A Motivating Example: Defaults in Knowledge Bases

1. employee(anne)
2. employee(bert)
3. employee(carla)
4. employee(detlef)
5. employee(thomas)
6. onUnpaidMPaternityLeave(thomas)
7. employee(X) $\wedge \neg$ onUnpaidMPaternityLeave(X) $\rightarrow$ gettingSalary(X)
8. typically employee(X) $\rightarrow$ $\neg$onUnpaidMPaternityLeave(X)

A Motivating Example: Defaults in Knowledge Bases

1. employee(anne)
2. employee(bert)
3. employee(carla)
4. employee(detlef)
5. employee(thomas)
6. onUnpaidMPaternityLeave(thomas)
7. employee(X) $\wedge \neg$ onUnpaidMPaternityLeave(X) $\rightarrow$ gettingSalary(X)
8. typically employee(X) $\rightarrow$ $\neg$onUnpaidMPaternityLeave(X)

A Motivating Example: Defaults in Knowledge Bases

1. employee(anne)
2. employee(bert)
3. employee(carla)
4. employee(detlef)
5. employee(thomas)
6. onUnpaidMPaternityLeave(thomas)
7. employee(X) $\land \neg$ onUnpaidMPaternityLeave(X) $\rightarrow$ gettingSalary(X)
8. typically employee(X) $\rightarrow \neg$onUnpaidMPaternityLeave(X)

A Motivating Example: Common Sense Reasoning

1. *Tweety* is a bird like other birds.
2. During the summer he stays in *Northern Europe*, in the winter he stays in Africa.

- Would you expect Tweety to be able to fly?
- How does Tweety get from Northern Europe to Africa?

How would you formalize this in formal logic so that you get the expected answers?

A Motivating Example: Common Sense Reasoning

1. *Tweety* is a bird like other birds.
2. During the summer he stays in *Northern Europe*, in the winter he stays in Africa.

- Would you expect Tweety to be able to fly?
- How does Tweety get from Northern Europe to Africa?

How would you formalize this in formal logic so that you get the expected answers?

A Motivating Example: Common Sense Reasoning

1. *Tweety* is a bird like other birds.
2. During the summer he stays in *Northern Europe*, in the winter he stays in Africa.

- Would you expect Tweety to be able to fly?
- How does Tweety get from Northern Europe to Africa?

How would you formalize this in formal logic so that you get the expected answers?

A Motivating Example: Common Sense Reasoning

1. *Tweety* is a bird like other birds.
2. During the summer he stays in *Northern Europe*, in the winter he stays in Africa.

- Would you expect Tweety to be able to fly?
- How does Tweety get from Northern Europe to Africa?

How would you formalize this in formal logic so that you get the expected answers?

A Motivating Example: Common Sense Reasoning

1. *Tweety* is a bird like other birds.
2. During the summer he stays in *Northern Europe*, in the winter he stays in Africa.

- Would you expect Tweety to be able to fly?
- How does Tweety get from Northern Europe to Africa?

How would you formalize this in formal logic so that you get the expected answers?

A Formalization . . .

1. bird(tweety)
2. spend-summer(tweety,northern-europe) $\wedge$ spend-winter(tweety,africa)
3. $\forall x$(bird(x) $\rightarrow$ can-fly(x))
4. faraway(northern-europe,africa)
5. $\forall xyz$(can-fly(x) $\wedge$ faraway(y, z) $\wedge$ spend-summer(x, y) $\wedge$ spend-winter(x, z) $\rightarrow$ flies(x, y, z))
6. The implication (3) is just a reasonable assumption
7. What if Tweety is an Emu?

A Formalization . . .

1. bird(tweety)
2. spend-summer(tweety,northern-europe) $\wedge$ spend-winter(tweety,africa)
3. $\forall x$(bird(x) $\rightarrow$ can-fly(x))
4. faraway(northern-europe,africa)
5. $\forall xyz$(can-fly(x) $\wedge$ faraway(y, z) $\wedge$ spend-summer(x, y) $\wedge$ spend-winter(x, z) $\rightarrow$ flies(x, y, z))
6. The implication (3) is just a reasonable assumption
7. What if Tweety is an Emu?

A Formalization . . .

1. bird(tweety)
2. spend-summer(tweety,northern-europe) $\wedge$ spend-winter(tweety,africa)
3. $\forall x$(bird(x) $\rightarrow$ can-fly(x))
4. faraway(northern-europe,africa)
5. $\forall xyz$(can-fly(x) $\wedge$ faraway(y, z) $\wedge$ spend-summer(x, y) $\wedge$ spend-winter(x, z) $\rightarrow$ flies(x, y, z))
6. The implication (3) is just a reasonable assumption
7. What if Tweety is an Emu?

A Formalization . . .

1. bird(tweety)
2. spend-summer(tweety,northern-europe) $\wedge$ spend-winter(tweety,africa)
3. $\forall x$(bird(x) $\rightarrow$ can-fly(x))
4. faraway(northern-europe,africa)
5. $\forall xyz$(can-fly(x) $\wedge$ faraway(y, z) $\wedge$ spend-summer(x, y) $\wedge$ spend-winter(x, z) $\rightarrow$ flies(x, y, z))
6. The implication (3) is just a reasonable assumption
7. What if Tweety is an Emu?

A Formalization . . .

1. bird(tweety)
2. spend-summer(tweety,northern-europe) $\wedge$ spend-winter(tweety,africa)
3. $\forall x$(bird(x) $\rightarrow$ can-fly(x))
4. faraway(northern-europe,africa)
5. $\forall xyz$(can-fly(x) $\wedge$ faraway(y, z) $\wedge$ spend-summer(x, y) $\wedge$ spend-winter(x, z) $\rightarrow$ flies(x, y, z))
6. The implication (3) is just a reasonable assumption
7. What if Tweety is an Emu?

A Formalization . . .

1. bird(tweety)
2. spend-summer(tweety,northern-europe) $\wedge$ spend-winter(tweety,africa)
3. $\forall x$(bird(x) $\rightarrow$ can-fly(x))
4. faraway(northern-europe,africa)
5. $\forall xyz$(can-fly(x) $\wedge$ faraway(y, z) $\wedge$ spend-summer(x, y) $\wedge$ spend-winter(x, z) $\rightarrow$ flies(x, y, z))
6. The implication (3) is just a reasonable assumption
7. What if Tweety is an Emu?

A Formalization . . .

1. bird(tweety)
2. spend-summer(tweety,northern-europe) $\wedge$ spend-winter(tweety,africa)
3. $\forall x$(bird(x) $\rightarrow$ can-fly(x))
4. faraway(northern-europe,africa)
5. $\forall xyz$(can-fly(x) $\wedge$ faraway(y, z) $\wedge$ spend-summer(x, y) $\wedge$ spend-winter(x, z) $\rightarrow$ flies(x, y, z))
6. The implication (3) is just a reasonable assumption
7. What if Tweety is an Emu?

Examples of Such Reasoning Patterns

Closed World Assumption: Data base of ground atoms. All ground atoms not present are assumed to be false.

Negation by Failure: In PROLOG, NOT(P) means “P is not provable” instead of “P is provably false”.

Non-strict Inheritance: An attribute value is inherited only if there is no more specialized information contradicting the attribute value.

Reasoning about Actions: When reasoning about actions, it is usually assumed that a property changes only if it has to change, i.e., properties by default do not change.

Examples of Such Reasoning Patterns

Closed World Assumption: Data base of ground atoms. All ground atoms not present are assumed to be false.

Negation by Failure: In PROLOG, NOT(P) means “P is not provable” instead of “P is provably false”.

Non-strict Inheritance: An attribute value is inherited only if there is no more specialized information contradicting the attribute value.

Reasoning about Actions: When reasoning about actions, it is usually assumed that a property changes only if it has to change, i.e., properties by default do not change.

Examples of Such Reasoning Patterns

Closed World Assumption: Data base of ground atoms. All ground atoms not present are assumed to be false.

Negation by Failure: In PROLOG, NOT(P) means “P is not provable” instead of “P is provably false”.

Non-strict Inheritance: An attribute value is inherited only if there is no more specialized information contradicting the attribute value.

Reasoning about Actions: When reasoning about actions, it is usually assumed that a property changes only if it has to change, i.e., properties by default do not change.

Examples of Such Reasoning Patterns

Closed World Assumption: Data base of ground atoms. All ground atoms not present are assumed to be false.

Negation by Failure: In PROLOG, NOT(P) means “P is not provable” instead of “P is provably false”.

Non-strict Inheritance: An attribute value is inherited only if there is no more specialized information contradicting the attribute value.

Reasoning about Actions: When reasoning about actions, it is usually assumed that a property changes only if it has to change, i.e., properties by default do not change.

Default, Defeasible and Nonmonotonic Reasoning

Default Reasoning: Jump to a conclusion if there is no information that contradicts the conclusion.

Defeasible Reasoning: Reasoning based on assumptions that can turn out to be wrong – i.e., conclusions are defeasible. In particular, default reasoning is defeasible.

Nonmonotonic Reasoning: In classical logic, the set of consequence *grows monotonically* with the set of premises. If reasoning becomes defeasible, then reasoning becomes non-monotonic.

Default, Defeasible and Nonmonotonic Reasoning

Default Reasoning: Jump to a conclusion if there is no information that contradicts the conclusion.

Defeasible Reasoning: Reasoning based on assumptions that can turn out to be wrong – i.e., conclusions are defeasible. In particular, default reasoning is defeasible.

Nonmonotonic Reasoning: In classical logic, the set of consequence *grows monotonically* with the set of premises. If reasoning becomes defeasible, then reasoning becomes non-monotonic.

Default, Defeasible and Nonmonotonic Reasoning

Default Reasoning: Jump to a conclusion if there is no information that contradicts the conclusion.

Defeasible Reasoning: Reasoning based on assumptions that can turn out to be wrong – i.e., conclusions are defeasible. In particular, default reasoning is defeasible.

Nonmonotonic Reasoning: In classical logic, the set of consequence *grows monotonically* with the set of premises. If reasoning becomes defeasible, then reasoning becomes non-monotonic.

Approaches to Non-Monotonic Reasoning

- Consistency-based: *Extend* classical theory by rules that test whether an assumption is consistent with existing beliefs.
- non-monotonic logics like DL (default logic), NMLP (non-monotonic logic programming)
- Entailment based on Normal Models: Models are ordered by *normality*. Entailment is determined by considering the most normal models only.
- Circumscription, Preferential and Cumulative Logics

Approaches to Non-Monotonic Reasoning

- Consistency-based: *Extend* classical theory by rules that test whether an assumption is consistent with existing beliefs.
- non-monotonic logics like DL (default logic), NMLP (non-monotonic logic programming)
- Entailment based on Normal Models: Models are ordered by *normality*. Entailment is determined by considering the most normal models only.
- Circumscription, Preferential and Cumulative Logics

NM Logic – Consistency-Based

If φ typically implies ψ, φ is given, and it is consistent to assume ψ, then conclude ψ.

1. Typically bird(x) implies can-fly(x)
2. $\forall x$(emu(x) $\rightarrow$ bird(x))
3. $\forall x$(emu(x) $\rightarrow$ $\neg$can-fly(x))
4. bird(tweety)

$\rightsquigarrow$ can-fly(tweety)

5. \+ emu(tweety)

$\rightsquigarrow$ $\neg$ can-fly(tweety)

NM Logic – Consistency-Based

If φ typically implies ψ, φ is given, and it is consistent to assume ψ, then conclude ψ.

1. Typically bird(x) implies can-fly(x)
2. $\forall x$(emu(x) $\rightarrow$ bird(x))
3. $\forall x$(emu(x) $\rightarrow$ $\neg$can-fly(x))
4. bird(tweety)

$\leadsto$ can-fly(tweety)

5. + emu(tweety)

$\leadsto$ $\neg$ can-fly(tweety)

NM Logic – Consistency-Based

If φ typically implies ψ, φ is given, and it is consistent to assume ψ, then conclude ψ.

1. Typically bird(x) implies can-fly(x)
2. $\forall x$(emu(x) $\rightarrow$ bird(x))
3. $\forall x$(emu(x) $\rightarrow$ $\neg$can-fly(x))
4. bird(tweety)

$\rightsquigarrow$ can-fly(tweety)

5. + emu(tweety)

$\rightsquigarrow$ $\neg$ can-fly(tweety)

NM Logic – Consistency-Based

If φ typically implies ψ, φ is given, and it is consistent to assume ψ, then conclude ψ.

