Nichtmonotone Vererbung und Default Logic

Wolfgang May

Freiburg, 6.12.2001

Nonmonotonic Reasoning

- seit späte 70er: Konzepte und Theoretische Untersuchungen
- 80er: Wissensrepräsentation
- 90er: Objektorientiertes Datenmodell
- heute: XML: Default-Attribute

Nichtmonotone Vererbung

- Vögel fliegen.
- Tweety ist ein Vogel.

Tweety fliegt

Nichtmonotone Vererbung

- Vögel fliegen.
- Tweety ist ein Vogel.

$\mathcal{D} \vdash \alpha$

Tweety fliegt

- Vögel fliegen.
- Pinguine fliegen nicht.
- Tweety ist ein Vogel.

Tweety fliegt

Nichtmonotone Vererbung

- Vögel fliegen.
- Tweety ist ein Vogel.

Tweety fliegt

$\mathcal{D} \vdash \alpha$

- Vögel fliegen.
- Pinguine fliegen nicht.
- Tweety ist ein Vogel.
- Tweety ist ein Pinguin!

Tweety fliegt nicht!

$\mathcal{D} \cup \mathcal{D}^{+} \nvdash \alpha$

Problematik

bird(X) $\rightarrow$ flies(X)

penguin(X) $\rightarrow$ not_flies(X)

bird(tweety)

bird(lora)

---

flies(tweety)

flies(lora)

Problematik

$\text{bird}(X) \rightarrow \text{flies}(X)$

$\text{penguin}(X) \rightarrow \text{not\_flies}(X)$

bird(tweety)

bird(lora)

penguin(tweety)

---

flies(tweety)

flies(lora)

not_flies(tweety)

inkonsistent

Problematik

~~bird(X) $\rightarrow$ flies(X)~~

penguin(X) $\rightarrow$ not_flies(X)

bird(tweety)

bird(lora)

penguin(tweety)

~~flies(tweety)~~

~~flies(lora)~~

not_flies(tweety)

Problematik

~~bird(X) → flies(X)~~
penguin(X) → not_flies(X)
bird(tweety)
bird(lora)
penguin(tweety)

~~flies(tweety)~~
~~flies(lora)~~
not_flies(tweety)

bird(X) ∧ ¬ penguin → flies(X)
penguin(X) → not_flies(X)
bird(tweety)
penguin(tweety)
bird(lora)

---

flies(lora)
not_flies(tweety).

Problematik

~~bird(X) → flies(X)~~

penguin(X) → not_flies(X)

bird(tweety)

bird(lora)

penguin(tweety)

~~flies(tweety)~~

~~flies(lora)~~

not_flies(tweety)

bird(X) ∧ ¬ penguin → flies(X)

penguin(X) → not_flies(X)

bird(tweety)

penguin(tweety)

bird(lora)

---

flies(lora) (?)

not_flies(tweety).

wissen wir, ob Lora kein Pinguin ist?

CWA

Inheritance Nets

- “Direkter”, grafischer Formalismus

Pfad: tweety – penguin – bird – fly

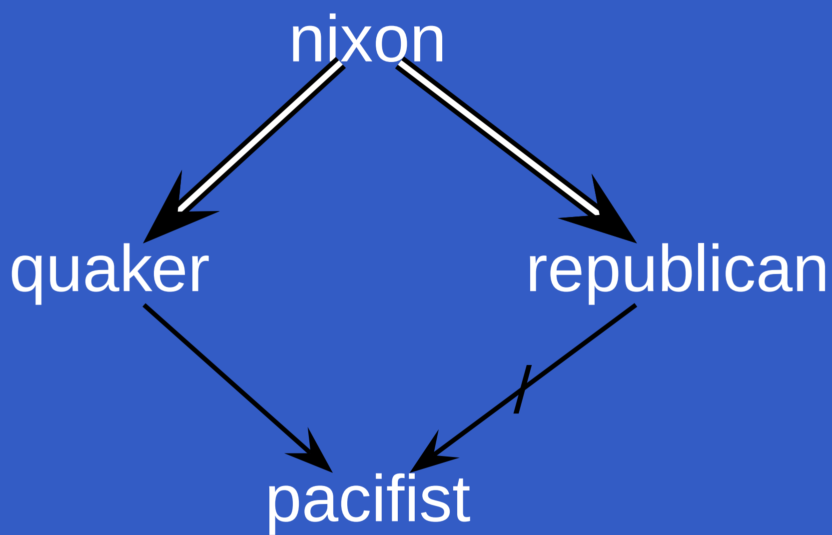
ist “preempted” durch tweety – penguin $\not\rightarrow$ fly

Pfad: lora – bird – fly

- *Extension* als theoretisches Modell-Konzept

Beispiel

Konflikt:



- zwei mögliche *Extensionen*

Default Logic

Erweitert First-Order Logic um zusätzliche “weiche” Schlussregeln [R.Reiter, AI 1980]

$$d = \frac{\alpha(\overline{x}) : \beta(\overline{x})}{w(\overline{x})} \qquad\qquad \frac{bird(x) : fly(x)}{fly(x)}$$

- *precondition* $p(d) = \alpha(\overline{x})$
- *justification* $J(d) = \beta(\overline{x}) = \{\beta_1(\overline{x}), \ldots, \beta_k(\overline{x})\}$
- *consequence* $c(d) = w(\overline{x})$
- α, β, w beliebige First-Order Formeln
- Ist $\alpha(\overline{c})$ beweisbar und die Annahme $\beta(\overline{c})$ konsistent, kann $w(\overline{c})$ daraus geschlossen werden.

