Robust Knowledge Representation

Better half an answer in time than a full answer too late

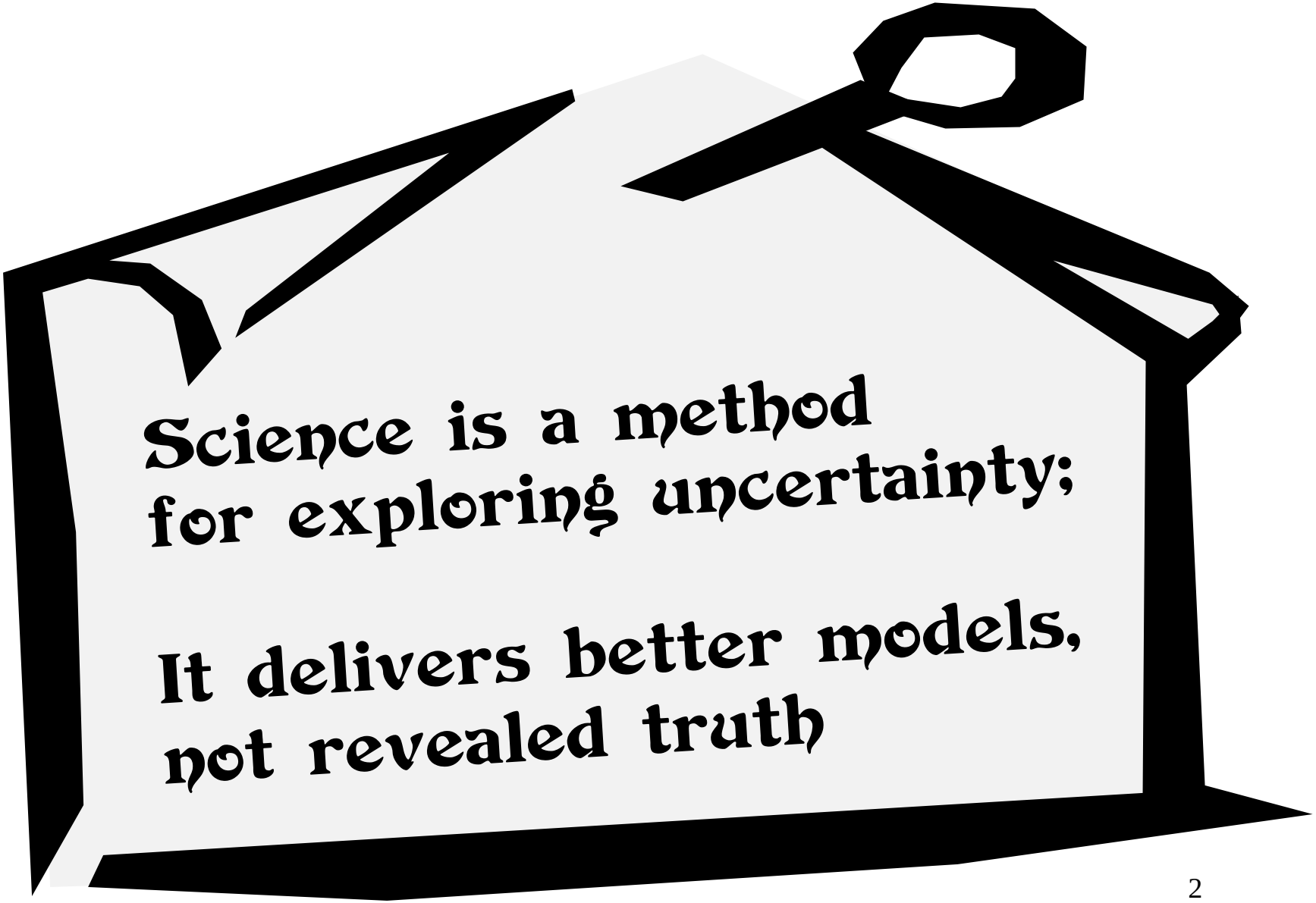
Frank van Harmelen

AI Department

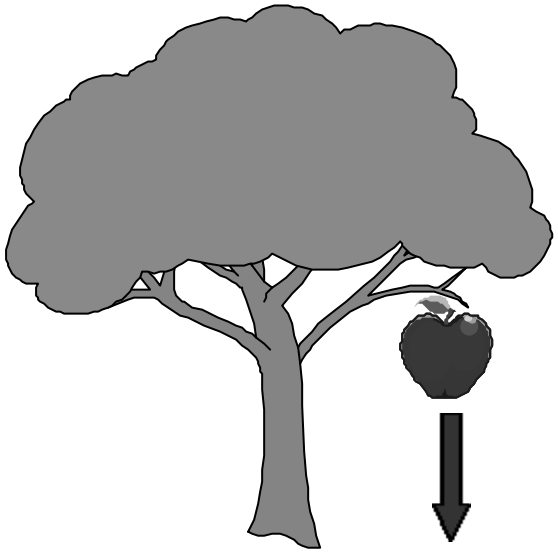
Vrije Universiteit Amsterdam

vrije Universiteit

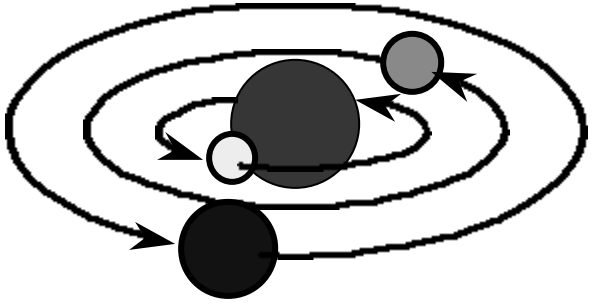
What is science about?



Science = making models



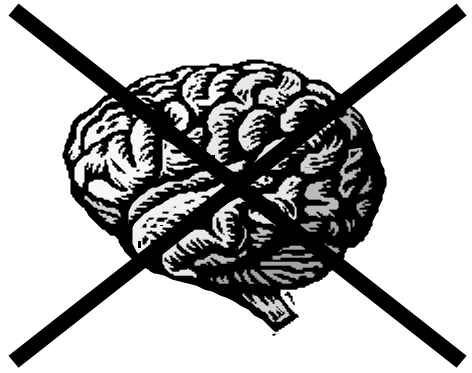
$$F = m \cdot a$$



$$F = \left(\frac{m}{1 - v^2 / c^2} \right) \cdot a$$

KR makes models of what?

