CSEP 573

Logic, Reasoning, and Uncertainty

What's on our menu today?

Propositional Logic

- Resolution
- WalkSAT

Reasoning with First-Order Logic

- Unification
- Forward/Backward Chaining
- Resolution
- Wumpus again

Uncertainty

- Bayesian networks

Recall from Last Time: Inference/Proof Techniques

Two kinds (roughly):

Successive application of inference rules

- Generate new sentences from old in a sound way
- Proof = a sequence of inference rule applications
- Use inference rules as *successor function* in a standard search algorithm
- E.g., Resolution

Model checking

- Done by checking satisfiability: the SAT problem
- Recursive depth-first enumeration of models using heuristics: DPLL algorithm (sec. 7.6.1 in text)
- Local search algorithms (sound but incomplete)

  e.g., randomized hill-climbing (WalkSAT)

Understanding **Resolution**

IDEA: To show KB $\models \alpha$, use proof by contradiction,

i.e., show *KB* $\wedge \neg \alpha$ unsatisfiable

KB is in *Conjunctive Normal Form (CNF):*

KB is conjunction of clauses

E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Literals

Clause

Generating new clauses

General Resolution inference rule (for CNF):

$$\frac{\ell_1 \vee \ldots \vee \ell_k \qquad\qquad m_1 \vee \ldots \vee m_n}{\ell_1 \vee \ldots \vee \ell_{i-1} \vee \ell_{i+1} \vee \ldots \vee \ell_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \ldots \vee m_n}$$

where ℓ_i and m_j are complementary literals ($\ell_i = \neg m_j$)

E.g., $$\frac{P_{1,3} \vee P_{2,2} \qquad\qquad \neg P_{2,2}}{P_{1,3}}$$

Why this is sound

Proof of soundness of resolution inference rule:

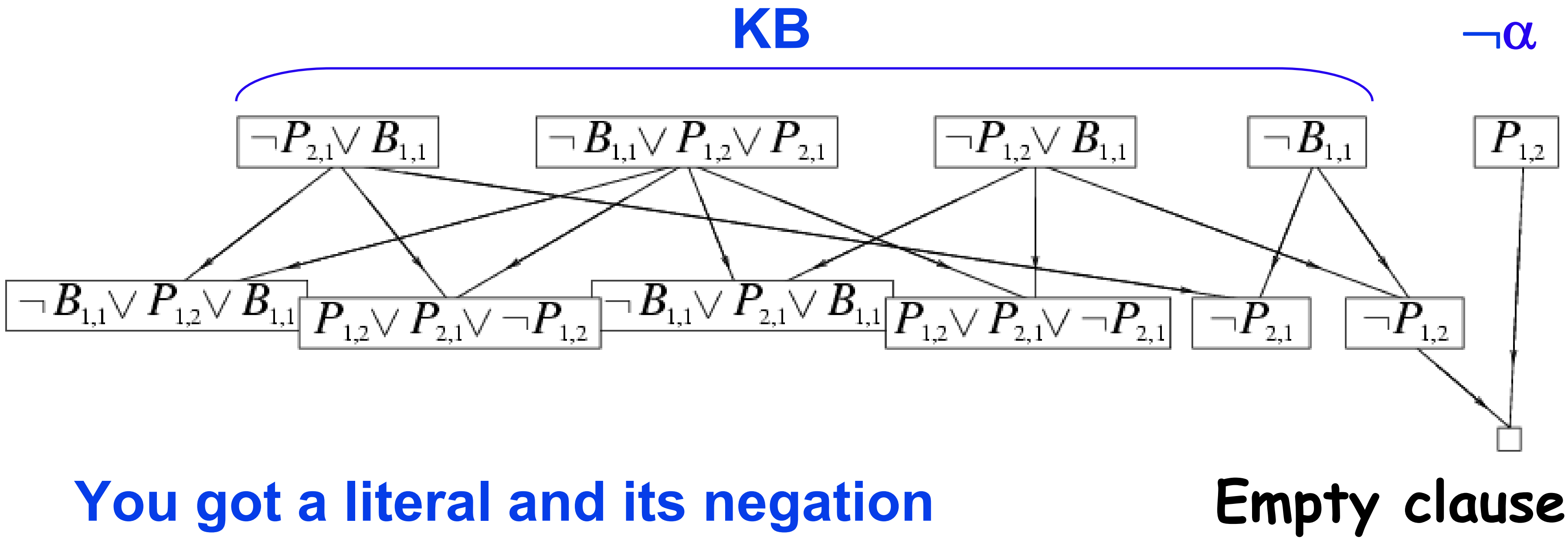
$\neg(\ell_1 \vee \ldots \vee \ell_{i-1} \vee \ell_{i+1} \vee \ldots \vee \ell_k) \Rightarrow \ell_i$

$\neg m_j \Rightarrow (m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)$

$\neg(\ell_i \vee \ldots \vee \ell_{i-1} \vee \ell_{i+1} \vee \ldots \vee \ell_k) \Rightarrow (m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n)$

(since $\ell_i = \neg m_j$)

Resolution example



You got a literal and its negation

What does this mean?

Recall that KB is a *conjunction* of all these clauses

Is $P_{1,2} \wedge \neg P_{1,2}$ satisfiable? No!

Therefore, KB $\wedge \neg \alpha$ is unsatisfiable, i.e., KB $\models \alpha$

Back to Inference/Proof Techniques

Two kinds (roughly):

Successive application of inference rules

- Generate new sentences from old in a sound way
- Proof = a sequence of inference rule applications
- Use inference rules as *successor function* in a standard search algorithm
- E.g., Resolution

Model checking

- Done by checking satisfiability: the SAT problem
- Recursive depth-first enumeration of models using heuristics: DPLL algorithm (sec. 7.6.1 in text)
- Local search algorithms (sound but incomplete)
  e.g., randomized hill-climbing (WalkSAT)

Why Satisfiability?

Can't get
$\neg$satisfaction

Why Satisfiability?

Recall: $KB \models \alpha$ iff $KB \wedge \neg\alpha$ is unsatisfiable

Thus, algorithms for satisfiability can be used for inference by showing $KB \wedge \neg\alpha$ is unsatisfiable

BUT… showing a sentence is satisfiable (the SAT problem) is NP-complete!

Finding a fast algorithm for SAT automatically yields fast algorithms for hundreds of difficult (NP-complete) problems

Satisfiability Examples

E.g. 2-CNF sentences (2 literals per clause):

$(\neg A \lor \neg B) \land (A \lor B) \land (A \lor \neg B)$

Satisfiable?

Yes (e.g., A = true, B = false)

$(\neg A \lor \neg B) \land (A \lor B) \land (A \lor \neg B) \land (\neg A \lor B)$

Satisfiable?

No

The WalkSAT algorithm

Local hill climbing search algorithm

- Incomplete: may not always find a satisfying assignment even if one exists

Evaluation function?

= Number of satisfied clauses

WalkSAT tries to *maximize* this function

Balance between greediness and randomness

The WalkSAT algorithm

```
function WALKSAT(clauses, p, max-flips) returns a satisfying model or failure
    inputs: clauses, a set of clauses in propositional logic
            p, the probability of choosing to do a "random walk" move
            max-flips, number of flips allowed before giving up

    model ← a random assignment of true/false to the symbols in clauses
    for i = 1 to max-flips do
        if model satisfies clauses then return model
        clause ← a randomly selected clause from clauses that is false in model
        with probability p flip the value in model of a randomly selected symbol
            from clause
        else flip whichever symbol in clause maximizes the number of satisfied clauses
    return failure
```

Greed

Randomness

Hard Satisfiability Problems

Consider random 3-CNF sentences. e.g.,

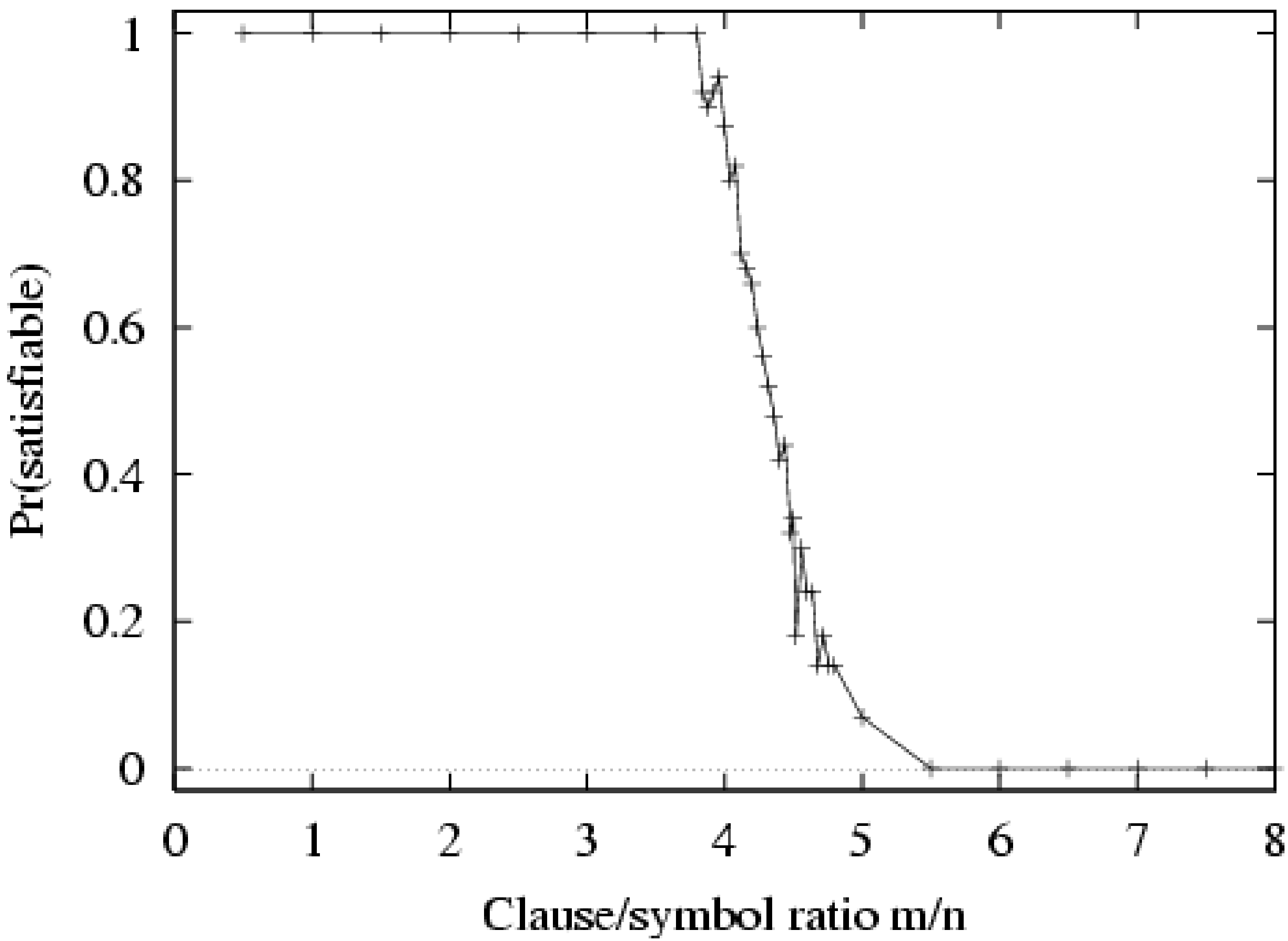
$(\neg D \lor \neg B \lor C) \land (B \lor \neg A \lor \neg C) \land (\neg C \lor \neg B \lor E) \land (E \lor \neg D \lor B) \land (B \lor E \lor \neg C)$