CWA: Implizites Negatives Wissen

- Datenbanken: nur positive Fakten gespeichert
  - CWA [Reiter 1978] ist ein negativer Default ohne Vorbedingung:

$$\frac{: \neg p(x_1, \ldots, x_n)}{\neg p(x_1, \ldots, x_n)}$$

  - negatives Wissen muss nicht explizit repräsentiert werden.
- Logic Programming: Negation as failure

Default Proofs

Eingabe: eine Menge D von Defaults und eine Menge S von Formeln ("Situation").
Ein Default Proof einer Formel γ bzgl. D und S ist eine Folge $d_1, \ldots, d_n$ von Defaults wenn

- für $i = 1 \ldots n$:
  - $p(d_i) \in \mathrm{Th}(S \cup c(\{d_1, \ldots, d_{i-1}\}))$
  - $\mathrm{Th}(S \cup \{J(d_i)\})$ ist konsistent
- $\gamma \in \mathrm{Th}(S \cup c(\{d_1, \ldots, d_n\}))$

- Aber:
  - $S \cup \{c(d_1), \ldots, c(d_n)\}$ kann inkonsistent sein
  - $J(d_i)$ kann mit vorhergehenden oder nachfolgenden $c(d_i)$ und $J(d_i)$ inkonsistent sein

Extensionen

- Gegeben: $\Delta = (D, F)$ wobei D eine Menge von Defaults und F eine Menge von geschlossenen Formeln.
- Sei S eine Menge geschlossener Formeln.
- $\Gamma(S)$ minimal so dass
  - $F \subseteq \Gamma(S)$
  - $\mathrm{Th}(\Gamma(S)) = \Gamma(S)$ (deduktiv abgeschlossen)
  - für alle γ, für die es einen Default Proof bzgl. Δ und S gibt, ist $\gamma \in \Gamma(S)$ (Abschluss gg. D).
  - $\Gamma(S)$ kann inkonsistent sein
- eine Menge E geschlossener Formeln ist eine *Extension* von Δ, falls $\Gamma(E) = E$.

Kriterium ist nicht konstruktiv.

Alternative Charakterisierung

[Reiter AI 1980] $\Delta = (D, F)$.
$S_0 = F, S_1, S_2, \ldots$ eine Folge von Mengen von Formeln so dass
$S = (\bigcup_{i=0}^{\infty} S_i)$ und

$$S_{i+1} = S_i \cup c(GD(S_i, S, D)) \ ,$$

$GD(S_i, S, D) := \{d \mid d$ ist eine Instanz eines Defaults in D,
$\mathrm{Th}(S_i) \models p(d)$ und $\mathrm{Th}(S \cup J(d))$ ist konsistent$\}$

Dann ist $\mathrm{Th}(S)$ eine *Extension* von Δ.

- “quasi-induktive” Charakterisierung

Eigenschaften

- Im allgemeinen besitzt Δ mehrere Extensionen
- für jedes γ gibt $S_0, S_1, \ldots$ eine “Herleitung” (“Default Proof”)
- Fragestellungen:
  - credulous: ist eine Formel/ein Faktum in irgendeiner Extension zu Δ enthalten?
  - sceptical: ist eine Formel/ein Faktum in jeder Extension zu Δ enthalten?
  - safe: besitzt eine Formel/ein Faktum in jeder Extension dieselbe Herleitung?

Eigenschaften (Cont'd)

- Alle wesentlichen Fragen sind Σ_2^P- oder Π_2^P-vollständig. Genauer: NP, wenn man ein SAT-Orakel verwendet
- Gibt es eine Extension von Δ, die γ enthält?
  - nicht semi-entscheidbar, d.h., die Menge aller Formeln die in irgendeiner Extension gelten ist nicht rekursiv aufzählbar.
- Aussagenlog. Default-Theorien ohne Disjunktion: NP [Kautz, Selman AI 91]
- Sceptical semantics: ist keine Extension, erfüllt *cumulative monotony* nicht.

Approximation durch Bottom-up-Iteration

$S_0 = F, S_1, S_2, \ldots$ eine Folge von Mengen von Formeln so dass $S = (\bigcup_{i=0}^{\infty} S_i)$ und

$$S_{i+1} = S_i \cup C_i \text{ wobei } C_i \subseteq c(GD(S_i, D)) \ ,$$

$$GD(S_i, D) := \{d \mid d \text{ ist eine Instanz eines Defaults in } D, \\ \mathrm{Th}(S_i) \models p(d) \text{ und } \mathrm{Th}(S_i \cup J(d)) \text{ ist konsistent}\}$$

Betrachte dann $\mathrm{Th}(S)$.

Bottom-Up

Risiko: Justification wird in einem späteren Schritt ungültig
Abhilfe:

- Mitführen der verwendeten Justifications [Brewka AI 1991]
- geeignete Struktur der Defaults

Bottom-Up

Risiko: Justification wird in einem späteren Schritt ungültig
Abhilfe:

- Mitführen der verwendeten Justifications [Brewka AI 1991]
- geeignete Struktur der Defaults

Normale Defaults

Form: $\frac{\alpha : w}{w}$ $\qquad$ $\frac{bird(x) : fly(x)}{fly(x)}$

- Jede normale Default-Theorie hat (mindestens) eine Extension.
- Jede Extension läßt sich bottom-up berechnen

Vererbung und Datenbanken

- Default Logic: Mengen von Formeln
- Deduktive, objektorientierte Datenbanken
  - Menge von (nur positiven) Fakten
  - Regeln *head* $\leftarrow$ *body*
  - Vererbung innerhalb der Klassenhierarchie
- bottom-up Auswertung, T_P-Operator ersetzt Theoriebildung
- dazu passen nur spezielle “Horn-style” Defaults
  - α, β vorzugsweise Literale (oder einfache Formeln)
  - w Atome
  - Horn Default-Theorien sind polynomial [Kautz, Selman AI 91]

Beispiel

penguin subcl *bird*

tweety isa *penguin*

$$d_1 : \frac{X \text{ isa } penguin : X[flies \rightarrow false]}{X[flies \rightarrow false]} \qquad d_2 : \frac{X \text{ isa } bird : X[flies \rightarrow true}{X[flies \rightarrow true]}$$

- Zwei mögliche Extensionen
  - Anwendung von d_1: $tweety[flies \rightarrow false]$
  - Anwendung von d_2: $tweety[flies \rightarrow true]$
    intuitiv nicht beabsichtigt
    Preemption wird nicht berücksichtigt!

