Knowledge Representation

Non-Monotonic Logic

- Classical logic
  - Permanent conclusions based on complete information
- Nonmonotonic logic
  - Drawing conclusions based on incomplete information
    You may need to retract them later!
  - (“Nonmonotonic” since learning new facts sometimes increases, sometimes *reduces* number of conclusions)

Non-Monotonic Logic

- Classical logic
  - Permanent conclusions based on complete information

- Nonmonotonic logic
  - Drawing conclusions based on incomplete information
    You may need to retract them later!

  - ("Nonmonotonic" since learning new facts sometimes increases, sometimes *reduces* number of conclusions)

Given: Turn key

Conclude: Car will start

Non-Monotonic Logic

- Classical logic
  – Permanent conclusions based on complete information

- Nonmonotonic logic
  – Drawing conclusions based on incomplete information
    You may need to retract them later!

  – ("Nonmonotonic" since learning new facts sometimes increases, sometimes *reduces* number of conclusions)

Given: Turn key BUT Battery dead?

Conclude: Car will start Then must retract conclusion. Car doesn't start

Non-Monotonic Logic

- Nonmonotonic logic

  Draw conclusions based on incomplete information

  You may need to retract them later!

- Want inferences based on what happen in the default case (e.g., battery works)

Again, this relates to the problem in Semantic Nets/Description Logics of how to implement subclass inheritance of properties by default---while still allowing the defaults to be overridden.

How to capture non-monotonic reasoning? Some approaches

- Default Logic
  - New type of inference rule

Default

$$\frac{\text{Turn key : Car starts}}{\text{Car starts}}$$

- Circumscription
  - New semantics

$\mathcal{C}(P \Rightarrow Q)$ *means*
(Q *typically* follows from P)

- Autoepistemic Logic, Modal NM
  - *If **P** were true, I'd know **P***

    $\neg$ Know(**P**) $\Rightarrow \neg$ **P**
  - *Negation as failure*:

    (Can't prove **P**) $\Rightarrow \neg$ **P**

How to capture non-monotonic reasoning? Some approaches

- I'll concentrate on the first two approaches, starting with Circumscription

Circumscription

- Given
  - "Birds fly, normally." $\forall x\ Bird(x) \wedge \neg Abnormal(x) \Rightarrow Fly(x)$
  - "A Bulbul is a bird." $Bird(B)$

- Want to conclude, provisionally (by default)
  - "Bulbuls fly" $Fly(B)$

- **FOL can't conclude this**

(since we don't know whether Bulbuls are normal)

- **Circumscription:** Maximum Normality Assumption

  Assume everything normal unless (until) you hear different

Circumscription

- Given
  - "Birds fly, normally." $\forall x\ Bird(x) \wedge \neg Abnormal(x) \Rightarrow Fly(x)$
  - "A Bulbul is a bird." $Bird(B)$
- Want to conclude, provisionally (by default)
  - "Bulbuls fly" $Fly(B)$
- **FOL can't conclude this**

(since we don't know whether Bulbuls are normal)

- **Circumscription:** Maximum Normality Assumption

  Assume as many things as possible are normal!

Circumscription

Assume something normal until you hear different (maximum normality)

- Given only:
  - "Birds fly, normally." $\forall x\ Bird(x) \wedge \neg Abnormal(x) \Rightarrow Fly(x)$
  - "Bulbul is a bird." $Bird(B)$
- *Consistent* to assume ***everything*** normal! $\forall x\ \neg Abnormal(x)$

(No information to the contrary)

Max Normality Assumption + Standard FOL $\neg Abnormal(B)$, $Bird(B)$

$\Downarrow$ $\Downarrow$

"Bulbuls fly" **$Fly(B)$**

Circumscription

Assume normal until you hear different

- Given
  - "Birds fly, normally." $\forall x\ Bird(x) \land \neg Abnormal(x) \Rightarrow Fly(x)$
  - "Bulbul is a bird." $Bird(B)$
  - "Ostrich is a bird and Ostrich doesn't fly" $Bird(O) \land \neg Fly(O)$

