FIRST LOGIC AND PROBABILISTIC INFRENCING

Inferencing in first order logic

- Use same rules as in propositional logic:

Rule Name	Premises	Derived Conclusion
Modus Ponens	$A, A \Rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A
Double Negation	$\neg\neg A$	A
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$A \vee B, \neg B \vee C$	$A \vee C$

- Additional rules to handle universal quantifier and existential quantifier.

Quantifier rules – Universal Elimination

- **UNIVERSAL ELIMINATION**
  - Given a sentence **P(X)** containing a universally quantified variable **X**:
  - any (and all) terms **g** <u>without variables</u> in the domain of **X** can be substituted for **X** and the result **P(g)** is TRUE.

- E.g., ∀ **X, EATS(BOB,X)**
  - we can infer **EATS(BOB,CHOCOLATE)** if *CHOCOLATE* is in the domain of X.

Quantifier rules – Existential Elimination

- **EXISTENTIAL ELIMINATION**
  - Any existentially quantified variable in a true sentence can be replaced by a constant symbol not already in the knowledge base.
  - The resulting symbol (sentence) is true.

- E.g., ∃ **X, EATS(BOB,X)**
  - we can infer **EATS(BOB,*C*)** if *C* is not already in the knowledge base.
  - May not know what Bob eats, but we can assign a name to it, in this case *C*.

Quantifier rules – Existential Introduction

- **EXISTENTIAL INTRODUCTION**
  - Given a sentence **P**, a variable **X** not in **P**, and a term **g** in **P** (without variables):
  - we can infer $\exists$ **X, P(X)** by replacing **g** with **X**
- E.g., **DRINKS(ANDREW,BEER)** we can infer $\exists$ **X, DRINKS(ANDREW,X)**
- Why can't we do universal introduction as well?
  - E.g., from **DRINKS(ANDREW,BEER)** infer $\forall$ **X, DRINKS(ANDREW,X)**?

Inferencing example

- Consider the following:
  - "All bill's dogs are brown. Bill owns fred. Fred is a dog".
  - Can we infer "fred is brown"?

- **TRANSLATE INTO FIRST ORDER LOGIC:**
  - "All bill's dogs are brown" -

    $$\forall X\ \mathrm{DOG}(X) \wedge \mathrm{OWNS}(\mathrm{bill}, X) \Rightarrow \mathrm{BROWN}(X)$$
  - "Bill owns fred" – **OWNS(bill, fred)**
  - "Fred is a dog" – **DOG(fred)**

- **APPLY INFERENCING RULES:**
  - AND-INTRODUCTION - $\mathrm{DOG}(\mathrm{fred}) \wedge \mathrm{OWNS}(\mathrm{bill}, \mathrm{fred})$
  - UNIVERSAL ELIMINATION -

    $$\mathrm{DOG}(\mathrm{fred}) \wedge \mathrm{OWNS}(\mathrm{bill}, \mathrm{fred}) \Rightarrow \mathrm{BROWN}(\mathrm{fred})$$
  - MODUS PONENS - **BROWN(fred)**

Generalized Modus Ponens

- Combines the steps of And-Introduction, Universal-Elimination and Modus Ponens into one step.
- Can build an inferencing mechanism using Generalized Modus Ponens (much like using Modus Ponens in propositional logic).
- Generalized Modus Ponens is not complete. There might be sentences entailed by a knowledge base we can not infer.
  - Has to do with structure of clauses.
  - Is complete for Horn clauses (positive literals, conjunctions only).
    - E.g., $\forall$ **X DOG(X)** $\wedge$ **OWNS(bill,X)** $\Rightarrow$ **BROWN(X)**

Forward Chaining

- Talked about this with Modus Ponens and propositional logic.
- Generalized Modus Ponens can be used with **forward chaining**.
- **Forward chaining**:
  - Start with a set of facts (KB) and derive new information.
  - Continue until goal is reached, always updating the KB with new conclusions.
  - **Data-driven** since no particular path to goal.
    - Cf. Uninformed search strategies.

Backward Chaining

- Talked about this with Modus Ponens and propositional logic.

- Generalized Modus Ponens can be used with **backward chaining**.

- **Backward chaining**:
  - Start with a set of facts (KB) and a goal to be proven.
  - Look for implication sentences that would allow our goal to be proven.
  - Consider these sentences to be <u>sub-goals</u>, and try to prove them.
  - **Goal-driven** since we always strive to prove something goal related.