Präferenzen

- “Ranked” Defaults
- Spezifischere Defaults automatisch bevorzugen
  [Poole IJCAI 85]
- hier: Klassenhierarchie
  “Wenn es keine dazwischenliegende Klasse gibt”

Defaults für Vererbung

Default Schemata:

$$\frac{X \text{ isa } C, C[M \bullet\!\!\rightarrow V] : X[M \rightarrow V] \wedge \neg \exists SC : (X \text{ isa } SC \wedge SC \text{ subcl } C)}{X[M \rightarrow V]}$$

Analog für Vererbung zu Subklassen.

- Bottom-up-Approximation
  - Risiko: Zwischenklassen, die erst in einem späteren Schritt abgeleitet werden und dem Default nicht entsprechen

Defaults für Vererbung

Default Schemata:

$$\frac{X \text{ isa } C, C[M \bullet\!\!\rightarrow V] : X[M \rightarrow V] \wedge \neg\exists SC : (X \text{ isa } SC \wedge SC \text{ subcl } C)}{X[M \rightarrow V]}$$

$$\neg(\exists SC : (X \text{ isa } SC \wedge SC \text{ subcl } C \wedge \neg SC[M \bullet\!\!\rightarrow V]))$$

Analog für Vererbung zu Subklassen.

- Bottom-up-Approximation
  - Risiko: Zwischenklassen, die erst in einem späteren Schritt abgeleitet werden und dem Default nicht entsprechen
  - kein Problem bei statischer Klassenhierarchie

Konflikte

Konflikt: *konsistente Teilmenge* anwenden.

Konflikte

Konflikt: *konsistente Teilmenge* anwenden.

Nixon Diamond:

$P = \{$ quaker[policy$\bullet\!\!\rightarrow$pacifist], republican[policy$\bullet\!\!\rightarrow$hawk],

nixon isa quaker, nixon isa republican$\}$.

$$S_0 = T_P^{\omega}(P) = P$$

$$GD(S_0, D) = \left\{ \begin{array}{l} \dfrac{\text{nixon isa quaker, quaker[policy}\bullet\!\!\rightarrow\text{pacifist]: nixon[policy}\rightarrow\text{pacifist]}}{\text{nixon[policy}\rightarrow\text{pacifist]}} \\[2ex] \dfrac{\text{nixon isa republican, quaker[policy}\bullet\!\!\rightarrow\text{hawk]: nixon[policy}\rightarrow\text{hawk]}}{\text{nixon[policy}\rightarrow\text{hawk]}} \end{array} \right.$$

Komplexität

- es genügt, in jedem Schritt einen anwendbaren Default anzuwenden
- ohne Objektgenerierung: eine Extension wird in polynomialer Zeit berechnet
- credulous/sceptical/safe: alle Extensionen müssen berechnet werden

Fazit:

- Anwendung von Defaults auf Vererbung in objektorientierten Datenbanken sinnvoll
- auch für XML vielversprechend

Kritik

- “sceptical” Semantik zu restriktiv:
  bereits ein einziger Default (z.B. $\frac{:\neg a}{a}$) kann dazu führen, dass keine Extension existiert
- “sceptical” Semantik ist *keine* Extension
- “sceptical” Semantik ist nicht kumulativ,erfüllt “Or/Distribution” nicht.

Weitere Aspekte

- Suche nach “besseren” Semantiken
- Logic Programming mit Negation und Default Logic
  - Übersetzung
- Metatheoretische Eigenschaften nichtmonotoner Systeme
  - Default-Logic verhält sich ziemlich “unerwünscht” (nicht kumulativ,erfüllt “Or” nicht)
  - Default-Logic ist nicht “Rational”
  - Anwendung für Vererbung in Bottom-up Evaluierung für Grundinstanzen ist problemlos

Die Suche nach “besseren” Semantiken

- gesucht: kumulative Semantik

Betrachte $\Gamma(S)$ imNixon-Diamond

- $\Gamma(\emptyset) = P \cup \{\text{nixon isa pacifist, nixon isa hawk}\} = E_{cred}$
- $\Gamma(\Gamma(\emptyset)) = P = E_{scept}$
- alternierend

Interpretationsmöglichkeiten:

- Dreiwertige Semantik wie für LP: dreiwertige Extensionen [Przymusinski AI 1991]
  (LP: 3-wertige Default-Semantik+CWA äquivalent zu WFS)

Stationary Default Extensions

Andere Interpretationsmöglichkeit:
"Stationary Default Extensions" [Przymusinska/-ski FI 94]

- Γ ist antimonoton, $\Gamma \circ \Gamma$ ist monoton
- Bedingung: $\Gamma(\Gamma(E)) = E$
- Extensionen $\subsetneq$ stationäre Extensionen
- kleinste stationäre Extension iterativ berechenbar: $\emptyset, \Gamma^2(\emptyset), \Gamma^4(\emptyset), \ldots, \Gamma^{2n}(\emptyset), \ldots$
- endlich falls Δ endlich und keine Funktionssymbole
- n Defaults, m Justifications $\rightarrow O(n^2 \cdot m)$ Erfüllbarkeitstests/Γ-Schritt,
  sinnvoll für Sprachen wo Erfüllbarkeit polynomiell
- "sceptical" erfüllt Kumulativität

