Slide 1

Lecture 2

- Computational complexity of default reasoning
- Default logic vs. other nonmonotonic logics
- A framework for automating default reasoning

Slide 2

Computational Complexity

- Main decision problems
  sceptical reasoning: is φ in every extension of $(\mathcal{D}, W)$?
  credulous reasoning: is φ in some extension of $(\mathcal{D}, W)$?
  extension existence: does $(\mathcal{D}, W)$ have an extension?
- First-order default reasoning is not even semi-decidable.
- Propositional sceptical reasoning is $\mathbf{\Pi_2^P}$-complete.
- Propositional credulous reasoning and extension existence are $\mathbf{\Sigma_2^P}$-complete.

Slide 3

Implications of the Results

- For first-order default reasoning no complete decision procedures.
- $\mathbf{\Pi_2^P}, \mathbf{\Sigma_2^P} \subseteq$ **PSPACE**:
  Propositional default reasoning can be done in polynomial space.
- If the polynomial hierarchy does not collapse,
  default reasoning strictly harder than classical reasoning.
- Highly unlikely that default reasoning can be implemented on top of a classical theorem proving method with only a polynomial overhead.

Slide 4

- $\mathbf{\Sigma_2^P} = \mathbf{NP}^{\mathbf{NP}}$:
  Default reasoning has two orthogonal sources of complexity:
  (i) classical reasoning
  (ii) default rule interaction
  Note: even if we assume that classical reasoning is fast (an oracle), **NP**-completeness remains.
- Tractable subclasses: consider both sources of complexity.
- A typical approach:
  (i) syntactic restriction on classical connectives,
  (ii) syntactic restriction on default rule interactions.
  Example. Stratified atomic defaults (logic programs with stratified negation): linear time complexity.

Slide 5

- Reducing complexity of default rule interaction:
  Use weaker semantics: stationary extension, well-founded semantics.
- But is high complexity of NMR a real drawback?
- NMR allows for more compact and natural representation of knowledge.
- A possible scenario:
  Compact representation using NMR techniques.
  No polynomial size translation to classical logic.
  ☞ The problem is "unsolvable" using classical logic (even the description does not fit the available memory)
  But solvable in polynomial space using NMR techniques!

Slide 6

NP vs. Σ_2^P

Many engineering problems are in **NP**
(a solution is checkable in polynomial time)

Problems on the second level of PH:

- Strategic companies
  A holding owns companies each producing some goods.
  Several companies may have joint control over another company.
  A company is strategic for a holding if it belongs to the strategic set: minimal set of companies such that
  (i) all goods can still be produced,
  (ii) if a company is controlled by companies in the set, it is included in the set.
- Conditional planning
- Diagnosis

Slide 7

Default Logic vs. Other Nonmonotonic Logics

Default Logic is very expressive and captures other nonmonotonic logics through modular translations:

- Circumscription
  To minimize an atom a, use $\frac{:\neg a}{\neg a}$
  To fix an atom a, use $\frac{:\neg a}{\neg a}, \frac{:a}{a}$
- Autoepistemic logic
  Treat belief formulae $L\varphi$ as new atoms and add
  (i) positive introspection: $\frac{\varphi :}{L\varphi}$
  (ii) negative introspection: $\frac{:\neg\varphi}{\neg L\varphi}$
  ☞ Iterative expansions
  Add $\neg L\neg L\varphi \rightarrow L\varphi$ to get stable expansions

Slide 8

Automating Default Reasoning

- Several approaches have been proposed
  – Methods based on generating defaults
  – Reduction to TMS/classical logic/constraint satisfaction
  – Tableaux/sequent calculus
- Our development is based on full sets
  – Davis-Putnam type procedure
  – General framework for implementing NMR
  – Efficient search space pruning methods

Slide 9

A Naive Procedure

S-answer:= Yes; C-answer:= No;
For every subset S of $\mathrm{just}(\mathcal{D})$ **do**
  If S is $(\mathcal{D}, W)$-full **then**
    If $\varphi \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},S)}(W)$ **then**
      C-answer:= Yes
    else
      S-answer:= No
    end if
  end if

- S-answer = Yes iff φ is a sceptical consequence of $(\mathcal{D}, W)$
- C-answer = Yes iff φ is a credulous consequence of $(\mathcal{D}, W)$.

Slide 10

- If $\mathrm{just}(\mathcal{D})$ is finite, then a finite number of iterations of the main loop.
- If $\mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},S)}(W)$ decidable, then $(\mathcal{D}, W)$-fullness decidable and hence default reasoning.
- Optimizations

  (i) backtracking search over candidate full sets,
  (ii) search space pruning by making assumptions about the justifications one at a time and
  (iii) before making a new assumption, an *approximation* of the extensions agreeing with the current assumptions is computed and *conflicts* are detected.

