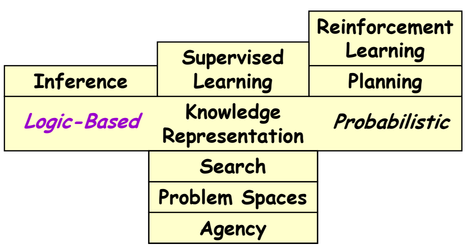
Knowledge Representation II

CSE 573

Logistics

- Reading for Wednesday
  - Ch 11 "Planning"
- Projects
  - Did we get everyone?
- Office Hour
  - Monday 3-4pm
  - Except…. Today only 3-3:20

573 Topics



Review of "Last Time"

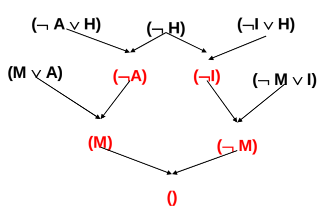
- Propositional Logic
  - Resolution
  - DPLL
  - WalkSAT
- Expressiveness *vs.* Tractability
- Randomly Generating SAT

Resolution

If the unicorn is mythical, then it is immortal, but if it is not mythical, it is a mammal. If the unicorn is either immortal or a mammal, then it is horned.

Prove: the unicorn is horned.

M = mythical
I = immortal
A = mammal
H = horned



DPLL (for real!)

Davis – Putnam – Loveland - Logemann

```
dpll(F, literal){
  remove clauses containing literal
  if (F contains no clauses) return true;
  shorten clauses containing ¬literal
  if (F contains empty clause)
     return false;
  if (F contains a unit or pure L)
     return dpll(F, L);
  choose V in F;
  if (dpll(F, ¬V))return true;
  return dpll(F, V);
}
```

WalkSat

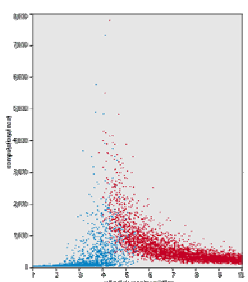
- Local search over space of ***complete*** truth assignments
  - With probability P: flip any variable in any unsatisfied clause
  - With probability (1-P): flip best variable in any unsat clause
    - Like fixed-temperature simulated annealing
- SAT encodings of N-Queens, scheduling
- Best algorithm for random K-SAT
  - Best DPLL: 700 variables
  - Walksat: 100,000 variables

[Slide #s from 2001]

Horn Theories

- Recall the special case of Horn clauses:
  - At most one positive literal / clause
  - E.g. from "If (fever) AND (vomiting) then FLU"
  - Unit propagation is refutation complete for Horn theories
  - Good implementation – linear time inference!
- Binary clauses
  - Linear-time inference

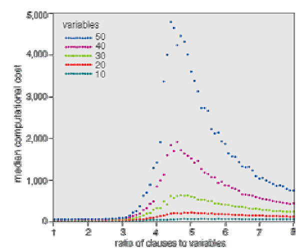
Random 3-SAT



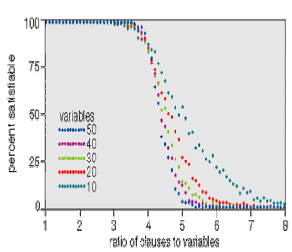
- Random 3-SAT
  - sample uniformly from space of all possible 3-clauses
  - *n* variables, *l* clauses
- Which are the hard instances?
  - around *l/n* = 4.3

Random 3-SAT

- Varying problem size, *n*
- Complexity peak appears to be largely invariant of algorithm
  - backtracking algorithms like Davis-Putnam
  - local search procedures like GSAT
- *What's so special about 4.3?*



Random 3-SAT



- Complexity peak coincides with solubility transition
  - l/n < 4.3 problems under-constrained and SAT
  - l/n > 4.3 problems over-constrained and UNSAT
  - l/n=4.3, problems on "knife-edge" between SAT and UNSAT

