# VU Einführung in Wissensbasierte Systeme

WS 2010/11

Hans Tompits

Institut für Informationssysteme
Arbeitsbereich Wissensbasierte Systeme

www.kr.tuwien.ac.at

# 5.3.2 Extensions

# Applicability relative to a context

$F$ deductively closed set of formulas
$K$ arbitrary set of formulas (called context)

A default $\delta = \frac{\varphi \ : \ \psi_1, \ldots, \psi_n}{\chi}$ is applicable to $F$ relative to $K$ iff

$$\boxed{\varphi \in F \text{ and } \neg\psi_1, \ldots, \neg\psi_n \notin K}$$

$K = F$: regular applicability of defaults (as defined earlier).

# The Operator $\Gamma_T$

Given: closed default theory $T = (W, \Delta)$, set of closed formulas $S$;

☞ $T$ is closed $:\Longleftrightarrow$ all formulas in $\Delta$ are closed.

We define $\Gamma_T(S)$ as the *smallest set* $F$ of closed formulas such that

1. $F$ is *deductively closed*,
2. $W \subseteq F$, and
3. $F$ is *closed under applications of defaults relative to context* $S$, i.e., for all $\delta = \frac{\varphi : \psi_1, \ldots, \psi_n}{\chi} \in \Delta$ it holds that if $\varphi \in F$ and $\neg\psi_1 \notin S, \ldots, \neg\psi_n \notin S$, then $\chi \in F$.

$E$ is an extension of $T = (W, \Delta)$ iff

$$\boxed{\Gamma_T(E) = E}$$

i.e., iff $E$ is a *fixed point of* $\Gamma_T$.

# The Operator $\Gamma_T$ (ctd.)

Intuitively, $\Gamma_T$ can be seen as a *logical closure operator*, representing a possible totality of knowledge of an agent.

- For constructing $\Gamma_T(E)$, $E$ serves as a "context" for testing the consistency conditions of the defaults in $\Delta$.
- $\Gamma_T(E)$ itself collects all formulas derivable from $W$ by means of
  - classical logic
  - and those defaults in $\Delta$ satisfying the consistency condition relative to $E$.
- A context $E$ is an *extension* of $T$ iff it *reproduces itself* under the closure operator $\Gamma_T$ (i.e., iff it is a fixed point of $\Gamma_T$).

# Extensions: Example

$$T = (\{water\_creature\}, \{\frac{water\_creature : fish}{fish}\})$$

$E = Cn(\{water\_creature, fish\})$ extension of $T$,
$E' = Cn(\{water\_creature, \neg fish\})$ *not* an extension of $T$, although

- $\{water\_creature\} \subseteq E'$,
- $E'$ is deductively closed, and
- $E'$ is closed under applications of defaults (trivially)

but $\Gamma_T(E') = Cn(\{water\_creature\}) \neq E'$!

# Computing extensions

Determining the operator $\Gamma_T$:

1. Classical reduct $\Delta_E$:
   $\Delta_E := \{\varphi/\gamma \mid (\varphi : \psi_1, \ldots, \psi_n/\gamma) \in \Delta \text{ and } \{\neg\psi_1, \ldots, \neg\psi_n\} \cap E = \emptyset\}$.
   - $\varphi/\gamma$ is the *residue* of $(\varphi : \psi_1, \ldots, \psi_n/\gamma)$.
2. $Cn^{\Delta_E}(W) := Cn(W \cup \bigcup_{i \geq 0} E_i)$, with
$$\begin{aligned} E_0 &:= \{\gamma \mid \varphi/\gamma \in \Delta_E \text{ and } W \vdash \varphi\}; \\ E_i &:= \{\gamma \mid \varphi/\gamma \in \Delta_E \text{ and } W \cup E_{i-1} \vdash \varphi\}. \end{aligned}$$
3. Then: $\Gamma_T(E) = Cn^{\Delta_E}(W)$.

# Computing extensions (ctd.)

Theorem: Let $T = (W, \Delta)$. Then:

$$E \text{ is extension of } T \iff Cn^{\Delta_E}(W) = E \qquad .$$

- Problem: Which sets are potential candidates for being extensions?
- Answer: All sets of form $Cn(W \cup \mathcal{C})$ s.t.

$$\mathcal{C} \subseteq \{\gamma \mid (\varphi : \psi_1, \ldots, \psi_n / \gamma) \in \Delta\}.$$

- N.B.: This yields a naive algorithm for computing extensions which is *exponential* in the size of the default theory in the worst case!
  - However, presumably, we can do no better in general as checking whether a given propositional default theory has an extension is $\Sigma_2^P$*-complete* [Gottlob, 1992].

# Example: Nixon diamond—Revisited

- Consider $T = (W, \Delta)$, where
$$\begin{aligned} W &= \{q, r\}, \\ \Delta &= \{(q : p/p), (r : \neg p/\neg p)\}, \end{aligned}$$
representing a propositional version of the Nixon diamond.

- $\Delta = \{(q : p/p), (r : \neg p/\neg p)\}$ has two defaults $\Longrightarrow$ four candidates:
$$\begin{array}{ll} E_1 = Cn(\{q, r\}) & E_3 = Cn(\{q, r, \neg p\}) \\ E_2 = Cn(\{q, r, p\}) & E_4 = Cn(\{q, r, p, \neg p\}) \end{array}$$

- Determining $E_i' := Cn^{\Delta_{E_i}}(W)$ $(=\Gamma_T(E_i))$:
$$\begin{array}{lll} \Delta_{E_1} = \{q/p, r/\neg p\} & E_1' = Cn(\{q, r, p, \neg p\}) & = E_4 \\ \Delta_{E_2} = \{q/p\} & E_2' = Cn(\{q, r, p\}) & = E_2 \\ \Delta_{E_3} = \{r/\neg p\} & E_3' = Cn(\{q, r, \neg p\}) & = E_3 \\ \Delta_{E_4} = \emptyset & E_4' = Cn(\{q, r\}) & = E_1 \end{array}$$

➥ $E_2$ and $E_3$ are extensions of $T$ (and there are no other extensions of $T$).