m = number of clauses

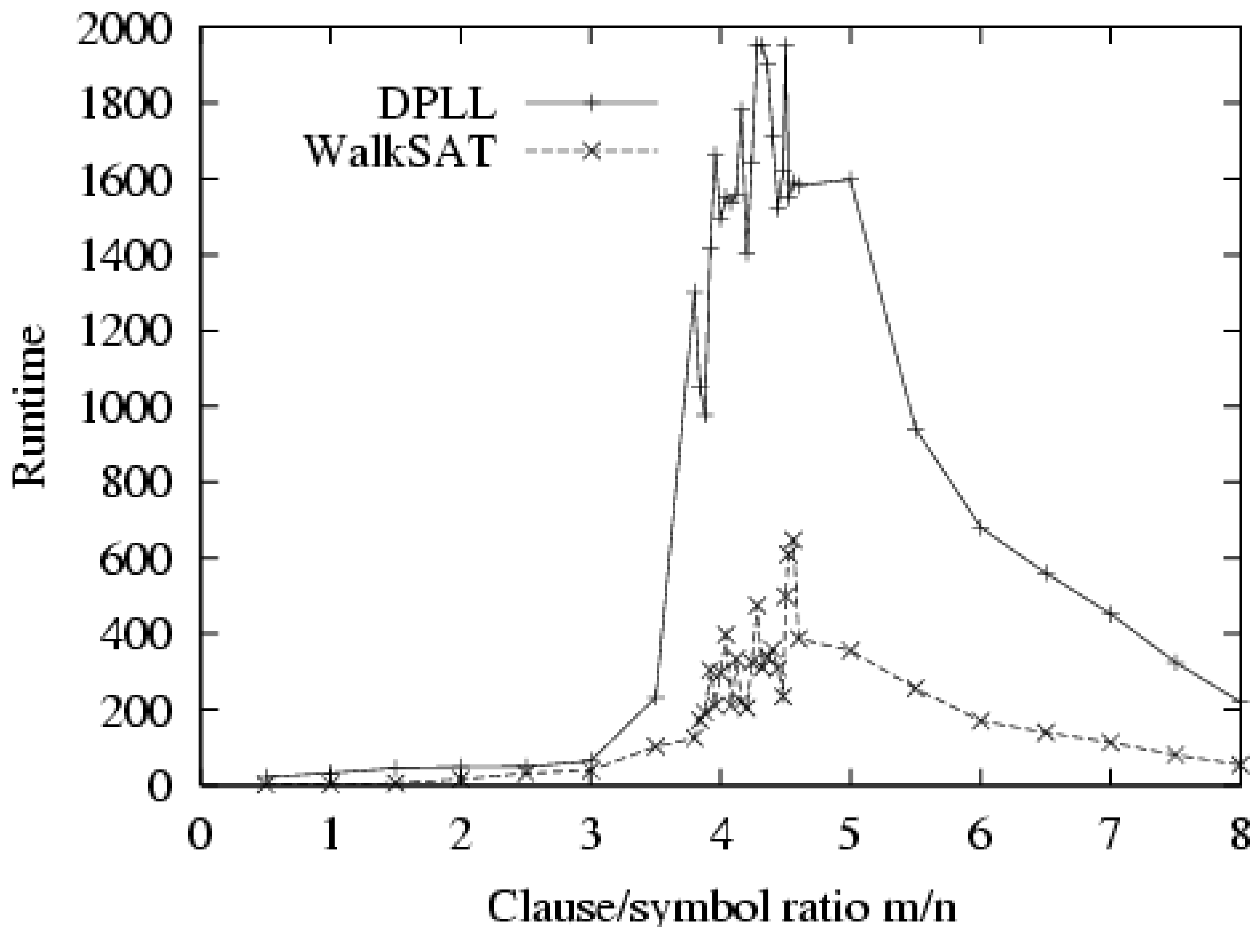
n = number of symbols

- Hard instances of SAT seem to cluster near m/n = 4.3 (critical point)

Hard Satisfiability Problems

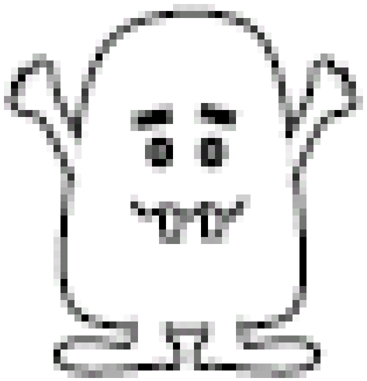


Hard Satisfiability Problems



Median runtime for random 3-CNF sentences, *n* = 50

What about me?



Putting it all together: Logical Wumpus Agents

A wumpus-world agent using propositional logic:

$\neg P_{1,1}$

$\neg W_{1,1}$

For x = 1, 2, 3, 4 and y = 1, 2, 3, 4, add (with appropriate boundary conditions):

$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$

$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$

$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$ At least 1 wumpus

$\neg W_{1,1} \vee \neg W_{1,2}$

$\neg W_{1,1} \vee \neg W_{1,3}$

…

At most 1 wumpus

$\Rightarrow$ 64 distinct proposition symbols, 155 sentences!

Limitations of propositional logic

KB contains "physics" sentences for every single square

For every time step t and every location $[x,y]$, we need to add to the KB:

$$L^{t}_{x,y} \wedge FacingRight^{\,t} \wedge Forward^{\,t} \Rightarrow L^{t+1}_{x+1,y}$$

Rapid proliferation of sentences!

What we'd like is a way to talk about *objects* and *groups* of objects, and to define relationships between them

Enter…First-Order Logic

(aka "Predicate logic")

Propositional vs. First-Order

Propositional logic

Facts: p, q, $\neg r$, $\neg P_{1,1}$, $\neg W_{1,1}$ etc.

$(p \wedge q) \vee (\neg r \vee q \wedge p)$

First-order logic

Objects: George, Monkey2, Raj, 573Student1, etc.

Relations:

Curious(George), Curious(573Student1), …

Smarter(573Student1, Monkey2)

Smarter(Monkey2, Raj)

Stooges(Larry, Moe, Curly)

PokesInTheEyes(Moe, Curly)

PokesInTheEyes(573Student1, Raj)

FOL Definitions

Constants: George, Monkey2, etc.
- Name a specific object.

Variables: X, Y.
- Refer to an object without naming it.

Functions: banana-of, grade-of, etc.
- Mapping from objects to objects.

Terms: banana-of(George), grade-of(stdnt1)
- Logical expressions referring to objects

Relations (predicates): Curious, PokesInTheEyes, etc.
- Properties of/relationships between objects.

More Definitions

Logical connectives: **and, or, not, $\Rightarrow$, $\Leftrightarrow$**

Quantifiers:

- $\forall$ **For all** **(Universal quantifier)**
- $\exists$ **There exists** **(Existential quantifier)**

Examples

- **George is a monkey and he is curious**

  Monkey(George) $\wedge$ Curious(George)
- **All monkeys are curious**

  $\forall$**m: Monkey(m) $\Rightarrow$ Curious(m)**
- **There is a curious monkey**

  $\exists$**m: Monkey(m) $\wedge$ Curious(m)**

Quantifier / Connective Interaction

$M(x)$ == "x is a monkey"
$C(x)$ == "x is curious"

$\forall x$: $M(x) \wedge C(x)$

"Everything is a curious monkey"

$\forall x$: $M(x) \Rightarrow C(x)$

"All monkeys are curious"

$\exists x$: $M(x) \wedge C(x)$

"There exists a curious monkey"

$\exists x$: $M(x) \Rightarrow C(x)$

"There exists an object that is *either* a curious monkey, *or* not a monkey at all"

Nested Quantifiers: Order matters!

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

Examples

Every monkey has a tail

$$\forall m \exists t \text{ has}(m,t)$$

Every monkey ***shares*** a tail!

$$\exists t \forall m \text{ has}(m,t)$$

Try:

Everybody loves somebody *vs.* Someone is loved by everyone

$$\forall x \exists y \text{ loves}(x, y) \qquad \exists y \forall x \text{ loves}(x, y)$$

Semantics

Semantics = what the arrangement of symbols *means* in the world

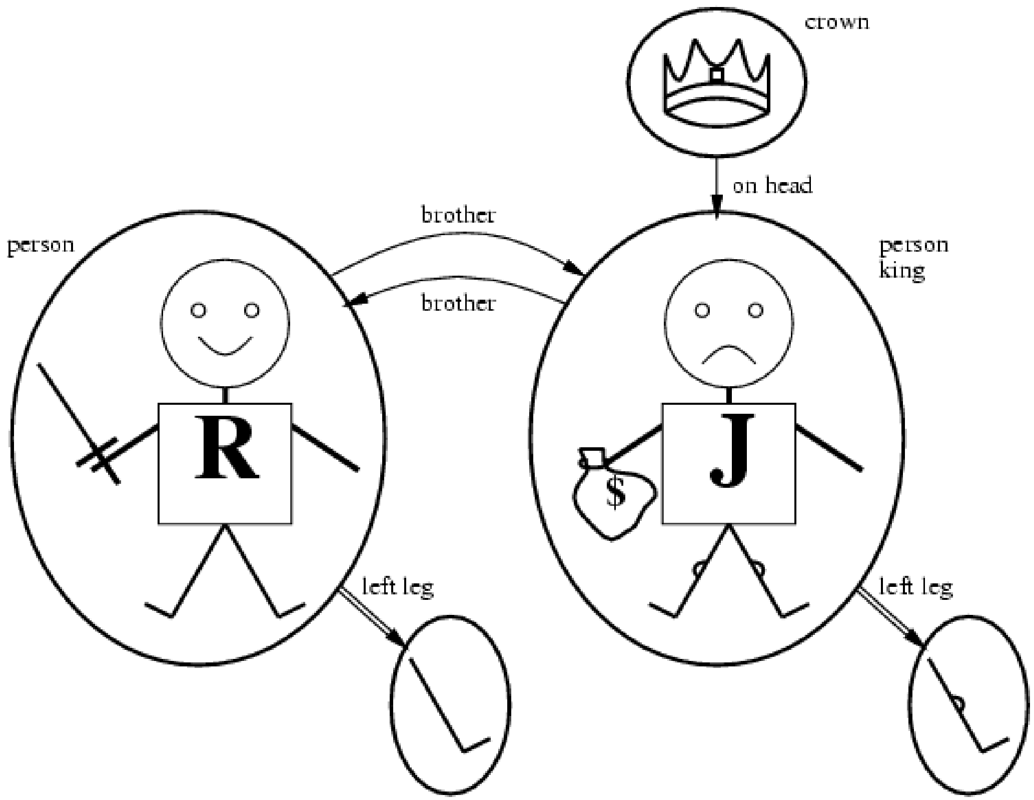
Propositional logic

- Basic elements are *variables*
  (refer to facts about the world)
- Possible worlds: mappings from variables to T/F

First-order logic

- Basic elements are *terms*
  (logical expressions that refer to objects)
- Interpretations: mappings from terms to real-world elements.