Real-World Phase Transition Phenomena

- Many NP-hard problem distributions show phase transitions -
  - job shop scheduling problems
  - TSP instances from TSPLib
  - exam timetables @ Edinburgh
  - Boolean circuit synthesis
  - Latin squares (alias sports scheduling)
- Hot research topic: predicting hardness of a given instance, & using hardness to control search strategy (Horvitz, Kautz, Ruan 2001-3)

Themes

- Expressiveness
  - Expressive but awkward
  - No notion of objects, properties, or relations
  - Number of propositions is fixed
- Tractability
  - NPC in general
  - Completeness / speed tradeoff
  - Horn clauses, binary clauses

Logic-Based KR

√ Propositional logic
- √ Syntax (CNF, Horn clauses, …)
- √ Semantics (Truth Tables)
- √ Inference (FC, Resolution, DPLL, WalkSAT)
- √ Restricted Subsets

First-order logic
- Syntax (quantifiers, skolem functions, …
- Semantics (Interpretations)
- Inference (FC, Resolution, Compilation)
- Restricted Subsets (e.g. Frame Systems)

Representing events, action & change

Propositional. Logic *vs.* First Order

Ontology	Facts (P, Q)	Objects, Properties, Relations
Syntax	Atomic sentences Connectives	Variables & quantification Sentences have structure: terms father-of(mother-of(X)))
Semantics	Truth Tables	Interpretations *(Much more complicated)*
Inference Algorithm	DPLL, GSAT Fast in practice	Unification Forward, Backward chaining Prolog, theorem proving
Complexity	NP-Complete	Semi-decidable

FOL Definitions

- ***Constants***: a,b, dog33.
  - Name a specific object.
- ***Variables***: X, Y.
  - Refer to an object without naming it.
- ***Functions***: dad-of
  - Mapping from objects to objects.
- ***Terms***: dad-of(dog33)
  - Refer to objects
- ***Atomic Sentences***: in(dad-of(dog33), food6)
  - Can be true or false
  - Correspond to propositional symbols P, Q

More Definitions

- **Logical connectives**: and, or, not, =>
- **Quantifiers**:
  - ∀ Forall
  - ∃ There exists
- **Examples**
  - Dumbo is grey
  - Elephants are grey
  - There is a grey elephant

Quantifier / Connective Interaction

E(x) == "x is an elephant"
G(x) == "x has the color grey"

1. $\forall x\ E(x) \wedge G(x)$
2. $\forall x\ E(x) \Rightarrow G(x)$
3. $\exists x\ E(x) \wedge G(x)$
4. $\exists x\ E(x) \Rightarrow G(x)$

Nested Quantifiers: Order matters!

$\forall x \exists y\ P(x,y) \neq \exists y \forall x\ P(x,y)$

- Examples

Every dog has a tail

$\forall d \exists t\ has(d,t)$

?

Every dog *shares* a tail!

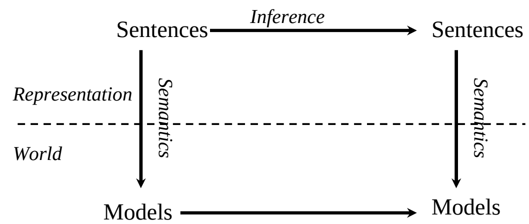
$\exists t \forall d\ has(d,t)$

Someone is loved by everyone

$\exists x \forall y\ loves(y, x)$

Semantics

- ***Syntax***: a description of the *legal* arrangements of symbols
  (Def "sentences")
- ***Semantics***: what the arrangement of symbols *means* in the world



Propositional Logic: SEMANTICS

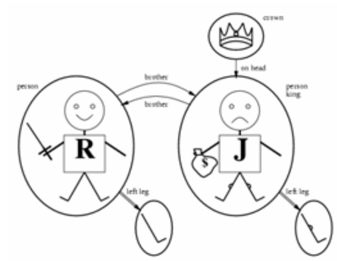
- "Interpretation" (or "possible world")
- Specifically, TRUTH TABLES
  Assignment to each variable either T or F
  Assignment of T or F to each connective