Slide 11

Partial full sets

- For the search we need to represent partially constructed full sets. This is done using not-formulae of the form $\mathrm{not}(\varphi)$.
- We reverse the representation of full sets:
  A partial full set FS is a set of formulae and not-formulae (not-literals) explicating the current assumptions about the **membership** in extensions for the **negations** of justifications.

Example.
$\mathrm{not}(\neg\chi) \in FS$: $\neg\chi$ does not belong to the extensions under consideration (χ belongs to the corresponding full sets)

$\neg\chi \in FS$: $\neg\chi$ belongs to the extensions under consideration (χ does not belong to the corresponding full sets)

Given justifications $\{\neg p, q, \neg r, s\}$,

$FS = \{\mathrm{not}(\neg\neg p), \mathrm{not}(\neg q), \mathrm{not}(\neg\neg r), \neg s\}$ represent the full set $\{\neg p, q, \neg r\}$.

Slide 12

We often use this representation directly.

For example,

$$\mathrm{Mon}(\mathcal{D}, FS) = \{\frac{pre(d)}{cons(d)} \mid d \in \mathcal{D}, \text{ for all } \psi \in \mathrm{just}(d), \mathrm{not}(\neg\psi) \in FS\}$$

which gives closure $\mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},FS)}(W)$.

Hence, the fullness condition for not-literals:

A set of not-literals FS is $(\mathcal{D}, W)$-full iff for all $\psi \in \mathrm{just}(\mathcal{D})$:

$$\begin{aligned} \mathrm{not}(\neg\psi) \in FS \quad &\text{iff} \quad \neg\psi \notin \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},FS)}(W) \\ \neg\psi \in FS \quad &\text{iff} \quad \neg\psi \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},FS)}(W) \end{aligned}$$

Slide 13

- A set of formulae S *agrees* with a set of not-literals FS if for every ordinary formula $\varphi \in FS$, $\varphi \in S$ and for all $\mathrm{not}(\varphi) \in FS$, $\varphi \notin S$.
  Example. $\{a, b\}$ agrees with $\{a, \mathrm{not}(c)\}$ but not with $\{a, \mathrm{not}(b)\}$.
- A set of not-literals FS *covers* a formula φ if either $\varphi \in FS$ or $\mathrm{not}(\varphi) \in FS$.
  A set of formulae S is covered by FS if each formula in S is covered by FS.
  Example. $\{\mathrm{not}(a), b\}$ covers a but not $\{a, c\}$.
- By $\mathrm{compl}(\chi)$ we denote the not-complement of an expression: for an ordinary formula χ, $\mathrm{compl}(\chi) = \mathrm{not}(\chi)$ and $\mathrm{compl}(\mathrm{not}(\chi)) = \chi$.

Slide 14

function extensions$((\mathcal{D}, W), FS, \varphi)$
$FS' :=$ **expand**$((\mathcal{D}, W), FS)$;
if conflict$((\mathcal{D}, W), FS')$ returns Yes **then** return No
else if FS' covers the negation of each justification in $\mathcal{D}$ **then**
  return **test**$((\mathcal{D}, W), FS', \varphi)$
else
  $\chi :=$ **choose**$((\mathcal{D}, W), FS')$;
  if extensions$((\mathcal{D}, W), FS' \cup \{\chi\}, \varphi)$ returns Yes **then** return Yes
  else return **extensions**$((\mathcal{D}, W), FS' \cup \{\mathrm{compl}(\chi)\}, \varphi)$
  end if
end if

extensions performs focused extension search:

extensions$((\mathcal{D}, W), FS, \varphi)$ returns Yes iff there is an extension E agreeing with FS such that **test**$((\mathcal{D}, W), FS', \varphi)$ returns Yes where FS' is the full set of E.

Slide 15

Key subroutines

- **expand**: approximations
- **conflict**: conflict detection.
- **choose**: search heuristics
- **test**: testing extensions

Slide 16

For soundness and completeness:

expand: expand$((\mathcal{D}, W), FS)$ returns a set FS' of not-literals which extends FS and with which every extension of $(\mathcal{D}, W)$ agreeing with FS agrees.

conflict: (i) If **conflict**$((\mathcal{D}, W), FS)$ returns Yes, there is no extension agreeing with FS and
(ii) if FS covers the negation of each justification in $(\mathcal{D}, W)$, and for some $\mathrm{not}(\chi) \in FS$, $\chi \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D}, FS)}(W)$ or for some $\chi \in FS$, $\chi \notin \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D}, FS)}(W)$, **conflict**$((\mathcal{D}, W), FS)$ returns Yes.

choose: choose$((\mathcal{D}, W), FS)$ returns either $\neg\psi$ or $\mathrm{not}(\neg\psi)$ for some justification ψ in $(\mathcal{D}, W)$ with $\neg\psi$ not covered by FS.