- E.g.,"Is fred brown?" which we represent as **BROWN(fred)?**
  - We match the implication $\forall$ **X DOG(X)** $\wedge$ **OWNS(bill,X)** $\Rightarrow$ **BROWN(X) if X=fred.**
  - We know we have two sub-goals, namely **DOG(fred)?** And **OWNS(bill,fred)?**

Unification

- Generalized Modus Ponens requires **pattern matching**.
  - E.g., in our previous example, we needed to make **DOG(fred)** and **DOG(X)** match by using **variable instantiation.**
  - **Variable instantiation** means finding a value for **X** to make expressions equivalent.

- **Unification** is the algorithm to match two expressions.
  - Returns the **variable substitutions** to make two sentences match, or **failure** if no match possible.
  - **UNIFY(p,q) =** θ are matched where θ is the list of substitutions in **p** and **q**.

- UNIFY(p,q) should return the matching that places the **least** restrictions on variable values.
  - Resulting substitution list is called the **most general unifier (MGU)**.

Unification rules

Rules to unify p and q:

- Function symbols and predicate symbols must have identical names and number of arguments.
- Constant symbols unify with only identical constant symbols.
- Variables unify with other variable symbols, constant symbols or function symbols.
- Variable symbols may not be unified with other terms in which the variable occurs. E.g., X cannot unify with G(X) since this will lead to G(G(G(G(…G(X))))). This is an **occur check error**..

Simplified pseudocode for unification

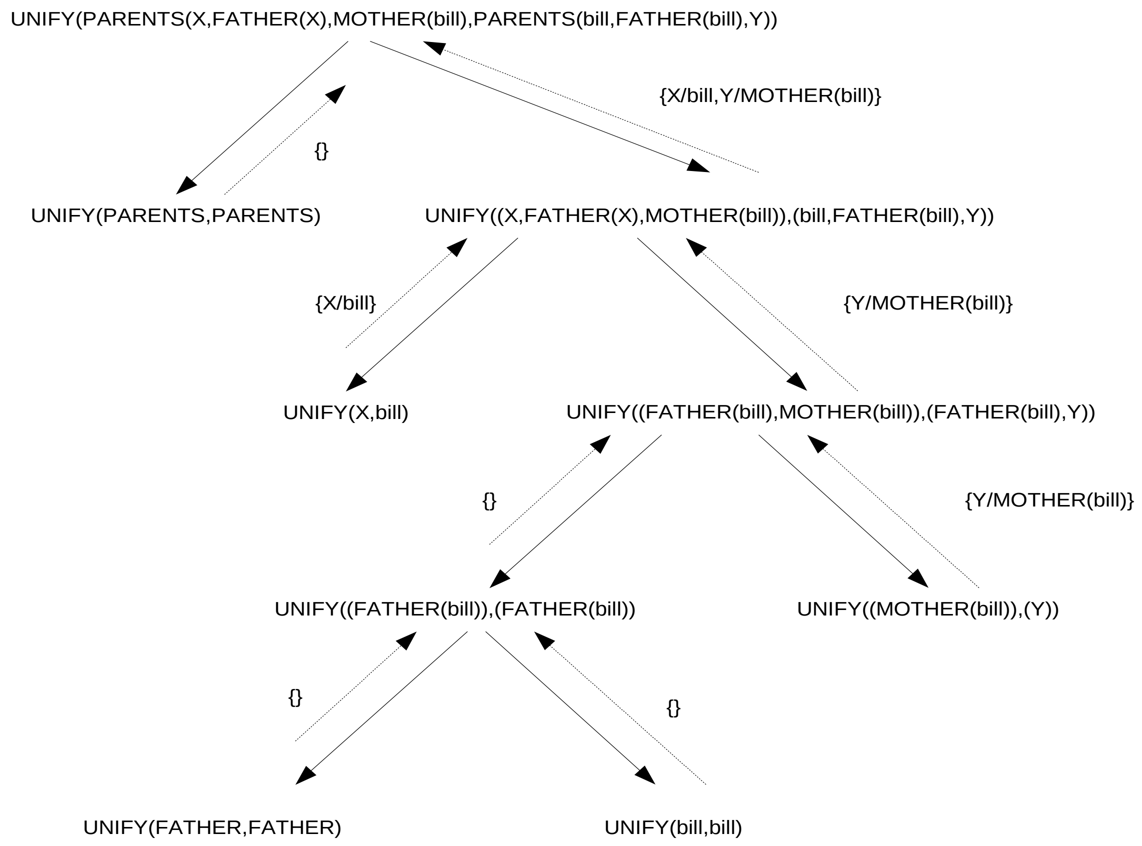
```
UNIFY(p,q,θ) {
  /* scan p and q left to right. */
  /* let r and s be terms where there is disagreement between p and q. */

  /* examine each r/s pair.
  if (IsVariable(r)) {
      if (r ∈ s) return failure; // occur check error.
      else {
              θ = θ ∧ r/s.
              /* apply substitutions to p and q */
              UNIFY(p,q,θ);
  }
  if (IsVariable(s)) {
      if (s ∈ r) return failure; // occur check error.
      else {
              θ = θ ∧ s/r.
              /* apply substitutions to p and q */
              UNIFY(p,q,θ);
  }
}
```

