AILog - Logical Reasoning

- Representation and reasoning system for logical reasoning
- Includes facilities normally found in tools for (logical) knowledge systems
- Supports various (sophisticated) reasoning methods
- Supports reasoning with uncertainty using ideas from probabilistic logics

However, experimental tool, not meant for development of real-world applications

Syntax and semantics

- Horn clauses (similar but not identical to Prolog):

$$A \;\texttt{<-}\; B_1 \,\&\, \cdots \,\&\, B_n.$$

- Negation by failure: $\sim$
- Directives:
  - Askable atoms: `askable p.`
  - Asumable atoms: `asumable p.`
- What follows from a knowledge bases:
  KB $\vdash \varphi$, with φ atom, becomes: ask φ.
  - No. φ doesn't follow from the knowledge base (what about undecidability?)
  - Answer: φ.

Ask-tell interface

- Tell: add a clause to the knowledge base (not needed if the knowledge based is loaded into AILog)
- Ask: query a knowledge base about whether the atom can be derived

Example:

```
ailog: tell parent(X, Y) <- mother(X,Y).
ailog: tell parent(X, Y) <- father(X, Y).
ailog: tell mother(bea,alex).
ailog: ask parent(U,V).
Answer: parent(bea, alex).
  [ok,more,how,help]:
```

(Note the differences with Prolog)

Askable atoms

During the reasoning process, atoms can be asked to users:

```
ailog: tell parent(X, Y) <- mother(X,Y).
ailog: tell parent(X, Y) <- father(X, Y).
ailog: askable mother(X,Y).
ailog: ask parent(bea,alex).
Is mother(bea, alex) true?
        [yes,no,unknown,why,help]: yes.
Answer: parent(bea, alex).
```

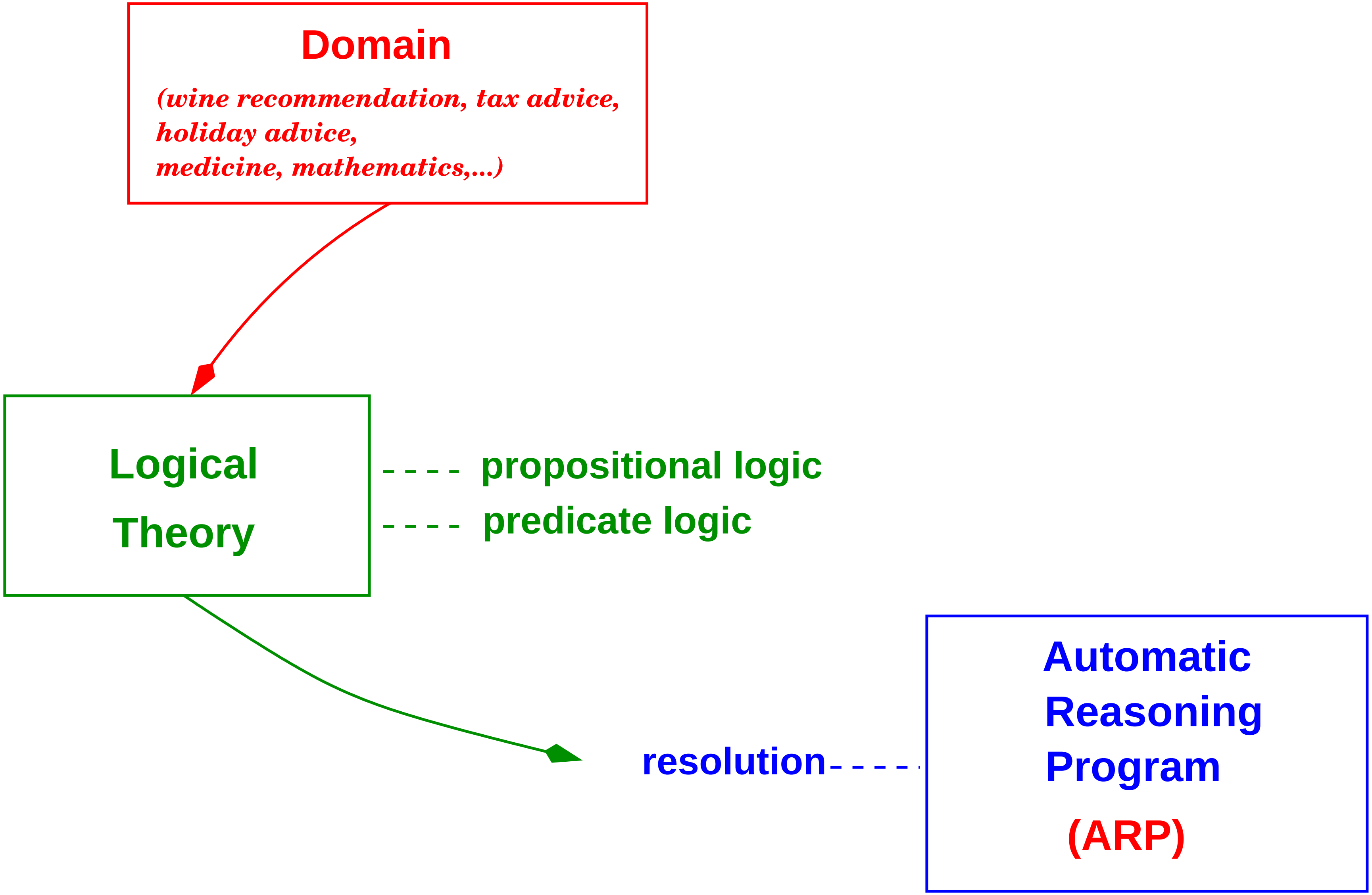
When the atom is asked, it should be fully instantiated when the result returns