    So Ostrich is necessarily abnormal

    (otherwise, it would fly)

- *Consistent* to assume Ostrich *only* abnormal thing: $\forall x\ Abnormal(x) \Rightarrow (x=O)$

  (No information to the contrary)

Again, Max Normality + FOL

$\Downarrow$

"Bulbuls fly"

Circumscription and the Frame Problem

- Circumscription
  - Method for making inferences about *normal* situations
    Don't need to rule out exceptions explicitly!
  - "Normally, birds fly." **NOT** "Birds fly, except Ostriches, Penguins, …"
- Frame Problem
  - Exploit Common Sense that things *normally* stay the same
    Don't want to explicitly cite all possible changes!
  - "Normally, guns stay loaded" **NOT** "Guns stay loaded, unless unloaded, fired, jam…

➔ Circumscription a natural solution for Frame Problem?

Circumscription and the Frame Problem

- Frame Problem
  - Exploit Common Sense that things *normally* stay the same

Circumscription a natural solution for Frame Problem?

Maybe, but not it’s not straightforward. Later I’ll try applying circumscription to the Turkey shooting problem.

Generalizing Circumscription

- Recall **Maximum Normality** $\forall x\ Bird(x) \land \neg Abnormal(x) \Rightarrow Fly(x)$
  - Assume Abnormal(x) true for as few things as possible, consistent with **Knowledge Base.**
- **Generalized Circumscription** for predicates **P**

  *Assume **P** true for the smallest set possible, consistent with **K**nowledge **B**ase*

Examples

- **KB** = Star(Sun)

Circumscribe for predicate Star.

Then only the Sun is a Star ⇔ $\forall x\ Star(x) \Leftrightarrow (x=Sun)$

Examples

- **KB** = Star(Sun)

Circumscribe for predicate Star.

Then only the Sun is a Star ⇔ $\forall x\ Star(x) \Leftrightarrow (x=Sun)$

- **KB** = ¬Fly(Penguin)

Circumscribe for predicate Fly

Then *nothing* flies $\neg\exists x\ Fly(x)$

Examples

- Circumscribe for fly
  - **KB** = $Fly(Wren) \wedge Fly(Robin)$

    Then only wrens + robins fly $\forall x \; Fly(x) \Leftrightarrow (x=Wren) \vee (x=Robin)$

Examples

- Circumscribe for fly
  - **KB** = Fly(Wren) ∧ Fly(Robin)

    Then only wrens + robins fly $\forall x \; Fly(x) \Leftrightarrow (x=Wren) \lor (x=Robin)$

  - More complex case: **KB** = Fly(Wren) ∨ Fly(Robin)

    Either: only wrens fly $\forall x \; Fly(x) \Leftrightarrow (x=Wren)$

    Or: only robins fly $\forall x \; Fly(x) \Leftrightarrow (x=Robin)$

Examples

- Circumscribe for fly
  - **KB** = Fly(Wren) $\wedge$ Fly(Robin)

    Then only wrens + robins fly $\forall x \; Fly(x) \Leftrightarrow (x=Wren) \vee (x=Robin)$

  - More complex case: **KB** = Fly(Wren) **V** Fly(Robin)

    Either: only wrens fly $\forall x \; Fly(x) \Leftrightarrow (x=Wren)$

    Or: only robins fly $\forall x \; Fly(x) \Leftrightarrow (x=Robin)$

Note: 2 Possible minimal definitions of Fly

Examples

- **KB =** RainAt(Hoboken) $\Rightarrow$ RainAt(Stevens)

Circumscribe for RainAt

Then it's not raining anywhere! $\forall x\ \neg$ RainAt(x)