- Additional checks for matching constant, function and predicate symbols.

Example of unification

- Unify: **PARENTS(X,FATHER(X),MOTHER(bill))** and **PARENTS(bill,father(bill),Y)**.



Resolution

- Sound and complete for first order logic.
- Idea is to prove something using **refutation** (proof by contradiction).
  - We want to prove something (clause/sentence).
  - We add its negation into the KB.
  - We attempt to reach a contradiction as a result of the addition of our negated goal.
  - If contradiction reached, our goal must be true.
- Note: same idea as in propositional logic, but need to handle **quantifiers** and **variables**.

Resolution

Resolution is step-by-step:

1. Convert problem into first order logic expressions.
2. Convert logic expressions into clause form.*
3. Add negation of what we intend to prove (in clause form) to our logic expressions.
4. Use resolution rule to produce new clauses that follow from what we know.
5. Produce a contradiction that proves our goal.

- **Any variable substitutions used during the above steps are those assignments for which the opposite of the negated goal is true.**
  - **Has to do with Unification, more later.**

Conversion to clause form

- Recall resolution for clauses $\alpha \vee \beta$, $\neg \beta \vee \gamma$ resolves to $\alpha \vee \gamma$.
- So resolution works with pairs (or groups) of disjuncts to produce new disjuncts.
- Therefore, need to convert first order logic into disjuncts (Conjunctive Normal Form, or clause form).
  - Individual clauses *conjuncted* together, each clause made up of *disjunctions* of literals.

Conversion procedure

1. Eliminate all $\Rightarrow$ using the fact that $A \Rightarrow B \equiv \neg A \vee B$.

2. Reduce scope of negation to the predicate level using:

   a) $\neg(\neg A) \equiv A$
   b) $\neg(\exists X), A(X) \equiv (\forall X), \neg A(X)$
   c) $\neg(\forall X), A(X) \equiv (\exists X), \neg A(X)$
   d) $\neg(A \wedge B) \equiv \neg A \vee \neg B$
   e) $\neg(A \vee B) \equiv \neg A \wedge \neg B$

3. Rename variables so that they are different if bound by different quantifiers.

Conversion procedure

4. Eliminate existential quantifiers. This is known as **Skolemization**.

   a) E.g., $\exists$ **X, DOG(X)** may be replaced by **DOG(fido)** where **fido** is a **Skolem constant**.

   b) E.g., $\forall$ **X (**$\exists$ **Y MOTHER(X,Y))** must be replaced by $\forall$ **X MOTHER(X,m(X))** where **m(X)** is a **Skolem function of X**.

5. Drop all $\forall$ universal quantifiers. Assume all variables to be universally quantified.

6. Convert each expression to a conjunction of disjuncts using $A \lor (B \land C) \equiv (A \lor B) \land (A \lor C)$.

7. Split conjuncts into separate clauses and rename variables in different clauses.

Example of conversion

- Convert: "Everyone who loves all animals is loved by someone"

  $\forall X\ [\ \forall Y\ \text{Animal}(Y) \Rightarrow \text{Loves}(X,Y)\ ] \Rightarrow [\ \exists Y\ \text{Loves}(Y,X)\ ]$

1. Remove $\Rightarrow$ from expression:

   $\forall X\ [\neg \forall Y\ \neg \text{Animal}(Y) \vee \text{Loves}(X,Y)\ ] \vee [\exists Y\ \text{Loves}(Y,X)\ ]$