Example. Trivial implementations:
expand$((\mathcal{D}, W), FS)$ returns FS.
conflict$((\mathcal{D}, W), FS)$ returns No except if FS covers the negations of the justifications in $(\mathcal{D}, W)$, then test (ii) is performed.

Slide 17

Example.

extensions$((\mathcal{D}, W), \emptyset, \varphi)$:

where

$\mathcal{D} = \{\frac{:p}{\neg q}, \frac{:q}{\neg p}\}$

$W = \{\}$

- 1.
  - 2. $\mathrm{not}(\neg p)$
    - 3. $\mathrm{not}(\neg q)$ $\times^a$
    - 4. $\neg q$ $\times^b$
  - 5. $\neg p$
    - 6. $\neg q$ $\times^c$
    - 7. $\mathrm{not}(\neg q)$ $\times^d$

a) $R_a = \mathrm{Mon}(\mathcal{D}, \{\mathrm{not}(\neg p), \mathrm{not}(\neg q)\}) = \{\frac{}{\neg q}, \frac{}{\neg p}\}$ and $\neg p, \neg q \in \mathrm{Cn}_{R_a}(W)$

b) **test**$((\mathcal{D}, W), \{\mathrm{not}(\neg p), \neg q\}, \varphi)$

c) $R_c = \mathrm{Mon}(\mathcal{D}, \{\neg p, \neg q\}) = \{\}$ and $\neg p, \neg q \notin \mathrm{Cn}_{R_c}(W)$

d) **test**$((\mathcal{D}, W), \{\neg p, \mathrm{not}(\neg q)\}, \varphi)$

Theorem. (Soundness and completeness)
Let $(\mathcal{D}, W)$ be a default theory and FS a set of not-literals. Then the function **extensions**$((\mathcal{D}, W), FS, \varphi)$ returns Yes iff there exists an extension of $(\mathcal{D}, W)$ agreeing with FS such that **test**$((\mathcal{D}, W), FS', \varphi)$ returns Yes where FS' is the corresponding full set.

Slide 18

Decision procedures for the different reasoning tasks:

Corollary. Let $(\mathcal{D}, W)$ be a default theory and φ a formula.

1. Let **test**$((\mathcal{D}, W), FS, \varphi)$ return Yes for all $(\mathcal{D}, W)$, FS, and φ. Then **extensions**$((\mathcal{D}, W), \emptyset, \varphi)$ returns Yes iff $(\mathcal{D}, W)$ has an extension.
2. Let **test**$((\mathcal{D}, W), FS, \varphi)$ return Yes iff $\varphi \in \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D}, FS)}(W)$. Then **extensions**$((\mathcal{D}, W), \emptyset, \varphi)$ returns Yes iff there is an extension of $(\mathcal{D}, W)$ containing φ.
3. Let **test**$((\mathcal{D}, W), FS, \varphi)$ return Yes iff $\varphi \notin \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D}, FS)}(W)$. Then **extensions**$((\mathcal{D}, W), \emptyset, \varphi)$ returns Yes iff there exists an extension of $(\mathcal{D}, W)$ not containing φ (i.e., **extensions**$((\mathcal{D}, W), \emptyset, \varphi)$ returns No iff φ is contained in every extension of $(\mathcal{D}, W)$).

Slide 19

Example. Decide whether r is a sceptical consequence of $(\mathcal{D}, W)$.

test$((\mathcal{D}, W), FS, \varphi)$ returns Yes iff $\varphi \notin \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D}, FS)}(W)$
Initially $FS = \{\}$

$\mathcal{D} = \{\frac{:p}{\neg q}, \frac{:q}{\neg p}\}$

$W = \{\neg p \rightarrow r\}$

- 1.
  - 2. $\mathrm{not}(\neg p)$
    - 3. $\mathrm{not}(\neg q)$ $\times^a$
    - 4. $\neg q$ $\times^b$

a) $R_a = \mathrm{Mon}(\mathcal{D}, \{\mathrm{not}(\neg p), \mathrm{not}(\neg q)\}) = \{\frac{}{\neg q}, \frac{}{\neg p}\}$ and $\neg p, \neg q \in \mathrm{Cn}_{R_a}(W)$

b) **test**$((\mathcal{D}, W), \{\mathrm{not}(\neg p), \neg q\}, r)$ returns Yes:
$R_b = \mathrm{Mon}(\mathcal{D}, \{\mathrm{not}(\neg p), \neg q\}) = \{\frac{}{\neg q}\}$ and $r \notin \mathrm{Cn}_{R_b}(W)$

Slide 20

Performance of extensions

Depends on

- the accuracy of the approximation (implemented by **expand**)
- the accuracy of the conflict detection method (implemented by **conflict**)
- the search heuristics (implemented by **choose**)

An interesting trade-off:

More accuracy in **expand** and **conflict** prunes the remaining backtracking search.