Examples

- **KB =** $RainAt(Hoboken) \Rightarrow RainAt(Stevens)$

  Circumscribe for RainAt

  Then it's not raining anywhere! $\forall x \ \neg RainAt(x)$

- **KB =** $Dog(Fido) \land \forall x \ Dog(x) \Rightarrow Dog(Child(x))$

  Circumscribe for Dog

  Then only Fido and all descendants are Dogs

  $Dog(Fido), Dog(Child(Fido)), Dog(Child(Child(Fido))), \ldots$

Circumscription: Problems

$\forall x \quad \text{Bird}(x) \wedge \neg \text{Ab}(x) \Rightarrow \text{Flies}(x)$ (Normal birds fly)

$\text{Bird}(\text{Tweety})$ (Tweety and Chilly are birds)

$\text{Bird}(\text{Chilly})$

$\neg \text{Fly}(\text{Chilly})$

Can't conclude Tweety flies!

Only $(\text{Tweety} \neq \text{Chilly}) \Rightarrow \text{Flies}(\text{Tweety})$

Circumscription: Problems

$\forall x \quad \text{Bird}(x) \land \neg \text{Ab}(x) \Rightarrow \text{Flies}(x)$ (Normal birds fly)

$\text{Bird}(\text{Tweety})$ (Tweety's a bird)

$\forall x \quad \text{Penguin}(x) \Rightarrow \text{Bird}(x) \land \neg \text{Flies}(x)$ (Penguins are birds and don't fly)

Conclude Tweety flies ....

Circumscription: Problems

$\forall x \quad Bird(x) \wedge \neg Ab(x) \Rightarrow Flies(x)$ (Normal birds fly)

$Bird(Tweety)$ (Tweety's a bird)

$\forall x \quad Penguin(x) \Rightarrow Bird(x) \wedge \neg Flies(x)$ (Penguins are birds and don't fly)

Conclude Tweety flies and no penguins!

Circumscription: Problems

$\forall x \quad \text{Bird}(x) \land \neg\text{Ab}(x) \Rightarrow \text{Flies}(x)$ (Normal birds fly)

$\text{Bird}(\text{Tweety})$ (Tweety's a bird)

$\forall x \quad \text{Penguin}(x) \Rightarrow \text{Bird}(x) \land \neg\text{Flies}(x)$ (Penguins are birds and don't fly)

Conclude Tweety flies and no penguins!

Possible solution: minimize *Ab* fixing #penguins

(idea: try to make Tweety normal and fly without doing anything to affect how many penguins there are)

Circumscription: Problems

$\forall x \quad \text{Bird}(x) \land \neg \text{Ab}(x) \Rightarrow \text{Flies}(x)$ (Normal birds fly)

$\text{Bird}(\text{Tweety})$ (Tweety's a bird)

$\forall x \quad \text{Penguin}(x) \Rightarrow \text{Bird}(x) \land \neg \text{Flies}(x)$ (Penguins are birds and don't fly)

Minimize *Ab* fixing #penguins...

But now we can't conclude Tweety flies!

(If Tweety is penguin, we don't care about minimizing its abnormality...
so can only conclude $\neg \text{Penguin}(\text{Tweety}) \Rightarrow \text{Flies}(\text{Tweety})$)

Circumscription: Problems

- Typical Canadians don't speak French $\forall x \ \neg Ab(x) \wedge Can(x) \Rightarrow \neg Fr(x)$
- All Quebecois are Canadians $\forall x \ Q(x) \Rightarrow Can(x)$
- Quebecois typically speak French $\forall x \ \neg Ab(x) \wedge Q(x) \Rightarrow Fr(x)$
- Robert is from Quebec $Q(Robt)$

Does Robert speak French?

Circumscription: Problems

- Typical Canadians don't speak French $\forall x \; \neg Ab(x) \wedge Can(x) \Rightarrow \neg Fr(x)$
- All Quebecois are Canadians $\forall x \;\; Q(x) \Rightarrow Can(x)$
- Quebecois typically speak French $\forall x \; \neg Ab(x) \wedge Q(x) \Rightarrow Fr(x)$
- Robert is from Quebec $Q(Robt)$

Does Robert speak French? We don't know.