Stärkere zweiwertige Semantiken

In vielen Fällen ist stationäre Semantik zu streng:
"Ungerade" Γ-Anwendung akzeptiertzu viel.
(Γ ist noch großzügiger als die bottom-up Approximation)

Stärkere zweiwertige Semantiken

In vielen Fällen ist stationäre Semantik zu streng:
"Ungerade" Γ-Anwendung akzeptiertzu viel.
(Γ ist noch großzügiger als die bottom-up Approximation)

- Γ basiert auf Default Proofs
- Default Proof kann inkonsistent sein
- Default Proof kann inkonsistente Justifications benutzt haben
  - einzeln inkonsistent zum Ergebnis
  - es gibt keine Extension, in der die verwendeten Justifications *gleichzeitig* zutreffen

Stärkere zweiwertige Semantiken

Strengere Γ_i-Operatoren [Brewka, Gottlob FI 1997]

- Bedingung: $\Gamma(\Gamma_i(E)) = E$
- $\Gamma_1(S) = \{p \mid \text{es gibt einen Default Proof für } p\}$
- Γ_2: Default Proof ist konsistent
- Γ_3: Justifications sind konsistent
- Γ_4: Default Proof muss mit einer Extension konsistent, d.h., nachvollziehbar sein

- Betrachte Fixpunkte von $\Gamma\Gamma_i$.
- $$\begin{aligned} WFS &= (\Gamma\Gamma_1)^\omega(\emptyset) \subseteq (\Gamma\Gamma_2)^\omega(\emptyset) \\ &\subseteq (\Gamma\Gamma_3)^\omega(\emptyset) \subseteq (\Gamma\Gamma_4)^\omega(\emptyset) = safe \end{aligned}$$

Gammas-Hierarchie

$$\bigcup(\text{Extensionen}) = \Gamma_4 \leq \Gamma_3 \leq \Gamma_2 \leq \Gamma_1 = \Gamma$$

$$\Gamma_4 \leq \bigcup(\text{bottom-up}^{\omega}) \leq \Gamma_2$$

$$WFS_{\Gamma_1}(\Delta) \subseteq WFS_{\Gamma_2}(\Delta) \subseteq WFS_{\Gamma_3}(\Delta) \subseteq WFS_{\Gamma_4}(\Delta) = Safe(\Delta)$$

Für normale Default-Theorien:

$\Gamma_2 = \Gamma_3 = \Gamma_4 = \bigcup(\text{bottom-up})^{\omega} = \bigcup(\text{Extensionen})$

Gammas-Hierarchie

- $\Gamma = \Gamma_1$: alles was irgendwie begründet werden kann.
- Γ_2: abgeleitete Formeln müssen konsistent sein.
- Γ_3: Justifications müssen mit abgeleiteten Formeln konsistent sein.
- bottom-up Iteration: Default-Proof muss in einer Extension nachvollziehbar sein – allerdings können Justifications später verloren gehen.
- Γ_4: Default-Proof muss in einer Extension nachvollziehbar sein.
- $\bigcup(\text{Extensionen}) = \Gamma_4$
- für normale Defaults: $\bigcup(\text{bottom-up})$, Γ_2, Γ_3, Γ_4 und $\bigcup(\text{Extensionen})$ äquivalent.

Weitere Aspekte

- Logic Programming mit Negation und Default Logic
  - Übersetzung
- Metatheoretische Eigenschaften nichtmonotoner Systeme
  - Default-Logik verhält sich ziemlich “unerwünscht” (nicht kumulativ, erfüllt “Or” nicht)
  - Default-Logic ist nicht “Rational”
  - Anwendung für Vererbung in Bottom-up Evaluierung für Grundinstanzen ist problemlos

Questions ??

LP mit Negation und Default Logic

Formulierung von LP in Default-Logic $P \mapsto \Delta(P)$
[Przymusinski 1988, Bidoit, Froidevaux I&C 1991]

$$p(\overline{x}) \ :- \ a_1(\overline{x}) \wedge \ldots \wedge a_n(\overline{x}), \neg b_1(\overline{x}) \wedge \ldots \wedge \neg b_m(\overline{x}).$$

$$\frac{a_1(\overline{x}) \wedge \ldots \wedge a_n(\overline{x}) : \neg b_1(\overline{x}) \wedge \ldots \wedge \neg b_m(\overline{x})}{p(\overline{x})}$$

Beispiel:

$$\frac{move(X,Y) : \neg win(Y)}{win(X)}$$

- stabile Modelle entsprechen Extensionen
- kleinstes stationäres Modell entspricht der WFS polynomiell

Default-Theorien als Logic Programs

umgekehrte Richtung ...

- Übersetzung von Default-Theorien in LPs [Li, You JCI 1991]

Defaults vs. Implikation

- Implikation: $(X \text{ isa } bird) \rightarrow fly(X)$
  Konsequenz: $\neg fly(tweety) \rightarrow \neg(tweety \text{ isa } bird)$

- Default: $\dfrac{bird(x) : fly(x)}{fly(x)}$

  Konsequenz: $\neg fly(tweety)$ bedeutet nur, dass der Default auf Tweety nicht anwendbar ist
  [vgl. Poole AI 1988 (Theorist)]

Generating Defaults

- S eine Menge geschlossener Formeln.

$$GD(S,D) := \{d \mid d \text{ ist eine Instanz eines Defaults in } D, \text{Th}(S) \models p(d) \text{ und } \text{Th}(S \cup J(d)) \text{ ist konsistent}\}$$

  - $GD(S,D)$ kann Konflikte enthalten.