Example: A World of Kings and Legs



Syntactic elements:

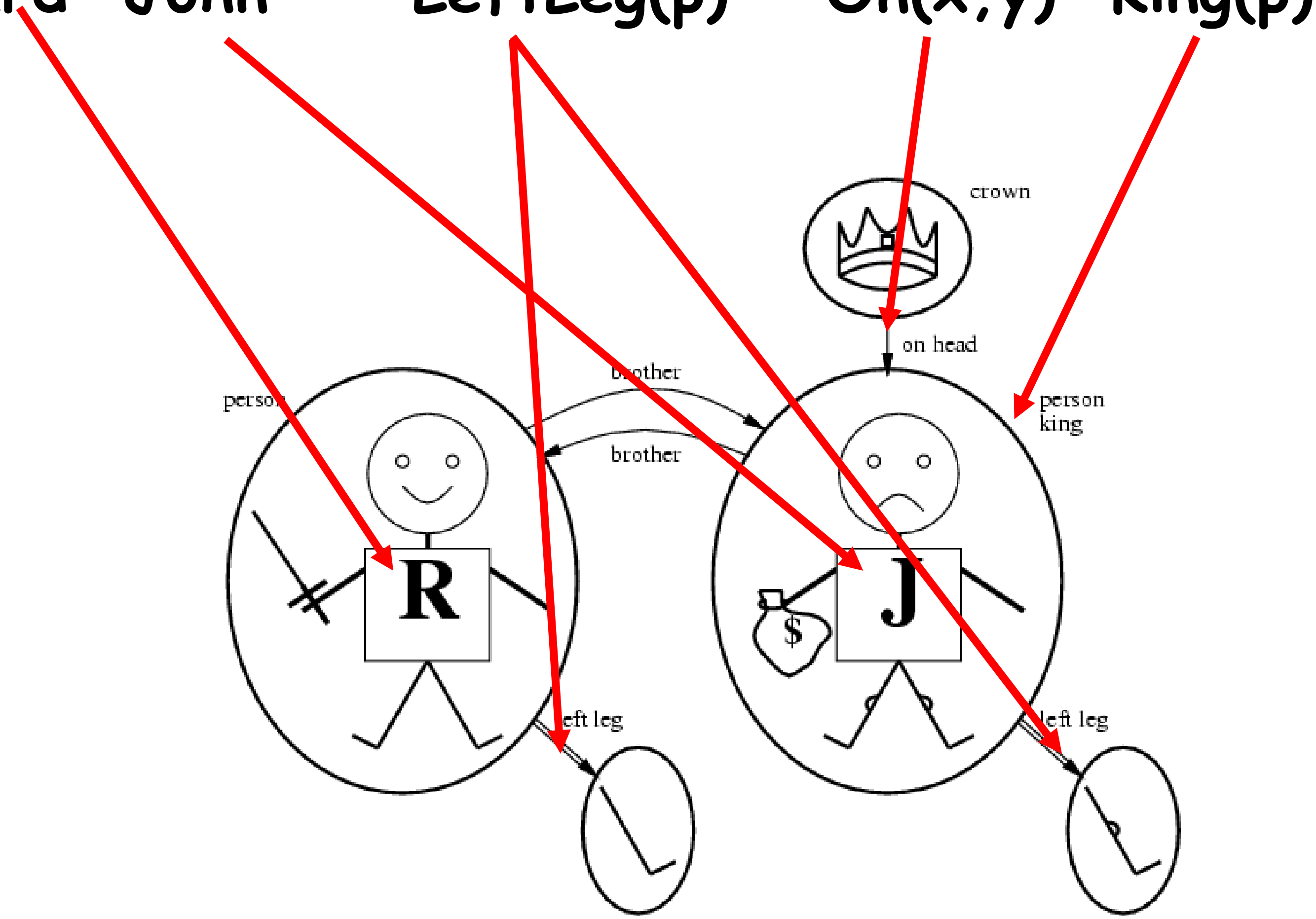
Constants:	Functions:	Relations:
Richard John	LeftLeg(p)	On(x,y) King(p)

Interpretation I

Interpretations map syntactic tokens to model elements

- Constants: Functions: Relations:
-

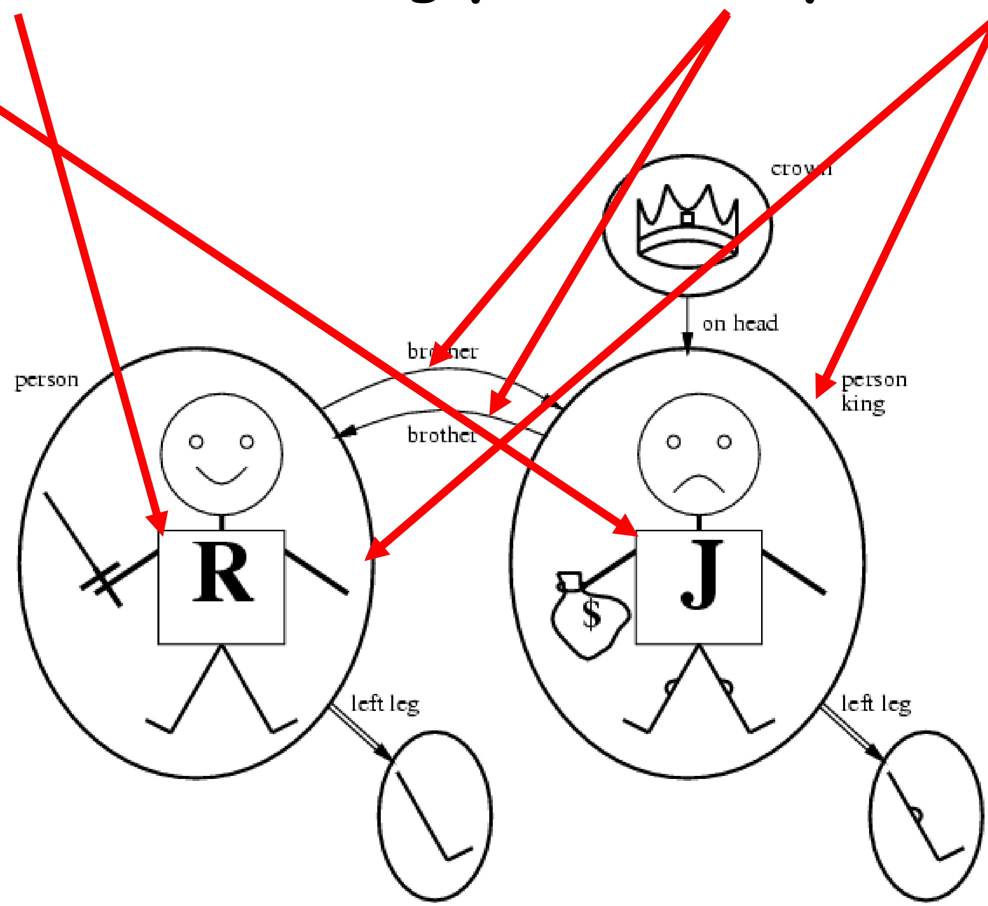
Richard John LeftLeg(p) On(x,y) King(p)



Interpretation II

- **Constants:**   **Functions:**   **Relations:**

Richard John LeftLeg(p) On(x,y) King(p)



How Many Interpretations?

Two constants (and 5 objects in world)

- Richard, John (R, J, crown, RL, JL)

$5^2 = 25$ object mappings

One unary relation

King(x)

Infinite number of values for x → infinite mappings

Even if we restricted x to: R, J, crown, RL, JL:

$2^5 = 32$ unary truth mappings

Two binary relations

- Leg(x, y); On(x, y)

Infinite. But even restricting x, y to five objects still yields 2^{25} mappings *for each* binary relation

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 in all interps

S1 $\models$ S2 (S1 entails S2) if

For all interps where S1 is true,

S2 is also true

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, WalkSAT Fast in practice	Unification Forward, Backward chaining Prolog, theorem proving
Complexity	NP-Complete	Semi-decidable

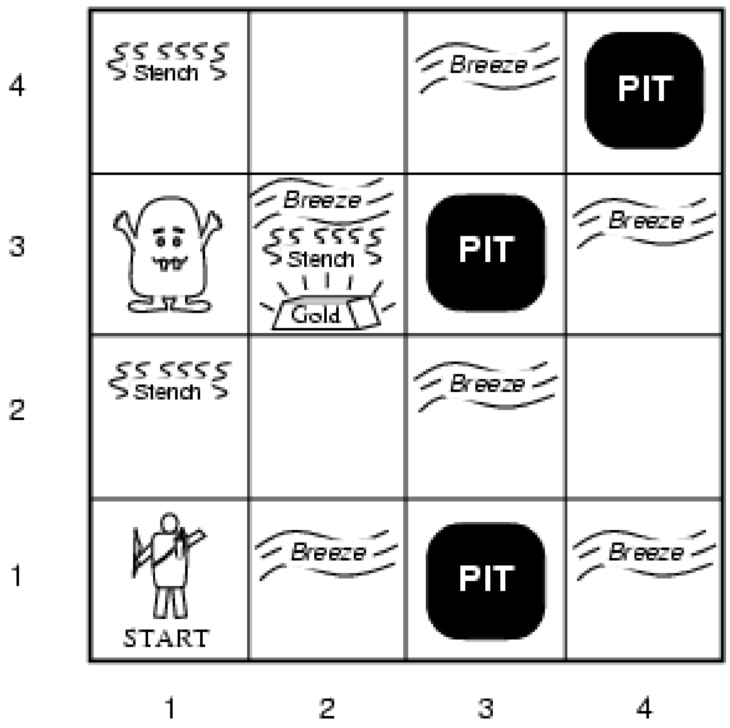
First-Order Wumpus World

Objects

- Squares, wumpuses, agents,
- gold, pits, stinkiness, breezes

Relations

- Square topology (adjacency),
- Pits/breezes,
- Wumpus/stinkiness



Wumpus World: Squares

- Each square as an object:

  $Square_{1,1}$, $Square_{1,2}$, …,
  $Square_{3,4}$, $Square_{4,4}$

- Square topology relations?

  $Adjacent(Square_{1,1}, Square_{2,1})$
  …
  $Adjacent(Square_{3,4}, Square_{4,4})$

Better: **Squares as lists:**

[1, 1], [1,2], …, [4, 4]

Square topology relations:

$\forall x, y, a, b$: Adjacent([x, y], [a, b]) ⇔
[a, b] ∈ {[x+1, y], [x-1, y], [x, y+1], [x, y-1]}

Wumpus World: Pits

- Each pit as an object:

  $Pit_{1,1}$, $Pit_{1,2}$, …,
  $Pit_{3,4}$, $Pit_{4,4}$
- Problem?

  Not all squares have pits

List only the pits we have?

$Pit_{3,1}$, $Pit_{3,3}$, $Pit_{4,4}$

Problem?

No reason to distinguish pits (same properties)

Better: **pit as unary predicate**

Pit(x)

Pit([3,1]); Pit([3,3]); Pit([4,4]) will be true

Wumpus World: Breezes

- Represent breezes like pits, as unary predicates:
  Breezy(x)

"Squares next to pits are breezy":

$\forall$**x, y, a, b:**

Pit([x, y]) $\wedge$ **Adjacent([x, y], [a, b])** $\Rightarrow$ **Breezy([a, b])**

Wumpus World: Wumpuses

- Wumpus as object:
  Wumpus

- Wumpus home as unary predicate:
  WumpusIn(x)

Better: **Wumpus's home as a function:**

Home(Wumpus) references the wumpus's home square.

FOL Reasoning: Outline

Basics of FOL reasoning

Classes of FOL reasoning methods

- Forward & Backward Chaining
- Resolution
- Compilation to SAT

Basics: Universal Instantiation

Universally quantified sentence:

- $\forall x$: Monkey(x) $\Rightarrow$ Curious(x)

Intutively, x can be anything:

- Monkey(George) $\Rightarrow$ Curious(George)
- Monkey(473Student1) $\Rightarrow$ Curious(473Student1)
- Monkey(DadOf(George)) $\Rightarrow$ Curious(DadOf(George))

Formally: (example)

$$\frac{\forall x \;\; S}{\text{Subst}(\{x/p\},\, S)} \qquad \frac{\forall x \;\; \text{Monkey}(x) \rightarrow \text{Curious}(x)}{\text{Monkey}(\text{George}) \rightarrow \text{Curious}(\text{George})}$$

x is replaced with p in S, and the quantifier removed

x is replaced with George in S, and the quantifier removed

Basics: Existential Instantiation

Existentially quantified sentence:

- $\exists x$: Monkey(x) $\wedge$ ¬Curious(x)

Intutively, x must name something. But what?

- Monkey(George) $\wedge$ ¬Curious(George) ???
- No! S might not be true for George!

Use a *Skolem Constant* :

- Monkey(K) $\wedge$ ¬Curious(K)

…where K is a completely new **symbol** (stands for the monkey for which the statement is true)

Formally:

$$\frac{\exists x \;\; S}{\text{Subst}(\{x/K\}, S)}$$

K is called a Skolem constant

Basics: Generalized Skolemization

What if our existential variable is nested?

- $\forall x\ \exists y$: Monkey(x) $\Rightarrow$ HasTail(x, y)

- $\forall x$: Monkey(x) $\Rightarrow$ HasTail(x, K_Tail) ???

Existential variables can be replaced by Skolem functions

- Args to function are all surrounding $\forall$ vars

$\forall x$: Monkey(x) $\Rightarrow$ HasTail(x, f(x))

"tail-of" function

Motivation for Unification

What if we want to use modus ponens?