1. Typically bird(x) implies can-fly(x)
2. $\forall x$(emu(x) $\rightarrow$ bird(x))
3. $\forall x$(emu(x) $\rightarrow$ $\neg$can-fly(x))
4. bird(tweety)

$\leadsto$ can-fly(tweety)

5. + emu(tweety)

$\leadsto$ $\neg$ can-fly(tweety)

NM Logic – Normal Models

- If φ typically implies ψ, then the models satisfying $\varphi \wedge \psi$ should be more normal than those satisfying $\varphi \wedge \neg\psi$.
- Similarly, try to *minimize* the interpretation of "*Ab*normality" predicates.
  $\forall x$(bird(x) $\wedge$ $\neg$*Ab*(x) $\rightarrow$ can-fly(x))
  $\forall x$(emu(x) $\rightarrow$ bird(x))
  $\forall x$(emu(x) $\rightarrow$ $\neg$can-fly(x))
  bird(tweety)
  Minimize interpretation of *Ab*. $\rightsquigarrow$ can-fly(tweety)
  + emu(tweety)$\rightsquigarrow$ Now in all models (including the normal ones) $\neg$ can-fly(tweety)

NM Logic – Normal Models

- If φ typically implies ψ, then the models satisfying $\varphi \wedge \psi$ should be more normal than those satisfying $\varphi \wedge \neg\psi$.
- Similarly, try to *minimize* the interpretation of "*Ab*normality" predicates.

  $\forall x(\text{bird}(x) \wedge \neg Ab(x) \rightarrow \text{can-fly}(x))$

  $\forall x(\text{emu}(x) \rightarrow \text{bird}(x))$

  $\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$

  bird(tweety)

  Minimize interpretation of *Ab*. $\rightsquigarrow$ can-fly(tweety)

  + emu(tweety)$\rightsquigarrow$ Now in all models (including the normal ones) ¬ can-fly(tweety)

NM Logic – Normal Models

- If φ typically implies ψ, then the models satisfying $\varphi \wedge \psi$ should be more normal than those satisfying $\varphi \wedge \neg\psi$.
- Similarly, try to *minimize* the interpretation of "*Ab*normality" predicates.
  $\forall x$(bird(x) $\wedge$ $\neg$*Ab*(x) $\rightarrow$ can-fly(x))
  $\forall x$(emu(x) $\rightarrow$ bird(x))
  $\forall x$(emu(x) $\rightarrow$ $\neg$can-fly(x))
  bird(tweety)
  Minimize interpretation of *Ab*. $\rightsquigarrow$ can-fly(tweety)
  + emu(tweety)$\rightsquigarrow$ Now in all models (including the normal ones) $\neg$ can-fly(tweety)

NM Logic – Normal Models

- If φ typically implies ψ, then the models satisfying $\varphi \wedge \psi$ should be more normal than those satisfying $\varphi \wedge \neg\psi$.
- Similarly, try to *minimize* the interpretation of "*Ab*normality" predicates.
  $\forall x(\text{bird}(x) \wedge \neg Ab(x) \rightarrow \text{can-fly}(x))$
  $\forall x(\text{emu}(x) \rightarrow \text{bird}(x))$
  $\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$
  bird(tweety)
  Minimize interpretation of *Ab*. $\rightsquigarrow$ can-fly(tweety)
  + emu(tweety)$\rightsquigarrow$ Now in all models (including the normal ones) ¬ can-fly(tweety)

NM Logic – Normal Models

- If φ typically implies ψ, then the models satisfying $\varphi \wedge \psi$ should be more normal than those satisfying $\varphi \wedge \neg\psi$.
- Similarly, try to *minimize* the interpretation of "*Ab*normality" predicates.
  $\forall x(\text{bird}(x) \wedge \neg Ab(x) \rightarrow \text{can-fly}(x))$
  $\forall x(\text{emu}(x) \rightarrow \text{bird}(x))$
  $\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$
  bird(tweety)
  Minimize interpretation of *Ab*. $\rightsquigarrow$ can-fly(tweety)
  + emu(tweety)$\rightsquigarrow$ Now in all models (including the normal ones) $\neg$ can-fly(tweety)

NM Logic – Normal Models

- If φ typically implies ψ, then the models satisfying $\varphi \wedge \psi$ should be more normal than those satisfying $\varphi \wedge \neg\psi$.
- Similarly, try to *minimize* the interpretation of "*Ab*normality" predicates.
  $\forall x(\text{bird}(x) \wedge \neg Ab(x) \rightarrow \text{can-fly}(x))$
  $\forall x(\text{emu}(x) \rightarrow \text{bird}(x))$
  $\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$
  bird(tweety)
  Minimize interpretation of *Ab*. $\rightsquigarrow$ can-fly(tweety)
  \+ emu(tweety) $\rightsquigarrow$ Now in all models (including the normal ones) $\neg$ can-fly(tweety)

NM Logic – Normal Models

- If φ typically implies ψ, then the models satisfying $\varphi \wedge \psi$ should be more normal than those satisfying $\varphi \wedge \neg\psi$.
- Similarly, try to *minimize* the interpretation of "*Ab*normality" predicates.
  $\forall x(\text{bird}(x) \wedge \neg \mathit{Ab}(x) \rightarrow \text{can-fly}(x))$
  $\forall x(\text{emu}(x) \rightarrow \text{bird}(x))$
  $\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$
  bird(tweety)
  Minimize interpretation of *Ab*. $\rightsquigarrow$ can-fly(tweety)
  + emu(tweety)$\rightsquigarrow$ Now in all models (including the normal ones) $\neg$ can-fly(tweety)

Motivation: Reiter's Default Logic

- We want to express something like "*typically birds fly*".
- Add non-logical inference rule

$$\frac{\text{bird}(x) : \text{can-fly}(x)}{\text{can-fly}(x)}$$

with the *intended meaning:*
If x is a bird and if it is consistent to assume that x can fly, then conclude that x can fly.

- *Exceptions* can be represented as formulae:

$$\forall x(\text{penguin}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{kiwi}(x) \rightarrow \neg\text{can-fly}(x))$$

Motivation: Reiter's Default Logic

- We want to express something like "*typically birds fly*".
- Add non-logical inference rule

$$\frac{\text{bird}(x) : \text{can-fly}(x)}{\text{can-fly}(x)}$$

with the *intended meaning:*
If x is a bird and if it is consistent to assume that x can fly, then conclude that x can fly.

- *Exceptions* can be represented as formulae:

$$\forall x(\text{penguin}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{kiwi}(x) \rightarrow \neg\text{can-fly}(x))$$

Motivation: Reiter's Default Logic

- We want to express something like "*typically birds fly*".
- Add non-logical inference rule

$$\frac{\text{bird}(x) : \text{can-fly}(x)}{\text{can-fly}(x)}$$

with the *intended meaning:*
If x is a bird and if it is consistent to assume that x can fly, then conclude that x can fly.

- *Exceptions* can be represented as formulae:

$$\forall x(\text{penguin}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{kiwi}(x) \rightarrow \neg\text{can-fly}(x))$$

Motivation: Reiter's Default Logic

- We want to express something like "*typically birds fly*".
- Add non-logical inference rule

$$\frac{\text{bird}(x) : \text{can-fly}(x)}{\text{can-fly}(x)}$$

with the *intended meaning:*
If x is a bird and if it is consistent to assume that x can fly, then conclude that x can fly.

- *Exceptions* can be represented as formulae:

$$\forall x(\text{penguin}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{emu}(x) \rightarrow \neg\text{can-fly}(x))$$
$$\forall x(\text{kiwi}(x) \rightarrow \neg\text{can-fly}(x))$$

Formal Framework

- *PL1* with classical provability relation $\vdash$ and *deductive closure:* $Th(\Phi) = \{\phi | \Phi \models \phi\}$
- **Default rules** $\dfrac{\alpha : \beta}{\gamma}$
  - α: Prerequisite – Must have been derived before rule can applied.
  - β: Consistency condition – The negation may not be derivable.
  - γ: Consequence – Will be concluded.
- A default rule is closed if it does not contain free variables.
- (Closed) Default Theory: A pair (D, W), where D is a countable set of (closed) default rules and W is a countable set of PL1 formulae.

Formal Framework

- *PL1* with classical provability relation $\vdash$ and *deductive closure: Th*$(\Phi) = \{\phi | \Phi \models \phi\}$
- **Default rules** $\dfrac{\alpha : \beta}{\gamma}$
  - α: Prerequisite – Must have been derived before rule can applied.
  - β: Consistency condition – The negation may not be derivable.
  - γ: Consequence – Will be concluded.
- A default rule is closed if it does not contain free variables.
- (Closed) Default Theory: A pair (D, W), where D is a countable set of (closed) default rules and W is a countable set of PL1 formulae.

Formal Framework

- *PL1* with classical provability relation $\vdash$ and *deductive closure:* $Th(\Phi) = \{\phi | \Phi \models \phi\}$
- **Default rules** $\dfrac{\alpha : \beta}{\gamma}$
  - α: Prerequisite – Must have been derived before rule can applied.
  - β: Consistency condition – The negation may not be derivable.
  - γ: Consequence – Will be concluded.
- A default rule is closed if it does not contain free variables.
- (Closed) Default Theory: A pair (D, W), where D is a countable set of (closed) default rules and W is a countable set of PL1 formulae.

Formal Framework

- *PL1* with classical provability relation $\vdash$ and *deductive closure:* $Th(\Phi) = \{\phi | \Phi \models \phi\}$
- **Default rules** $\dfrac{\alpha : \beta}{\gamma}$
  - α: Prerequisite – Must have been derived before rule can applied.
  - β: Consistency condition – The negation may not be derivable.
  - γ: Consequence – Will be concluded.
- A default rule is closed if it does not contain free variables.
- (Closed) Default Theory: A pair (D, W), where D is a countable set of (closed) default rules and W is a countable set of PL1 formulae.

Formal Framework

- *PL1* with classical provability relation $\vdash$ and *deductive closure: Th*$(\Phi) = \{\phi | \Phi \models \phi\}$
- **Default rules** $\dfrac{\alpha : \beta}{\gamma}$
  - α: Prerequisite – Must have been derived before rule can applied.
  - β: Consistency condition – The negation may not be derivable.
  - γ: Consequence – Will be concluded.
- A default rule is closed if it does not contain free variables.
- (Closed) Default Theory: A pair (D, W), where D is a countable set of (closed) default rules and W is a countable set of PL1 formulae.

Formal Framework

- *PL1* with classical provability relation $\vdash$ and *deductive closure:* $Th(\Phi) = \{\phi | \Phi \models \phi\}$
- **Default rules** $\dfrac{\alpha : \beta}{\gamma}$
  - α: Prerequisite – Must have been derived before rule can applied.
  - β: Consistency condition – The negation may not be derivable.
  - γ: Consequence – Will be concluded.
- A default rule is closed if it does not contain free variables.
- (Closed) Default Theory: A pair (D, W), where D is a countable set of (closed) default rules and W is a countable set of PL1 formulae.

Formal Framework

- *PL1* with classical provability relation $\vdash$ and *deductive closure: Th*$(\Phi) = \{\phi | \Phi \models \phi\}$
- **Default rules** $\dfrac{\alpha : \beta}{\gamma}$
  - α: Prerequisite – Must have been derived before rule can applied.
  - β: Consistency condition – The negation may not be derivable.
  - γ: Consequence – Will be concluded.
- A default rule is closed if it does not contain free variables.
- (Closed) Default Theory: A pair (D, W), where D is a countable set of (closed) default rules and W is a countable set of PL1 formulae.

Extensions of Default Theories

Default theories extend the theories given by W using the default rules $D \rightsquigarrow$ extensions. There may be zero, one, or many extensions.

Example

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

One extension contains b, the other contains c.

Intuitively: A extension is a set of *beliefs* resulting from W and D.

Extensions of Default Theories

Default theories extend the theories given by W using the default rules $D \leadsto$ extensions. There may be zero, one, or many extensions.

Example

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

One extension contains b, the other contains c.

Intuitively: A extension is a set of *beliefs* resulting from W and D.

Extensions of Default Theories

Default theories extend the theories given by W using the default rules $D \rightsquigarrow$ extensions. There may be zero, one, or many extensions.

Example

$$
\begin{aligned}
W &= \{a, \neg b \lor \neg c\} \\
D &= \left\{ \frac{a : b}{b}, \frac{a : c}{c} \right\}
\end{aligned}
$$

One extension contains b, the other contains c.

Intuitively: A extension is a set of *beliefs* resulting from W and D.

Decision Problems about Extensions in Default Logic

Existence of extensions: Does a default theory have an extension?

Credulous reasoning: If φ is in at least one extension, φ is a credulous default conclusion.

Skeptical Reasoning: If φ is in all extensions, φ is a skeptical default conclusion.