- **Representation**: Structure of knowledge
  - Symbolic representation of knowledge
- **Inference**: Patterns of reasoning
  - Deriving new information from existing
  - algorithms, implementations



- **Examples**:
  - Traditional First Order Logic: Truth
  - Modalities: Knowledge, Belief
  - Non-monotonic reasoning: reasoning with exceptions
  - etc.

KR models are based on logic

An ideal reasoner
under ideal circumstances

- Reasoner makes no mistakes (sound & complete)
- Reasoner has unlimited resources
- All knowledge is available
- All knowledge is correct

KR models are based on logic

Reliance on logic is a **strength**

- Strong theoretical basis
- Well known properties
- Well known implementation techniques

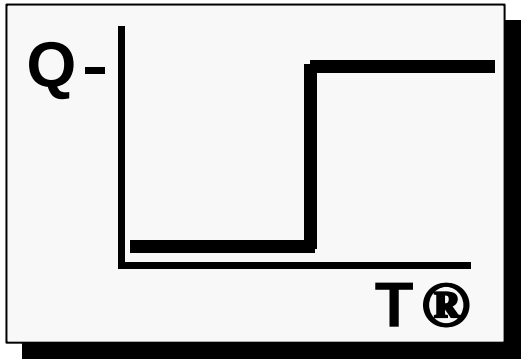
Reliance on logic is a **weakness**

- **Crisp** (no approximate answers)
- **Abrupt** (no intermediate answers)
- **Inefficient** (no time/quality trade-off)

Desiderata for Robust Knowledge Representation

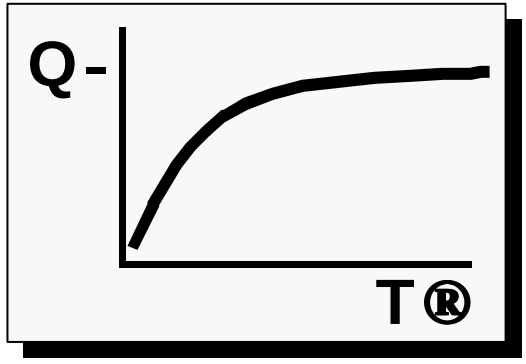
Reliance on logic is a weakness

- Crisp
- Abrupt
- Inefficient



Instead, we would want:

- **Approximate** answers
- **Incremental** computation
- **Anytime** cost/quality trade-off

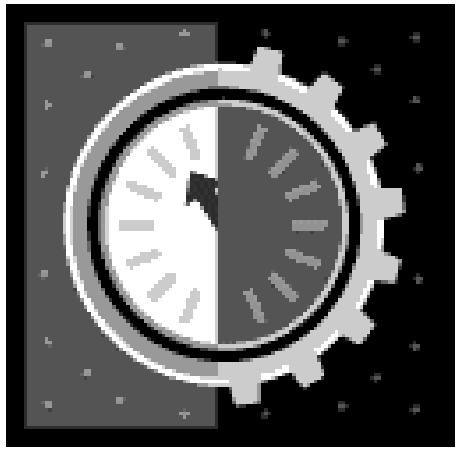


Can this be done *in logic* ?

- **YES!!**:
  I. Approximate deduction in diagnosis
  II. Qualitative performance profiles
  III. Empirical performance profiles

- Don’t abandon logic:
  - Neural Networks
  - Genetic Algorithms
  - Statistical models

Approximate Deduction: Intuition



- Turn the knob on the reasoning engine
  - exchange precision for cost
  - anytime reasoning = turn the knob gradually
  - characterise the effect of the approximation

  can we be precise about the imprecision ?

Part I: Approximate Deduction...

WHAT

- not yes/no answers, but
- optimise a quality measure
- NB: not necessarily numeric

WHY

- AI problems are intractable
- often approximate solutions suffice
- anytime behaviour

HOW

- define reasoning method using `
- replace ` by approximate deduction

… in diagnosis

- Dealing with
  - “no diagnosis”,
  - “too many diagnoses”
- Sometimes not interested in exact diagnosis (e.g. safe over-diagnosis)
- Prefer cheap approximation over expensive exact solution (time-pressure)
- Anytime algorithms

1,3-S (Cadoli & Schaerf)

- S = set of propositional letters
- classical inference on letters in S
- 1-S: unsound on letters outside S
- 3-S: incomplete on letters outside S

- 1-S-assignment:
  if $x \in S$ then x and $\neg x$ classical
  if $x \notin S$ then x and $\neg x$ both false
  $\Rightarrow$ for $x \notin S$ only 1 assignment
- 3-S-assignment:
  if $x \in S$ then x and $\neg x$ classical
  if $x \notin S$ then x and $\neg x$ not both false
  $\Rightarrow$ for $x \notin S$ 3 assignments:
  (1,0) ; (0,1) ; (1,1)

Intuitions for clausal form

- 1-S-assignment =
  - remove parts of clause outside S
  - if $a \notin S$: $\{a \vee b, \neg b \vee c\}$ becomes $\{b, \neg b \vee c\}$
  - theory might become $\bot$.
- 3-S-assignment =
  - remove entire clause if part outside S
  - if $a \notin S$: $\{a \vee b, \neg b \vee c\}$ becomes $\{\neg b \vee c\}$
  - theory might become $\top$.

Main result of Cadoli/Schaerf

$$\vdash_3^{\emptyset} \Rightarrow \vdash_3^{S} \Rightarrow \vdash_3^{S'} \Rightarrow \vdash_2 \Rightarrow \vdash_1^{S'} \Rightarrow \vdash_1^{S} \Rightarrow \vdash_1^{\emptyset}$$
$$\nvdash_2 \Leftarrow \nvdash_1^{S'} \Leftarrow \nvdash_1^{S} \Leftarrow \nvdash_1^{\emptyset}$$

- $\vdash_3^S$ is an incomplete approx. of $\vdash_2$
- $\vdash_1^S$ is an unsound approx. of $\vdash_2$, or:
  $\nvdash_1^S$ is an incomplete approx. of $\nvdash_2$

efficient incremental anytime algorithms:

cost of iterated computation is

never higher than computing $\vdash_2$ once!