Propositional Logic:

$$\frac{a \wedge b, \qquad a \wedge b \Rightarrow c}{c}$$

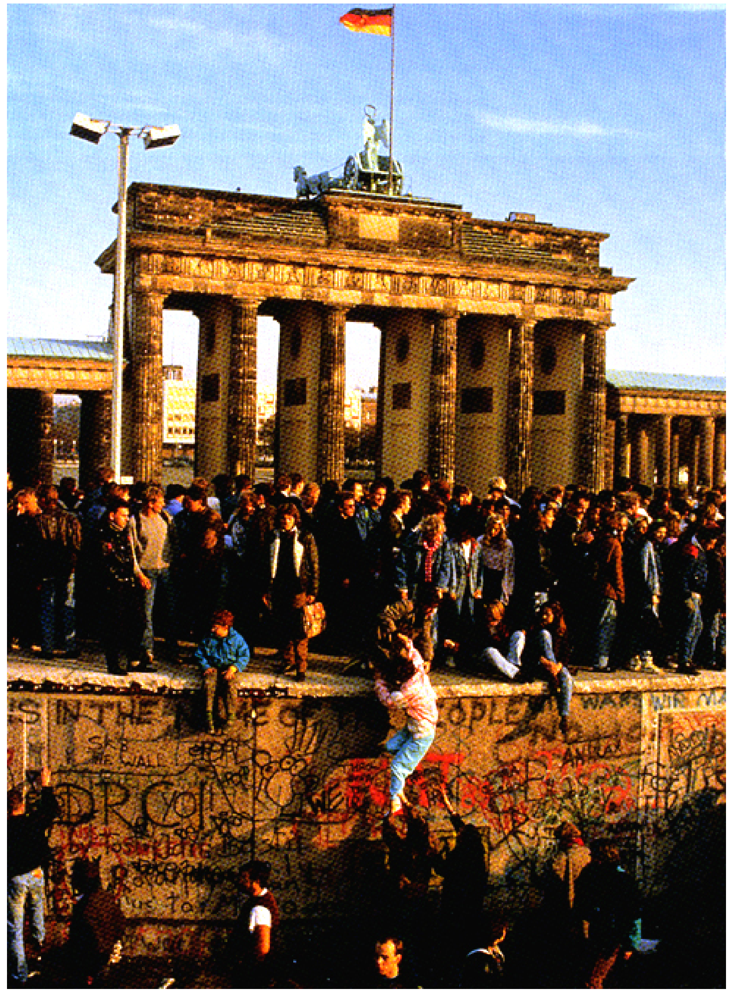
In First-Order Logic?

$$\frac{\begin{array}{l} Monkey(x) \Rightarrow Curious(x) \\ Monkey(George) \end{array}}{????}$$

Must "*unify*" x with George:

- Need to substitute {x/George} in Monkey(x) $\Rightarrow$ Curious(x) to infer Curious(George)

What is Unification?



CITY PRICES

London Herald

LATE EDITION

SATURDAY 11th NOVEMBER 1989

BERLIN WALL TUMBLES

'Beginning of the End' for Communism

Germany Re-unified

Breaching the Wall

OTHER NEWS

Not this kind of unification...

What is Unification?

Match up expressions by *finding variable values that make the expressions identical*

Unify(x, y) returns most general unifier (MGU).

MGU places fewest restrictions on values of variables

Examples:

- **Unify(city(x), city(seattle)) returns {x/seattle}**
- **Unify(PokesInTheEyes(Moe,x), PokesInTheEyes(y,z)) returns {y/Moe,z/x}**
  - **{y/Moe,x/Moe,z/Moe} possible but not MGU**

Unification and Substitution

- Unification produces a mapping from variables to values (e.g., {x/kent,y/seattle})
- Substitution: Subst(mapping,sentence) returns new sentence with variables replaced by values
  - Subst({x/kent,y/seattle}),connected(x, y)),
    returns connected(kent, seattle)

Unification Examples I

Unify(road(x, kent), road(seattle, y))

- Returns {x / seattle, y / kent}
- When substituted in both expressions, the resulting expressions match:
- Each is (road(seattle, kent))

Unify(road(x, x), road(seattle, kent))

- Not possible – Fails!
- x can't be seattle and kent at the same time!

Unification Examples II

Unify(f(g(x, dog), y)), f(g(cat, y), dog)

- {x / cat, y / dog}

Unify(f(g(x)), f(x))

- Fails: no substitution makes them identical.
- E.g. {x / g(x) } yields f(g(g(x))) and f(g(x)) which are not identical!

Unification Examples III

Unify(f(g(cat, y), y), f(x, dog))

- {x / g(cat, dog), y / dog}

Unify(f(g(y)), f(x))

- {x / g(y)}

Back to curious monkeys:

$$\frac{\begin{array}{l}\text{Monkey}(x) \rightarrow \text{Curious}(x)\\ \text{Monkey}(\text{George})\end{array}}{\text{Curious}(\text{George})}$$

Unify and then use modus ponens =

generalized modus ponens

("Lifted" version of modus ponens)

Inference I: Forward Chaining

The algorithm:

- Start with the KB
- Add any fact you can generate with GMP (i.e., unify expressions and use modus ponens)
- Repeat until: goal reached or generation halts.

Inference II: Backward Chaining

The algorithm:

- Start with KB and goal.
- Find all rules whose *results* unify with goal:
  - Add the *premises* of these rules to the goal list
  - Remove the corresponding result from the goal list
- Stop when:
  - Goal list is empty (SUCCEED) or
  - Progress halts (FAIL)

Inference III: Resolution
[Robinson 1965]

$\{ (p \vee q), (\neg p \vee r \vee s) \} \longrightarrow (q \vee r \vee s)$

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

First-Order Resolution
[Robinson 1965]

$$\{\ (p(x) \lor q(A),\quad (\neg\ p(B) \lor r(x) \lor s(y))\ \}$$

$$(q(A) \lor r(B) \lor s(y))$$

Substitute MGU $\{x/B\}$ in all literals

- Literal in one clause
- Negation of *something which unifies* in other
- Result is disjunction of all other literals with substitution based on MGU

Inference using First-Order Resolution

As before, use "proof by contradiction"

To show $KB \models \alpha$, show $KB \wedge \neg\alpha$ unsatisfiable

Method

- Let $S = KB \wedge \neg goal$
- Convert S to clausal form
  - Standardize apart variables (change names if needed)
  - Move quantifiers to front, skolemize to remove $\exists$
  - Replace $\Rightarrow$ with $\vee$ and $\neg$
  - DeMorgan's laws to get CNF (ands-of-ors)
- Resolve clauses in S until empty clause (unsatisfiable) or no new clauses added

First-Order Resolution Example

Given

- $\forall x$ man(x) $\Rightarrow$ human(x)
- $\forall x$ woman(x) $\Rightarrow$ human(x)
- $\forall x$ singer(x) $\Rightarrow$ man(x) $\vee$ woman(x)
- singer(M)

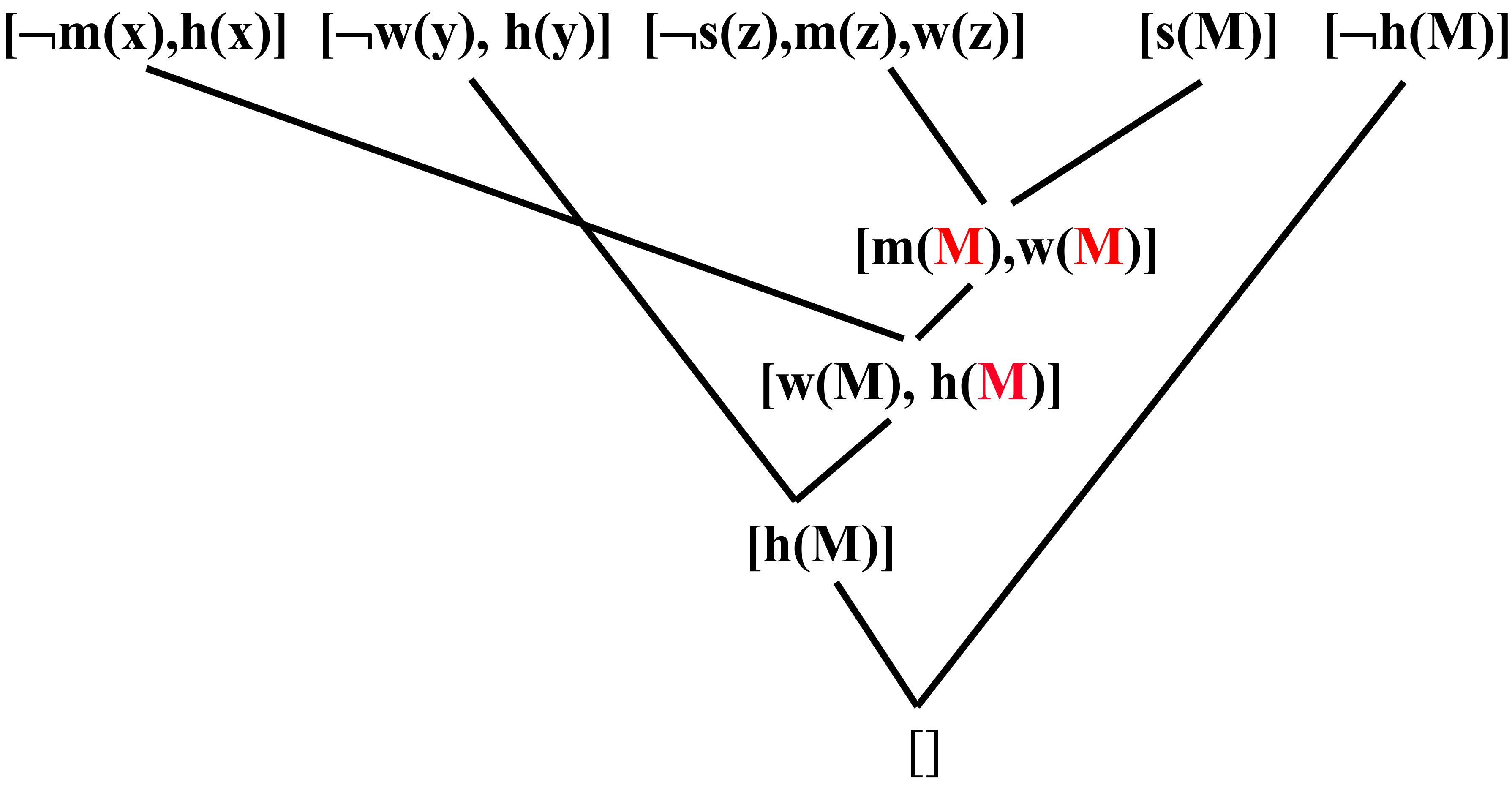
Prove

- human(M)

CNF representation (list of clauses):

[¬m(x),h(x)] [¬w(y), h(y)] [¬s(z),m(z),w(z)] [s(M)] [¬h(M)]

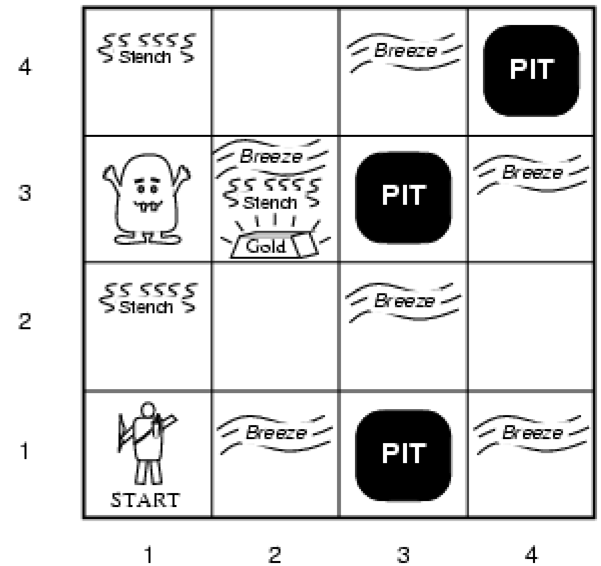
FOL Resolution Example



Back To the Wumpus World

Recall description:

- Squares as lists: [1,1] [3,4] etc.
- Square adjacency as binary predicate.
- Pits, breezes, stenches as unary predicates:
  Pit(x)
- Wumpus, gold, homes as functions:
  Home(Wumpus)



Back To the Wumpus World

"Squares next to pits are breezy":

$\forall x, y, a, b:$
$Pit([x, y]) \wedge Adjacent([x, y], [a, b]) \Rightarrow Breezy([a, b])$

"Breezes happen *only* and *always* next to pits":

- $\forall a,b\ Breezy([a, b]) \Leftrightarrow$
  $\exists\ x,y\ Pit([x, y]) \wedge Adjacent([x, y], [a, b])$

That's nice but these algorithms assume complete knowledge of the world!