P \ Q	T	F
T		
F		

$P \wedge Q$

Models

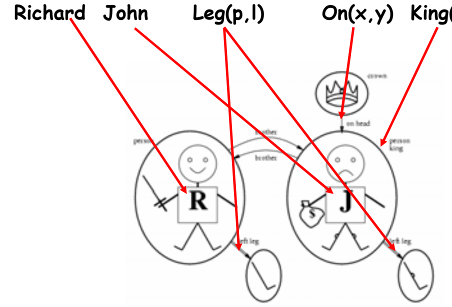
- Depiction of one possible "real-world" model



Interpretations=Mappings syntactic tokens → model elements

Depiction of one possible interpretation, assuming

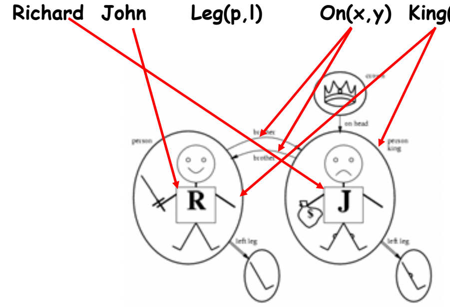
Constants: Richard John
Functions: Leg(p,l)
Relations: On(x,y) King(p)



Interpretations=Mappings syntactic tokens → model elements

Another interpretation, same assumptions

Constants: Richard John
Functions: Leg(p,l)
Relations: On(x,y) King(p)



Satisfiability, Validity, & Entailment

- S is **valid** if it is true in all interpretations
- S is **satisfiable** if it is true in some interp
- S is **unsatisfiable** if it is false all interps
- S1 **entails** S2 if (|=)
  forall interps where S1 is true,
  S2 is also true

Skolemization

- Existential quantifiers aren't necessary!
  Existential variables can be replaced by
  - Skolem functions (or constants)
  - Args to function are all surrounding ∀ vars
- ∀d ∃t has(d, t)

  ∀d has(d, f(d))
- ∃*x* ∀*y* loves(*y*, *x*)

  ∀y loves(y, f())

  ∀y loves(y, f_{97})

FOL Reasoning

- FO Forward & Backward Chaining
- FO Resolution
- Many other types of theorem proving
- Restricted representations
  Description logics
  Horn Clauses
- Compilation to SAT

Forward Chaining

- Given
  ∀?x lifeform(?x) => mortal(?x)
  ∀?x mammal(?x) => lifeform(?x)
  ∀?x dog(?x) => mammal(?x)
  dog(fido)
- Prove
  mortal(fido)

∀?x dog(?x) => mammal(?x)
dog(fido)
―――――――――――――― ?
mammal(fido)

Unification

- Emphasize variables with ?
- Useful for FO inference (modus ponens, …)
  Also for compilation of FOPC -> propositional
- Unify(Φ, Ψ) returns "mgu"
  Unify(city(?a), city(kent)) returns ?a/kent
- Substitute(expr, mapping) returns new expr
  Substitute(connected(?a, ?b), {?a/kent})
  returns connected(kent, ?b)

Unification Examples

- Unify(road(?a, kent), road(seattle, ?b))
- Unify(road(?a, ?a), road(seattle, kent))
- Unify(f(g(?x, dog), ?y)), f(g(cat, ?y), dog)
- Unify(f(g(?x)), f(?x))

Resolution
[Robinson 1965]

{ (p ∨ α), (¬ p ∨ β ∨ γ) } |-$_R$ (α ∨ β ∨ γ)

Recall Propositional Case:
- **Literal in one clause**
- **Its negation in the other**
- **Result is disjunction of *other* literals**

First-Order Resolution
[Robinson 1965]

{ (p(?x) ∨ a(a), (¬ p(q) ∨ b(?x) ∨ c(?y)) }

|-$_R$

(a(a) ∨ b(q) ∨ c(?y))

- **Literal in one clause**
- **The negation of *something which unifies* in the other**
- **Result is disjunction of other literals / mgu**