Circumscription: Problems

- Typical Canadians don't speak French $\forall x\ \neg Ab(x) \wedge Can(x) \Rightarrow \neg Fr(x)$
- All Quebecois are Canadians $\forall x\ Q(x) \Rightarrow Can(x)$
- Quebecois typically speak French $\forall x\ \neg Ab(x) \wedge Q(x) \Rightarrow Fr(x)$
- Robert is from Quebec $Q(Robt)$

Does Robert speak French? We don't know.

2 ways to minimize the abnormality

- Robert normal Canadian, abnormal Quebecois ➔ no French
- Robert abnormal Canadian, normal Quebecois ➔ French

Frame Problem

- Recall Successor-State Axioms solving Frame Problem

SS: (Fluent true after action) $\Leftrightarrow$ (Action *made it* true, OR
Already true, Action didn't affect it)

- This assumes all change comes from Actions.

Circumscribing "Fluent changes after action"gives **SS axioms**

Yale Shooting Problem + Circumscription

Action Axioms

- Circumscription: normally, properties of the world *don't* change

$$\neg\, ab(fluent, action, S) \Leftrightarrow (Holds(fluent, S) \Leftrightarrow Holds(fluent, Result(action, S)))$$

- $Holds(Loaded, Result(Load, S))$ (Loading makes the Gun loaded)

- $Holds(Loaded, S) \Rightarrow \neg\, Holds(Alive, Result(Fire, S)) \wedge \neg\, Holds(Loaded, Result(Fire, S))$

  (Firing loaded gun kills turkey and unloads gun)

Yale Shooting Problem + Circumscription

Action Axioms

- Circumscription: normally, properties of the world *don't* change

$$\neg ab(fluent, action, S) \Leftrightarrow (Holds(fluent, S) \Leftrightarrow Holds(fluent, Result(action, S)))$$

- Holds(Loaded, Result(Load,S)) (Loading makes the Gun loaded)

- Holds(Loaded,S) $\Rightarrow$ $\neg$Holds(Alive,Result(Fire,S)) $\wedge$ $\neg$Holds(Loaded,Result(Fire,S))
  (Firing loaded gun kills turkey and unloads gun)

Action/Situation Sequence

S_1 = Result(Load, S_0)

S_2 = Result(Sneeze, S_1), … S_{N+1}= Result(Sneeze, S_N),

S_{N+2} = Result(Fire, S_{N+1})

Yale Shooting Problem + Circumscription

Action Axioms

- Circumscription: normally, properties of the world *don't* change

$\neg$ ab(fluent,action,S) $\Leftrightarrow$ (Holds(fluent,S) $\Leftrightarrow$ Holds(fluent,Result(action,S)))

- Holds(Loaded, Result(Load,S)) (Loading makes the Gun loaded)

- Holds(Loaded,S) $\Rightarrow$ $\neg$ Holds(Alive,Result(Fire,S)) $\wedge$ $\neg$ Holds(Loaded,Result(Fire,S))

  (Firing loaded gun kills turkey and unloads gun)

Now we figure out what happens to turkey by minimizing the *abnormality*

Yale Shooting Problem + Circumscription

Action Axioms

- Circumscription: normally, properties of the world *don't* change

$\neg$ ab(fluent,action,S) $\Leftrightarrow$ (Holds(fluent,S) $\Leftrightarrow$ Holds(fluent,Result(action,S)))

- Holds(Loaded, Result(Load,S)) (Loading makes the Gun loaded)

- Holds(Loaded,S) $\Rightarrow$ $\neg$ Holds(Alive,Result(Fire,S)) $\wedge$ $\neg$ Holds(Loaded,Result(Fire,S))
  (Firing loaded gun kills turkey and unloads gun)

Now we figure out what happens to turkey by minimizing the *abnormality*

- Common sense scenario in which turkey dies:
  Abnormality = 1 (gun becomes loaded) + 2 (turkey dies, gun unloads) = 3 (total)