Hard to achieve in most cases

Enter…

Uncertainty

Example: Catching a flight

Suppose you have a flight at 6pm

When should you leave for SEATAC?

- What are the traffic conditions?
- How crowded is security?

Leaving time before 6pm	P(arrive-in-time)
20 min	0.05
30 min	0.25
45 min	0.50
60 min	0.75
120 min	0.98
1 day	0.99999

Probability Theory: Beliefs about events

Utility theory: Representation of preferences

Decision about when to leave depends on both:

Decision Theory = Probability + Utility Theory

What Is Probability?

Probability: **Calculus for dealing with nondeterminism and uncertainty**

Probabilistic model: **Says how often we expect different things to occur**

Where do the numbers for probabilities come from?

- **Frequentist view** (numbers from experiments)
- **Objectivist view** (numbers inherent properties of universe)
- **Subjectivist view** (numbers denote agent's beliefs)

Why Should You Care?

The world is full of uncertainty

- Logic is not enough
- Computers need to be able to handle uncertainty

Probability: new foundation for AI (& CS!)

Massive amounts of data around today

- Statistics and CS are both about data
- Statistics lets us summarize and understand it
- Statistics is the basis for most learning

Statistics lets data do our work for us

Logic vs. Probability

Symbol: Q, R …	Random variable: Q …
Boolean values: T, F	Values/Domain: you specify e.g. {heads, tails}, [1,6]
State of the world: Assignment of T/F to all Q, R … Z	Atomic event: a complete assignment of values to Q… Z • Mutually exclusive • Exhaustive
	Prior probability aka Unconditional prob: P(Q)
	Joint distribution: Prob. of every atomic event

Types of Random Variables

Propositional or Boolean random variables

e.g., $Cavity$ (do I have a cavity?)

Discrete random variables (*finite* or *infinite*)

e.g., $Weather$ is one of $\langle sunny, rain, cloudy, snow \rangle$

$Weather = rain$ is a proposition

Values must be exhaustive and mutually exclusive

Continuous random variables (*bounded* or *unbounded*)

e.g., $Temp = 21.6$; also allow, e.g., $Temp < 22.0$.

Arbitrary Boolean combinations of basic propositions

Axioms of Probability Theory

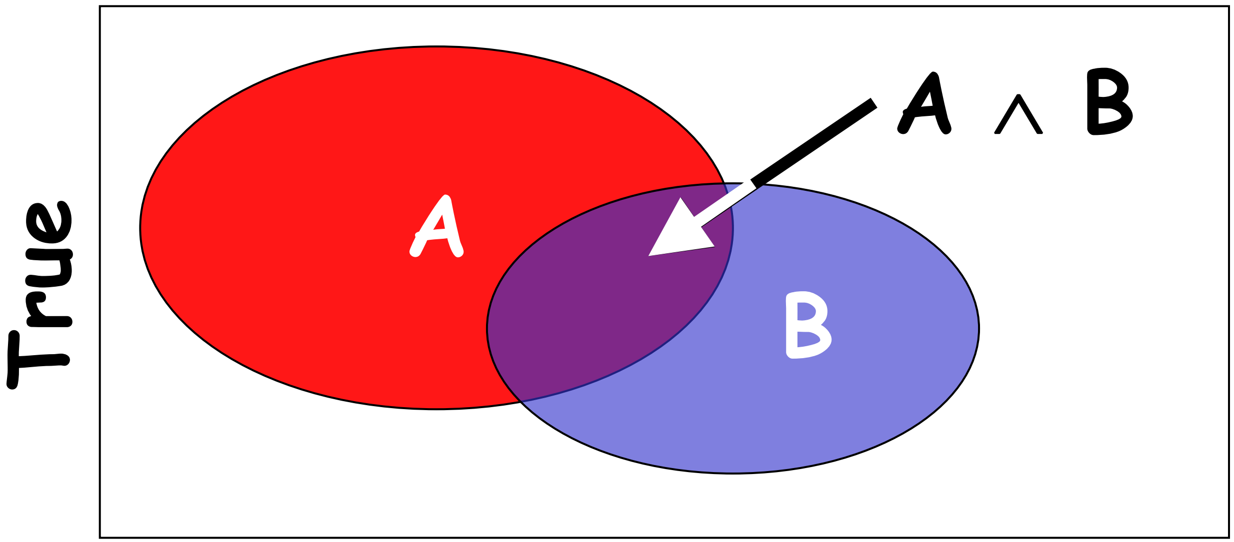
Just 3 are enough to build entire theory!

1. All probabilities between 0 and 1

   $0 \leq P(A) \leq 1$

2. P(true) = 1 and P(false) = 0
3. Probability of disjunction of events is:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$



Prior and Joint Probability

Prior or unconditional probabilities of propositions

e.g., $P(Cavity = true) = 0.2$ and $P(Weather = sunny) = 0.72$

correspond to belief prior to arrival of any (new) evidence

Probability distribution gives values for all possible assignments:

$\mathbf{P}(Weather) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$ (*normalized*, i.e., sums to 1)

sunny, rain, cloudy, snow

Joint probability distribution for a set of r.v.s gives the probability of every atomic event on those r.v.s

$\mathbf{P}(Weather, Cavity) =$ a 4×2 matrix of values:

$Weather =$	$sunny$	$rain$	$cloudy$	$snow$
$Cavity = true$	0.144	0.02	0.016	0.02
$Cavity = false$	0.576	0.08	0.064	0.08

We will see later how any question can be answered by the joint distribution

Conditional Probability

Conditional probabilities

e.g., P(*Cavity = true* | *Toothache = true*) = probability of *cavity* given *toothache*

Notation for conditional distributions:

P(*Cavity* | *Toothache*) = 2-element vector of 2-element vectors (2 P values when Toothache is true and 2 P values when false)

If we know more, e.g., *cavity* is also given (i.e. *Cavity = true*), then we have

P(*cavity* | *toothache, cavity*) = 1

New evidence may be irrelevant, allowing simplification:

P(*cavity* | *toothache, sunny*) = P(*cavity* | *toothache*) = 0.8

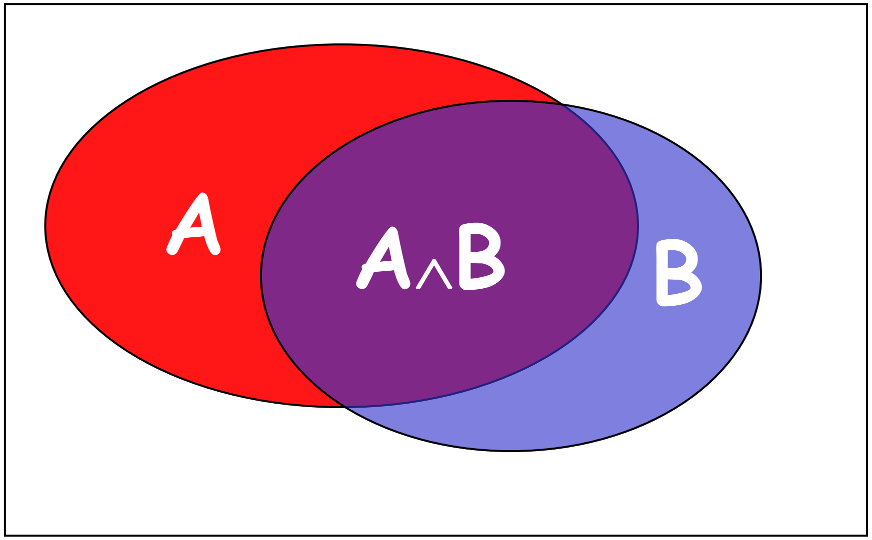
Conditional Probability

P(*A* | *B*) is the probability of *A* given *B*

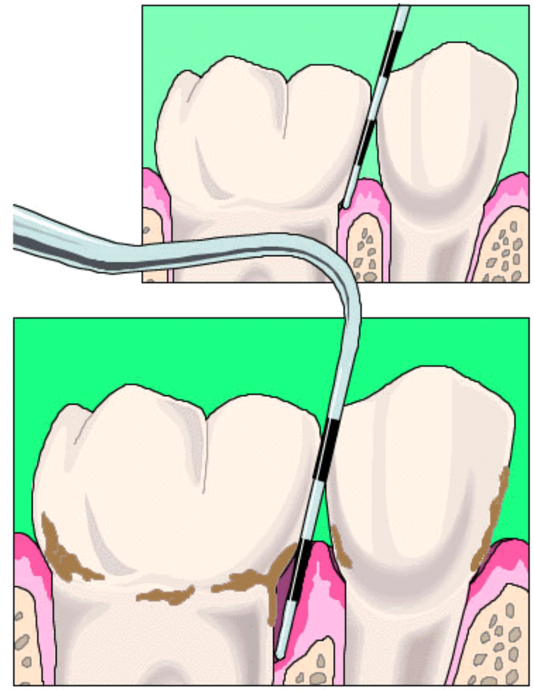
Assumes that *B* is the only info known.

Defined as:

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(A \wedge B)}{P(B)}$$



Dilemma at the Dentist's



What is the probability of a cavity given a toothache?
What is the probability of a cavity given the probe catches?

Probabilistic Inference by Enumeration

Start with the joint distribution:

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	.108	.012	.072	.008
¬ *cavity*	.016	.064	.144	.576

For any proposition ϕ, sum the atomic events where it is true:

$$P(\phi) = \Sigma_{\omega:\omega \models \phi} P(\omega)$$

P(toothache)= .108+.012+.016+.064
= .20 or 20%

Inference by Enumeration

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

P(toothache∨cavity) = ?

.2 + .108 + .012 + .072 + .008 - (.108+.012)

= .28

Inference by Enumeration

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

Can also compute conditional probabilities:

$$P(\neg cavity | toothache) = \frac{P(\neg cavity \wedge toothache)}{P(toothache)}$$

$$= \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4$$

Problems with Enumeration

Worst case time: $O(d^n)$

where d = max arity of random variables
e.g., d = 2 for Boolean (T/F)

and n = number of random variables

Space complexity also $O(d^n)$

- Size of joint distribution

Problem: Hard/impossible to estimate all $O(d^n)$ entries for large problems

Independence

A **and** ***B*** **are** ***independent*** **iff:**

$$P(A|B) = P(A)$$

$$P(B|A) = P(B)$$

These two constraints are logically equivalent

Therefore, if ***A*** **and** ***B*** **are independent:**

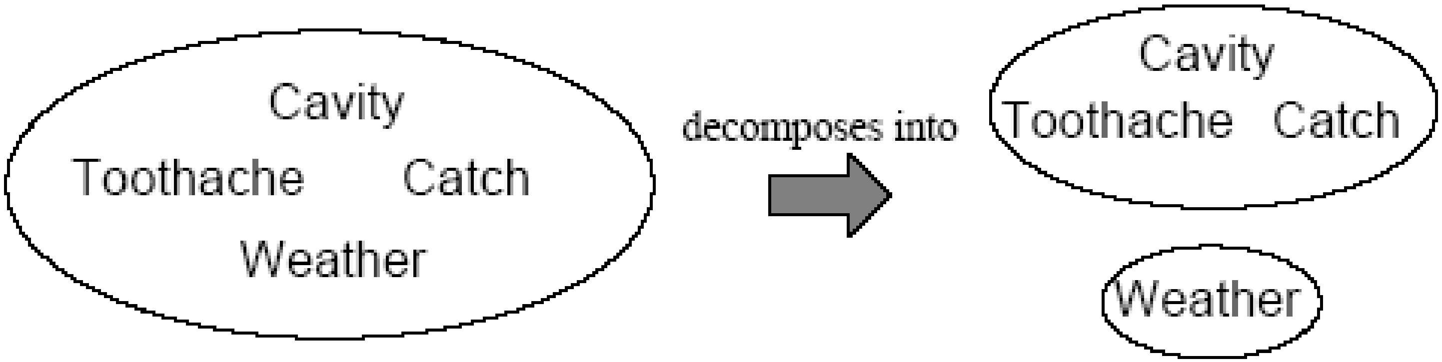
$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = P(A)$$

$$P(A \wedge B) = P(A)P(B)$$

Independence

A and B are independent iff

$$\mathbf{P}(A|B) = \mathbf{P}(A) \quad \text{or} \quad \mathbf{P}(B|A) = \mathbf{P}(B) \quad \text{or} \quad \mathbf{P}(A,B) = \mathbf{P}(A)\mathbf{P}(B)$$



$$\mathbf{P}(Toothache, Catch, Cavity, Weather)$$
$$= \mathbf{P}(Toothache, Catch, Cavity)\mathbf{P}(Weather)$$

2 values

4 values

32 entries reduced to 12; for n independent biased coins, $2^n \rightarrow n$

Complete independence is powerful but rare

What to do if it doesn't hold?