Notice: approximate, incremental, anytime

Definition of diagnosis

- Given:
  - Behaviour model BM
  - Observations O
- Find
  - Explanation E
- Such that:

$$BM \cup E \vdash O$$

$$BM \cup E \nvdash \bot$$

written $ABD(E)$

- Replace ` by $`^{S}_{1,3}$

Main results

$$\begin{aligned} \emptyset = \{ABD_1^{\emptyset}\} &\subseteq \{ABD_1^{S}\} \subseteq^{*} \{ABD^{2}\} \\ \{ABD^{2}\} &\supseteq^{*} \{ABD_3^{S}\} \supseteq \{ABD_3^{\emptyset}\} = \emptyset \end{aligned}$$

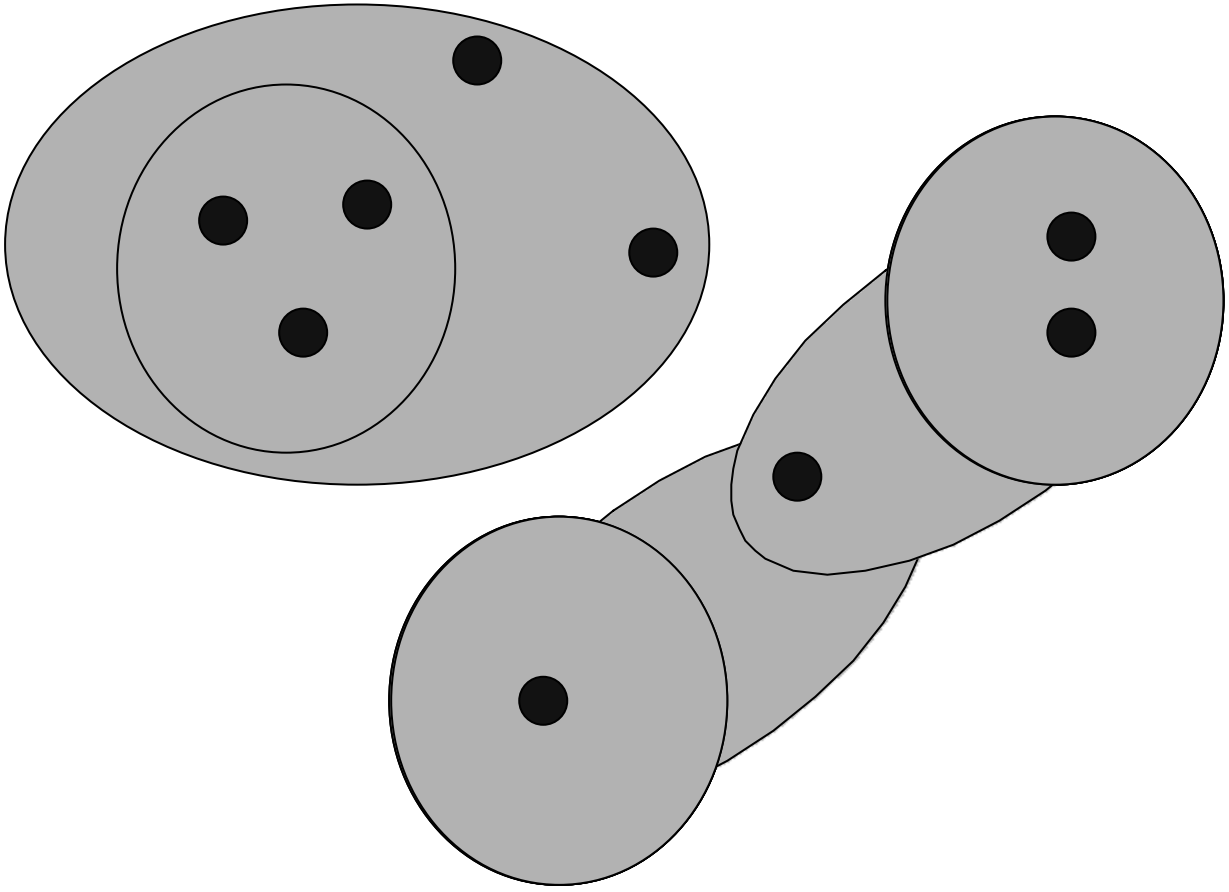
- ABD_1^S diagnoses are contained in classical diagnoses
- ABD_3^S diagnoses contain classical diagnoses

- When S grows
- ABD_1^S no new subdiagnoses
- ABD_3^S no new superdiagnoses

$$\begin{aligned} 0 = |\{ABD_1^{\emptyset}\}| &\leq |\{ABD_1^{S}\}| \leq |\{ABD_2\}| \\ |\{ABD_2\}| &\geq |\{ABD_3^{S}\}| \geq |\{ABD_3^{\emptyset}\}| = 0 \end{aligned}$$

Intuition

ABD_3^S

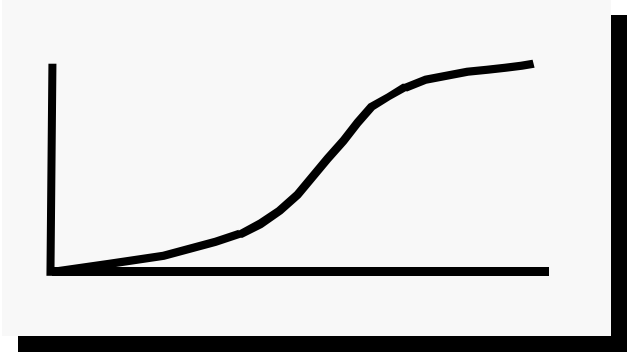


Strategies for choosing S

- ABD_1^S = all urgent subsets of classical diagnoses
- ABD_3^S = all classical diagnoses that are entirely urgent
- Increase S with less urgent causes, interrupt when
  - No time left:
    only non-urgent diagnoses lost
  - First diagnosis found:
    most urgent diagnosis