2. Reduce scope of negation:

   $\forall X\ [\ \exists Y\ \ \text{Animal}(Y) \wedge \neg \text{Loves}(X,Y)\ ] \vee [\exists Y\ \text{Loves}(Y,X)\ ]$

3. Rename variables:

   $\forall X\ [\ \exists Y\ \ \text{Animal}(Y) \wedge \neg \text{Loves}(X,Y)\ ] \vee [\exists Z\ \text{Loves}(Z,X)\ ]$

4. Eliminate $\exists$ using Skolemization:

   $\forall X\ [\ \text{Animal}(\ F(X)\ ) \wedge \neg \text{Loves}(\ X,\ F(X)\ )\ ] \vee [\ \text{Loves}(\ G(X),\ X)\ ]$

Example of conversion

5. Drop universal quantifiers:
   [Animal(F(X)) ∧ ¬Loves(X, F(X))] ∨ [Loves(G(X), X)]

6. Convert to conjuction of disjuncts:
   [Animal(F(X)) ∨ Loves(G(X), X)] ∧ [¬Loves(X, F(X)) ∨ Loves(G(X), X)]

7. Separate into clauses:
   [Animal(F(X)) ∨ Loves(G(X), X)]
   [¬Loves(X, F(X)) ∨ Loves(G(X), X)]

8. Rename variables (again) in different clauses:
   [Animal(F(X)) ∨ Loves(G(X) ,X)]
   [¬Loves(W, F(W)) ∨ Loves(G(W), W)]

Example of resolution refutation

- Consider the following story:

  "Anyone passing his or her artificial intelligence exam and winning the lottery is happy. But anyone who studies or is lucky can pass all his exams. Pete did not study but is lucky. Anyone who is lucky wins the lottery. Is Pete happy?".

Example of resolution refutation

Step 1: Change sentences to first order logic:

1. "Anyone passing his or her artificial intelligence exam and winning the lottery is happy"

   $(\forall X)\ (PASS(X,ai) \wedge WIN(X,lottery) \Rightarrow HAPPY(X))$

2. "Anyone who studies or is lucky can pass all his exams"

   $(\forall X \forall Y)\ (STUDIES(X) \vee LUCKY(X) \Rightarrow PASS(X,Y))$

3. "Pete did not study but is lucky"

   $\neg STUDY(pete) \wedge LUCKY(pete)$

4. "Anyone who is lucky wins the lottery"

   $(\forall X)\ (LUCKY(X) \Rightarrow WINS(X,lottery))$

Example of Resolution Refutation

Step 2: Convert all sentences into clause form:

1. $(\forall X)\ (PASS(X,ai) \wedge WIN(X,lottery) \Rightarrow HAPPY(X))$ gives:

   $\neg PASS(X,ai) \vee \neg WIN(X,lottery) \vee HAPPY(X)$

2. $(\forall X\ \forall Y)\ (\neg STUDIES(X) \vee LUCKY(X) \Rightarrow PASS(X,Y))$ gives:

   $\neg STUDIES(Y) \vee PASS(Y,Z)$

   $\neg LUCKY(W) \vee PASS(W,V)$

3. $\neg STUDY(pete) \wedge LUCKY(pete)$ gives:

   $\neg STUDIES(pete)$

   $LUCKY(pete)$

4. $(\forall X)\ (LUCKY(X) \Rightarrow WINS(X,lottery))$ gives:

   $\neg LUCKY(U) \vee WINS(U,lottery)$

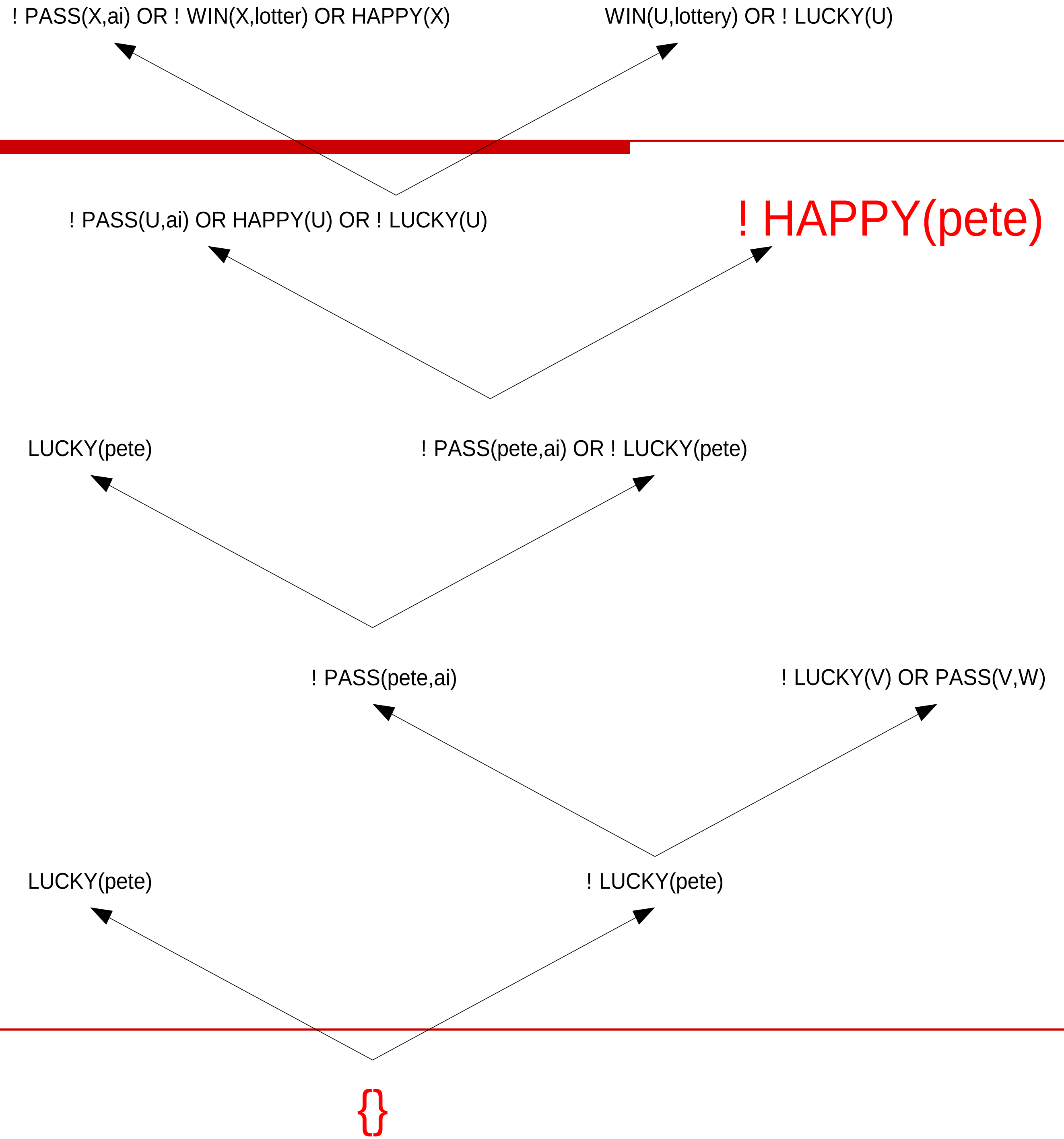
Example of resolution refutation

Step 3: Add negation (in clause form) of what we want to know:

$\neg$ HAPPY(pete)

Step 4: Use resolution (and our negated goal) to build a resolution refutation graph to prove a contradiction.

- When building the graph to prove the contradiction, it should be the case that the negative of the goal gets used somewhere in the proof.
- Contradiction occurs when resolve clauses like A, $\neg$ A
-
- result = {} (NULL set) since A cannot be true and false at the same time!



! PASS(X,ai) OR ! WIN(X,lotter) OR HAPPY(X)
WIN(U,lottery) OR ! LUCKY(U)
! PASS(U,ai) OR HAPPY(U) OR ! LUCKY(U)
! HAPPY(pete)
LUCKY(pete)
! PASS(pete,ai) OR ! LUCKY(pete)
! PASS(pete,ai)
! LUCKY(V) OR PASS(V,W)
LUCKY(pete)
! LUCKY(pete)
{}

Probabilistic Approach to Uncertainty

- Logic agents almost never have access to the whole truth about their environment.
- There will always be questions to which a categorical answer (i.e., TRUE of FALSE) cannot be found.
- Will try to apply **probability theory** to deal with degree of belief about things.