Extensions – Informally

Desirable properties of an extension E of (D, W):

1. Contains all facts $W \subseteq E$.
2. Is deductively closed: $E = \mathrm{Th}(E)$.
3. All applicable default rules have been applied:
   If
   1. $(\frac{\alpha:\beta}{\gamma}) \in D$,
   2. $\alpha \in E$,
   3. $\neg\beta \notin E$

   then $\gamma \in E$.
4. Requirement: Application of default rules must follow in sequence (groundedness).

Extensions – Informally

Desirable properties of an extension E of (D, W):

1. Contains all facts $W \subseteq E$.
2. Is deductively closed: $E = \text{Th}(E)$.
3. All applicable default rules have been applied:
   If
   1. $(\frac{\alpha:\beta}{\gamma}) \in D$,
   2. $\alpha \in E$,
   3. $\neg\beta \notin E$

   then $\gamma \in E$.
4. Requirement: Application of default rules must follow in sequence (groundedness).

Extensions – Informally

Desirable properties of an extension E of (D, W):

1. Contains all facts $W \subseteq E$.
2. Is deductively closed: $E = \mathrm{Th}(E)$.
3. All applicable default rules have been applied:
   If
   1. $(\frac{\alpha:\beta}{\gamma}) \in D$,
   2. $\alpha \in E$,
   3. $\neg\beta \notin E$

   then $\gamma \in E$.
4. Requirement: Application of default rules must follow in sequence (groundedness).

Extensions – Informally

Desirable properties of an extension E of (D, W):

1. Contains all facts $W \subseteq E$.
2. Is deductively closed: $E = \mathrm{Th}(E)$.
3. All applicable default rules have been applied:
   If
   1. $(\frac{\alpha:\beta}{\gamma}) \in D$,
   2. $\alpha \in E$,
   3. $\neg\beta \notin E$

   then $\gamma \in E$.
4. Requirement: Application of default rules must follow in sequence (groundedness).

Groundedness

Example

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

Question: Should *Th*$(\{a, b\})$ be an extension?

Answer: No!

a can only be derived if we already have derived b.

b can only be derived if we already have derived a.

Groundedness

Example

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

Question: Should $Th(\{a, b\})$ be an extension?

Answer: No!

a can only be derived if we already have derived b.

b can only be derived if we already have derived a.

Groundedness

Example

$$\begin{aligned} W &= \emptyset \\ D &= \left\{ \frac{a : b}{b}, \frac{b : a}{a} \right\} \end{aligned}$$

Question: Should *Th*$(\{a, b\})$ be an extension?
Answer: No!
a can only be derived if we already have derived b.
b can only be derived if we already have derived a.

Extensions – Formally

Definition

Let $\Delta = (D, W)$ be a closed default theory and let E be a set of closed formulae. Let

$$E_0 = W$$
$$E_i = \mathrm{Th}(E_{i-1}) \cup \left\{ \gamma \,\middle|\, \frac{\alpha : \beta}{\gamma} \in D, \alpha \in E_{i-1}, \neg\beta \notin E \right\}$$

Then E is an extension of Δ *iff*

$$E = \bigcup_{i=0}^{\infty} E_i.$$

Extensions – Formally

Definition

Let $\Delta = (D, W)$ be a closed default theory and let E be a set of closed formulae. Let

$$
\begin{aligned}
E_0 &= W \\
E_i &= \mathrm{Th}(E_{i-1}) \cup \left\{ \gamma \,\middle|\, \frac{\alpha : \beta}{\gamma} \in D, \alpha \in E_{i-1}, \neg\beta \notin E \right\}
\end{aligned}
$$

Then E is an extension of Δ *iff*

$$E = \bigcup_{i=0}^{\infty} E_i.$$

Extensions – Formally

Definition

Let $\Delta = (D, W)$ be a closed default theory and let E be a set of closed formulae. Let

$$\begin{aligned} E_0 &= W \\ E_i &= \mathrm{Th}(E_{i-1}) \cup \left\{ \gamma \,\middle|\, \frac{\alpha : \beta}{\gamma} \in D, \alpha \in E_{i-1}, \neg\beta \notin E \right\} \end{aligned}$$

Then E is an extension of Δ *iff*

$$E = \bigcup_{i=0}^{\infty} E_i.$$

How to Use This Definition?

- The definition does not tell us how to *construct* an extension.
- However, it tells us how to *check* whether a set is an extension.
- Guess a set E.
- Then construct sets E_i by starting with W.
- If $\boxed{E = \bigcup_{i=0}^{\infty} E_i}$, then E is an extension of (D, W).

How to Use This Definition?

- The definition does not tell us how to *construct* an extension.
- However, it tells us how to *check* whether a set is an extension.
- Guess a set E.
- Then construct sets E_i by starting with W.
- If $\boxed{E = \bigcup_{i=0}^{\infty} E_i}$, then E is an extension of (D, W).

How to Use This Definition?

- The definition does not tell us how to *construct* an extension.
- However, it tells us how to *check* whether a set is an extension.
- Guess a set E.
- Then construct sets E_i by starting with W.
- If $\boxed{E = \bigcup_{i=0}^{\infty} E_i}$, then E is an extension of (D, W).

Examples

$$D = \left\{\frac{a : b}{b}, \frac{b : a}{a}\right\} \qquad W = \{(a \vee b)\}$$

$$D = \left\{\frac{a : b}{\neg b}\right\} \qquad W = \emptyset$$

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

$$D = \left\{\frac{: a}{a}, \frac{: b}{b}, \frac{: c}{c}\right\} \qquad W = \{b \rightarrow \neg a \wedge \neg c\}$$

$$D = \left\{\frac{: c}{\neg d}, \frac{: d}{\neg e}, \frac{: e}{\neg f}\right\} \qquad W = \emptyset$$

$$D = \left\{\frac{: c}{\neg d}, \frac{: d}{\neg c}\right\} \qquad W = \emptyset$$

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

Examples

$$D = \left\{\frac{a : b}{b}, \frac{b : a}{a}\right\} \qquad W = \{(a \vee b)\}$$

$$D = \left\{\frac{a : b}{\neg b}\right\} \qquad W = \emptyset$$

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

$$D = \left\{\frac{:a}{a}, \frac{:b}{b}, \frac{:c}{c}\right\} \qquad W = \{b \rightarrow \neg a \wedge \neg c\}$$

$$D = \left\{\frac{:c}{\neg d}, \frac{:d}{\neg e}, \frac{:e}{\neg f}\right\} \qquad W = \emptyset$$

$$D = \left\{\frac{:c}{\neg d}, \frac{:d}{\neg c}\right\} \qquad W = \emptyset$$

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

Examples

$$D = \left\{\frac{a : b}{b}, \frac{b : a}{a}\right\} \qquad W = \{(a \vee b)\}$$

$$D = \left\{\frac{a : b}{\neg b}\right\} \qquad W = \emptyset$$

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

$$D = \left\{\frac{: a}{a}, \frac{: b}{b}, \frac{: c}{c}\right\} \qquad W = \{b \rightarrow \neg a \wedge \neg c\}$$

$$D = \left\{\frac{: c}{\neg d}, \frac{: d}{\neg e}, \frac{: e}{\neg f}\right\} \qquad W = \emptyset$$

$$D = \left\{\frac{: c}{\neg d}, \frac{: d}{\neg c}\right\} \qquad W = \emptyset$$

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

Examples

$$D = \left\{\frac{a : b}{b}, \frac{b : a}{a}\right\} \qquad W = \{(a \vee b)\}$$
$$D = \left\{\frac{a : b}{\neg b}\right\} \qquad W = \emptyset$$
$$D = \left\{\frac{a : b}{\neg b}\right\} \qquad W = \{a\}$$
$$D = \left\{\frac{: a}{a}, \frac{: b}{b}, \frac{: c}{c}\right\} \qquad W = \{b \rightarrow \neg a \wedge \neg c\}$$
$$D = \left\{\frac{: c}{\neg d}, \frac{: d}{\neg e}, \frac{: e}{\neg f}\right\} \qquad W = \emptyset$$
$$D = \left\{\frac{: c}{\neg d}, \frac{: d}{\neg c}\right\} \qquad W = \emptyset$$
$$D = \left\{\frac{a : b}{c}, \frac{a : d}{e}\right\} \qquad W = \{a, \neg b \vee \neg d\}$$

Examples

$$D = \left\{\frac{a:b}{b}, \frac{b:a}{a}\right\} \qquad W = \{(a \vee b)\}$$

$$D = \left\{\frac{a:b}{\neg b}\right\} \qquad W = \emptyset$$

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

$$D = \left\{\frac{:a}{a}, \frac{:b}{b}, \frac{:c}{c}\right\} \qquad W = \{b \rightarrow \neg a \wedge \neg c\}$$

$$D = \left\{\frac{:c}{\neg d}, \frac{:d}{\neg e}, \frac{:e}{\neg f}\right\} \qquad W = \emptyset$$

$$D = \left\{\frac{:c}{\neg d}, \frac{:d}{\neg c}\right\} \qquad W = \emptyset$$

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

Examples

$$D = \left\{ \frac{a : b}{b}, \frac{b : a}{a} \right\} \qquad W = \{(a \vee b)\}$$
$$D = \left\{ \frac{a : b}{\neg b} \right\} \qquad W = \emptyset$$
$$D = \left\{ \frac{a : b}{\neg b} \right\} \qquad W = \{a\}$$
$$D = \left\{ \frac{:a}{a}, \frac{:b}{b}, \frac{:c}{c} \right\} \qquad W = \{b \rightarrow \neg a \wedge \neg c\}$$
$$D = \left\{ \frac{:c}{\neg d}, \frac{:d}{\neg e}, \frac{:e}{\neg f} \right\} \qquad W = \emptyset$$
$$D = \left\{ \frac{:c}{\neg d}, \frac{:d}{\neg c} \right\} \qquad W = \emptyset$$
$$D = \left\{ \frac{a : b}{c}, \frac{a : d}{e} \right\} \qquad W = \{a, \neg b \vee \neg d\}$$

Examples

$$
\begin{array}{lll}
D = \left\{\frac{a : b}{b}, \frac{b : a}{a}\right\} & & W = \{(a \vee b)\} \\
D = \left\{\frac{a : b}{\neg b}\right\} & & W = \emptyset \\
D = \left\{\frac{a : b}{\neg b}\right\} & & W = \{a\} \\
D = \left\{\frac{:a}{a}, \frac{:b}{b}, \frac{:c}{c}\right\} & & W = \{b \rightarrow \neg a \wedge \neg c\} \\
D = \left\{\frac{:c}{\neg d}, \frac{:d}{\neg e}, \frac{:e}{\neg f}\right\} & & W = \emptyset \\
D = \left\{\frac{:c}{\neg d}, \frac{:d}{\neg c}\right\} & & W = \emptyset \\
D = \left\{\frac{a : b}{c}, \frac{a : d}{e}\right\} & & W = \{a, \neg b \vee \neg d\}
\end{array}
$$

Questions, Questions, Questions . . .

- What can we say about the existence of extensions?
- How do the different extensions relate to each other?
  - Can one extension be a subset of another one?
  - Are extensions pairwise incompatible (i.e. jointly inconsistent)?
- Can an extension be inconsistent?

Questions, Questions, Questions . . .

- What can we say about the existence of extensions?
- How do the different extensions relate to each other?
  - Can one extension be a subset of another one?
  - Are extensions pairwise incompatible (i.e. jointly inconsistent)?
- Can an extension be inconsistent?

Questions, Questions, Questions . . .

- What can we say about the existence of extensions?
- How do the different extensions relate to each other?
  - Can one extension be a subset of another one?
  - Are extensions pairwise incompatible (i.e. jointly inconsistent)?
- Can an extension be inconsistent?

Questions, Questions, Questions . . .

- What can we say about the existence of extensions?
- How do the different extensions relate to each other?
  - Can one extension be a subset of another one?
  - Are extensions pairwise incompatible (i.e. jointly inconsistent)?
- Can an extension be inconsistent?

Questions, Questions, Questions . . .

- What can we say about the existence of extensions?
- How do the different extensions relate to each other?
  - Can one extension be a subset of another one?
  - Are extensions pairwise incompatible (i.e. jointly inconsistent)?
- Can an extension be inconsistent?

Properties of Extensions

Theorem

1. *If W is inconsistent, there is only one extension.*
2. *A closed default theory (D, W) has an inconsistent extension iff W is inconsistent.*

Proof idea.

1. If W is inconsistent, no default rule is applicable and Th(W) is the only extension.
2. Claim 1 $\Longrightarrow$ the *if*-part. For *only if*: If W is consistent, there is a consistent E_i s. t. E_{i+1} is inconsistent. Let $\{\gamma_1, \ldots, \gamma_n\} = E_{i+1} \backslash$Th($E_i$) (the conclusions of applied defaults.) Now $\{\neg\beta_1, \ldots, \neg\beta_n\} \cap E = \emptyset$ because otherwise the defaults are not applicable. But this contradicts the inconsistency of E.