Conditional Independence

$\mathbf{P}(Toothache, Cavity, Catch)$ has $2^3 - 1 = 7$ independent entries

If I have a cavity, the probability that the probe catches in it doesn't depend on whether I have a toothache:

(1) $P(catch|toothache, cavity) = P(catch|cavity)$

The same independence holds if I haven't got a cavity:

(2) $P(catch|toothache, \neg cavity) = P(catch|\neg cavity)$

$Catch$ is *conditionally independent* of $Toothache$ given $Cavity$:

$\mathbf{P}(Catch|Toothache, Cavity) = \mathbf{P}(Catch|Cavity)$

Instead of 7 entries, only need 5 (why?)

Conditional Independence II

P(catch | toothache, cavity) = P(catch | cavity)
P(catch | toothache,¬cavity) = P(catch |¬cavity)

Equivalent statements:

$$\mathbf{P}(Toothache|Catch, Cavity) = \mathbf{P}(Toothache|Cavity)$$
$$\mathbf{P}(Toothache, Catch|Cavity) = \mathbf{P}(Toothache|Cavity)\mathbf{P}(Catch|Cavity)$$

Why only 5 entries in table?

Write out full joint distribution using chain rule:

$$\begin{aligned}
&\mathbf{P}(Toothache, Catch, Cavity) \\
&= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch, Cavity) \\
&= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity) \\
&= \mathbf{P}(Toothache|Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity)
\end{aligned}$$

I.e., $2 + 2 + 1 = 5$ independent numbers (equations 1 and 2 remove 2)

Power of Cond. Independence

Often, using conditional independence reduces the storage complexity of the joint distribution from exponential to linear!!

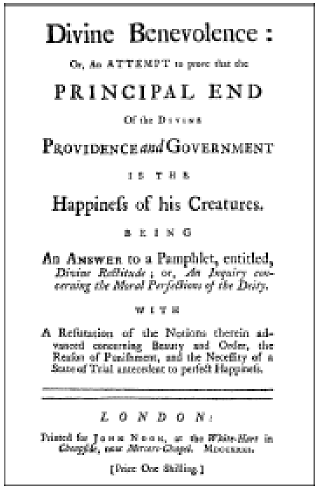
Conditional independence is the most basic & robust form of knowledge about uncertain environments.

Thomas Bayes

Reverand Thomas Bayes
Nonconformist minister
(1702-1761)

Publications:

- *Divine Benevolence, or an Attempt to Prove That the Principal End of the Divine Providence and Government is the Happiness of His Creatures* (1731)



Divine Benevolence:
Or, An ATTEMPT to prove that the
PRINCIPAL END
Of the DIVINE
PROVIDENCE and GOVERNMENT
IS THE
Happiness of his Creatures.
BEING
An ANSWER to a Pamphlet, entitled, *Divine Rectitude; or, An Inquiry concerning the Moral Perfections of the Deity.*
WITH
A Refutation of the Notions therein advanced concerning Beauty and Order, the Reason of Punishment, and the Necessity of a State of Trial antecedent to perfect Happiness.

LONDON:
Printed for JOHN NOON, at the *White-Hart* in *Cheapside*, near *Mercers-Chapel.* MDCCXXXI.
[Price One Shilling.]

- *An Introduction to the Doctrine of Fluxions* (1736)
- *An Essay Towards Solving a Problem in the Doctrine of Chances* (1764)

Recall: Conditional Probability

P(x | y) is the probability of x given y

Assumes that y is the only info known.

Defined as:

$$P(x \mid y) = \frac{P(x, y)}{P(y)}$$

$$P(y \mid x) = \frac{P(y, x)}{P(x)} = \frac{P(x, y)}{P(x)}$$

Therefore?

Bayes' Rule

$$P(x, y) = P(x \mid y)P(y) = P(y \mid x)P(x)$$

$$\Rightarrow$$

$$P(x \mid y) = \frac{P(y \mid x)\; P(x)}{P(y)}$$

What this useful for?

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

Bayes' rule is used to Compute *Diagnostic* Probability from *Causal* Probability

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

E.g. let M be meningitis, S be stiff neck

P(M) = 0.0001,
P(S) = 0.1,
P(S|M)= 0.8 (note: these can be estimated from patients)

P(M|S) = $\frac{P(s|m)P(m)}{P(s)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$

Note: posterior probability of meningitis still very small!

Normalization in Bayes' Rule

$$P(x|y) = \frac{P(y|x)\ P(x)}{P(y)} = \alpha\ P(y|x)\,P(x)$$

$$\alpha = \frac{1}{P(y)} = \frac{1}{\sum_x P(y,x)} = \frac{1}{\sum_x P(y|x)P(x)}$$

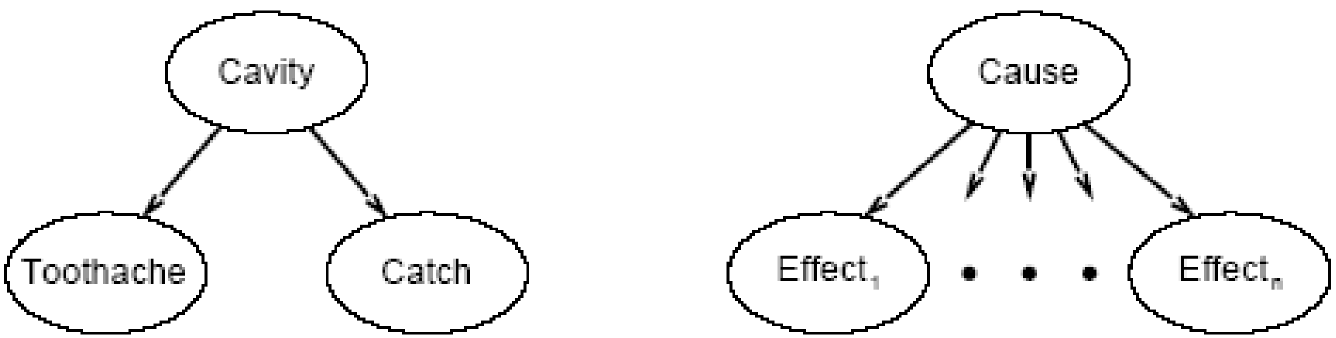
α is called the normalization constant

Cond. Independence and the Naïve Bayes Model

$$\mathbf{P}(Cavity|toothache \wedge catch)$$
$$= \alpha\, \mathbf{P}(toothache \wedge catch|Cavity)\mathbf{P}(Cavity)$$
$$= \alpha\, \mathbf{P}(toothache|Cavity)\mathbf{P}(catch|Cavity)\mathbf{P}(Cavity)$$

This is an example of a *naive Bayes* model:

$$\mathbf{P}(Cause, Effect_1, \ldots, Effect_n) = \mathbf{P}(Cause)\Pi_i \mathbf{P}(Effect_i|Cause)$$

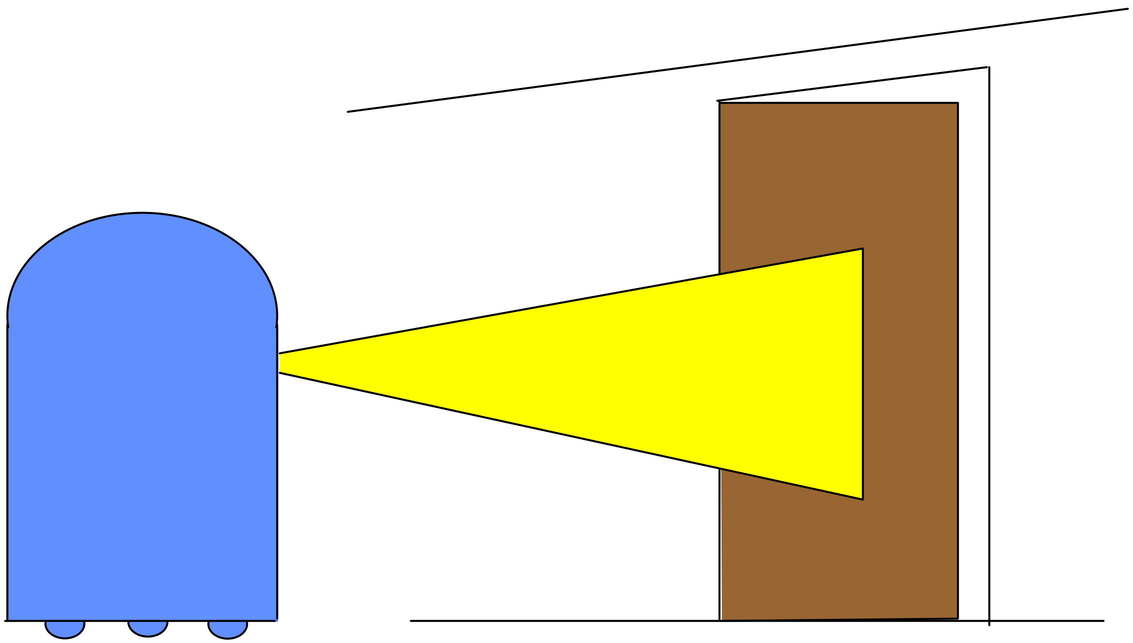


Total number of parameters is *linear* in n

Example 1: State Estimation

Suppose a robot obtains **measurement z**

What is *P(doorOpen|z)?*



Causal vs. Diagnostic Reasoning

P(open|z) is diagnostic.

P(z|open) is causal.

Often causal knowledge is easier to obtain.

Bayes rule allows us to use causal knowledge:

count frequencies!

$$P(open \mid z) = \frac{P(z \mid open)P(open)}{P(z)}$$

State Estimation Example

$P(z|open) = 0.6$ $P(z|\neg open) = 0.3$

$P(open) = P(\neg open) = 0.5$

$$P(open \mid z) = \frac{P(z \mid open)P(open)}{P(z \mid open)\,p(open) + P(z \mid \neg open)\,p(\neg open)}$$

$$P(open \mid z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

Measurement z raises the probability that the door is open from 0.5 to 0.67

Combining Evidence

Suppose our robot obtains another observation z_2.

How can we integrate this new information?

More generally, how can we estimate $P(x| z_1...z_n)$?

Recursive Bayesian Updating

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1})\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

Markov assumption: z_n is independent of $z_1, \ldots, z_{n-1}$ given x.

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1})\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

$$= \frac{P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

$$= \alpha\, P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})$$

Recursive!