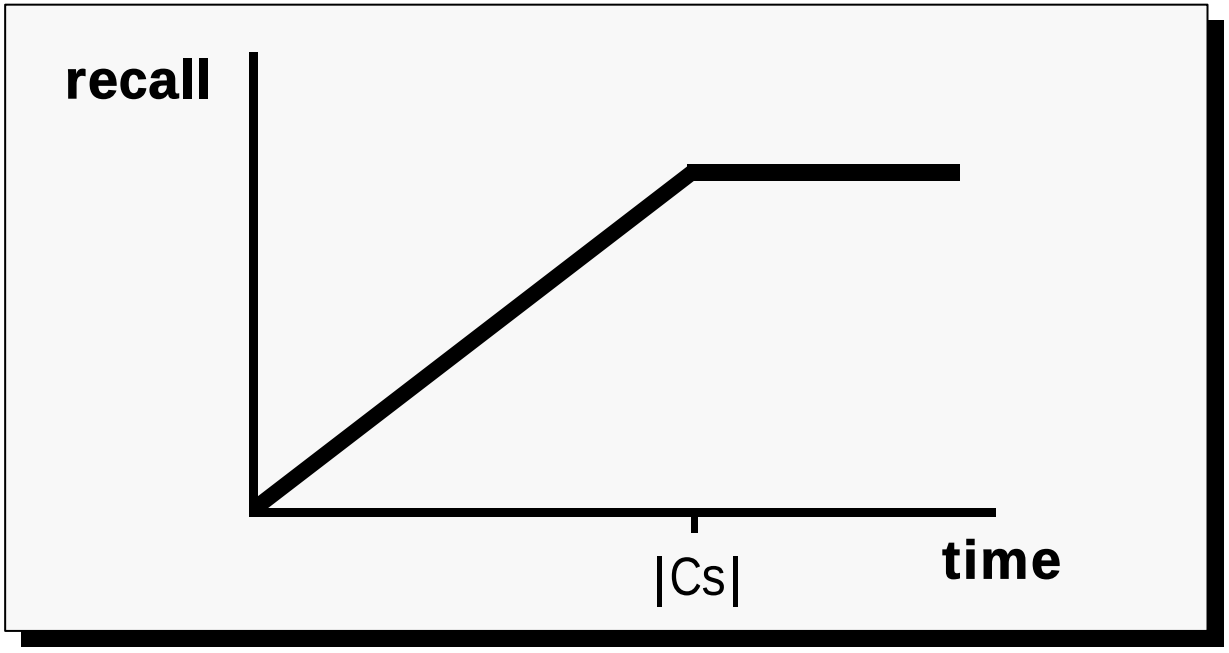
Part II:
Qualitative Performance profiles

- Output-quality is function of some varying resource
  - reasoning time,
  - inference accuracy,
  - representational precision
- This function is (ideally)
  - monotonic
  - diminishing returns
  - characterised by a performance profile



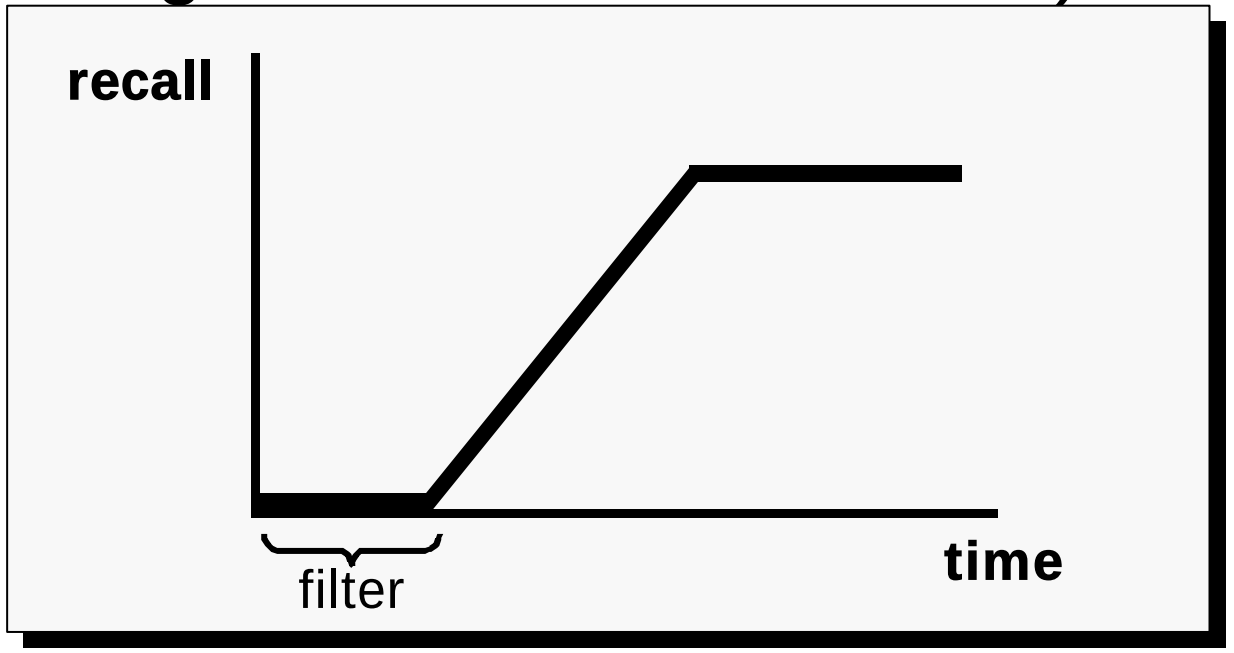
Classification by linear candidate confirmation

1. Iterate over all classes
2. Check every class with the observations; (leading to confirmation or not)



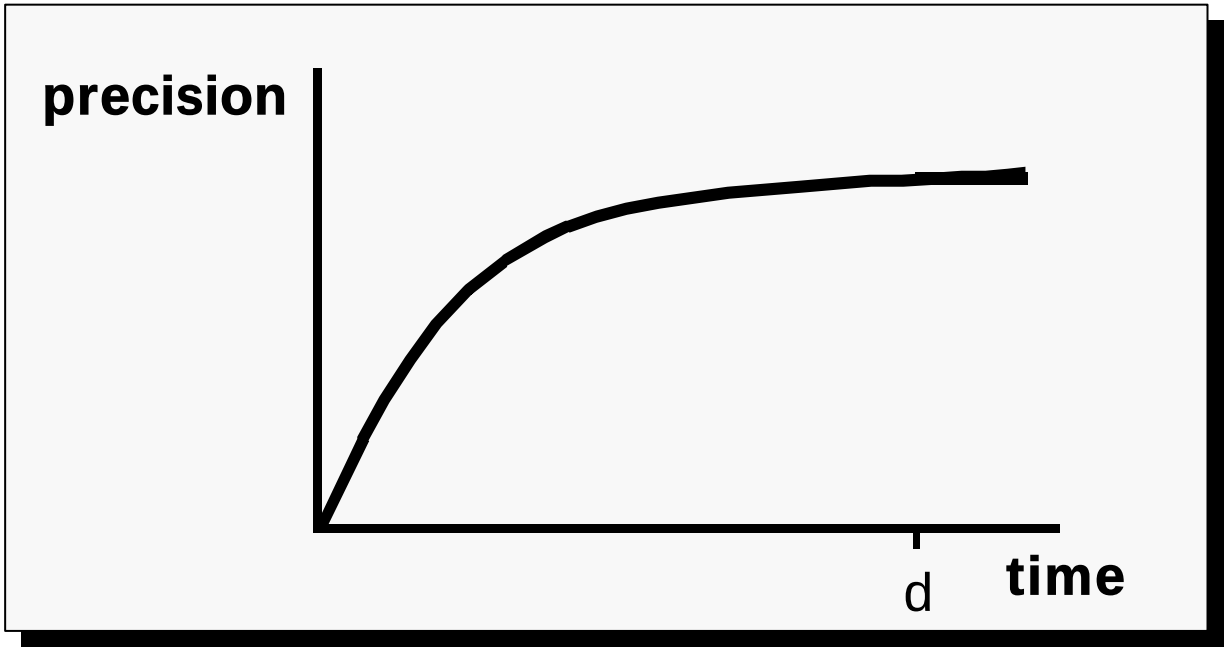
Classification by confirmation with filtering