Gammas-Hierarchie

- Skeptical reasoning in Reiter's Default Logik: Π_2^P-vollständig.
- $\Gamma = \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$: WFS Π_2^P-hart.
- Γ_4: WFS Σ_3^P-vollständig.

Gammas-Hierarchie

- Skeptical reasoning in Reiter's Default Logik: Π_2^P-vollständig.
- $\Gamma = \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$: WFS Π_2^P-hart.
- Γ_4: WFS Σ_3^P-vollständig.

wobei

- Π_2^P: polynomiell nichtdeterministisch lösbar, wenn man ein NP-Orakel hat
- Σ_2^P: Probleme, deren Komplement in Π_2^P ist.

Beispiel für stationäre Extension

[Przymusinska, Przymusinski FI 1994]

$$sleep \leftarrow \neg work. \qquad \frac{: \neg work}{sleep}$$

$$work \leftarrow \neg tired. \qquad \frac{: \neg tired}{work}$$

$$tired \leftarrow \neg sleep. \qquad \frac{: \neg sleep}{tired}$$

$$paid.$$

$$angry \leftarrow work, \neg paid \qquad \frac{work : \neg paid}{angry}$$

analog: Nixon Diamond

Beispiel für stationäre Extension (Cont'd)

- mehrere Extensionen $\{paid, \neg angry, sleep\}$, $\{paid, \neg angry, work\}$, $\{paid, \neg angry, tired\}$
- Sceptical Semantics: $E_{scept} = \{paid, \neg angry\}$
- ist keine Extension
- $\Gamma(\emptyset) = \{paid, \neg angry, sleep, work, tired\}$
- $\Gamma(\{paid, \neg angry, sleep, work, tired\}) = E_{scept}$
- $\Gamma^2(E_{scept}) = E_{scept}$

Beispiel zu WFS-2

$$D = \left\{ \frac{: b}{b}, \frac{: a}{a}, \frac{: \neg a}{\neg a} \right\}$$

Beispiel zu WFS-2

$$D = \left\{ \frac{:b}{b}, \frac{:a}{a}, \frac{:\neg a}{\neg a} \right\}$$

$\Gamma(\emptyset) = \mathrm{Th}(\{b, a, \neg a\}) = Lang$
$\Gamma_2(\emptyset) = \mathrm{Th}(\{b, a\}) \cup \mathrm{Th}(\{b, \neg a\})$
enthält $\neg b$ nicht.
Damit ist die Annahme von b konsistent:
$\Gamma(\Gamma_2(\emptyset)) = \mathrm{Th}(\{b\})$
Fixpunkt.

Beispiel zu Gamma-Hierarchie

$$D = \left\{ \frac{:a}{a}, \frac{:\neg a}{\neg a}, \frac{:b}{c}, \frac{:a}{d}, \frac{\neg a : b}{\neg b} \right\}$$

Beispiel zu Gamma-Hierarchie

$$D = \left\{ \frac{:a}{a}, \frac{:\neg a}{\neg a}, \frac{:b}{c}, \frac{:a}{d}, \frac{\neg a : b}{\neg b} \right\}$$

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$\Gamma_i(\emptyset)$	$Lang$	$\mathrm{Th}(\{a,c,d\}) \cup \mathrm{Th}(\{\neg a, c, d, \neg b\})$	$\mathrm{Th}(\{a,c,d\}) \cup \mathrm{Th}(\{\neg a, c\})$	$\mathrm{Th}(\{a,c,d\})$
$\Gamma\Gamma_i(\emptyset)$	$\mathrm{Th}(\emptyset)$	$\mathrm{Th}(\emptyset)$	$\mathrm{Th}(\{c\})$	$\mathrm{Th}(\{a,c,d\})$

WFS$_i$ für Logic Programming

$$P = \{\ a \leftarrow \neg d \qquad c \leftarrow \neg b \qquad b \leftarrow \neg b, d$$
$$d \leftarrow \neg a \qquad f \leftarrow \neg d\}$$

- Stabiles Modell: $\{a, c, f\}$
- $WFS(P) = WFS_2(P) = \mathrm{Th}(\emptyset)$
- $WFS_3(P) = \mathrm{Th}(\{c\})$
- $WFS_4(P) = \mathrm{Th}(\{a, c, f\})$

WFS$_i$ für Logic Programming

$$P = \{ \begin{array}{lll} a \leftarrow \neg d & c \leftarrow \neg b & b \leftarrow \neg b, d \\ d \leftarrow \neg a & f \leftarrow \neg d \} & \end{array}$$

- Stabiles Modell: $\{a, c, f\}$
- $WFS(P) = WFS_2(P) = \mathrm{Th}(\emptyset)$
- $WFS_3(P) = \mathrm{Th}(\{c\})$ (b-Regel ist self-defeating)
- $WFS_4(P) = \mathrm{Th}(\{a, c, f\})$

Metatheoretische Eigenschaften

[Kraus,Lehmann,Magidor AI 1990]
Eigenschaften von Konsequenzrelationen:
Was soll man aus einer Menge von "$\alpha \mid\!\sim \beta$" schließen?
Ganz notwendig:

- Reflexivity: $\alpha \mid\!\sim \alpha$
- Left Logical Equivalence: $\dfrac{\models \alpha \leftrightarrow \beta\ ,\ \alpha \mid\!\sim \gamma}{\beta \mid\!\sim \gamma}$
- Right Weakening: $\dfrac{\models \alpha \rightarrow \beta\ ,\ \gamma \mid\!\sim \alpha}{\gamma \mid\!\sim \beta}$

Kumulativität

System **C** (Cumulative): Eigenschaften aus [Gabbay 1985]