# Extending default theories

$$T: \quad W = \emptyset, \; \Delta = \left\{ \delta_0 = \frac{\top \;:\; a}{a} \right\}$$

$T$ has exactly one extension: $E = Cn(\{a\})$

- Let $\Delta_1 = \{\delta_0, \delta_1 = \frac{\top : b}{\neg b}\}$. $T_1 = (W, \Delta_1)$ has *no extension*.
- Let $\Delta_2 = \{\delta_0, \delta_2 = \frac{b : c}{c}\}$. $T_2 = (W, \Delta_2)$ has *still $E$ as single extension*.
- Let $\Delta_3 = \{\delta_0, \delta_3 = \frac{\top : \neg a}{\neg a}\}$. $T_3 = (W, \Delta_3)$ has *two extensions*, namely $E$ and $Cn(\{\neg a\})$.
- Let $\Delta_4 = \{\delta_0, \delta_4 = \frac{a : b}{b}\}$. $T_4 = (W, \Delta_4)$ has the *extension $Cn(\{a, b\})$, containing $E$*.

# Extending default theories (ctd.)

Extending a default theory can thus

- eliminate extensions,
- modify extensions, or
- yield new extensions.

# Normal defaults

- A default is normal iff it is of the form

$$\boxed{\frac{\varphi \; : \; \psi}{\psi}}$$

- Important property:
  - Normal default theories (i.e., containing only normal defaults) always possess extensions.

- Many common-sense reasoning patterns can be modeled in terms of normal defaults.
  - Example: birds typically fly

$$\frac{\mathit{bird} \; : \; \mathit{can\_fly}}{\mathit{can\_fly}}$$

# Open default theories

- We defined extensions only for *closed* default theories, i.e., where all defaults are closed—containing only closed formulas.
- In case a default theory $T$ is *open* (i.e., not closed), one uses a method similar to grounding in logic programming to obtain the *closure of $T$*
  - i.e., one replaces all open defaults by instantiating them with the terms constructible from the terms mentioned by $T$.
- However, there is a catch:
  - a default theory may also determine objects *only implicitly mentioned*
  - e.g., objects determined by existential quantification.
  - In such a case, one uses *skolemisation* to eliminate existential quantifiers, making the implicitly mentioned objects explicit by introducing new *Skolem terms*.

# Open default theories—Example

- Let $T = \langle W, D \rangle$ be the following default theory:

$$W = \{\exists x \, Kryptonian(x)\};$$
$$D = \left\{ d = \frac{Kryptonian(x) \; : \; Superpowers(x)}{Superpowers(x)} \right\}.$$

- $T$ makes implicit reference to an object being a Kryptonian.
- We expect that this object possesses superpowers, i.e.,

$$\exists x \, (Kryptonian(x) \land Superpowers(x))$$

should be contained in an extension of $T$.

# Open default theories—Example (ctd.)

- To achieve this, we replace the premiss $\exists x\, \mathit{Kryptonian}(x)$ by its skolemisation $\mathit{Kryptonian}(a)$, introducing a new Skolem constant $a$.
- The closure of $T$ is then given as follows:

$$\mathit{Kryptonian}(a),\quad \frac{\mathit{Kryptonian}(a)\ :\ \mathit{Superpowers}(a)}{\mathit{Superpowers}(a)}$$

- This default theory has one extension, namely

$$E = \mathit{Cn}(\{\mathit{Kryptonian}(a), \mathit{Superpowers}(a)\})$$

and it holds that $\exists x\, (\mathit{Kryptonian}(x) \wedge \mathit{Superpowers}(x)) \in E$.

## 5.3.3 Glimpses Beyond

# Modal Nonmonotonic Logics

Additional important nonmonotonic formalisms: modal nonmonotonic logics.

- Based on the language of modal logic.
- Model the behaviour of an ideally rational agent reasoning about his own beliefs.
- Modal operators:
  - $LA$: $A$ is believed
  - $MA$: $A$ can be consistently assumed.
- E.g., "Birds typically fly" can be expressed by

$$\forall x\big((Bird(x) \wedge MFlies(x)) \rightarrow Flies(x)\big).$$

- Important modal nonmonotonic logic:
  - autoepistemic logic [Moore, 1983].

# Answer-Set Semantics

- Implementing nonmonotonic reasoning:
  - logic programs with default negation under the answer-set semantics, containing rules of form
    $$a \leftarrow b_1, \ldots, b_n, not\ c_1, \ldots, not\ c_m$$
    - $a, b_1, \ldots, b_n, c_1, \ldots, c_m$ are atoms from a finite vocabulary;
    - *not* denotes *default negation* (a.k.a. *negation as failure*);
    - rule "fires" if $b_1, \ldots, b_n$ is derivable but $c_1, \ldots, c_m$ are *not* derivable.

- The answer-set semantics is the result of associating logic programs with default theories in a canonical way:
  - For rule $r$ as above, let $\delta(r)$ be the following default:
    $$\delta(r) = \frac{b_1 \wedge \cdots \wedge b_n : \neg c_1, \ldots, \neg c_m}{a}.$$
  - ➥ The answer sets of a program $P$ are in a one-to-one correspondence to the extensions of the default theory $\langle \emptyset, \{\delta(r) \mid r \in P\} \rangle$.