□

Properties of Extensions

Theorem

1. *If W is inconsistent, there is only one extension.*
2. *A closed default theory (D, W) has an inconsistent extension iff W is inconsistent.*

Proof idea.

1. If W is inconsistent, no default rule is applicable and Th(W) is the only extension.
2. Claim 1 $\Longrightarrow$ the *if*-part. For *only if*: If W is consistent, there is a consistent E_i s. t. E_{i+1} is inconsistent. Let $\{\gamma_1, \ldots, \gamma_n\} = E_{i+1} \setminus \text{Th}(E_i)$ (the conclusions of applied defaults.) Now $\{\neg\beta_1, \ldots, \neg\beta_n\} \cap E = \emptyset$ because otherwise the defaults are not applicable. But this contradicts the inconsistency of E.

□

Properties of Extensions

Theorem

1. *If W is inconsistent, there is only one extension.*
2. *A closed default theory (D, W) has an inconsistent extension iff W is inconsistent.*

Proof idea.

1. If W is inconsistent, no default rule is applicable and Th(W) is the only extension.
2. Claim 1 $\Longrightarrow$ the *if*-part. For *only if*: If W is consistent, there is a consistent E_i s. t. E_{i+1} is inconsistent. Let $\{\gamma_1, \ldots, \gamma_n\} = E_{i+1} \backslash \text{Th}(E_i)$ (the conclusions of applied defaults.) Now $\{\neg\beta_1, \ldots, \neg\beta_n\} \cap E = \emptyset$ because otherwise the defaults are not applicable. But this contradicts the inconsistency of E.

□

Properties of Extensions

Theorem

1. *If W is inconsistent, there is only one extension.*
2. *A closed default theory (D, W) has an inconsistent extension iff W is inconsistent.*

Proof idea.

1. If W is inconsistent, no default rule is applicable and Th(W) is the only extension.
2. Claim 1 $\Longrightarrow$ the *if*-part. For *only if*: If W is consistent, there is a consistent E_i s. t. E_{i+1} is inconsistent. Let $\{\gamma_1, \ldots, \gamma_n\} = E_{i+1} \backslash$Th$(E_i)$ (the conclusions of applied defaults.) Now $\{\neg\beta_1, \ldots, \neg\beta_n\} \cap E = \emptyset$ because otherwise the defaults are not applicable. But this contradicts the inconsistency of E.

□

Properties of Extensions

Theorem

1. *If W is inconsistent, there is only one extension.*
2. *A closed default theory (D, W) has an inconsistent extension iff W is inconsistent.*

Proof idea.

1. If W is inconsistent, no default rule is applicable and Th(W) is the only extension.
2. Claim 1 $\Longrightarrow$ the *if*-part. For *only if*: If W is consistent, there is a consistent E_i s. t. E_{i+1} is inconsistent. Let $\{\gamma_1, \ldots, \gamma_n\} = E_{i+1} \backslash$Th$(E_i)$ (the conclusions of applied defaults.) Now $\{\neg\beta_1, \ldots, \neg\beta_n\} \cap E = \emptyset$ because otherwise the defaults are not applicable. But this contradicts the inconsistency of E.

□

Properties of Extensions

Theorem

1. *If W is inconsistent, there is only one extension.*
2. *A closed default theory (D, W) has an inconsistent extension iff W is inconsistent.*

Proof idea.

1. If W is inconsistent, no default rule is applicable and Th(W) is the only extension.
2. Claim 1 $\Longrightarrow$ the *if*-part. For *only if*: If W is consistent, there is a consistent E_i s. t. E_{i+1} is inconsistent. Let $\{\gamma_1, \ldots, \gamma_n\} = E_{i+1} \backslash$Th$(E_i)$ (the conclusions of applied defaults.) Now $\{\neg\beta_1, \ldots, \neg\beta_n\} \cap E = \emptyset$ because otherwise the defaults are not applicable. But this contradicts the inconsistency of E.

□

Properties of Extensions

Theorem

If E and F are extensions of (D, W) such that $E \subseteq F$, then $E = F$.

Proof sketch.

$E = \bigcup_{i=0}^{\infty} E_i$ and $F = \bigcup_{i=0}^{\infty} F_i$. Use induction to show $F_i \subseteq E_i$.

Base case $i = 0$: Trivially $E_0 = F_0 = W$.

Inductive case $i \geq 1$: Assume $\gamma \in F_{i+1}$. Two cases:

1. $\gamma \in \mathrm{Th}(F_i)$ implies $\gamma \in \mathrm{Th}(E_i)$ (because $F_i \subseteq E_i$ by IH), and therefore $\gamma \in E_{i+1}$.
2. Otherwise $\frac{\alpha : \beta}{\gamma} \in D$, $\alpha \in F_i$, $\neg\beta \notin F$. However, then we have $\alpha \in E_i$ (because $F_i \subseteq E_i$) and $\neg\beta \notin E$ (because of $E \subseteq F$), i.e., $\gamma \in E_{i+1}$.

□

Properties of Extensions

Theorem

If E and F are extensions of (D, W) such that $E \subseteq F$, then $E = F$.

Proof sketch.

$E = \bigcup_{i=0}^{\infty} E_i$ and $F = \bigcup_{i=0}^{\infty} F_i$. Use induction to show $F_i \subseteq E_i$.

Base case $i = 0$: Trivially $E_0 = F_0 = W$.

Inductive case $i \geq 1$: Assume $\gamma \in F_{i+1}$. Two cases:

1. $\gamma \in \mathrm{Th}(F_i)$ implies $\gamma \in \mathrm{Th}(E_i)$ (because $F_i \subseteq E_i$ by IH), and therefore $\gamma \in E_{i+1}$.
2. Otherwise $\frac{\alpha : \beta}{\gamma} \in D$, $\alpha \in F_i$, $\neg\beta \notin F$. However, then we have $\alpha \in E_i$ (because $F_i \subseteq E_i$) and $\neg\beta \notin E$ (because of $E \subseteq F$), i.e., $\gamma \in E_{i+1}$.

□

Properties of Extensions

Theorem

If E and F are extensions of (D, W) such that $E \subseteq F$, then $E = F$.

Proof sketch.

$E = \bigcup_{i=0}^{\infty} E_i$ and $F = \bigcup_{i=0}^{\infty} F_i$. Use induction to show $F_i \subseteq E_i$.

Base case $i = 0$: Trivially $E_0 = F_0 = W$.

Inductive case $i \geq 1$: Assume $\gamma \in F_{i+1}$. Two cases:

1. $\gamma \in \text{Th}(F_i)$ implies $\gamma \in \text{Th}(E_i)$ (because $F_i \subseteq E_i$ by IH), and therefore $\gamma \in E_{i+1}$.
2. Otherwise $\frac{\alpha : \beta}{\gamma} \in D$, $\alpha \in F_i$, $\neg\beta \notin F$. However, then we have $\alpha \in E_i$ (because $F_i \subseteq E_i$) and $\neg\beta \notin E$ (because of $E \subseteq F$), i.e., $\gamma \in E_{i+1}$.

Properties of Extensions

Theorem

If E and F are extensions of (D, W) such that $E \subseteq F$, then $E = F$.

Proof sketch.

$E = \bigcup_{i=0}^{\infty} E_i$ and $F = \bigcup_{i=0}^{\infty} F_i$. Use induction to show $F_i \subseteq E_i$.

Base case $i = 0$: Trivially $E_0 = F_0 = W$.

Inductive case $i \geq 1$: Assume $\gamma \in F_{i+1}$. Two cases:

1. $\gamma \in \mathrm{Th}(F_i)$ implies $\gamma \in \mathrm{Th}(E_i)$ (because $F_i \subseteq E_i$ by IH), and therefore $\gamma \in E_{i+1}$.
2. Otherwise $\frac{\alpha : \beta}{\gamma} \in D$, $\alpha \in F_i$, $\neg\beta \notin F$. However, then we have $\alpha \in E_i$ (because $F_i \subseteq E_i$) and $\neg\beta \notin E$ (because of $E \subseteq F$), i.e., $\gamma \in E_{i+1}$.

□

Properties of Extensions

Theorem

If E and F are extensions of (D, W) such that $E \subseteq F$, then $E = F$.

Proof sketch.

$E = \bigcup_{i=0}^{\infty} E_i$ and $F = \bigcup_{i=0}^{\infty} F_i$. Use induction to show $F_i \subseteq E_i$.

Base case $i = 0$: Trivially $E_0 = F_0 = W$.

Inductive case $i \geq 1$: Assume $\gamma \in F_{i+1}$. Two cases:

1. $\gamma \in \mathrm{Th}(F_i)$ implies $\gamma \in \mathrm{Th}(E_i)$ (because $F_i \subseteq E_i$ by IH), and therefore $\gamma \in E_{i+1}$.
2. Otherwise $\frac{\alpha : \beta}{\gamma} \in D$, $\alpha \in F_i$, $\neg\beta \notin F$. However, then we have $\alpha \in E_i$ (because $F_i \subseteq E_i$) and $\neg\beta \notin E$ (because of $E \subseteq F$), i.e., $\gamma \in E_{i+1}$.

□

Normal Default Theories

All defaults in a normal default theory are normal:

$$\frac{\alpha : \beta}{\beta}.$$

Theorem

Normal default theories have at least one extension.

Proof sketch.

If W inconsistent, trivial. Otherwise construct

$$\begin{array}{rcl} E_0 & = & W \\ E_{i+1} & = & \mathrm{Th}(E_i) \cup T_i \end{array} \qquad E = \bigcup_{i=0}^{\infty} E_i$$

where T_i is a maximal set s.t. (1) $E_i \cup T_i$ is consistent and (2) if $\beta \in T_i$ then there is $\frac{\alpha : \beta}{\beta} \in D$ and $\alpha \in E_i$.

Show: $T_i = \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$ for all $i \geq 0$. □

Normal Default Theories

All defaults in a normal default theory are normal:

$$\frac{\alpha : \beta}{\beta}.$$

Theorem

Normal default theories have at least one extension.

Proof sketch.

If W inconsistent, trivial. Otherwise construct

$$\begin{aligned} E_0 &= W \\ E_{i+1} &= \mathrm{Th}(E_i) \cup T_i \end{aligned} \qquad E = \bigcup_{i=0}^{\infty} E_i$$

where T_i is a maximal set s.t. (1) $E_i \cup T_i$ is consistent and (2) if $\beta \in T_i$ then there is $\frac{\alpha : \beta}{\beta} \in D$ and $\alpha \in E_i$.

Show: $T_i = \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$ for all $i \geq 0$. □

Normal Default Theories

All defaults in a normal default theory are normal:

$$\frac{\alpha : \beta}{\beta}.$$

Theorem

Normal default theories have at least one extension.

Proof sketch.

If W inconsistent, trivial. Otherwise construct

$$\begin{aligned} E_0 &= W \\ E_{i+1} &= \mathrm{Th}(E_i) \cup T_i \end{aligned} \qquad E = \bigcup_{i=0}^{\infty} E_i$$

where T_i is a maximal set s.t. (1) $E_i \cup T_i$ is consistent and (2) if $\beta \in T_i$ then there is $\frac{\alpha : \beta}{\beta} \in D$ and $\alpha \in E_i$.

Show: $T_i = \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$ for all $i \geq 0$. □

Normal Default Theories

All defaults in a normal default theory are normal:

$$\frac{\alpha : \beta}{\beta}.$$

Theorem

Normal default theories have at least one extension.

Proof sketch.

If W inconsistent, trivial. Otherwise construct

$$\begin{aligned} E_0 &= W \\ E_{i+1} &= \text{Th}(E_i) \cup T_i \end{aligned} \qquad E = \bigcup_{i=0}^{\infty} E_i$$

where T_i is a maximal set s.t. (1) $E_i \cup T_i$ is consistent and (2) if $\beta \in T_i$ then there is $\frac{\alpha : \beta}{\beta} \in D$ and $\alpha \in E_i$.

Show: $T_i = \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$ for all $i \geq 0$. □

Normal Default Theories

All defaults in a normal default theory are normal:

$$\frac{\alpha : \beta}{\beta}.$$

Theorem

Normal default theories have at least one extension.

Proof sketch.

If W inconsistent, trivial. Otherwise construct

$$\begin{aligned} E_0 &= W \\ E_{i+1} &= \text{Th}(E_i) \cup T_i \end{aligned} \qquad E = \bigcup_{i=0}^{\infty} E_i$$

where T_i is a maximal set s.t. (1) $E_i \cup T_i$ is consistent and (2) if $\beta \in T_i$ then there is $\frac{\alpha : \beta}{\beta} \in D$ and $\alpha \in E_i$.