First-Order Resolution

- Is it the case that Σ |= Φ ?
- Method
  - Let ϑ = Σ ∧ ¬Φ
  - Convert ϑ to clausal form
    - Standardize variables
    - Move quantifiers to front, skolemize to remove ∃
    - Replace ⇒ with ∨ and ¬
    - Demorgan's laws...
  - Resolve until get empty clause

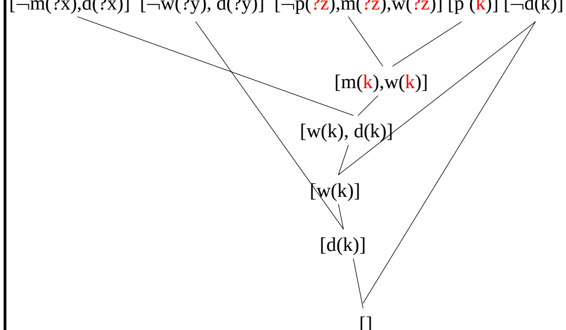
Example

- Given
  - ∀?x man(?x) => mortal(?x)
  - ∀?x woman(?x) => mortal(?x)
  - ∀?x person(?x) => man(?x) ∨ woman(?x)
  - person(kelly)
- Prove
  - mortal(kelly)

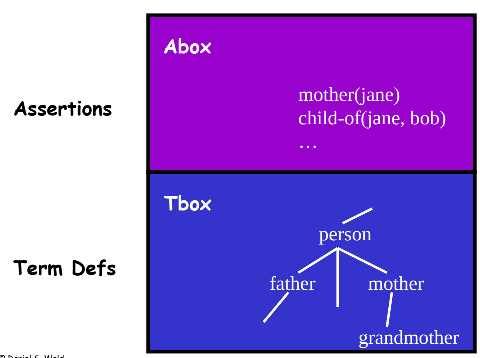
[¬m(?x),d(?x)] [¬w(?y), d(?y)] [¬p(?z),m(?z),w(?z)] [p (k)][¬d(k)]

Example Continued

[¬m(?x),d(?x)] [¬w(?y), d(?y)] [¬p(?z),m(?z),w(?z)] [p (k)] [¬d(k)]



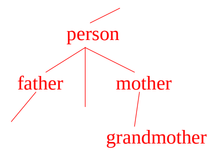
KR with Description Logics



Tbox

- Term definitions
- FO Language + inference organized into a taxonomy, e.g:
  - father(x) = person(x) ∧male(x) ∧ ∃y childof(y,x)
  - parent(x) = person(x) ∧ ∃y childof(y,x)
- Complexity of classifying new terms
  - subsumption

Subsumption hierarchy →



Abox

- Assertions
- Abox – separate language + inference for "propositional" assertions using Tbox terms
  - e.g. person(kelley)

Debate

- Restricted language thesis
  - Disjunction, negation, particularization, order…
  - Natural kinds
- Restricted classification thesis
  - Concepts using contingent information:
  - Treatable disease, democratic country, illegal act
- Counterargument
- Constructs: Omit vs limit
  - Completeness
  - Efficiency

Compilation to Prop. Logic I

- Typed Logic
  - $\forall_{city}\ a,b$ connected(a,b)
- Universe
  - Cities: seattle, tacoma, enumclaw
- Equivalent propositional formula:

Compilation to Prop. Logic II

- Universe
  - Cities: seattle, tacoma, enumclaw
  - Firms: IBM, Microsoft, Boeing
- First-Order formula
  - $\forall_{city}\ c\ \exists_{firm}\ f$ hasHQ(c, f)
- Equivalent propositional formula

Hey!

- You said FO Inference is semi-decidable
- But you compiled it to SAT
  - Which is NP Complete
- So now we can always do the inference?!?
  - Tho it might take exponential time…
- Something seems wrong here….????

Restricted Forms of FO Logic

- Known, Finite Universes
  - Compile to SAT
- Frame Systems
  - Ban certain types of expressions
- Horn Clauses
  - Aka Prolog
- Function-Free Horn Clauses
  - Aka Datalog