Trace and explanation facilities

- Trace: as in Prolog meant to follow the reasoning process step-by-step

- Expanation facilities:
  - Why: why is a particular question asked? (Will go through a potential derivation)
  - How: how was a particular fact derived? (Will go through a derivation that was already successful)
  - Whynot: why is it not possible to derive a fact?

AILog and knowledge systems



Domain

(wine recommendation, tax advice,
holiday advice,
medicine, mathematics,...)

Logical Theory

- - - - propositional logic

- - - - predicate logic

resolution- - - - -

Automatic Reasoning Program

(ARP)

Example knowledge base

```
askable age(Patient) = V.
askable duration(Patient,heart,pain) = V.
askable diabetic(Patient).
               ...
sclerotic_arteries(Patient) <-
 age(Patient) = A &
 A > 50.
               ...

disorder(Patient,stable_angina_pectoris) <-
  state(Patient,atherosclerosis) &
  state(Patient,ischaemia,reversible) &
  o2_demand(Patient,heart,increased).

condition(Patient,emergency) <-
  disorder(Patient,myocardial_infarction).

condition(Patient,no_emergency) <-
  disorder(Patient,stable_angina_pectoris)
```

Why

```
What is the value of duration(john, heart, pain)?
        [value,unknown,why,help]: why.
   duration(john, heart, pain)=A is used in the rule
   state(john, ischaemia, irreversible) <-
      1: ischaemia(john, heart)
   ** 2: duration(john, heart, pain)=A
      3: A>30
      4: ~state(john, coronary_vasodilatation)
   [Number,why,help,ok]: why.
   state(john, ischaemia, irreversible) is used in the rule
   disorder(john, unstable_angina_pectoris) <-
      1: o2_demand(john, heart, normal)
   ** 2: state(john, ischaemia, irreversible)
      3: state(john, atherosclerosis)
   [Number,why,help,ok]:
```

How

```
ailog: ask disorder(john,Y).
Is pain(john) true? [yes,no,unknown,why,help]: yes.
What is the value of duration(john, heart, pain)?
    [value,unknown,why,help]: 50.
Is pattern(john, pain, left_lateral_thoracic) true?
    [yes,no,unknown,why,help]: no.
Is pattern(john, pain, right_lateral_thoracic) true?
    [yes,no,unknown,why,help]: why.
   pattern(john, pain, right_lateral_thoracic) is used in the rule
   ischaemia(john, heart) <-
      1: pain(john)
   ** 2: pattern(john, pain, right_lateral_thoracic)
   [Number,why,help,ok]: how 1.
   You told me pain(john) is true.
   pattern(john, pain, right_lateral_thoracic) is used in the rule
   ischaemia(john, heart) <-
      1: pain(john)
   ** 2: pattern(john, pain, right_lateral_thoracic)
   [Number,why,help,ok]:
```

Methods

- Deductive solution: S is a *deductive solution* of a problem with associated set of observed findings F iff

  $$\mathrm{KB} \cup F \vDash S$$

  where S is a set of solution formulae
- Abductive/inductive solution: S is an *abductive solution* of a problem with observed findings F

  $$\mathrm{KB} \cup S \cup K \vDash F$$

  is satisfied *covering condition*
- Consistency-based solution: S is a *consistency-based solution* with observed findings F:

  $$\mathrm{KB} \cup S \cup F \nvDash \Box$$

Conclusions

- Prolog is a simple and efficient (logic) programming language in which knowledge representation and reasoning systems can be implemented easily

- AILog is such a knowledge representation and reasoning system:
  - sophisticated reasoning methods (deductive, abductive, consistency-based)
  - reasoning with uncertainty (including Bayesian networks!)
  - user interaction