Show: $T_i = \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$ for all $i \geq 0$. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \mathrm{Th}(E_i) \cup \left\{ \beta \;\middle|\; \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \mathrm{Th}(E_i) \cup \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \text{Th}(E_i) \cup \left\{ \beta \;\middle|\; \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \mathrm{Th}(E_i) \cup \left\{ \beta \;\middle|\; \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \mathrm{Th}(E_i) \cup \left\{ \beta \;\middle|\; \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \mathrm{Th}(E_i) \cup \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \mathrm{Th}(E_i) \cup \left\{ \beta \;\middle|\; \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Normal Default Theories: Extensions are Orthogonal

Theorem (Orthogonality)

Let E and F be two extensions of a normal default theory. Then $E \cup F$ is inconsistent.

Proof.

Let $E = \bigcup E_i$ and $F = \bigcup F_i$ with

$$E_{i+1} = \mathrm{Th}(E_i) \cup \left\{ \beta \,\middle|\, \frac{\alpha : \beta}{\beta} \in D, \alpha \in E_i, \neg\beta \notin E \right\}$$

and the same for F. Since $E \neq F$, there exists a smallest i such that $E_{i+1} \neq F_{i+1}$. This means there exists $\frac{\alpha : \beta}{\beta} \in D$ with $\alpha \in E_i = F_i$ but $\beta \in E_{i+1}$ and $\beta \notin F_{i+1}$. This is only possible if $\neg\beta \in F$. This means $\beta \in E$ and $\neg\beta \in F$, i.e., $E \cup F$ is inconsistent. □

Default Proofs in Normal Default Theories

Definition

A default proof of γ in a normal default theory (D, W) is a finite sequence of defaults $(\delta_i = \frac{\alpha_i : \beta_i}{\beta_i})_{i=1,\dots,n}$ such that

1. $W \cup \{\beta_1, \dots, \beta_n\} \vdash \gamma$,
2. $W \cup \{\beta_1, \dots, \beta_n\}$ is consistent,and
3. $W \cup \{\beta_1, \dots, \beta_k\} \vdash \alpha_{k+1}$, for $0 \leq k \leq n-1$.

Theorem

Let $\Delta = \langle D, W \rangle$ be a normal default theory so that W is consistent. Then γ has a default proof in Δ iff there exists an extension E of Δ such that $\gamma \in E$.

Test 2 (consistency) in the proof procedure suggests that default provability is not even semi-decidable.

Default Proofs in Normal Default Theories

Definition

A default proof of γ in a normal default theory (D, W) is a finite sequence of defaults $(\delta_i = \frac{\alpha_i : \beta_i}{\beta_i})_{i=1,\ldots,n}$ such that

1. $W \cup \{\beta_1, \ldots, \beta_n\} \vdash \gamma$,
2. $W \cup \{\beta_1, \ldots, \beta_n\}$ is consistent, and
3. $W \cup \{\beta_1, \ldots, \beta_k\} \vdash \alpha_{k+1}$, for $0 \leq k \leq n - 1$.

Theorem

Let $\Delta = \langle D, W \rangle$ be a normal default theory so that W is consistent. Then γ has a default proof in Δ iff there exists an extension E of Δ such that $\gamma \in E$.

Test 2 (consistency) in the proof procedure suggests that default provability is not even semi-decidable.

Default Proofs in Normal Default Theories

Definition

A **default proof of γ** in a normal default theory (D, W) is a finite sequence of defaults $(\delta_i = \frac{\alpha_i : \beta_i}{\beta_i})_{i=1,\dots,n}$ such that

1. $W \cup \{\beta_1, \dots, \beta_n\} \vdash \gamma$,
2. $W \cup \{\beta_1, \dots, \beta_n\}$ is consistent,and
3. $W \cup \{\beta_1, \dots, \beta_k\} \vdash \alpha_{k+1}$, for $0 \leq k \leq n-1$.

Theorem

Let $\Delta = \langle D, W \rangle$ be a normal default theory so that W is consistent. Then γ has a default proof in Δ iff there exists an extension E of Δ such that $\gamma \in E$.

Test 2 (consistency) in the proof procedure suggests that default provability is not even semi-decidable.

Default Proofs in Normal Default Theories

Definition

A default proof of γ in a normal default theory (D, W) is a finite sequence of defaults $(\delta_i = \frac{\alpha_i : \beta_i}{\beta_i})_{i=1,\ldots,n}$ such that

1. $W \cup \{\beta_1, \ldots, \beta_n\} \vdash \gamma$,
2. $W \cup \{\beta_1, \ldots, \beta_n\}$ is consistent,and
3. $W \cup \{\beta_1, \ldots, \beta_k\} \vdash \alpha_{k+1}$, for $0 \leq k \leq n-1$.

Theorem

Let $\Delta = \langle D, W \rangle$ be a normal default theory so that W is consistent. Then γ has a default proof in Δ iff there exists an extension E of Δ such that $\gamma \in E$.

Test 2 (consistency) in the proof procedure suggests that default provability is not even semi-decidable.

Decidability

Theorem

It is not semi-decidable to test whether a formula follows (skeptically or credulously) from a default theory.

Proof.

Let (D, W) be a default theory with $W = \emptyset$ and $D = \left\{ \frac{:\beta}{\beta} \right\}$ with β an arbitrary closed PL1 formula. Clearly, β is in some/all extensions of (D, W) if and only if β is satisfiable. The existence of a semi-decision procedure for default proofs implies that there is a semi-decision procedure for satisfiability in PL1.

But this is not possible because PL1 validity is semi-decidable and this together with semi-decidability of PL1 satisfiability would imply decidability of PL1, which is not the case. □

Decidability

Theorem

It is not semi-decidable to test whether a formula follows (skeptically or credulously) from a default theory.

Proof.

Let (D, W) be a default theory with $W = \emptyset$ and $D = \left\{ \frac{:\beta}{\beta} \right\}$ with β an arbitrary closed PL1 formula. Clearly, β is in some/all extensions of (D, W) if and only if β is satisfiable. The existence of a semi-decision procedure for default proofs implies that there is a semi-decision procedure for satisfiability in PL1.

But this is not possible because PL1 validity is semi-decidable and this together with semi-decidability of PL1 satisfiability would imply decidability of PL1, which is not the case. □

Decidability

Theorem

It is not semi-decidable to test whether a formula follows (skeptically or credulously) from a default theory.

Proof.

Let (D, W) be a default theory with $W = \emptyset$ and $D = \left\{ \frac{:\beta}{\beta} \right\}$ with β an arbitrary closed PL1 formula. Clearly, β is in some/all extensions of (D, W) if and only if β is satisfiable. The existence of a semi-decision procedure for default proofs implies that there is a semi-decision procedure for satisfiability in PL1.

But this is not possible because PL1 validity is semi-decidable and this together with semi-decidability of PL1 satisfiability would imply decidability of PL1, which is not the case. □

Propositional Default Logic

- Propositional DL is decidable.
- How difficult is reasoning in propositional DL?
- The skeptical default reasoning problem (does φ follow from Δ skeptically: $\Delta \mid\!\sim \varphi$?) is called PDS, credulous reasoning is called LPDS.
- (L)PDS is co-NP-hard (let $D = \emptyset$, $W = \emptyset$) and NP-hard (let $W = \emptyset$, $D = \left\{ \frac{:\beta}{\beta} \right\}$).

Propositional Default Logic

- Propositional DL is decidable.
- How difficult is reasoning in propositional DL?
- The skeptical default reasoning problem (does φ follow from Δ skeptically: $\Delta \mathrel{|\!\sim} \varphi$?) is called PDS, credulous reasoning is called LPDS.
- (L)PDS is co-NP-hard (let $D = \emptyset$, $W = \emptyset$) and NP-hard (let $W = \emptyset$, $D = \left\{ \frac{:\beta}{\beta} \right\}$).

Propositional Default Logic

- Propositional DL is decidable.
- How difficult is reasoning in propositional DL?
- The skeptical default reasoning problem (does φ follow from Δ skeptically: $\Delta \mid\!\sim \varphi$?) is called PDS, credulous reasoning is called LPDS.
- (L)PDS is co-NP-hard (let $D = \emptyset$, $W = \emptyset$) and NP-hard (let $W = \emptyset$, $D = \left\{ \frac{:\beta}{\beta} \right\}$).

Propositional Default Logic

- Propositional DL is decidable.
- How difficult is reasoning in propositional DL?
- The skeptical default reasoning problem (does φ follow from Δ skeptically: $\Delta \mid\!\sim \varphi$?) is called PDS, credulous reasoning is called LPDS.
- (L)PDS is co-NP-hard (let $D = \emptyset$, $W = \emptyset$) and NP-hard (let $W = \emptyset$, $D = \left\{ \frac{:\beta}{\beta} \right\}$).

Oracle Turing Machines

- Oracle Turing machine is a Turing machine (DTM, NDTM) with the possibility to query an oracle, another Turing machine without resource restrictions, whether it accepts or reject a given string. Computation by the oracle does not cost anything!
- Formalization:
  - a tape onto which strings for the oracle are written,
  - a yes/no answer from the oracle depending on whether it accepts or rejects the input string.

Oracle Turing Machines

- Oracle Turing machine is a Turing machine (DTM, NDTM) with the possibility to query an oracle, another Turing machine without resource restrictions, whether it accepts or reject a given string. Computation by the oracle does not cost anything!
- Formalization:
  - a tape onto which strings for the oracle are written,
  - a yes/no answer from the oracle depending on whether it accepts or rejects the input string.

Complexity Classes Defined in Terms of Oracle TMs

1. P^{NP} = decision problems solved by poly-time DTMs with an oracle for a decision problem in NP.
2. NP^{NP} = decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
3. co-NP^{NP} = complements of decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
4. $NP^{NP^{NP}}$ = ...
   and so on

Complexity Classes Defined in Terms of Oracle TMs

1. P^{NP} = decision problems solved by poly-time DTMs with an oracle for a decision problem in NP.
2. NP^{NP} = decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
3. $co\text{-}NP^{NP}$ = complements of decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
4. $NP^{NP^{NP}}$ = ...
   and so on

Complexity Classes Defined in Terms of Oracle TMs

1. P^{NP} = decision problems solved by poly-time DTMs with an oracle for a decision problem in NP.
2. NP^{NP} = decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
3. $co\text{-}NP^{NP}$ = complements of decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
4. $NP^{NP^{NP}}$ = ...
   and so on

Complexity Classes Defined in Terms of Oracle TMs

1. P^{NP} = decision problems solved by poly-time DTMs with an oracle for a decision problem in NP.
2. NP^{NP} = decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
3. co-NP^{NP} = complements of decision problems solved by poly-time NDTMs with an oracle for a decision problem in NP.
4. $NP^{NP^{NP}}$ = ...
   and so on

The Polynomial Hierarchy

There are problems that are NP-hard and co-NP-hard and in PSPACE, but do not seem to be complete for NP, co-NP or PSPACE.

The polynomial hierarchy PH

An infinite hierarchy of complexity classes:

$$\begin{array}{rclrclrcl} \Sigma_0^p & = & P & \Pi_0^p & = & P & \Delta_0^p & = & P \\ \Sigma_{i+1}^p & = & \mathrm{NP}^{\Sigma_i^p} & \Pi_{i+1}^p & = & \text{co-}\Sigma_{i+1}^p & \Delta_{i+1}^p & = & P^{\Sigma_i^p} \end{array}$$

- PH = $\bigcup_{i\geq 0}(\Sigma_i^p \cup \Pi_i^p \cup \Delta_i^p) \subseteq$PSPACE
- NP = Σ_1^p, co-NP = Π_1^p

The Polynomial Hierarchy

There are problems that are NP-hard and co-NP-hard and in PSPACE, but do not seem to be complete for NP, co-NP or PSPACE.

The polynomial hierarchy PH

An infinite hierarchy of complexity classes:

$$\Sigma_0^p = P \qquad \Pi_0^p = P \qquad \Delta_0^p = P$$
$$\Sigma_{i+1}^p = NP^{\Sigma_i^p} \qquad \Pi_{i+1}^p = \text{co-}\Sigma_{i+1}^p \qquad \Delta_{i+1}^p = P^{\Sigma_i^p}$$

- PH = $\bigcup_{i\geq 0}(\Sigma_i^p \cup \Pi_i^p \cup \Delta_i^p) \subseteq$PSPACE
- NP = Σ_1^p, co-NP = Π_1^p

The Polynomial Hierarchy

There are problems that are NP-hard and co-NP-hard and in PSPACE, but do not seem to be complete for NP, co-NP or PSPACE.