- Cut: $$\frac{\alpha \wedge \beta \mathrel{|\!\sim} \gamma\ ,\ \alpha \mathrel{|\!\sim} \beta}{\alpha \mathrel{|\!\sim} \gamma}$$

- Weak Monotonicity/Cautious Monotonicity/Cumulative Monotonicity: $$\frac{\alpha \mathrel{|\!\sim} \beta\ ,\ \alpha \mathrel{|\!\sim} \gamma}{\alpha \wedge \beta \mathrel{|\!\sim} \gamma}$$

Kumulativität

System **C** (Cumulative): Eigenschaften aus [Gabbay 1985]

- Cut: $\dfrac{\alpha \wedge \beta \mid\!\sim \gamma \,,\; \alpha \mid\!\sim \beta}{\alpha \mid\!\sim \gamma}$

  Um $\alpha \mid\!\sim \gamma$ zu zeigen, kann man temporär β dazunehmen, wenn ...

- Weak Monotonicity/Cautious Monotonicity/Cumulative Monotonicity: $\dfrac{\alpha \mid\!\sim \beta \,,\; \alpha \mid\!\sim \gamma}{\alpha \wedge \beta \mid\!\sim \gamma}$

  Wenn man β erfährt und es vorher schon geglaubt hat, bleiben alle Schlüsse gültig

Beide zusammen:
Wenn $\alpha \mid\!\sim \beta$, dann stimmen die Schlüsse aus α und $\alpha \wedge \beta$ überein.

Metatheoretische Eigenschaften (Cont'd)

- "Sceptical" Default Semantik:
  - erfüllt Cut [Makinson NMR 89]
  - nicht kumulativ [Makinson NMR 89]; auch nicht für normale Defaults
  - erfüllt "Or" nicht
- WFS ist kumulativ, WFS_2, WFS_3 und WFS_4 sind nicht kumulativ.

System C (Cont'd)

Abgeleitete Regeln für System **C**:

- Equivalence: $$\frac{\alpha \mathrel{|\!\sim} \beta \,,\; \beta \mathrel{|\!\sim} \alpha \,,\; \alpha \mathrel{|\!\sim} \gamma}{\beta \mathrel{|\!\sim} \gamma}$$
- And: $$\frac{\alpha \mathrel{|\!\sim} \beta \,,\; \alpha \mathrel{|\!\sim} \gamma}{\alpha \mathrel{|\!\sim} \beta \wedge \gamma}$$
- Modus Ponens Cumulative: $$\frac{\alpha \mathrel{|\!\sim} \beta \rightarrow \gamma \,,\; \alpha \mathrel{|\!\sim} \beta}{\alpha \mathrel{|\!\sim} \gamma}$$
- schwache Transitivität: $$\frac{\alpha \vee \beta \mathrel{|\!\sim} \alpha \,,\; \alpha \mathrel{|\!\sim} \gamma}{\alpha \vee \beta \mathrel{|\!\sim} \gamma}$$

System CM

System **CM** (Schwächer als Classical Monotonic Logic):
(abgelehnt für nichtmonotone Systeme)

- Monotonicity: $\dfrac{\models \alpha \rightarrow \beta \,,\; \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma}$

- Easy Half of Deduction Theorem: $\dfrac{\alpha \mid\!\sim \beta \rightarrow \gamma}{\alpha \wedge \beta \mid\!\sim \gamma}$

- Transitivität: $\dfrac{\alpha \mid\!\sim \beta \,,\; \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma}$

- Contraposition: $\dfrac{\alpha \mid\!\sim \beta}{\neg\beta \mid\!\sim \neg\alpha}$

System CM

System **CM** (Schwächer als Classical Monotonic Logic):
(abgelehnt für nichtmonotone Systeme)

- Monotonicity: $\dfrac{\models \alpha \rightarrow \beta \ , \ \beta \mathrel{|\!\sim} \gamma}{\alpha \mathrel{|\!\sim} \gamma}$

  $penguin \rightarrow bird \ , \ bird \mathrel{|\!\sim} flies$

- Easy Half of Deduction Theorem: $\dfrac{\alpha \mathrel{|\!\sim} \beta \rightarrow \gamma}{\alpha \wedge \beta \mathrel{|\!\sim} \gamma}$

- Transitivität: $\dfrac{\alpha \mathrel{|\!\sim} \beta \ , \ \beta \mathrel{|\!\sim} \gamma}{\alpha \mathrel{|\!\sim} \gamma}$

- Contraposition: $\dfrac{\alpha \mathrel{|\!\sim} \beta}{\neg\beta \mathrel{|\!\sim} \neg\alpha}$

Präferentielle Systeme

System **P** (Preferential): **C** und

- Or: $\dfrac{\alpha \mid\!\sim \gamma \,,\; \beta \mid\!\sim \gamma}{\alpha \vee \beta \mid\!\sim \gamma}$

Ableitbar:

- Hard Half of Deduction Theorem: $\dfrac{\alpha \wedge \beta \mid\!\sim \gamma}{\alpha \mid\!\sim \beta \rightarrow \gamma}$
- Proof by Cases/Distribution: $\dfrac{\alpha \wedge \beta \mid\!\sim \gamma \,,\; \alpha \wedge \neg\beta \mid\!\sim \gamma}{\alpha \mid\!\sim \gamma}$

Präferentielle Modelle und Hülle

[Shoham LICS 87; Kraus, Lehmann, Magidor AI 90]
Partiell geordnete Menge von Strukturen/Theorien ("Welten"), die angibt, welche "mehr normal" als andere sind.