Slide 21

Systematic basis for optimizations

- The concepts of lower and upper bound are used.
- Lower bound $\mathbf{LB}((\mathcal{D},W),FS)$:
  every extension E of $(\mathcal{D},W)$ agreeing with FS agrees with $\mathbf{LB}((\mathcal{D},W),FS)$
- Upper bound $\mathbf{UB}((\mathcal{D},W),FS)$:
  for every extension E of $(\mathcal{D},W)$ which agrees with FS, $E \subseteq \mathbf{UB}(P,FS)$.

Example. Trivial bounds:
$\mathbf{LB}((\mathcal{D},W),FS) = FS$
$\mathbf{UB}((\mathcal{D},W),FS) = \mathcal{L}$

Slide 22

Implementing expand and conflict

- **expand**$((\mathcal{D},W),FS)$ returns the smallest set of formulae FS' extending FS and closed under the following rules:
  - if $\neg\varphi \in \mathbf{LB}((\mathcal{D},W),FS')$ for $\varphi \in \mathrm{just}(\mathcal{D})$, then include $\neg\varphi$ in FS' and
  - if $\mathrm{not}(\neg\varphi) \in \mathbf{LB}((\mathcal{D},W),FS')$ for $\varphi \in \mathrm{just}(\mathcal{D})$, then include $\mathrm{not}(\neg\varphi)$ in FS' and
  - if $\neg\varphi \notin \mathbf{UB}((\mathcal{D},W),FS')$ for $\varphi \in \mathrm{just}(\mathcal{D})$, then include $\mathrm{not}(\neg\varphi)$ in FS'.
- **conflict**$((\mathcal{D},W),FS)$ returns Yes iff for some formula φ, $\mathrm{not}(\varphi) \in FS$ but $\varphi \in \mathbf{LB}((\mathcal{D},W),FS)$ or $\varphi \in FS$ but $\varphi \notin \mathbf{UB}((\mathcal{D},W),FS)$.

Slide 23

Tight bounds:

- $\mathbf{LB}((\mathcal{D},W),FS) = \mathrm{Cn}_{\mathrm{Mon}(\mathcal{D},FS)}(W \cup FS^{+})$
  where FS^{+} is set of the ordinary formulae in FS .
  Example. $\mathcal{D} = \{\frac{:p}{\neg q}, \frac{:q}{\neg p}\}$ For $FS = \{\mathrm{not}(\neg q)\}$
  $W = \{\}$ $\mathrm{Mon}(\mathcal{D},FS) = \{\frac{}{\neg p}\}$
  $\neg p \in \mathbf{LB}((\mathcal{D},W),FS)$
- $\mathbf{UB}((\mathcal{D},W),FS) = \mathrm{Cn}_{\mathcal{D}^{FS}}(W)$ where
  $$\mathcal{D}^{FS} = \{\frac{\mathrm{pre}(d)}{\mathrm{cons}(d)} \mid d \in \mathcal{D},\ \text{for all } \psi \in \mathrm{just}(\mathcal{D}), \neg\psi \notin FS\}.$$
  Example. For $FS = \{\neg q\}$, $\mathcal{D}^{FS} = \{\frac{}{\neg q}\}$ and $\neg p \notin \mathbf{UB}((\mathcal{D},W),FS)$
- To implement these efficiently a decision method for $W \vdash_R \varphi$ is needed.
  This is **co-NP**-complete (similar to classical consequence).

Slide 24

Goal-directedness

- **extensions** considers all defaults
- Appears to be unavoidable (holistic nature of default reasoning)
- Some amount of goal-directedness can be achieved.
- Search heuristics
- Using the query to guide the search:
- When deciding whether φ is a sceptical consequence of $(\mathcal{D},W)$:
  Start with $FS = \{\mathrm{not}(\varphi)\}$
  (i) can prune the search space considerably
  (ii) all extensions need not be generated fully in order to answer the query (sometimes none at all).
- For credulous consequence use $FS = \{\varphi\}$.
- For limited subclasses (e.g., semi-normal) more can be done.