The polynomial hierarchy PH

An infinite hierarchy of complexity classes:

$$\begin{array}{lll} \Sigma_0^p = P & \Pi_0^p = P & \Delta_0^p = P \\ \Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p} & \Pi_{i+1}^p = \text{co-}\Sigma_{i+1}^p & \Delta_{i+1}^p = P^{\Sigma_i^p} \end{array}$$

- PH = $\bigcup_{i \geq 0}(\Sigma_i^p \cup \Pi_i^p \cup \Delta_i^p) \subseteq$PSPACE
- NP = Σ_1^p, co-NP = Π_1^p

The Polynomial Hierarchy

There are problems that are NP-hard and co-NP-hard and in PSPACE, but do not seem to be complete for NP, co-NP or PSPACE.

The polynomial hierarchy PH

An infinite hierarchy of complexity classes:

$$\begin{array}{rclrclrcl} \Sigma_0^p & = & P & \Pi_0^p & = & P & \Delta_0^p & = & P \\ \Sigma_{i+1}^p & = & \mathrm{NP}^{\Sigma_i^p} & \Pi_{i+1}^p & = & \text{co-}\Sigma_{i+1}^p & \Delta_{i+1}^p & = & P^{\Sigma_i^p} \end{array}$$

- PH = $\bigcup_{i\geq 0}(\Sigma_i^p \cup \Pi_i^p \cup \Delta_i^p) \subseteq$PSPACE
- NP = Σ_1^p, co-NP = Π_1^p

The Polynomial Hierarchy

There are problems that are NP-hard and co-NP-hard and in PSPACE, but do not seem to be complete for NP, co-NP or PSPACE.

The polynomial hierarchy PH

An infinite hierarchy of complexity classes:

$$\begin{array}{rclrclrcl} \Sigma_0^p & = & P & \Pi_0^p & = & P & \Delta_0^p & = & P \\ \Sigma_{i+1}^p & = & \mathrm{NP}^{\Sigma_i^p} & \Pi_{i+1}^p & = & \text{co-}\Sigma_{i+1}^p & \Delta_{i+1}^p & = & P^{\Sigma_i^p} \end{array}$$

- PH = $\bigcup_{i \geq 0}(\Sigma_i^p \cup \Pi_i^p \cup \Delta_i^p) \subseteq$ PSPACE
- NP = Σ_1^p, co-NP = Π_1^p

The Polynomial Hierarchy

There are problems that are NP-hard and co-NP-hard and in PSPACE, but do not seem to be complete for NP, co-NP or PSPACE.

The polynomial hierarchy PH

An infinite hierarchy of complexity classes:

$$\Sigma_0^p = P \qquad \Pi_0^p = P \qquad \Delta_0^p = P$$
$$\Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p} \qquad \Pi_{i+1}^p = \text{co-}\Sigma_{i+1}^p \qquad \Delta_{i+1}^p = P^{\Sigma_i^p}$$

- PH = $\bigcup_{i\geq 0}(\Sigma_i^p \cup \Pi_i^p \cup \Delta_i^p)$ $\subseteq$PSPACE
- NP = Σ_1^p, co-NP = Π_1^p

Quantified Boolean Formulae: Definition

- If ϕ is a propositional formula and σ is a sequence of $\exists p$ and $\forall p$, one for every $p \in P$, then $\sigma\phi$ is a QBF.
- A formula $\exists x \phi$ is true if and only if $\phi[\top/x] \vee \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true or* $\phi[\bot/x]$ *is true*.)
- A formula $\forall x \phi$ is true if and only if $\phi[\top/x] \wedge \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true and* $\phi[\bot/x]$ *is true*.)
- This definition directly leads to an AND/OR tree traversal algorithm for evaluating QBF.

Quantified Boolean Formulae: Definition

- If ϕ is a propositional formula and σ is a sequence of $\exists p$ and $\forall p$, one for every $p \in P$, then $\sigma\phi$ is a QBF.
- A formula $\exists x\phi$ is true if and only if $\phi[\top/x] \vee \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true or* $\phi[\bot/x]$ *is true*.)
- A formula $\forall x\phi$ is true if and only if $\phi[\top/x] \wedge \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true and* $\phi[\bot/x]$ *is true*.)
- This definition directly leads to an AND/OR tree traversal algorithm for evaluating QBF.

Quantified Boolean Formulae: Definition

- If ϕ is a propositional formula and σ is a sequence of $\exists p$ and $\forall p$, one for every $p \in P$, then $\sigma\phi$ is a QBF.
- A formula $\exists x\phi$ is true if and only if $\phi[\top/x] \vee \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true or* $\phi[\bot/x]$ *is true*.)
- A formula $\forall x\phi$ is true if and only if $\phi[\top/x] \wedge \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true and* $\phi[\bot/x]$ *is true*.)
- This definition directly leads to an AND/OR tree traversal algorithm for evaluating QBF.

Quantified Boolean Formulae: Definition

- If ϕ is a propositional formula and σ is a sequence of $\exists p$ and $\forall p$, one for every $p \in P$, then $\sigma\phi$ is a QBF.
- A formula $\exists x \phi$ is true if and only if $\phi[\top/x] \vee \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true or* $\phi[\bot/x]$ *is true*.)
- A formula $\forall x \phi$ is true if and only if $\phi[\top/x] \wedge \phi[\bot/x]$ is true. (Equivalently, $\phi[\top/x]$ *is true and* $\phi[\bot/x]$ *is true*.)
- This definition directly leads to an AND/OR tree traversal algorithm for evaluating QBF.

Quantified Boolean Formulae: Definition

The evaluation problem of QBF generalizes both the *satisfiability* and *validity/tautology problems* of the propositional logic.
The latter are respectively NP-complete and co-NP-complete whereas the former is PSPACE-complete.

Example

The formulae $\forall x \exists y (x \leftrightarrow y)$ and $\exists x \exists y (x \wedge y)$ are true.

Example

The formulae $\exists x \forall y (x \leftrightarrow y)$ and $\forall x \forall y (x \vee y)$ are false.

Quantified Boolean Formulae: Definition

The evaluation problem of QBF generalizes both the *satisfiability* and *validity/tautology problems* of the propositional logic.

The latter are respectively NP-complete and co-NP-complete whereas the former is PSPACE-complete.

Example

The formulae $\forall x \exists y(x \leftrightarrow y)$ and $\exists x \exists y(x \wedge y)$ are true.

Example

The formulae $\exists x \forall y(x \leftrightarrow y)$ and $\forall x \forall y(x \vee y)$ are false.

Quantified Boolean Formulae: Definition

The evaluation problem of QBF generalizes both the *satisfiability* and *validity/tautology problems* of the propositional logic.
The latter are respectively NP-complete and co-NP-complete whereas the former is PSPACE-complete.

Example

The formulae $\forall x \exists y (x \leftrightarrow y)$ and $\exists x \exists y (x \wedge y)$ are true.

Example

The formulae $\exists x \forall y (x \leftrightarrow y)$ and $\forall x \forall y (x \vee y)$ are false.

The Polynomial Hierarchy: Connection to QBF

Truth of QBFs with prefix $\overbrace{\forall\exists\forall\cdots}^{i}$ is Π_i^p-complete.

Truth of QBFs with prefix $\overbrace{\exists\forall\exists\cdots}^{i}$ is Σ_i^p-complete.

Special cases corresponding to SAT and TAUT:
The truth of QBFs with prefix $\exists x_1^1 \cdots x_n^1$ is NP$=\Sigma_1^p$-complete.
The truth of QBFs with prefix $\forall x_1^1 \cdots x_n^1$ is co-NP$=\Pi_1^p$-complete.

Skeptical Reasoning in Propositional DL

Lemma

PDS $\in \Pi_2^p$.

Proof.

We show that the complementary problem UNPDS (is there an extension E such that $\varphi \notin E$) is in Σ_2^p.

The *algorithm*: Guess set $T \subseteq D$ of defaults: those that are applied.

Verify that defaults in T lead to E, using a SAT oracle and the guessed $E = \text{Th}\left(\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W\right)$.

Verify that $\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W \not\models \varphi$ (SAT oracle).

$\rightsquigarrow$ UNPDS $\in \Sigma_2^p$. □

Note: LPDS $\in \Sigma_2^p$.

Skeptical Reasoning in Propositional DL

Lemma

$PDS \in \Pi_2^p$.

Proof.

We show that the complementary problem UNPDS (is there an extension E such that $\varphi \notin E$) is in Σ_2^p.

The *algorithm*: Guess set $T \subseteq D$ of defaults: those that are applied.

Verify that defaults in T lead to E, using a SAT oracle and the guessed $E = \text{Th}\left(\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W\right)$.

Verify that $\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W \not\models \varphi$ (SAT oracle).

$\rightsquigarrow$ UNPDS $\in \Sigma_2^p$. □

Note: LPDS $\in \Sigma_2^p$.

Skeptical Reasoning in Propositional DL

Lemma

PDS $\in \Pi_2^p$.

Proof.

We show that the complementary problem UNPDS (is there an extension E such that $\varphi \notin E$) is in Σ_2^p.

The *algorithm*: Guess set $T \subseteq D$ of defaults: those that are applied.

Verify that defaults in T lead to E, using a SAT oracle and the guessed $E = \text{Th}\left(\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W\right)$.

Verify that $\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W \not\models \varphi$ (SAT oracle).

$\rightsquigarrow$ UNPDS $\in \Sigma_2^p$. □

Note: LPDS $\in \Sigma_2^p$.

Skeptical Reasoning in Propositional DL

Lemma

PDS $\in \Pi_2^p$.

Proof.

We show that the complementary problem UNPDS (is there an extension E such that $\varphi \notin E$) is in Σ_2^p.

The *algorithm*: Guess set $T \subseteq D$ of defaults: those that are applied.

Verify that defaults in T lead to E, using a SAT oracle and the guessed $E = \text{Th}\left(\left\{\gamma \middle| \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W\right)$.

Verify that $\left\{\gamma \middle| \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W \not\models \varphi$ (SAT oracle).

$\rightsquigarrow$ UNPDS $\in \Sigma_2^p$. □

Note: LPDS $\in \Sigma_2^p$.

Skeptical Reasoning in Propositional DL

Lemma

PDS $\in \Pi_2^p$.

Proof.

We show that the complementary problem UNPDS (is there an extension E such that $\varphi \notin E$) is in Σ_2^p.

The *algorithm*: Guess set $T \subseteq D$ of defaults: those that are applied.

Verify that defaults in T lead to E, using a SAT oracle and the guessed $E = \text{Th}\left(\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W\right)$.

Verify that $\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W \not\models \varphi$ (SAT oracle).

$\rightsquigarrow$ UNPDS $\in \Sigma_2^p$. □

Note: LPDS $\in \Sigma_2^p$.

Skeptical Reasoning in Propositional DL

Lemma

PDS $\in \Pi_2^p$.

Proof.

We show that the complementary problem UNPDS (is there an extension E such that $\varphi \notin E$) is in Σ_2^p.

The *algorithm*: Guess set $T \subseteq D$ of defaults: those that are applied.

Verify that defaults in T lead to E, using a SAT oracle and the guessed $E = \text{Th}\left(\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W\right)$.

Verify that $\left\{\gamma | \frac{\alpha:\beta}{\gamma} \in T\right\} \cup W \not\vdash \varphi$ (SAT oracle).

$\leadsto$ UNPDS $\in \Sigma_2^p$. □

Note: LPDS $\in \Sigma_2^p$.

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists\vec{a}\,\forall\vec{b}\phi(\vec{a},\vec{b})$ with $\vec{a} = a_1,\dots,a_n$ and $\vec{b} = b_1,\dots,b_m$ construct $\Delta = (D, W)$ with

$$D = \left\{ \frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\mathrel{|}\!\sim \neg\phi(\vec{a},\vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a},\vec{b}) \notin E$
iff there is E s.t. $\phi(\vec{a},\vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)
iff there is $A \subset \{a_1, \neg a_1, \dots, a_n, \neg a_n\}$ s.t. $A \models \phi(\vec{a},\vec{b})$
iff $\exists\vec{a}\forall\vec{b}\phi(\vec{a},\vec{b})$ is true.

□

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists\vec{a}\,\forall\vec{b}\phi(\vec{a},\vec{b})$ with $\vec{a} = a_1, \ldots, a_n$ and $\vec{b} = b_1, \ldots, b_m$ construct $\Delta = (D, W)$ with

$$D = \left\{ \frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\models \neg\phi(\vec{a},\vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a},\vec{b}) \notin E$

iff there is E s.t. $\phi(\vec{a},\vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)

iff there is $A \subseteq \{a_1, \neg a_1, \ldots, a_n, \neg a_n\}$ s.t. $A \models \phi(\vec{a},\vec{b})$

iff $\exists\vec{a}\forall\vec{b}\phi(\vec{a},\vec{b})$ is true.