Prior (or Unconditional) Probabilities

- The prior or unconditional probability is the probability that an event **A** is **TRUE** in the **absence of any other information**. Denoted with **P(A)**.

- Axioms for probabilities:

  1. **P(A)** should be a number between 0 and 1.

  2. **P(A) = 1** if ***A*** is a necessarily TRUE event and **P(A) = 0** if ***A*** is a necessarily FALSE event.

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

- **Mutually exclusive** means that events A and B cannot both be true at the same time. In this case, we find $P(A \vee B) = P(A) + P(B)$.

Mutual Exclusive Events

- Given two events **A** and **B**, the events are **Mutually Exclusive Events** if they **cannot both** be true at the same time.

- In this case, we find **$P(A \vee B) = P(A) + P(B)$ – (since $P(A \wedge B)$** must be **0).**

Other Properties of Probabilities

- From axioms, can derive other properties too.
- **Example**: Let **B** be $\neg$**A** and attempt to find **P(B) = P($\neg$A).**
- **Solution:**

$$
\begin{aligned}
P(A \vee B) &= P(A \vee \neg A) \\
&= P(A) + P(\neg A) - P(A \wedge \neg A) \\
&= P(A) + P(\neg A) - P(FALSE) \\
&= P(A) + P(\neg A) \\
&= P(TRUE) \\
&= 1.
\end{aligned}
$$

- **Therefore, $P(A) + P(\neg A) = 1$, and $P(\neg A) = 1 - P(A)$.**

Random Variables

- In **P(A)**, ***A*** can be a proposition, and logical connection, or a **random variable**.
- Domains can be **discrete** or **continuous**.
- **Example #1**: Consider we are interested in the number of faults in a system. Best to think of *A* as a variable with a domain of {0,1,2,3,…}
- **Example #2**: Consider we are interested in the outdoor temperature. Best to think if T as a random variable with domain of {cold, cool, warm, hot}.

$$
\begin{aligned}
P(T = cold) &= 0.2 \\
P(T = cool) &= 0.3 \\
P(T = warm) &= 0.4 \\
P(T = hot) &= 0.1
\end{aligned}
$$

Random Variables and Probability Distributions

- Given a random variable *A* and its domain $\{a_1, a_2, \cdots, a_n\}$, its probability distribution is $\{P(a_1), P(a_2), \cdots, P(a_n)\}$.

- Events in the domain are mutually exclusive. Therefore…

$$\sum_{1}^{n} P(A = a_i) = 1$$

- When we talk about P(A) for a random variable, we are actually talking about its probability distribution.

Joint Probabilities and Marginalization

- Assume we are given two random variables **A** and **B**.
- We denote by **P(A,B)** the **joint probability distribution** of **A** and **B**.
- Specifically, for two random variables we have a table of probabilities where **$P(a_i,b_j)$** is the probability that **$A=a_i$ and $B=b_j$**.
- If we know the joint probability distribution P(A,B), we can compute P(A) using:

$$P(a_i) = \sum_j P(a_i, b_j)$$

- This works since the ev… , $\cdots$ are mutually exclusive.
- Removal of **B** to compute **P(A)** from **P(A,B)** is called **marginalization**.

Posterior (or Conditional) Probabilities

- **Posterior or conditional probability** – the probability that an event **A** is **TRUE** given some other evidence.

- Notation is **P(A|B)** meaning the probability of A given that B is known.

- Defined in terms of prior and joint probabilities **$P(A|B)P(B) = P(A,B)$** and by symmetry **$P(B|A)P(A) = P(A,B)$** where **$P(A,B) = P(A \wedge B)$**.

Bayes Theorem

- We have that P(A,B) = P(A|B) P(B) = P(B|A) P(A) and therefore we can remove the joint probability to find that:

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

- Called **Bayes' Theorem** and provides a way to determine a conditional probability without the joint probability of A and B.

Bayes Theorem

- Common to think of Bayes' Theorem in terms of updating our belief about a hypothesis A in the light of new evidence B.
  - Specifically, *our posterior belief P(A|B) is calculated by multiplying our prior belief P(A) by the likelihood P(B|A) that B will occur if A is true.*
- In many situations it is difficult to compute P(A|B) directly, yet we might have information about P(B|A).
  - Bayes' Theorem enables us to compute P(A|B) in terms of P(B|A).
  - This is what makes Bayes' Theorem so powerful.