Yale Shooting Problem + Circumscription

Action Axioms

- Circumscription: normally, properties of the world *don't* change

  $\neg\, ab(fluent, action, S) \Leftrightarrow (Holds(fluent, S) \Leftrightarrow Holds(fluent, Result(action, S)))$

- $Holds(Loaded, Result(Load, S))$ (Loading makes the Gun loaded)

- $Holds(Loaded, S) \Rightarrow \neg\, Holds(Alive, Result(Fire, S)) \wedge \neg\, Holds(Loaded, Result(Fire, S))$

  (Firing loaded gun kills turkey and unloads gun)

Now we figure out what happens to turkey by minimizing the *abnormality*

- Common sense scenario in which turkey dies:

  Abnormality = 1 (gun becomes loaded) + 2 (turkey dies, gun unloads) = 3 (total)

- Counterintuitive scenario: turkey survives

  Abnormality = 1 (gun becomes loaded) + 1 (gun unloads on sneeze, so it doesn't fire later) = 2 (total)

Yale Shooting Problem + Circumscription

Action Axioms

- Circumscription: normally, properties of the world *don't* change

$$\neg\, ab(fluent, action, S) \Leftrightarrow (Holds(fluent, S) \Leftrightarrow Holds(fluent, Result(action, S)))$$

- $Holds(Loaded, Result(Load, S))$ (Loading makes the Gun loaded)

- $Holds(Loaded, S) \Rightarrow \neg\, Holds(Alive, Result(Fire, S)) \wedge \neg\, Holds(Loaded, Result(Fire, S))$
  (Firing loaded gun kills turkey and unloads gun)

Now we figure out what happens to turkey by minimizing the *abnormality*

- Common sense scenario in which turkey dies:
  Abnormality = 1 (gun becomes loaded) + 2 (turkey dies, gun unloads) = 3 (total)
- Counterintuitive scenario: turkey survives because less abnormality!
  Abnormality = 1 (gun becomes loaded) + 1 (gun unloads on sneeze, so it doesn't fire later) = 2 (total)

Yale Shooting Problem + Circumscription

- Using circumscription to solve Frame Problem isn't easy

- Similar to Canadian Robert example: there's an unexpected way to minimize abnormality

- Historical note
  - Frame problem took 20 years to solve
  - Shooting Problem was turning point that showed problem's difficulty

- Basic issue still open: how to make our inference procedure accord with common sense on what happens by default?

Circumscription

- Circumscription really 2nd order logic...
  but often it can be translated into FOL

So can use inference methods of FOL!
Resolution, Backward Chaining, …

Default Logic

- Instead of restricting "meaning" of predicate as in Circumscription, *use new inference rules*.

- **Default Rule**

  Given**: P**
  and **Q** *consistent with all inferences from* **KB**,
  Deduce**: R**

$$\frac{P \;:\; Q}{R}$$

- **Example:** Does Tweety fly?
- **KB ={** Bird(Tweety), ∀x Penguin(x) ⇒ Bird(x);

  Bird(x) : Fly(x) ⇒ Fly(x), Penguin(x) : ¬Fly(x) ⇒ ¬ Fly(x) **}**

Default inference rules

Default Logic

- Instead of restricting "meaning" of predicate as in Circumscription, *use new inference rules*.