□

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists\vec{a}\,\forall\vec{b}\phi(\vec{a},\vec{b})$ with $\vec{a} = a_1, \ldots, a_n$ and $\vec{b} = b_1, \ldots, b_m$ construct $\Delta = (D, W)$ with

$$D = \left\{ \frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\models \neg\phi(\vec{a},\vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a},\vec{b}) \notin E$

iff there is E s.t. $\phi(\vec{a},\vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)

iff there is $A \subseteq \{a_1, \neg a_1, \ldots, a_n, \neg a_n\}$ s.t. $A \models \phi(\vec{a},\vec{b})$

iff $\exists\vec{a}\forall\vec{b}\phi(\vec{a},\vec{b})$ is true.

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists\vec{a}\,\forall\vec{b}\phi(\vec{a},\vec{b})$ with $\vec{a}=a_1,\ldots,a_n$ and $\vec{b}=b_1,\ldots,b_m$ construct $\Delta=(D,W)$ with

$$D = \left\{\frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})}\right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\approx \neg\phi(\vec{a},\vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a},\vec{b}) \notin E$

iff there is E s.t. $\phi(\vec{a},\vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)

iff there is $A \subset \{a_1,\neg a_1,\ldots,a_n,\neg a_n\}$ s.t. $A \models \phi(\vec{a},\vec{b})$

iff $\exists\vec{a}\forall\vec{b}\phi(\vec{a},\vec{b})$ is true.

□

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists\vec{a}\,\forall\vec{b}\phi(\vec{a},\vec{b})$ with $\vec{a} = a_1, \ldots, a_n$ and $\vec{b} = b_1, \ldots, b_m$ construct $\Delta = (D, W)$ with

$$D = \left\{ \frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\models \neg\phi(\vec{a},\vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a},\vec{b}) \notin E$

iff there is E s.t. $\phi(\vec{a},\vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)

iff there is $A \subset \{a_1, \neg a_1, \ldots, a_n, \neg a_n\}$ s.t. $A \models \phi(\vec{a},\vec{b})$

iff $\exists\vec{a}\forall\vec{b}\phi(\vec{a},\vec{b})$ is true.

□

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists\vec{a}\,\forall\vec{b}\phi(\vec{a},\vec{b})$ with $\vec{a} = a_1, \ldots, a_n$ and $\vec{b} = b_1, \ldots, b_m$ construct $\Delta = (D, W)$ with

$$D = \left\{ \frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\models \neg\phi(\vec{a},\vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a},\vec{b}) \notin E$

iff there is E s.t. $\phi(\vec{a},\vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)

iff there is $A \subset \{a_1, \neg a_1, \ldots, a_n, \neg a_n\}$ s.t. $A \models \phi(\vec{a},\vec{b})$

iff $\exists\vec{a}\forall\vec{b}\phi(\vec{a},\vec{b})$ is true.

□

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists \vec{a}\,\forall \vec{b} \phi(\vec{a}, \vec{b})$ with $\vec{a} = a_1, \ldots, a_n$ and $\vec{b} = b_1, \ldots, b_m$ construct $\Delta = (D, W)$ with

$$D = \left\{ \frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\models \neg\phi(\vec{a}, \vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a}, \vec{b}) \notin E$

iff there is E s.t. $\phi(\vec{a}, \vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)

iff there is $A \subset \{a_1, \neg a_1, \ldots, a_n, \neg a_n\}$ s.t. $A \models \phi(\vec{a}, \vec{b})$

iff $\exists \vec{a} \forall \vec{b} \phi(\vec{a}, \vec{b})$ is true.

□

Π_2^p-Hardness

Lemma

PDS is Π_2^p-hard.

Proof.

Reduction from 2QBF to UNPDS: For $\exists \vec{a}\, \forall \vec{b} \phi(\vec{a}, \vec{b})$ with $\vec{a} = a_1, \ldots, a_n$ and $\vec{b} = b_1, \ldots, b_m$ construct $\Delta = (D, W)$ with

$$D = \left\{ \frac{:a_i}{a_i}, \frac{:\neg a_i}{\neg a_i}, \frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \right\}, \quad W = \emptyset$$

No extension contains both a_i and $\neg a_i$. Now

$\Delta \not\mid\!\sim \neg\phi(\vec{a}, \vec{b})$ iff there is extension E s.t. $\neg\phi(\vec{a}, \vec{b}) \notin E$
iff there is E s.t. $\phi(\vec{a}, \vec{b}) \in E$ (by $\frac{:\neg\phi(\vec{a},\vec{b})}{\neg\phi(\vec{a},\vec{b})} \in D$)
iff there is $A \subset \{a_1, \neg a_1, \ldots, a_n, \neg a_n\}$ s.t. $A \models \phi(\vec{a}, \vec{b})$
iff $\exists \vec{a} \forall \vec{b} \phi(\vec{a}, \vec{b})$ is true.

□

Conclusions & Remarks

Theorem

PDS is Π_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

Theorem

LPDS is Σ_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

- PDS is *"easier"* than reasoning in most modal logics.
- General and normal defaults have the same complexity.
- Polynomial special cases cannot be achieved by restricting, for example, to Horn clauses (satisfiability testing in polynomial time).
- It is necessary to restrict the underlying monotonic reasoning problem and the *number of extensions*.
- Similar results hold for other nonmonotonic logics.

Conclusions & Remarks

Theorem

PDS is Π_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

Theorem

LPDS is Σ_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

- PDS is *"easier"* than reasoning in most modal logics.
- General and normal defaults have the same complexity.
- Polynomial special cases cannot be achieved by restricting, for example, to Horn clauses (satisfiability testing in polynomial time).
- It is necessary to restrict the underlying monotonic reasoning problem and the *number of extensions*.
- Similar results hold for other nonmonotonic logics.

Conclusions & Remarks

Theorem

PDS is Π_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

Theorem

LPDS is Σ_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

- PDS is *"easier"* than reasoning in most modal logics.
- General and normal defaults have the same complexity.
- Polynomial special cases cannot be achieved by restricting, for example, to Horn clauses (satisfiability testing in polynomial time).
- It is necessary to restrict the underlying monotonic reasoning problem and the *number of extensions*.
- Similar results hold for other nonmonotonic logics.

Conclusions & Remarks

Theorem

PDS is Π_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

Theorem

LPDS is Σ_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

- PDS is *"easier"* than reasoning in most modal logics.
- General and normal defaults have the same complexity.
- Polynomial special cases cannot be achieved by restricting, for example, to Horn clauses (satisfiability testing in polynomial time).
- It is necessary to restrict the underlying monotonic reasoning problem and the *number of extensions*.
- Similar results hold for other nonmonotonic logics.

Conclusions & Remarks

Theorem

PDS is Π_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

Theorem

LPDS is Σ_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

- PDS is *"easier"* than reasoning in most modal logics.
- General and normal defaults have the same complexity.
- Polynomial special cases cannot be achieved by restricting, for example, to Horn clauses (satisfiability testing in polynomial time).
- It is necessary to restrict the underlying monotonic reasoning problem and the *number of extensions*.
- Similar results hold for other nonmonotonic logics.

Conclusions & Remarks

Theorem

PDS is Π_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

Theorem

LPDS is Σ_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

- PDS is *"easier"* than reasoning in most modal logics.
- General and normal defaults have the same complexity.
- Polynomial special cases cannot be achieved by restricting, for example, to Horn clauses (satisfiability testing in polynomial time).
- It is necessary to restrict the underlying monotonic reasoning problem and the *number of extensions*.
- Similar results hold for other nonmonotonic logics.

Conclusions & Remarks

Theorem

PDS is Π_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

Theorem

LPDS is Σ_2^p-complete, even for defaults of the form $\frac{:\alpha}{\alpha}$.

- PDS is *"easier"* than reasoning in most modal logics.
- General and normal defaults have the same complexity.
- Polynomial special cases cannot be achieved by restricting, for example, to Horn clauses (satisfiability testing in polynomial time).
- It is necessary to restrict the underlying monotonic reasoning problem and the *number of extensions*.
- Similar results hold for other nonmonotonic logics.

Semi-Normal Defaults (1)

Semi-normal defaults are sometimes useful:

$$\frac{\alpha : \beta \wedge \gamma}{\beta}$$

Important when one has *interacting* defaults:

$$\frac{\texttt{Adult}(x) : \texttt{Employed}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x) : \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

$$\frac{\texttt{Student}(x) : \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

For `Student(TOM)` we get two extensions: one with `Employed(Tom)` and the other one with $\neg$`Employed(Tom)`.

Since the third rule is *"more specific"*, we may prefer it.

Semi-Normal Defaults (1)

Semi-normal defaults are sometimes useful:

$$\frac{\alpha : \beta \wedge \gamma}{\beta}$$

Important when one has *interacting* defaults:

$$\frac{\texttt{Adult}(x) : \texttt{Employed}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x) : \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

$$\frac{\texttt{Student}(x) : \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

For `Student(TOM)` we get two extensions: one with `Employed(Tom)` and the other one with ¬`Employed(Tom)`.
Since the third rule is *"more specific"*, we may prefer it.

Semi-Normal Defaults (1)

Semi-normal defaults are sometimes useful:

$$\frac{\alpha : \beta \wedge \gamma}{\beta}$$

Important when one has *interacting* defaults:

$$\frac{\texttt{Adult}(x) : \ \texttt{Employed}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x) : \ \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

$$\frac{\texttt{Student}(x) : \ \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

For `Student(TOM)` we get two extensions: one with `Employed(Tom)` and the other one with $\neg$`Employed(Tom)`.

Since the third rule is *"more specific"*, we may prefer it.

Semi-Normal Defaults (1)

Semi-normal defaults are sometimes useful:

$$\frac{\alpha : \beta \wedge \gamma}{\beta}$$

Important when one has *interacting* defaults:

$$\frac{\texttt{Adult}(x): \quad \texttt{Employed}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x): \quad \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

$$\frac{\texttt{Student}(x): \quad \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

For `Student(TOM)` we get two extensions: one with `Employed(Tom)` and the other one with ¬`Employed(Tom)`.

Since the third rule is *"more specific"*, we may prefer it.

Semi-Normal Defaults (1)

Semi-normal defaults are sometimes useful:

$$\frac{\alpha : \beta \wedge \gamma}{\beta}$$

Important when one has *interacting* defaults:

$$\frac{\texttt{Adult}(x) : \quad \texttt{Employed}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x) : \quad \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

$$\frac{\texttt{Student}(x) : \quad \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

For `Student(TOM)` we get two extensions: one with `Employed(Tom)` and the other one with $\neg$`Employed(Tom)`.

Since the third rule is *“more specific”*, we may prefer it.

Semi-Normal Defaults (2)

- Since being a student is an exception, we could use a semi-normal default to exclude students from employed adults:

$$\frac{\texttt{Student}(x): \quad \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

$$\frac{\texttt{Adult}(x): \quad \texttt{Employed}(x) \wedge \neg\texttt{Student}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x): \quad \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

- Representing conflict-resolution by semi-normal defaults becomes clumsy when the number of default rules becomes high.
- A scheme for assigning *priorities* would be more elegant (there are indeed such schemes).

Semi-Normal Defaults (2)

- Since being a student is an exception, we could use a semi-normal default to exclude students from employed adults:

$$\frac{\mathtt{Student}(x):\ \neg\mathtt{Employed}(x)}{\neg\mathtt{Employed}(x)}$$

$$\frac{\mathtt{Adult}(x):\ \mathtt{Employed}(x)\wedge\neg\mathtt{Student}(x)}{\mathtt{Employed}(x)}$$

$$\frac{\mathtt{Student}(x):\ \mathtt{Adult}(x)}{\mathtt{Adult}(x)}$$

- Representing conflict-resolution by semi-normal defaults becomes clumsy when the number of default rules becomes high.
- A scheme for assigning *priorities* would be more elegant (there are indeed such schemes).

Semi-Normal Defaults (2)

- Since being a student is an exception, we could use a semi-normal default to exclude students from employed adults:

$$\frac{\texttt{Student}(x) : \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

$$\frac{\texttt{Adult}(x) : \texttt{Employed}(x) \wedge \neg\texttt{Student}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x) : \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

- Representing conflict-resolution by semi-normal defaults becomes clumsy when the number of default rules becomes high.
- A scheme for assigning *priorities* would be more elegant (there are indeed such schemes).

Semi-Normal Defaults (2)

- Since being a student is an exception, we could use a semi-normal default to exclude students from employed adults:

$$\frac{\texttt{Student}(x):\ \ \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

$$\frac{\texttt{Adult}(x):\ \ \texttt{Employed}(x)\ \wedge\neg\texttt{Student}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x):\ \ \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

- Representing conflict-resolution by semi-normal defaults becomes clumsy when the number of default rules becomes high.
- A scheme for assigning *priorities* would be more elegant (there are indeed such schemes).