Example of Bayes Theorem

- Let A represent the event "Person has cancer" and let B represent the event "Person is a smoker".

- We know the probability of the prior event A is 0.1 based on past data (10% of patients entering the clinic turn out to have cancer). Thus $P(A)=0.1$

- Want to compute the probability of the posterior event $P(A|B)$.

Example of Bayes Theorem

- It is difficult to find this out directly. However, …
  - We are likely to know P(B) by considering the percentage of patients who smoke. Suppose P(B)=0.5.
  - We are also likely to know P(B|A) by checking from our records the proportion of smokers among those diagnosed with cancer. Suppose P(B|A)=0.8.

- We can now use Bayes' Theorem to compute P(A|B) = (0.8 0.1)/0.5 = 0.16

- Thus, in the light of *evidence* that the person is a smoker we revise our prior probability from 0.1 to a posterior probability of 0.16. This is a significance increase, but it is still unlikely that the person has cancer.

Bayes Theorem In a Different Form

- The denominator of Bayes' Theorem P(A|B) = P(B|A) P(A) / P(B) is P(B) and is a normalizing constant which can be computed using marginalization.

$$P(B) = \sum_{1}^{n} P(B|A_i)P(A_i)$$

- Hence we can state Bayes' Theorem is a different form:

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

Example of Bayes Theorem In a Different Form

- Suppose that we have two bags each containing black and white balls.

- One bag contains three times as many white balls as blacks. The other bag contains three times as many black balls as white.

- Suppose we choose one of these bags at random. For this bag we select five balls at random, replacing each ball after it has been selected. The result is that we find 4 white balls and one black.

- What is the probability that we were using the bag with mainly white balls?

Example of Bayes Theorem In a Different Form

- Let A be the random variable "bag chosen" then A={a1,a2} where a1 represents "bag with mostly white balls" and a2 represents "bag with mostly black balls" . We know that P(a1)=P(a2)=1/2 since we choose the bag at random.

- Let B be the event "4 white balls and one black ball chosen from 5 selections". We have to calculate P(a1|B).

- For the bag with mostly white balls, we find P(B|a1) = 405/1024. For the bag with mostly black balls we find P(B|a2) = 15/1024.

- Hence, P(a1|B) = P(B|a1)P(a1)/(P(B|a1)P(a1)|P(B|a2)P(a2)) = ((405/1024)/(405/1024)+(15/1024)) = 0.964.

Chain Rule

- Recall that $P(A,B) = P(A|B)\,P(B)$.
- Can extend this formula to more variables.
- Example: For 3-variables:

  $$P(A,B,C) = P(A|\,B,C)\,P(B,C) = P(A|B,C)\,P(B|C)\,P(C)$$

- Example: For n-variables:

  $$P(A1,\,A2,\,\ldots,\,An) = P(A1|\,A2,\,\ldots,\,An)\,P(A2|\,A3,\,\ldots,\,An)\,P(An\text{-}1|An)\,P(An)$$

Chain Rule

- In general we refer to this as the **chain rule**.
- This formula provides a means of calculating the full joint probability distribution.
- This formula is especially significant for **Bayesian Belief Nets** where/when many of the variables are conditionally independent (and the formula can be simplified.

Independence and Conditional Independence

- Two events are independent if $P(A \wedge B) = P(A)\,P(B)$.

- If both events have a positive probability, then the statement of independence of events is equivalent to a statement of conditional independence:

$$P(A|B) = P(A) \text{ if and only if } P(B|A) = P(B) \text{ if and only if } P(A \wedge B) = P(A)\,P(B)$$

- Can think of independence in the following way: knowledge that one event has occurred does not change the probability assigned to the other event.

Bayesian Networks (Belief Networks)

- Recall - given **random variable v**, possible that only a small set of other variables **directly affect v** in the sense that other variables are **independent of v given values for those directly affecting variables**.

- A Bayesian Belief Network is a graphical representation of the dependencies (and conditional independencies) between random variables.

Bayesian Networks (Belief Networks)

- Belief Network is:
  - A directed acyclic graph G(N,A).
  - Nodes represent random variables with domains.
  - Directed edges between pairs of nodes $\mathbf{n_i}$, $\mathbf{n_j} \in \mathbf{N}$ and $\langle \mathbf{n_i}, \mathbf{n_j} \rangle \in \mathbf{A}$, then $\mathbf{n_i}$ is a **parent** of $\mathbf{n_j}$ and $\mathbf{n_j}$ is a **child** of $\mathbf{n_i}$.
  - A set of conditional probability distributions (expressed as **conditional probability tables – CPTs**) for nodes with parents and prior probabilities for nodes without parents.