- **Default Rule**

  Given**: P**

  and **Q** *consistent with all inferences from* **KB**,

  Deduce**: R**

$$\frac{\mathbf{P} \ : \ \mathbf{Q}}{\mathbf{R}}$$

- **Example:** Does Tweety fly?
- **KB =**{ Bird(Tweety), ∀x Penguin(x) ⇒ Bird(x);

  Bird(x) : Fly(x) ⇒ Fly(x), Penguin(x) : ¬Fly(x) ⇒ ¬ Fly(x) }

  (If consistent, assume Birds fly) (If consistent, assume Penguins *don't* fly)

Default Logic

- **KB ={** Bird(Tweety), $\forall$x Penguin(x) $\Rightarrow$ Bird(x);
  Bird(x) : Fly(x) $\Rightarrow$ Fly(x), Penguin(x) : $\neg$Fly(x) $\Rightarrow$ $\neg$ Fly(x) }

- **Default Rule** Bird(T) : Fly(T) $\Rightarrow$Fly(T)

  Given: Tweety is a bird,

  Deduce: Tweety flies if this is consistent with *all conclusions from* ***KB*****.**

- **“*All conclusions from*** *KB*” means a maximal set of conclusions: you can’t add any more.

  ➔ So before we can decide whether Tweety flies, we must compute a maximal *consistent* set of conclusions.

Default Logic

- **KB ={** Bird(Tweety), ∀x Penguin(x) ⇒ Bird(x);
  Bird(x) : Fly(x) ⇒ Fly(x), Penguin(x) : ¬Fly(x) ⇒ ¬ Fly(x) }

- A Maximal Set of Conclusions?

1. Bird(Tweety), ∀x Penguin(x) ⇒ Bird(x)	Tweety is Bird, Penguins are Birds
2. Fly(Tweety),	Applying first default rule (consistent so far)
3. ∀x Penguin(x) ⇒ ¬Fly(x),	Applying 2nd default rule (consistent so far)
4. ¬ Penguin(Tweety)	2+3

Can't draw any more conclusions. These gives maximal consistent set

So yes! Tweety flies.

Default Logic

- **KB =** { Penguin(Tweety), $\forall x$ Penguin(x) $\Rightarrow$ Bird(x);
  Bird(x) : Fly(x) $\Rightarrow$ Fly(x), Penguin(x) : $\neg$Fly(x) $\Rightarrow$ $\neg$ Fly(x) }

Slightly different Knowledge Base: Tweety is a Penguin

Default Logic

- **KB ={** Penguin(Tweety), ∀x Penguin(x) ⇒ Bird(x);
  Bird(x) : Fly(x) ⇒ Fly(x), Penguin(x) : ¬Fly(x) ⇒ ¬ Fly(x) }

- A Maximal Set of Conclusions?
  1. Penguin(Tweety), ∀x Penguin(x) ⇒ Bird(x) — Tweety is Penguin, Penguins are Birds
  2. Bird(Tweety) — From 1
  3. Fly(Tweety), — Applying first default rule (consistent so far)

  Can't apply 2nd default rule, since it's no longer consistent that Tweety doesn't fly. No more conclusions possible; a maximal consistent set

  So yes! Tweety flies.

Default Logic

- **KB ={** Penguin(Tweety), $\forall x$ Penguin(x) $\Rightarrow$ Bird(x);
  Bird(x) : Fly(x) $\Rightarrow$ Fly(x), Penguin(x) : $\neg$Fly(x) $\Rightarrow$ $\neg$ Fly(x) }

- But now try applying rules in a different order!

1. Penguin(Tweety), $\forall x$ Penguin(x) $\Rightarrow$ Bird(x)	Tweety is Penguin, Penguins are Birds
2. Bird(Tweety)	From 1
3. $\neg$ Fly(Tweety),	1a + 2nd default rule (consistent so far)

Can't apply first default rule, since it's no longer consistent that Tweety flies. No more conclusions possible; a maximal consistent set

So Tweety doesn't fly!

Default Logic

- Can derive *“Tweety flies”* or *“Tweety doesn’t fly”*

- The maximal consistent Conclusions *differ* in these two cases.

- Similar to the Canadian Robert example for Circumscription.

  (In fact, circumscription + default logic give *same result* on this example)

- Common sense: use *more specific* default first

  (i.e., use Penguin(x) : $\neg$Fly(x) $\Rightarrow$ $\neg$ Fly(x) first and derive Tweety flies)

  Methods exist for prioritizing defaults