1. Filter the classes,
   based on a subset of the observations
2. Iterate over all classes
3. Check every class with the observations;
   (leading to confirmation or not)



Hierarchical classification

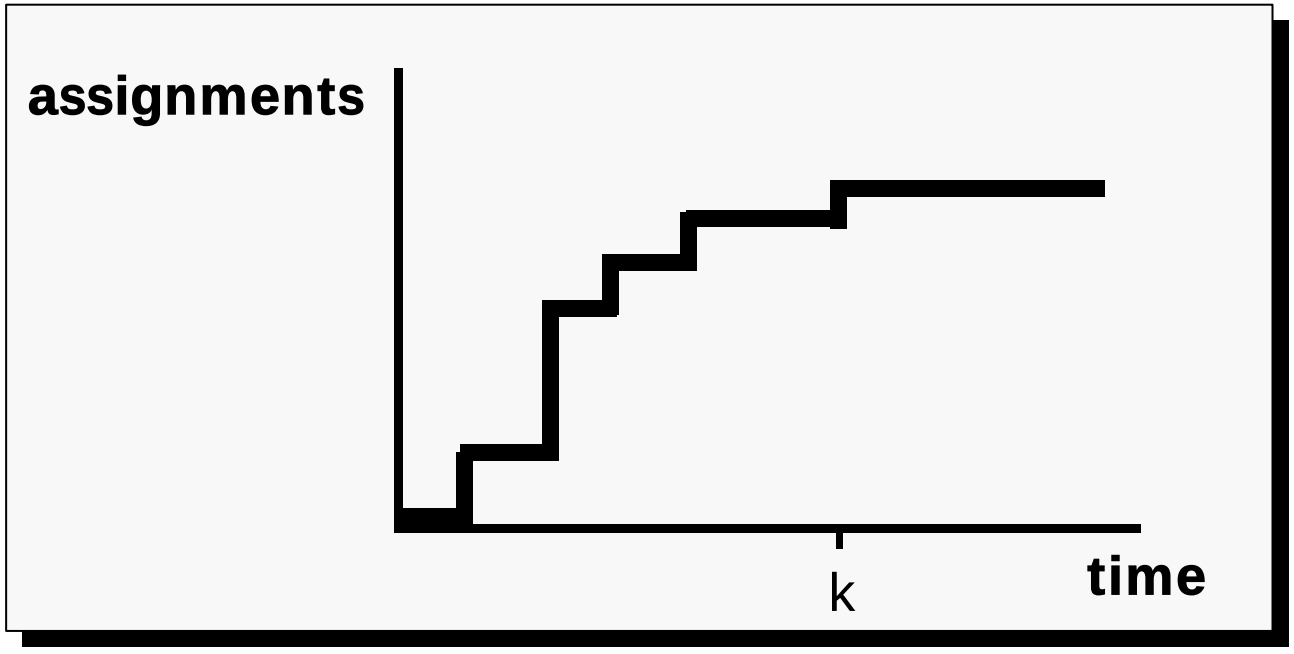
1. First consider all classes as solutions
2. Descend a classification hierarchy (depth d), eliminating all classes on entire branches



Design by Constraint clustering

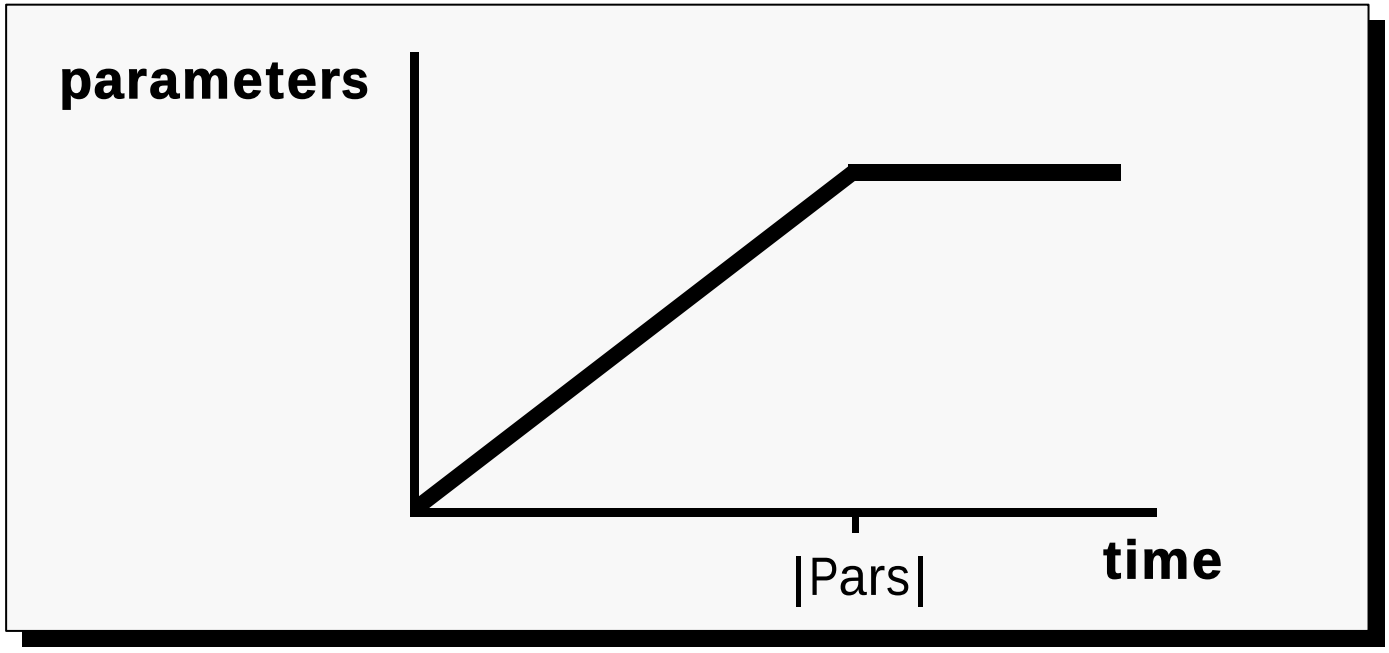
Group constraints in non-interacting clusters