- $\alpha \mid\!\sim \beta$ gilt, wenn alle Welten, die α erfüllen und "am normalsten" (minimal) sind, auch β erfüllen.
- Def: $\alpha \mid\!\sim \beta \in \mathbf{K}^p$ wenn es in allen präferentielle Modellen zu $\mathbf{K}$ gilt.
- Die präferentielle Folgerungsrelation ist co-NP [Lehmann, Magidor AI 1992]

Rationalität

Zusätzlich Aussagen, was *nicht* geschlossen werden soll:

- Negation Rationality: $\dfrac{\alpha \wedge \gamma \not\mathrel{|\!\sim} \beta, \ \alpha \wedge \neg\gamma \not\mathrel{|\!\sim} \beta}{\alpha \not\mathrel{|\!\sim} \beta}$
- Disjunctive Rationality: $\dfrac{\alpha \not\mathrel{|\!\sim} \gamma, \ \beta \not\mathrel{|\!\sim} \gamma}{\alpha \vee \beta \not\mathrel{|\!\sim} \gamma}$
- Rational Monotonicity: $\dfrac{\alpha \wedge \beta \not\mathrel{|\!\sim} \gamma, \ \alpha \not\mathrel{|\!\sim} \neg\beta}{\alpha \not\mathrel{|\!\sim} \gamma}$ äq. $\dfrac{\alpha \mathrel{|\!\sim} \gamma, \ \alpha \not\mathrel{|\!\sim} \neg\beta}{\alpha \wedge \beta \mathrel{|\!\sim} \gamma}$
- R.M. impliziert mit System **C** D.R., und das wiederum N.R.
- “Rational”: Systeme, die Rational Monotonicity erfüllen.
- wenn ein Faktum, dessen Negation vorher nicht abgeleitet werden konnte, dazugelernt wird, wird kein vorheriger Schluss widerrufen

Rationalität (Cont'd)

für rationale Systeme gelten schwächere Formen der bei $\mathbf{C}M$ genannten Regeln:

- Weak Transitivity: $$\frac{\alpha \mathrel{|\!\sim} \beta\,,\ \beta \mathrel{|\!\sim} \gamma\,,\ \beta \mathrel{|\!\not\sim} \neg\alpha}{\alpha \mathrel{|\!\sim} \gamma}$$
- Weak Contraposition: $$\frac{\alpha \wedge \gamma \mathrel{|\!\sim} \beta\,,\ \gamma \mathrel{|\!\not\sim} \beta}{\gamma \wedge \neg\beta \mathrel{|\!\sim} \neg\alpha}$$

Rationale Konsequenzrelationen

[Lehmann, Magidor AI 1992]

- Ranking von Formeln: beschreibt, wie sehr “Ausnahme” eine Formel ist.
- Rationale Konsequenzrelationen können durch “Ranked Models” repräsentiert werden:
- “Ranked Models” sind präferentielle Modelle, deren Ordnung bestimmte Bedingungen erfüllt (kleinerer Rank $\rightarrow$ weniger unnormal).
- trotzdem ist *Ranked Entailment* nur *Preferential entailment*: Schnitt aller rationalen Extensionen ist nur $\mathbf{K}^p$.

Rationale Hülle

[Lehmann, Magidor AI 1992]

- Ordnung auf rationalen Extensionen (nach Normalität)
- falls minimale rationale Extension $\overline{\mathbf{K}}$ existiert, ist das die rationale Hülle
  (existiert wenn ein sinnvolles Ranking der Formeln möglich ist, z.B. für alle endlichen $\mathbf{K}$)
- $\alpha \mid\!\sim \beta \in \overline{\mathbf{K}}$ falls
  - $rank(\alpha) < rank(\alpha \wedge \neg\beta)$, oder
  - $rank(\alpha)$ existiert nicht (dann ist α inkonsistent zu $\mathbf{K}$)

Rationale Hülle: Komplexität

[Lehmann, Magidor AI 1992]

- $\overline{\mathbf{K}}$ iterativ berechenbar aus $\mathbf{K}$, indem man solange $E(C)$ (Menge aller Ausnahmeformeln) bildet, bis man bei α ankommt. Dann wird geprüft ob β noch mehr Ausnahme ist.
- Test, ob eine Formel eine Ausnahme beschreibt: reduzierbar auf SAT in der zugrundeliegenden Logik.
- Man braucht $O(n^2)$ Iterationen.
- Horn-Fall: polynomial.

Default-Logic und Kumulativität

Default-Logic erfüllt Kumulativität nicht
[Makinson LPNMR 89]

$$\frac{:p}{p} \quad , \quad \frac{p \vee q : \neg p}{\neg p}$$

- hat genau eine Extension: Th$\{p\}$, enthält also auch $p \vee q$
- nimmt man $p \vee q$ als Prämisse an, bekommt man eine zweite Extension Th$(\{\neg p, q\})$ enthält.

Anderes, normales Beispiel [Makinson Handbook 1994]:

$$\frac{:a}{a} \quad , \quad \frac{a:b}{b} \quad , \quad \frac{b:\neg a}{\neg a}$$

Default-Logic und Kumulativität

Beispiel mit normalen Defaults [Brewka AI 91]

$$F = \{dog \vee bird \rightarrow pet \ , \ dog \rightarrow \neg bird \ , \ sings\}$$
$$D = \left\{ \frac{pet : dog}{dog} \ , \ \frac{sings : bird}{bird} \right\}$$

Extension: Th$(F \cup \{bird\})$ contains pet.
Nimmt man pet zu den Fakten dazu, erhält man eine zusätzliche Extension Th$(F \cup \{dog\})$

Default-Logic und OR

aus [Poole KR89]

$$\frac{: usable(X) \wedge \neg broken(X)}{usable(X)}$$

Prämisse: $broken(left\_arm) \vee broken(right\_arm)$

Die einzige Extension enthält

$$usable(left\_arm) \wedge usable(right\_arm)$$

(jedes $usable(X)$ kann einzeln abgeleitet werden)

- Lösung: Buchführung über verwendete Justifications [Lukaszewicz CI 1988, Brewka AI 1991, Delgrande 1994]
- womit man auch das Problem des bottom-up Verfahrens löst

Default-Logic und OR

[Poole Handbook 1994]

$$\frac{employed(X) : get\_paid(X) \wedge works(X)}{get\_paid(X)}$$

Fakten: $employed(david)$, $employed(john)$,
$\neg works(david) \vee \neg works(john)$

Hier ist es sinnvoll, dass die Extension

$$get\_paid(david) \wedge get\_paid(john)$$

ableitet.