Incorporating a Second Measurement

$P(z_2|open) = 0.5$ $P(z_2|\neg open) = 0.6$

$P(open|z_1)=2/3 = 0.67$

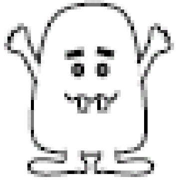
$$P(open \mid z_2, z_1) = \frac{P(z_2 \mid open)\, P(open \mid z_1)}{P(z_2 \mid open)\, P(open \mid z_1) + P(z_2 \mid \neg open)\, P(\neg open \mid z_1)}$$

$$= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

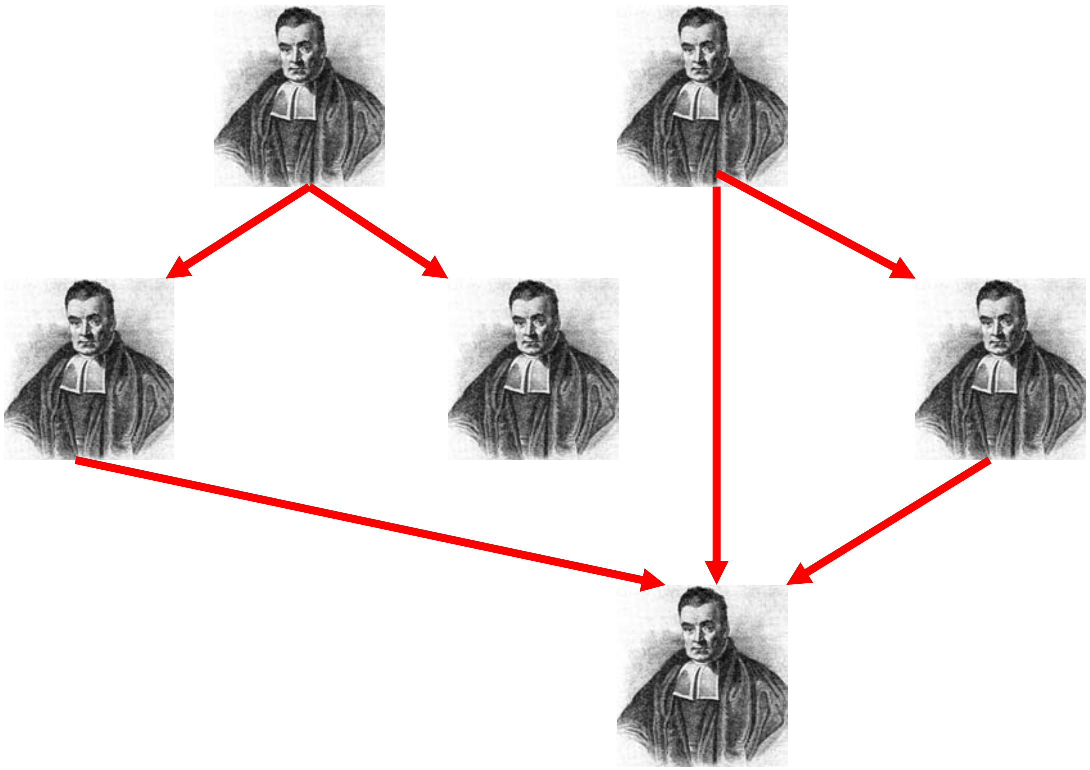
- z_2 lowers the probability that the door is open.

These calculations seem laborious to do for each problem domain – is there a general representation scheme for probabilistic inference?

Yes!



Enter…Bayesian networks



What are Bayesian networks?

Simple, graphical notation for conditional independence assertions

Allows compact specification of full joint distributions

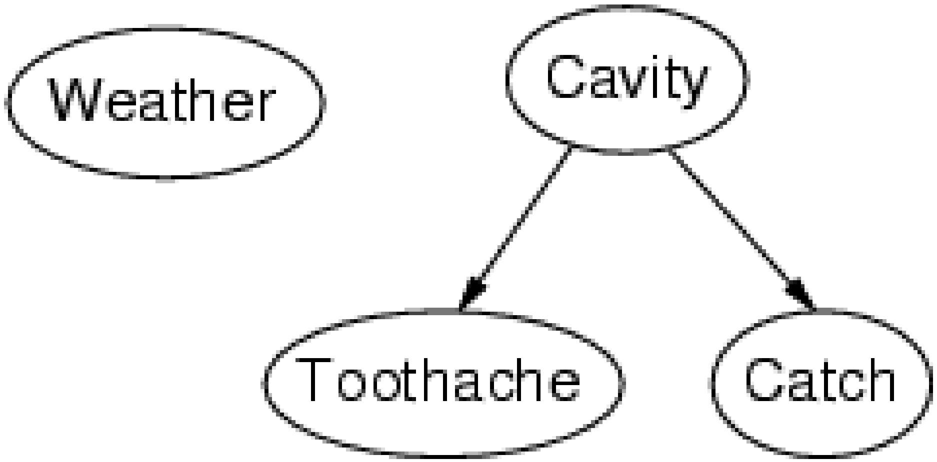
Syntax:

- a set of nodes, one per random variable
- a directed, acyclic graph (link ≈ "directly influences")
- a conditional distribution for each node given its parents: $P(X_i \mid Parents(X_i))$

For discrete variables, conditional distribution = conditional probability table (CPT) = distribution over X_i for each combination of parent values

Back at the Dentist's

Topology of network encodes conditional independence assertions:



Weather is independent of the other variables

Toothache and *Catch* are conditionally independent of each other given *Cavity*

Example 2: Burglars and Earthquakes

You are at a "Done with the AI class" party.

Neighbor John calls to say your home alarm has gone off (but neighbor Mary doesn't).

Sometimes your alarm is set off by minor earthquakes.

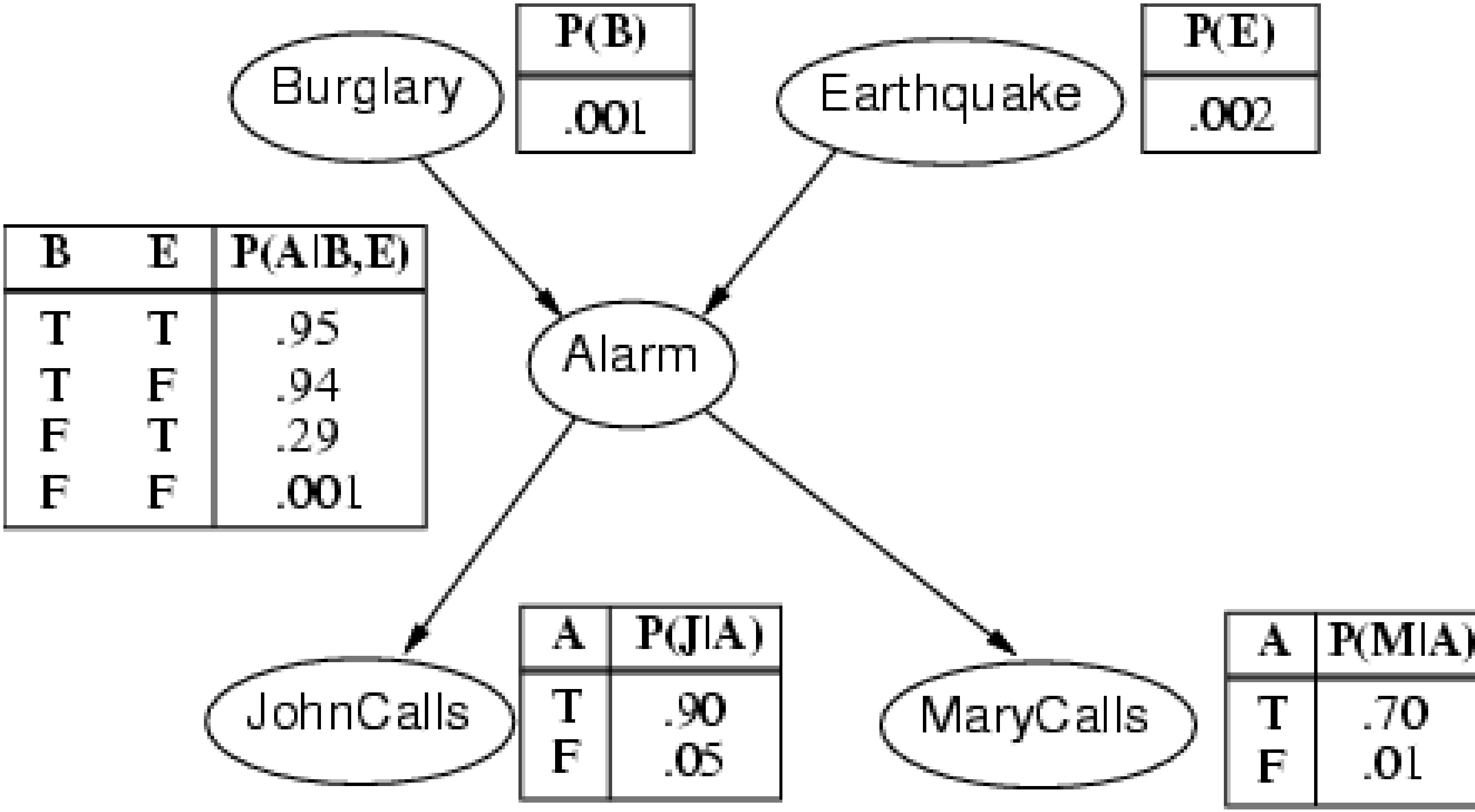
Question: Is your home being burglarized?

Variables: *Burglary, Earthquake, Alarm, JohnCalls, MaryCalls*

Network topology reflects "causal" knowledge:

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

Burglars and Earthquakes



P(B)
.001

P(E)
.002

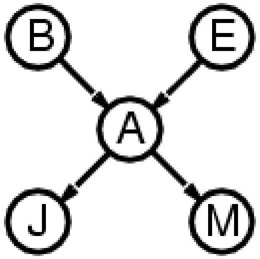
B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Compact Representation of Probabilities in Bayesian Networks

A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values



Each row requires one number p for X_i = *true* (the other number for X_i = *false* is just $1-p$)

If each variable has no more than k parents, an n-variable network requires $O(n \cdot 2^k)$ numbers

- This grows linearly with n vs. $O(2^n)$ for full joint distribution

For our network, 1+1+4+2+2 = 10 numbers (vs. 2^5-1 = 31)

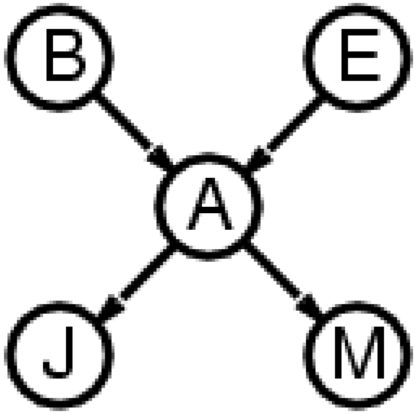
Semantics

Full joint distribution is defined as product of local conditional distributions:

$$P(X_1, \ldots, X_n) = \pi_{i=1}^{n} P(X_i \mid Parents(X_i))$$

e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$= P(j \mid a)\, P(m \mid a)\, P(a \mid \neg b, \neg e)\, P(\neg b)\, P(\neg e)$