1. Iterate over all k clusters
2. Find an assignment per cluster



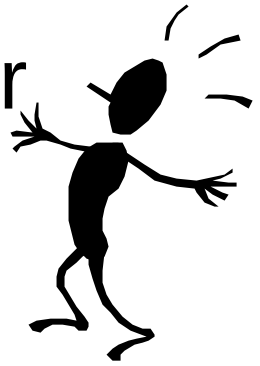
Design by Propose & Revise

- Assign successive parameters
- Test partial designs
- Re-assign earlier parameters if needed



Summarising

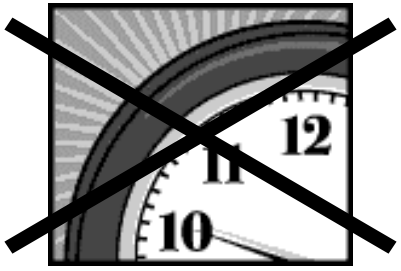
- Many inference methods have surprisingly natural anytime behaviour



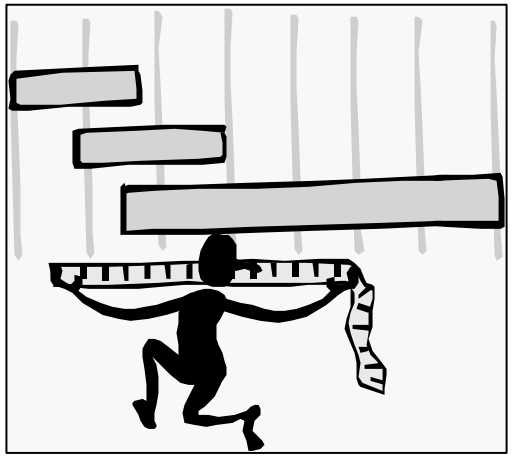
- Problem:
  - Only upper/lower bound, but no quantitative measures
  - Do search methods really behave like this in practice?

Part III:
Quantitative Performance Profiles

- How does quality of output change as function of
  - quality of input ?
  - quality of knowledge base ?

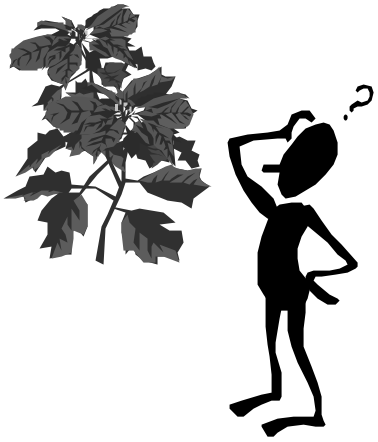


- Measure quantitative profiles



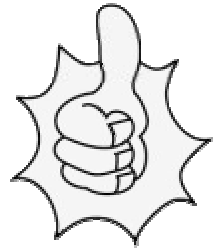
Experimental setting

- Vegetation classification system
  - 93 plant names
  - 40 observables (max. 30 per case)
  - 7586 rules
  - 150 test cases
- Use recall and precision as quality measures
- Incomplete input
- Incomplete knowledge base
- Incorrect knowledge base