Default-Logic und Proof-by-cases

aus [Makinson Handbook 94]

$$\frac{a : c}{c} \quad , \quad \frac{\neg a : c}{c}$$

Es gilt $a \mid\!\sim c$ und $\neg a \mid\!\sim c$ aber nicht $true \mid\!\sim c$.

Cumulative Default Logic

Buchführung über verwendete Justifications [Brewka AI 1991]:
Formel ϕ kann unter Verwendung der Justifications $\gamma_1, \ldots, \gamma_n$ begründet werden:

$$\langle \phi, \{r_1, \ldots, r_n\} \rangle$$

$S_0 = F, S_1, S_2, \ldots$ eine Folge von Mengen von (annotierten) Formeln so dass $S = (\bigcup_{i=0}^{\infty} S_i)$ und

$$S_{i+1} = S_i \cup \{ \langle C, R \cup \beta \cup \{c(d)\} \rangle \mid d \in D, \mathrm{Th}(S_i) \models p(d) \text{ und } \mathrm{Th}(S \cup Supp(S) \cup J(d) \cup \beta \cup \{c(d)\}) \text{ ist konsistent} \}$$

Dann ist $\mathrm{Th}(S)$ eine *CDL-Extension* von Δ.
Kumulativ, und es existiert immer eine CDL-Extension.

Lokalität normaler Defaults

Form: $\frac{\alpha : w}{w}$

- Jede Extension läßt sich bottom-up berechnen
- Semi-Monotonie: $D \subseteq D'$, E eine Extension von (D, F). Dann hat (D', F) eine Extension E' so dass
  - $E \subseteq E'$
  - $GD(E, D) \subseteq GD(E', D')$

  $\Rightarrow$ “Lokalität”
- Vollständigkeit von Top-Down Default Proofs [Reiter AI 1980].

Seminormale Defaults

Es gibt Dinge, die nicht als normale Defaults ausdrückbar sind:

$$\frac{has\_motive(X) : suspect(X) \wedge guilty(X)}{suspect(X)}$$

Komplexität: Basic Notions

- SAT für propositional Logic ist NP-vollständig
- SAT für first-order ist nicht rekursiv aufzählbar
- $QBF(2,\exists) = SAT(\exists\ldots\exists\forall\ldots\forall\phi)$ ist Σ_2^P-vollständig (d.h., NP-vollst., wenn man auf ein Σ_1^P- oder NP-Orakel zurückgreifen kann).
- $\Sigma_{k+1}^P = NP^{\Sigma_k^P}$, $\Sigma_0^P = P$
- $\Sigma_1^P = NP$

Komplexität von Default Reasoning

Σ_P^2 oder Π_P^2-vollständig sind für endliche aussagenlogische Default-Theorien:

- Existenz einer [konsistenten] Extension.
- gilt γ in [einer|allen] [konsistenten] Extensionen [nicht]?
- dasselbe bereits für normale Defaults ohne Prerequisites.
- "Sceptical" für normale Defaults ist $P^{NP[\log n]}$-vollständig (LFP-Berechnung der stationären Semantik muss nur bis Γ^2 ausgeführt werden) [Gottlob IC 1995]

Default-Theorien ohne Disjunktion: NP
[Kautz, Selman AI 91]
Horn Default Theorien: linear time [Kautz, Selman AI 91]

Beweistheorie

Gibt es eine Extension von Δ, die γ enthält?

1. zeige γ mit $F \cup c(D)$
   (R Grundinstanzen der verwendeten Defaults)
2. zeige alle Preconditions der verwendeten Defaults
   (rekursiv, R^+ alle verwendeten Grundinstanzen)
3. teste Konsistenz von $F \cup J(R^+)$ (SAT, nicht r.a.)

Es gibt keine Prozedur, um das im allgemeinen Fall zu berechnen.

Beweistheorie

Für normale Defaults existiert eine vollständige Beweistheorie [Reiter AI 1980]:

- F, γ in Horn-Form
- jedes $c(D)$ in Horn-Form:
  Menge $(C, \{\delta\})$ von Paaren von annotierten Klauseln
  ($\delta = \emptyset$ für $C \in F$)
- Resolution: (C_1, D_1) (C_2, D_2) zu $(R, D_1 \cup D_2)$

Beweistheorie

Lineare Resolution

- Startklausel: $R_0 =$ eine negierte Klausel in γ
- R_{i-1} mit einem C_i zu R_i, wobei C_i
  - ein $(C, \{d\})$ vom Input
  - eine negierte Klausel in γ
  - ein vorhergehendes R_j.
- $R_n = (\Box, D)$ für eine Menge D_0 von Defaults

- Gezeigt: Aus $c(D_0)$ lässt sich γ ableiten.
- rekursiv: Herleitungen für die Preconditions in D_k suchen, ergibt D_{k+1}.
- Zuletzt bleibt zu zeigen: $F \cup \bigcup_k c(D_k)$ ist erfüllbar (SAT(Horn) ist polynomial)