Probabilistic Inference in BNs

The graphical independence representation yields efficient inference schemes

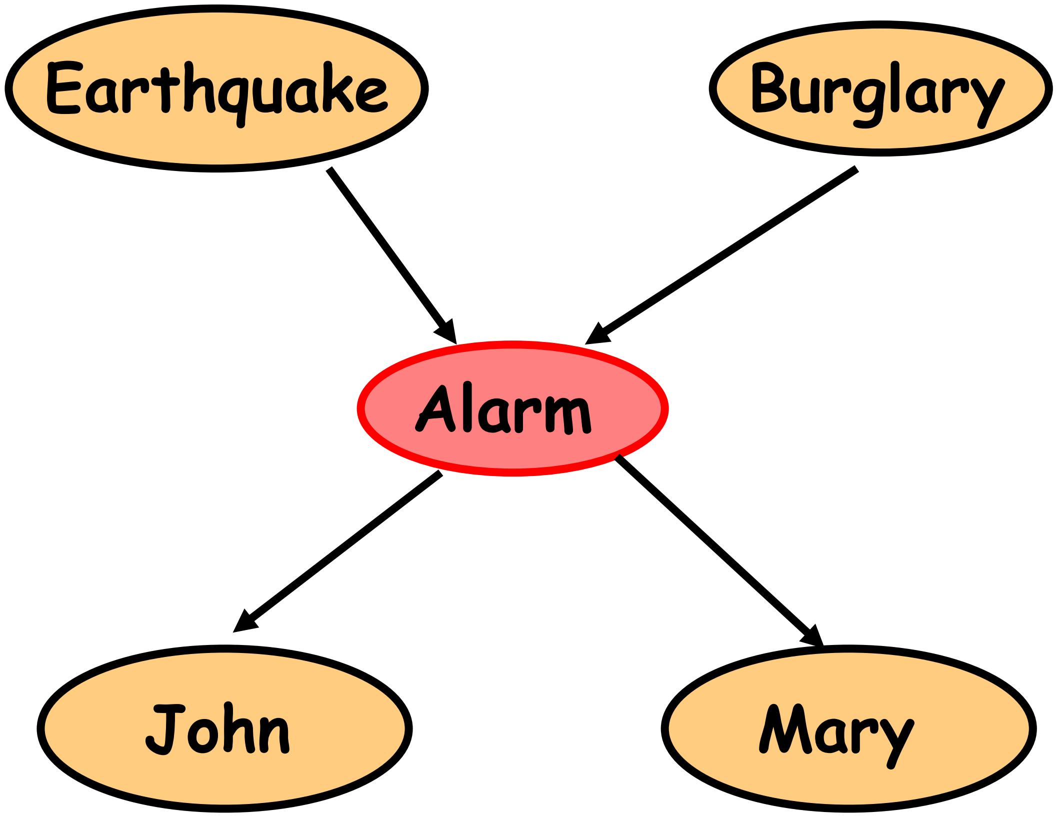
We generally want to compute

- *P(X|E)* where *E* is evidence from sensory measurements etc. (known values for variables)
- Sometimes, may want to compute just *P(X)*

One simple algorithm:

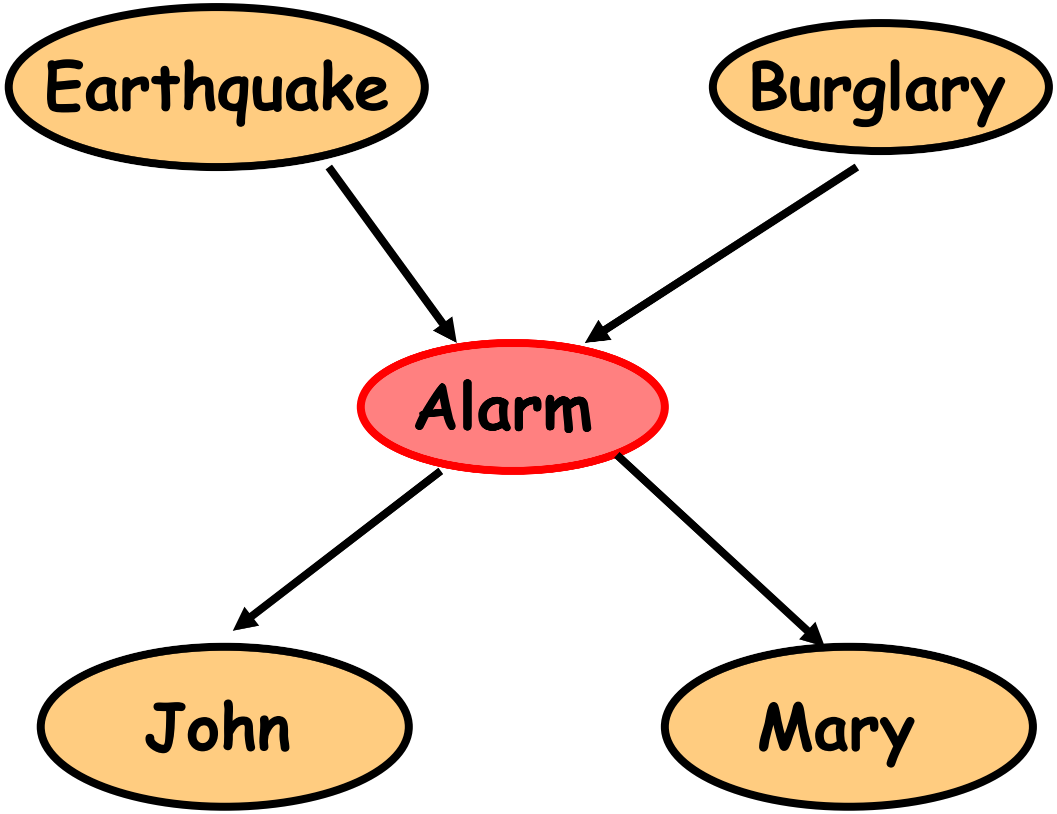
- *variable elimination (VE)*

P(B | J=true, M=true)



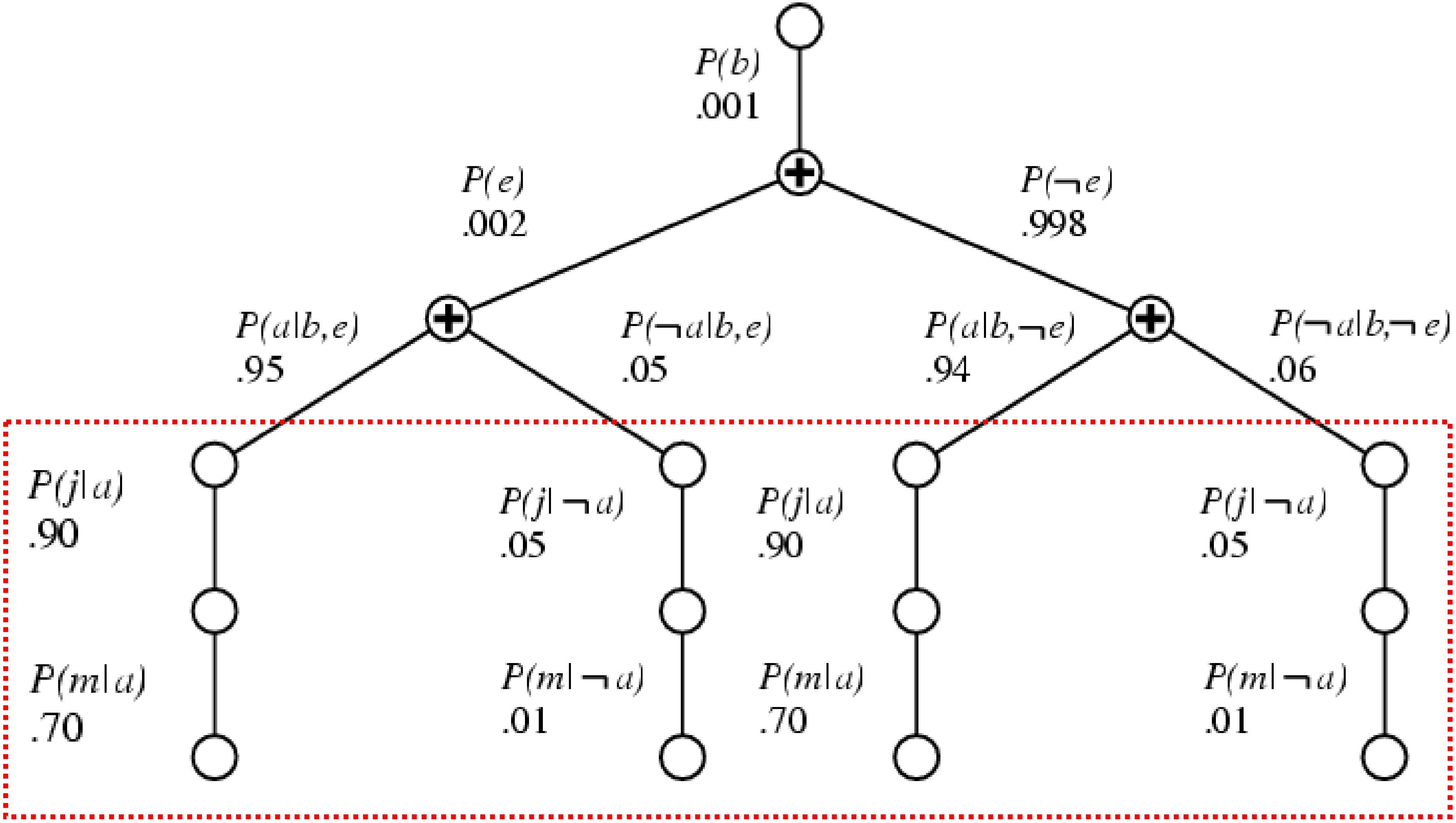
$P(b|j,m) = \alpha\, P(b,j,m) = \alpha \sum_{e,a} P(b,j,m,e,a)$

Computing P(B | J=true, M=true)



$$
\begin{aligned}
P(b|j,m) &= \alpha \textstyle\sum_{e,a} P(b,j,m,e,a) \\
&= \alpha \textstyle\sum_{e,a} P(b)\, P(e)\, P(a|b,e)\, P(j|a)\, P(m|a) \\
&= \alpha\, P(b) \textstyle\sum_{e} P(e) \textstyle\sum_{a} P(a|b,e)P(j|a)P(m|a)
\end{aligned}
$$

Structure of Computation



Repeated computations $\Rightarrow$ use dynamic programming?

Variable Elimination

A *factor* is a function from some set of variables to a specific value: e.g., *f(E,A,Mary)*

- CPTs are factors, e.g., *P(A|E,B)* function of *A,E,B*

VE works by *eliminating* all variables in turn until there is a factor with only the query variable

To eliminate a variable:

1. *join* all factors containing that variable (like DBs/SQL), multiplying probabilities
- 2. *sum out* the influence of the variable on new factor

$$P(b|j,m) = \alpha\, P(b) \sum_e P(e) \overbrace{\sum_a P(a|b,e)P(j|a)P(m|a)}$$

Example of VE: P(J)

$P(J)$

$= \Sigma_{M,A,B,E}\ P(J,M,A,B,E)$

$= \Sigma_{M,A,B,E}\ P(J|A)P(M|A)\ P(B)P(A|B,E)P(E)$

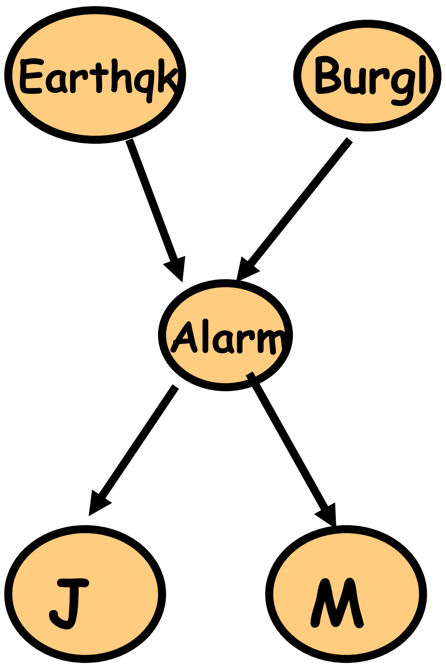
$= \Sigma_A P(J|A)\ \Sigma_M P(M|A)\ \Sigma_B P(B)\ \Sigma_E P(A|B,E)P(E)$

$= \Sigma_A P(J|A)\ \Sigma_M P(M|A)\ \Sigma_B P(B)\ f1(A,B)$

$= \Sigma_A P(N1|A)\ \Sigma_M P(M|A)\ f2(A)$

$= \Sigma_A P(J|A)\ f3(A)$

$= f4(J)$



Other Inference Algorithms

Direct Sampling:

- Repeat N times:
  - Use random number generator to generate sample values for each node
  - Start with nodes with no parents
  - Condition on sampled parent values for other nodes
- Count frequencies of samples to get an approximation to joint distribution

Other variants: **Rejection sampling, likelihood weighting, Gibbs sampling and other MCMC methods (see text)**

Belief Propagation: **A "message passing" algorithm for approximating P(X|evidence) for each node variable X**

Variational Methods: **Approximate inference using distributions that are more tractable than original ones**

(see text for details)

Summary

Bayesian networks provide a natural way to represent conditional independence

Network topology + CPTs = compact representation of joint distribution

Generally easy for domain experts to construct

BNs allow inference algorithms such as VE that are efficient in many cases

Next Time

Guest lecture by Dieter Fox on Applications of Probabilistic Reasoning

To Do: Work on homework #2