Logic in Practice

Horn Clauses

$x \wedge y \rightarrow z \equiv \neg x \vee \neg y \vee z$

Horn clause: at most one positive literal (exactly one is definite clause)

Cat(x) $\rightarrow$ Furry(x), Meows(x).

Cat(y) $\rightarrow$ Feline(y).

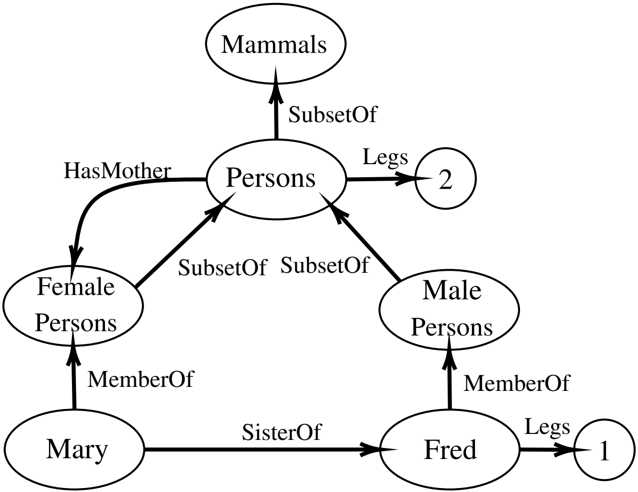
Furry(A).

Meows(A).

? Cat(z).

Undecidable query

Semantic Networks



Semantic Networks

Multiple Aspects:

- Visual notation
- Restricted Logic
- Set of implementation tricks

Typically in implementation:

- Efficient indexing
- Precomputation
- Methods for defaults

Also known as: frames, inheritance networks, semantic graphs, description logics, ontologies

Description Logic

Computing categories and membership, including:

1. subsumption
2. classification
3. inheritance

Missing many things:

- negation
- disjunction
- nested functions
- existentials
- intractability

Description Logic

- concepts (primitive and derived), instances
- roles and properties
- subsumptions: $subsumes(x, y)$ iff:
  - x is a concept and
  - x, y have same ancestor and
  - for each role of x with restriction r_x, y has the same role with a restriction r_y that r_x subsumes

Satisfiability

Boolean Satisfiability

Given a boolean logic formula, is there a T/F assignment to all variables that make the formula true?

$$(a \vee b \vee c) \wedge (\neg a \vee b \vee \neg c) \wedge (\neg a \vee \neg b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$$

Davis-Putnam-Logemann-Loveland Algorithm

DPLL(α)

if α has no clauses, **return** true
if α has an empty clause, **return** false
if α contains a unit clause, **return** DPLL(SIMPLIFY(α, *literal*))
$v \leftarrow$ choose a literal in α
if DPLL(SIMPLIFY(α, v)) is true, **return** true
else **return** DPLL(SIMPLIFY($\alpha, \neg v$))

SIMPLIFY(α, *literal*)

remove clauses in α where *literal* is positive
remove $\neg$*literal* from clauses where it appears
return new α

Local SAT search

1. Start with random solution
2. Repeatedly flip a variable to satisfy the most clauses
3. After a while, restart

GSAT, WalkSAT choose next variable differently

Both are much more efficient in practice than DPLL

Logics of Action

Event Calculus

Events and fluents are reified:

$$T(Equals(President(USA), George), Begin(1790), End(1790))$$

Situation Calculus

World state (situation) is reified:

$$Result(GoForward, s_0) = s_1$$

$$Result(Turn(Left), s_1) = s_2$$

$$\forall s, a, b \; Clear(a, s) \land Clear(b, s) \rightarrow On(a, b, Result(PutOn(a, b), s))$$

Problems with Logic

- **Defaults**: difficult to build coherent semantics and efficient inference (default logics, probabilistic logic)
- **Ramification problem**: chosing what to infer (specialized systems)
- **Retraction**: when something changes from true to false (truth maintenance systems)
- **Qualification Problem**: making rules correct (probabilistic logic)