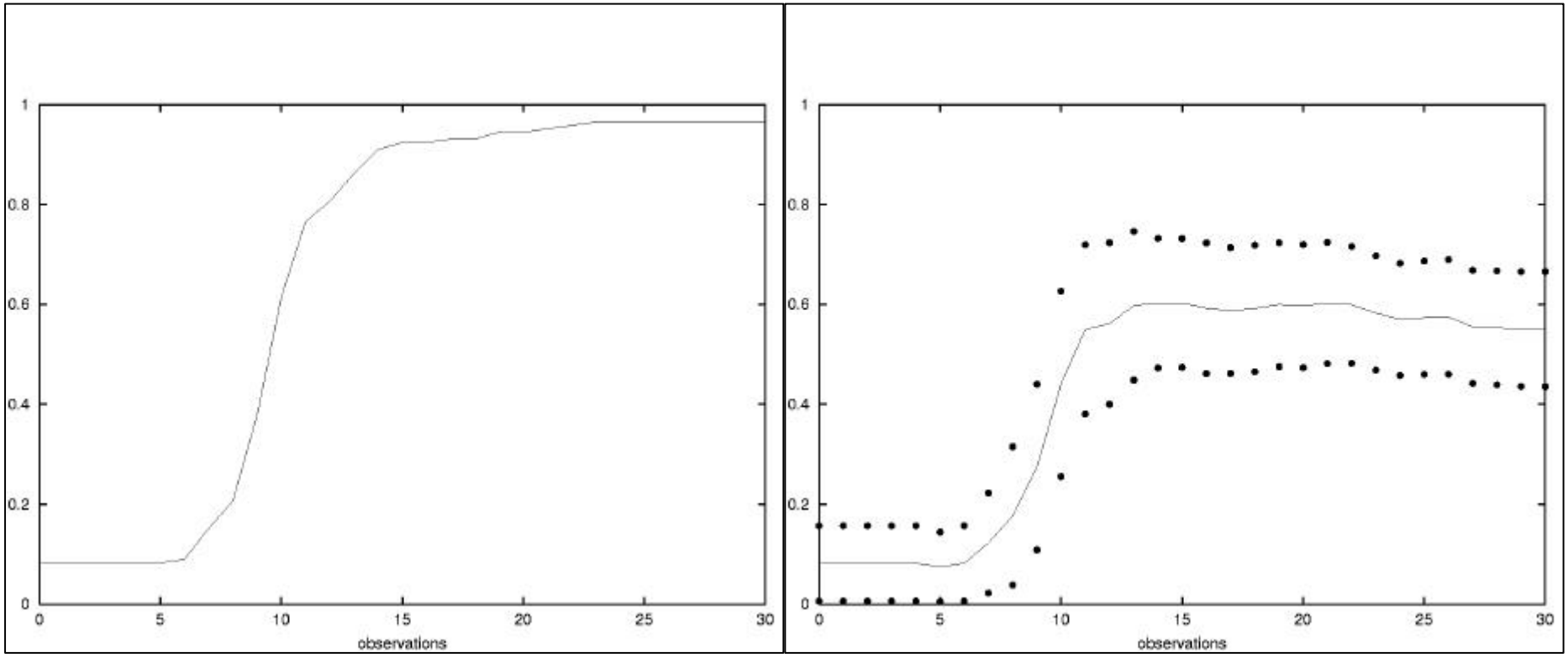


Experimental results (1)

Incomplete input:

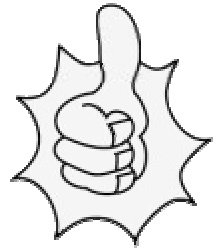


Recall: precision:

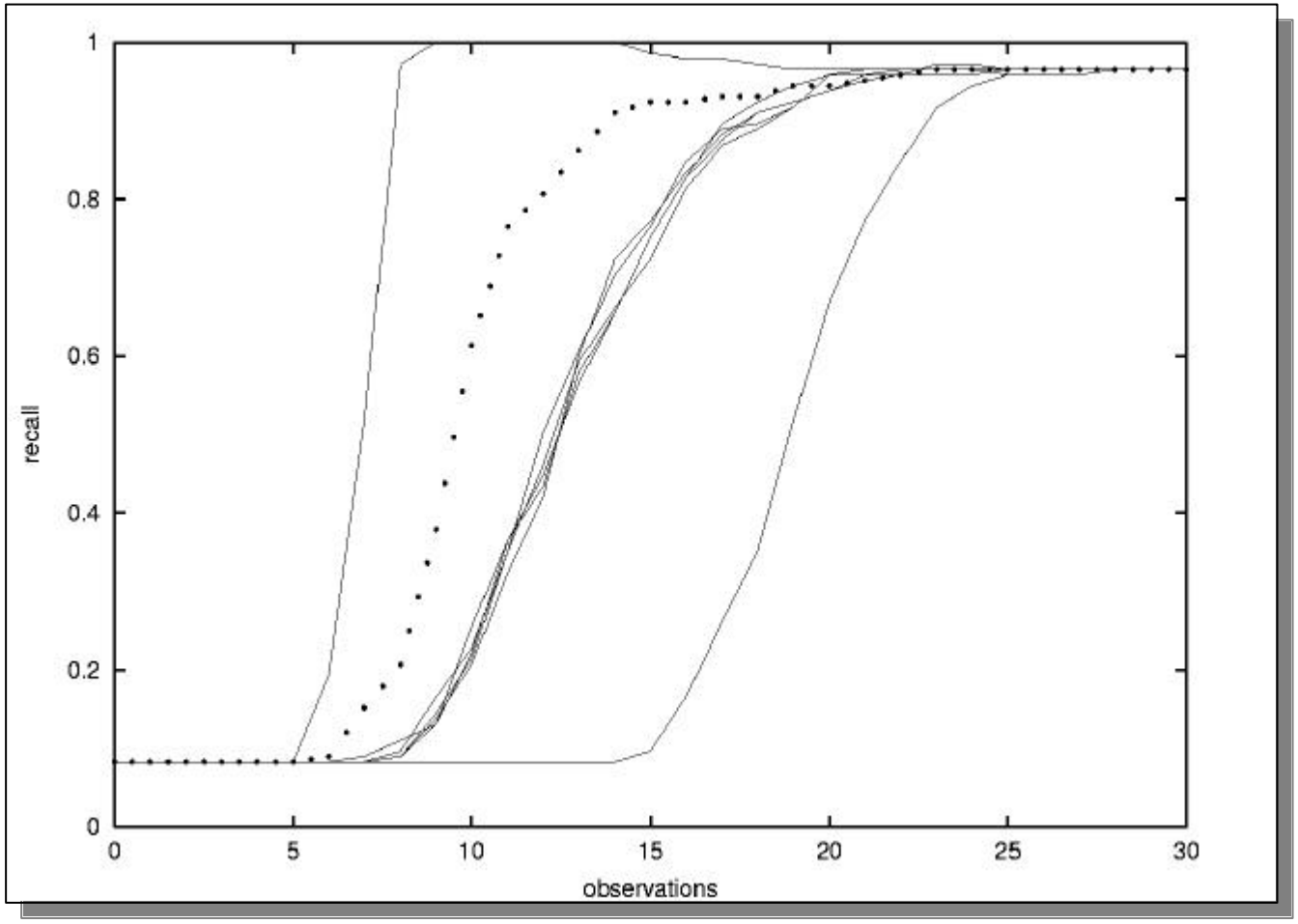


Experimental results (2)

Incomplete input:

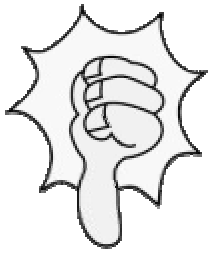


Recall with different input orderings:

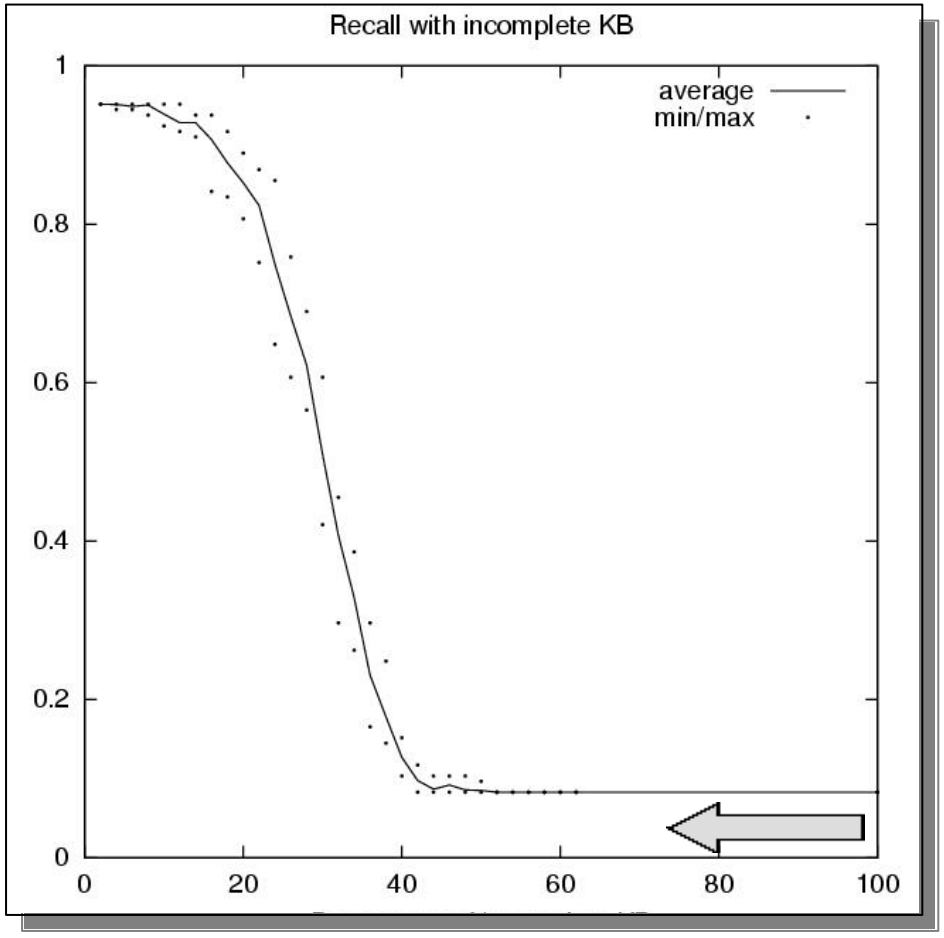


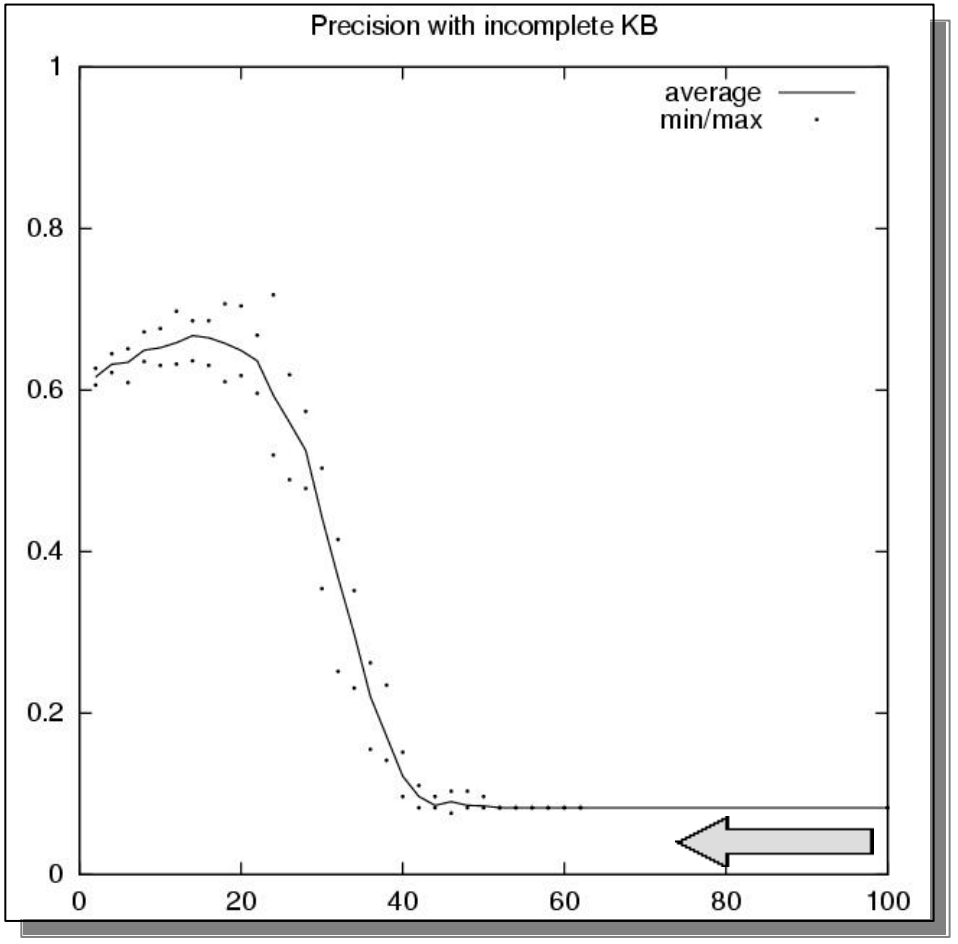
Experimental results (3)

Incomplete knowledge base:

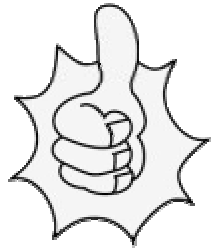


(with realistic removal model)





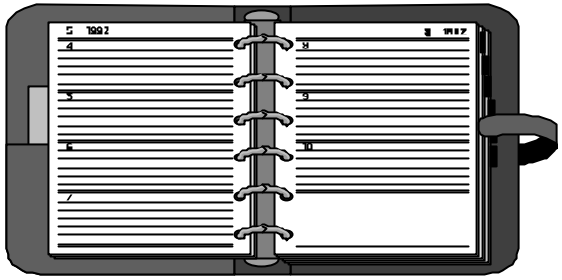
Can this be done *in logic* ?



An ideal reasoner
under ideal circumstances ?

- Reasoner makes no mistakes (sound & complete)
  - ➔ Cadoli & Schaerf (part I)
- Reasoner has unlimited resources
  - ➔ Qualitative performance profiles (part II)
- All knowledge is available & correct
  - ➔ Quantitative performance profiles (part III)

Research agenda



- Other approximate deduction relations?
- Exploit other methods:
  - Knowledge compilation (Kautz & Selman)
  - Language weakening
- Relations between these?
- New application areas:
  - Semantic Web
    (approximate Description Logics)
  - Agent communication
    (approximate terminology mappings)
  - Software retrieval
    (approximate pre/post-conditions, Web services)