : \neg \textit{sp\_engl(x)}}{\neg \textit{sp\_engl(x)}}$

F = {*quebecian(Pierre)*, ∀ x *quebecian (x)* ⇒ canadian*(x)*}



Verhinderungsproblem

Präemptionsgesichtspunkte können bei Verwendung normaler Defaults nicht adäquat berücksichtigt werden.

- Es werden „zu viele" Extensionen bereitgestellt.
- Die Extensionen stellen gleichwertige Alternativen dar.

- Nicht normale Defaults würden andere Extensionsberechnungen ermöglichen, z.B.

$$\frac{\textit{\textbf{canadian(x)}} : \neg \textit{\textbf{quebecian(x)}}}{\textit{\textbf{speaks\_english(x)}}}$$

Prioritäten zwischen Defaults

Die Basisidee von Preemption:

Spezifischere Knoten der Vererbungshierarchien sind weniger spezifischen Knoten in ihrem Beitrag zum Schließen vorzuziehen.

Definition:

Eine Default-Theorie mit Prioritäten ist ein Tripel $\Delta = \langle F, D, \prec \rangle$, wobei $\Delta = \langle F, D \rangle$ eine Default-Theorie ist, und $\prec$ eine strikte, partielle Ordnung über der Menge *D* der Defaults ist.

Literatur

zu Prioritäten zwischen Defaults:

James Delgrande & Torsten Schaub. (2001). The Role of Default Logic in Knowledge Representation. In J. Minker (ed.), *Logic-Based Artificial Intelligence*, (pp. 107-126), Kluwer: Dordrecht.

Gerhard Brewka & Thomas Eiter (2000). Prioritizing Default Logic. In *Intellectics and Computational Logic, Papers in Honor of Wolfgang Bibel*, (pp. 27-45). Kluwer Academic Publishers: Dordrecht.

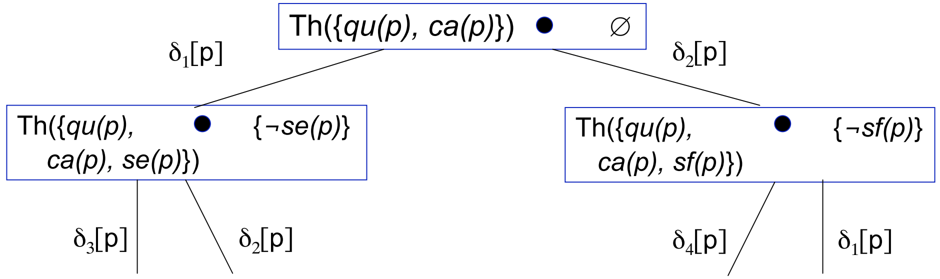
Priorisierungen in Default-Theorien: Beispiel

D $\delta_1 = \dfrac{canadian(x) :\ speaks\_english(x)}{speaks\_english(x)}$ $\delta_2 = \dfrac{quebecian(x) : speaks\_french(x)}{speaks\_french(x)}$

$\delta_3 = \dfrac{sp\_engl(x) : \neg\ sp\_french(x)}{\neg\ sp\_french(x)}$ $\delta_4 = \dfrac{sp\_french(x) : \neg\ sp\_engl(x)}{\neg\ sp\_engl(x)}$

F = {*quebecian(Pierre),* $\forall$ x *quebecian (x)* $\Rightarrow$ canadian*(x)*}

$\prec$: $\delta_3,\ \delta_4 \prec \delta_2 \prec \delta_1$



Priorisierungen in Default-Theorien

Alternative Realisierungen.

- Priorisierungen schränken die Zulässigkeit von Ableitungen ein:
  Ableitungen müssen die Ordnung über den Defaults berücksichtigen.
- Priorisierungen führen zu einer Ordnung auf der Menge der Extensionen:
  - Ableitungen können bewertet werden im Hinblick auf die Anzahl der Anordnungsverletzungen.
  - Extensionen können bewertet werden, indem die Bewertung der erzeugenden Ableitung verwendet wird.

Argumentieren mit Defaults

Skeptische Argumentation (sceptical argumentation)

- Die Default-Theorie $\Delta = \langle F, D\rangle$ liefert eine skeptische Rechtfertigung einer Formel α, wenn α in allen Extensionen zu Δ enthalten ist.
- Beispiel Auto-Diamond: *discussCar(Pete)*

Mutige Argumentation (brave argumentation)

- Die Default-Theorie $\Delta = \langle F, D\rangle$ liefert eine mutige Rechtfertigung einer Formel α, wenn α in einer Extension zu Δ enthalten ist. [Vorsicht: Orthogonalität der Extensionen bei normalen Default-Theorien]
- Beispiel Auto-Diamond: *discussCar(Pete), likesCar(Pete)*

➢ **Schließen über Extensionen vs. Schließen in Extensionen.**

Wissensbasiertes System mit priorisierender Default Logik