Semi-Normal Defaults (2)

- Since being a student is an exception, we could use a semi-normal default to exclude students from employed adults:

$$\frac{\texttt{Student}(x): \neg\texttt{Employed}(x)}{\neg\texttt{Employed}(x)}$$

$$\frac{\texttt{Adult}(x): \texttt{Employed}(x) \wedge \neg\texttt{Student}(x)}{\texttt{Employed}(x)}$$

$$\frac{\texttt{Student}(x): \texttt{Adult}(x)}{\texttt{Adult}(x)}$$

- Representing conflict-resolution by semi-normal defaults becomes clumsy when the number of default rules becomes high.
- A scheme for assigning *priorities* would be more elegant (there are indeed such schemes).

Open Defaults (1)

- Our examples included open defaults, but the theory covers only closed defaults.
- If we have $\frac{\alpha(\vec{x}):\beta(\vec{x})}{\gamma(\vec{x})}$, then the variables should stand for all nameable objects.
- Problem: What about objects that have been introduced implicitly: $\boxed{\exists x P(x)}$.
- Solution by Reiter: Skolemization of all formulae in W and D.
- Interpretation: An open default stands for all the closed defaults resulting from substituting ground terms for the variables.

Open Defaults (1)

- Our examples included open defaults, but the theory covers only closed defaults.
- If we have $\frac{\alpha(\vec{x}):\beta(\vec{x})}{\gamma(\vec{x})}$, then the variables should stand for all nameable objects.
- Problem: What about objects that have been introduced implicitly: $\exists x P(x)$.
- Solution by Reiter: Skolemization of all formulae in W and D.
- Interpretation: An open default stands for all the closed defaults resulting from substituting ground terms for the variables.

Open Defaults (1)

- Our examples included open defaults, but the theory covers only closed defaults.
- If we have $\frac{\alpha(\vec{x}):\beta(\vec{x})}{\gamma(\vec{x})}$, then the variables should stand for all nameable objects.
- Problem: What about objects that have been introduced implicitly: $\boxed{\exists x P(x)}$.
- Solution by Reiter: Skolemization of all formulae in W and D.
- Interpretation: An open default stands for all the closed defaults resulting from substituting ground terms for the variables.

Open Defaults (1)

- Our examples included open defaults, but the theory covers only closed defaults.
- If we have $\frac{\alpha(\vec{x}):\beta(\vec{x})}{\gamma(\vec{x})}$, then the variables should stand for all nameable objects.
- Problem: What about objects that have been introduced implicitly: $\boxed{\exists x P(x)}$.
- Solution by Reiter: Skolemization of all formulae in W and D.
- Interpretation: An open default stands for all the closed defaults resulting from substituting ground terms for the variables.

Open Defaults (1)

- Our examples included open defaults, but the theory covers only closed defaults.
- If we have $\frac{\alpha(\vec{x}):\beta(\vec{x})}{\gamma(\vec{x})}$, then the variables should stand for all nameable objects.
- Problem: What about objects that have been introduced implicitly: $\boxed{\exists x P(x)}$.
- Solution by Reiter: Skolemization of all formulae in W and D.
- Interpretation: An open default stands for all the closed defaults resulting from substituting ground terms for the variables.

Open Defaults (2)

Skolemization can create problems because it preserves satisfiability, but it is not an equivalence transformation.

Example

$$\forall x(\mathtt{Man}(x) \leftrightarrow \neg \mathtt{Woman}(x))$$
$$\forall x(\mathtt{Man}(x) \rightarrow (\exists y(\mathtt{Spouse}(x, y) \wedge \mathtt{Woman}(y)) \vee \mathtt{Bachelor}(x)))$$
$$\mathtt{Man}(\mathtt{TOM})$$
$$\mathtt{Spouse}(\mathtt{TOM}, \mathtt{MARY})$$
$$\mathtt{Woman}(\mathtt{MARY})$$
$$\frac{: \mathtt{Man}(x)}{\mathtt{Man}(x)}$$

Skolemization of $\exists y: \ldots$ enables concluding $\mathtt{Bachelor}(\mathtt{TOM})$! The reason is that for $g(\mathtt{TOM})$ we get $\mathtt{Man}(g(\mathtt{TOM}))$ by default (g is the Skolem function).

Open Defaults (2)

Skolemization can create problems because it preserves satisfiability, but it is not an equivalence transformation.

Example

$$\forall x(\mathtt{Man}(x) \leftrightarrow \neg\mathtt{Woman}(x))$$
$$\forall x(\mathtt{Man}(x) \rightarrow (\exists y(\mathtt{Spouse}(x, y) \wedge \mathtt{Woman}(y)) \vee \mathtt{Bachelor}(x)))$$
$$\mathtt{Man}(\mathtt{TOM})$$
$$\mathtt{Spouse}(\mathtt{TOM}, \mathtt{MARY})$$
$$\mathtt{Woman}(\mathtt{MARY})$$
$$\frac{:\mathtt{Man}(x)}{\mathtt{Man}(x)}$$

Skolemization of $\exists y:$ ... enables concluding $\mathtt{Bachelor}(\mathtt{TOM})$! The reason is that for $g(\mathtt{TOM})$ we get $\mathtt{Man}(g(\mathtt{TOM}))$ by default (g is the Skolem function).

Open Defaults (2)

Skolemization can create problems because it preserves satisfiability, but it is not an equivalence transformation.

Example

$$\forall x(\mathtt{Man}(x) \leftrightarrow \neg\mathtt{Woman}(x))$$
$$\forall x(\mathtt{Man}(x) \rightarrow (\exists y(\mathtt{Spouse}(x,y) \wedge \mathtt{Woman}(y)) \vee \mathtt{Bachelor}(x)))$$
$$\mathtt{Man}(\mathtt{TOM})$$
$$\mathtt{Spouse}(\mathtt{TOM}, \mathtt{MARY})$$
$$\mathtt{Woman}(\mathtt{MARY})$$
$$\frac{:\mathtt{Man}(x)}{\mathtt{Man}(x)}$$

Skolemization of $\exists y : \ldots$ enables concluding $\mathtt{Bachelor}(\mathtt{TOM})$! The reason is that for $g(\mathtt{TOM})$ we get $\mathtt{Man}(g(\mathtt{TOM}))$ by default (g is the Skolem function).

Open Defaults (3)

It is even worse. Logically equivalent theories can have different extensions.

$$
\begin{aligned}
W_1 &= \{\exists x(P(C,x) \vee Q(C,x))\} \\
W_2 &= \{\exists x P(C,x) \vee \exists x Q(C,x)\} \\
D &= \left\{ \frac{P(x,y) \vee Q(x,y) \colon R}{R} \right\}
\end{aligned}
$$

W_1 and W_2 are logically equivalent. However, the Skolemization of W_1, symbolically $s(W_1)$, is not equivalent with $s(W_2)$. The only extension of (D, W_1) is $\mathrm{Th}(s(W_1) \cup R)$. The only extension of (D, W_2) is $\mathrm{Th}(s(W_2))$.

Note: Skolemization is not the right method to deal with open defaults in the general case.

Open Defaults (3)

It is even worse. Logically equivalent theories can have different extensions.

$$\begin{aligned} W_1 &= \{\exists x(P(C,x) \vee Q(C,x))\} \\ W_2 &= \{\exists x P(C,x) \vee \exists x Q(C,x)\} \\ D &= \left\{\frac{P(x,y) \vee Q(x,y) : R}{R}\right\} \end{aligned}$$

W_1 and W_2 are logically equivalent. However, the Skolemization of W_1, symbolically $s(W_1)$, is not equivalent with $s(W_2)$. The only extension of (D, W_1) is $\text{Th}(s(W_1) \cup R)$. The only extension of (D, W_2) is $\text{Th}(s(W_2))$.

Note: Skolemization is not the right method to deal with open defaults in the general case.

Open Defaults (3)

It is even worse. Logically equivalent theories can have different extensions.

$$\begin{aligned} W_1 &= \{\exists x(P(C,x) \vee Q(C,x))\} \\ W_2 &= \{\exists x P(C,x) \vee \exists x Q(C,x)\} \\ D &= \left\{\frac{P(x,y) \vee Q(x,y) \colon R}{R}\right\} \end{aligned}$$

W_1 and W_2 are logically equivalent. However, the Skolemization of W_1, symbolically $s(W_1)$, is not equivalent with $s(W_2)$. The only extension of (D, W_1) is $\mathrm{Th}(s(W_1) \cup R)$. The only extension of (D, W_2) is $\mathrm{Th}(s(W_2))$.

Note: Skolemization is not the right method to deal with open defaults in the general case.

Open Defaults (3)

It is even worse. Logically equivalent theories can have different extensions.

$$
\begin{aligned}
W_1 &= \{\exists x(P(C,x) \vee Q(C,x))\} \\
W_2 &= \{\exists x P(C,x) \vee \exists x Q(C,x)\} \\
D &= \left\{ \frac{P(x,y) \vee Q(x,y) : R}{R} \right\}
\end{aligned}
$$

W_1 and W_2 are logically equivalent. However, the Skolemization of W_1, symbolically $s(W_1)$, is not equivalent with $s(W_2)$. The only extension of (D, W_1) is $\mathrm{Th}(s(W_1) \cup R)$. The only extension of (D, W_2) is $\mathrm{Th}(s(W_2))$.

Note: Skolemization is not the right method to deal with open defaults in the general case.

Open Defaults (3)

It is even worse. Logically equivalent theories can have different extensions.

$$W_1 = \{\exists x(P(C,x) \vee Q(C,x))\}$$
$$W_2 = \{\exists x P(C,x) \vee \exists x Q(C,x)\}$$
$$D = \left\{ \frac{P(x,y) \vee Q(x,y) \colon R}{R} \right\}$$

W_1 and W_2 are logically equivalent. However, the Skolemization of W_1, symbolically $s(W_1)$, is not equivalent with $s(W_2)$. The only extension of (D, W_1) is $\mathrm{Th}(s(W_1) \cup R)$.
The only extension of (D, W_2) is $\mathrm{Th}(s(W_2))$.
Note: Skolemization is not the right method to deal with open defaults in the general case.

Open Defaults (3)

It is even worse. Logically equivalent theories can have different extensions.

$$\begin{aligned} W_1 &= \{\exists x(P(C,x) \vee Q(C,x))\} \\ W_2 &= \{\exists x P(C,x) \vee \exists x Q(C,x)\} \\ D &= \left\{ \frac{P(x,y) \vee Q(x,y) \colon R}{R} \right\} \end{aligned}$$

W_1 and W_2 are logically equivalent. However, the Skolemization of W_1, symbolically $s(W_1)$, is not equivalent with $s(W_2)$. The only extension of (D, W_1) is $\mathrm{Th}(s(W_1) \cup R)$. The only extension of (D, W_2) is $\mathrm{Th}(s(W_2))$.

Note: Skolemization is not the right method to deal with open defaults in the general case.

Open Defaults (3)

It is even worse. Logically equivalent theories can have different extensions.

$$
\begin{aligned}
W_1 &= \{\exists x(P(C,x) \vee Q(C,x))\} \\
W_2 &= \{\exists x P(C,x) \vee \exists x Q(C,x)\} \\
D &= \left\{\frac{P(x,y) \vee Q(x,y) \colon R}{R}\right\}
\end{aligned}
$$

W_1 and W_2 are logically equivalent. However, the Skolemization of W_1, symbolically $s(W_1)$, is not equivalent with $s(W_2)$. The only extension of (D, W_1) is $\mathrm{Th}(s(W_1) \cup R)$. The only extension of (D, W_2) is $\mathrm{Th}(s(W_2))$.
Note: Skolemization is not the right method to deal with open defaults in the general case.

Outlook

Although Reiter's definition of DL makes sense, one can of course come up with a number of variations and extend the investigation . . .

- Extensions can be defined differently (e.g., by remembering consistency conditions).
- . . . or by removing the groundedness condition.
- Open defaults can be handled differently (more model-theoretically).
- General proof methods for the finite, decidable case
- Applications of default logic:
  - Diagnosis
  - Reasoning about actions

Literature

(*) Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1):81–132, April 1980.

(*) Georg Gottlob. Complexity Results for Nonmonotonic Logics. *Journal for Logic and Computation*, 2(3), 1992.

Marco Cadoli and Marco Schaerf. A Survey of Complexity Results for Non-monotonic Logics. *The Journal of Logic Programming* 17: 127–160, 1993.

Gerhard Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press, Cambridge, UK, 1991.

Franz Baader and Bernhard Hollunder. Embedding defaults into terminological knowledge representation formalisms. In B. Nebel, W. Swartout, and C. Rich, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference*, pages 306–317, Cambridge, MA, October 1992. Morgan Kaufmann.