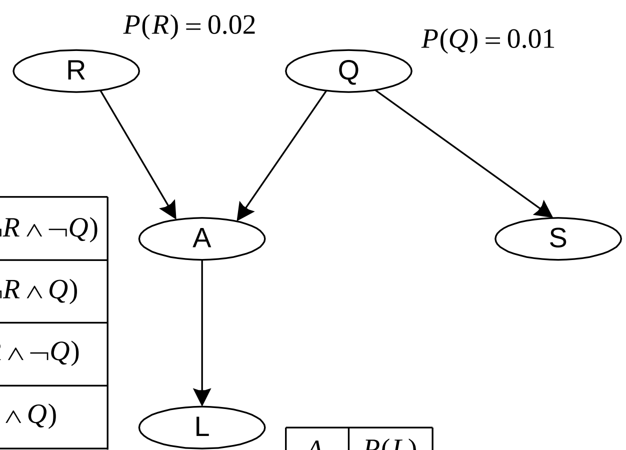
Bayesian Networks (Belief Networks)

- Topology of a belief network – nodes and edges – specifies conditional independence relationships that hold.

- Belief networks give a concise representation of **any full joint probability distribution for n-variables**; i.e., full representation of $\mathbf{P(x_1, x_2, \cdots, x_n)}$.

Belief Network Example

- CPTs shown for nodes with parents (CPTs for a node depend on parents). Prior probabilities shown for nodes without parents.



R	Q	$P(A)$	
F	F	0.0001	$P(A \mid \neg R \wedge \neg Q)$
F	T	0.99	$P(A \mid \neg R \wedge Q)$
T	F	0.85	$P(A \mid R \wedge \neg Q)$
T	T	0.50	$P(A \mid R \wedge Q)$

Q	$P(S)$	
F	0.01	$P(S \mid \neg Q)$
T	0.9	$P(S \mid Q)$

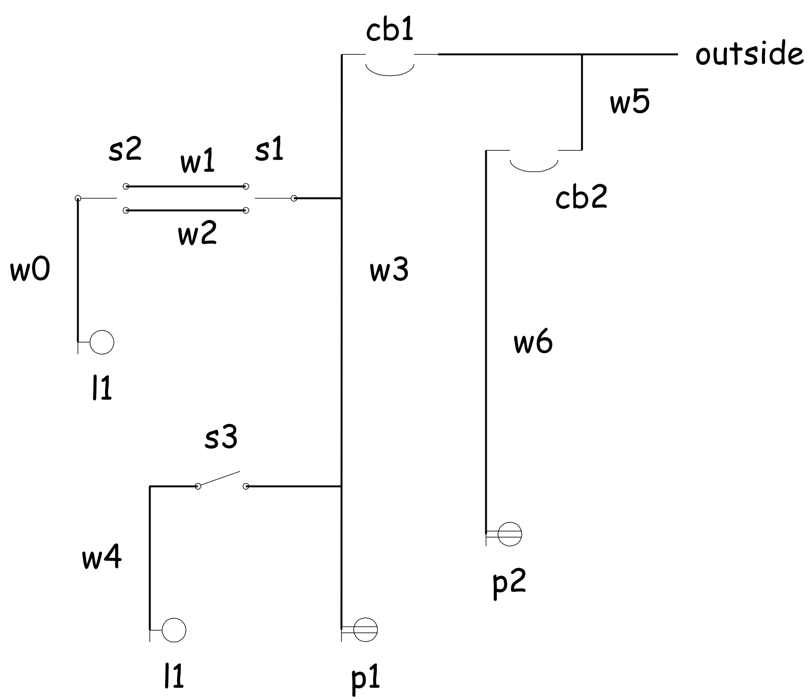
A	$P(L)$	
F	0.001	$P(L \mid \neg A)$
T	0.88	$P(L \mid A)$

- Note that each row in the CPT must sum to 1 since entries are exhaustive. For Boolean variables, once you know the probability of a true value is p, then the value it is false must be 1-p.

- So, $P(\neg L|A) = 1 - 0.88 = 0.12$.

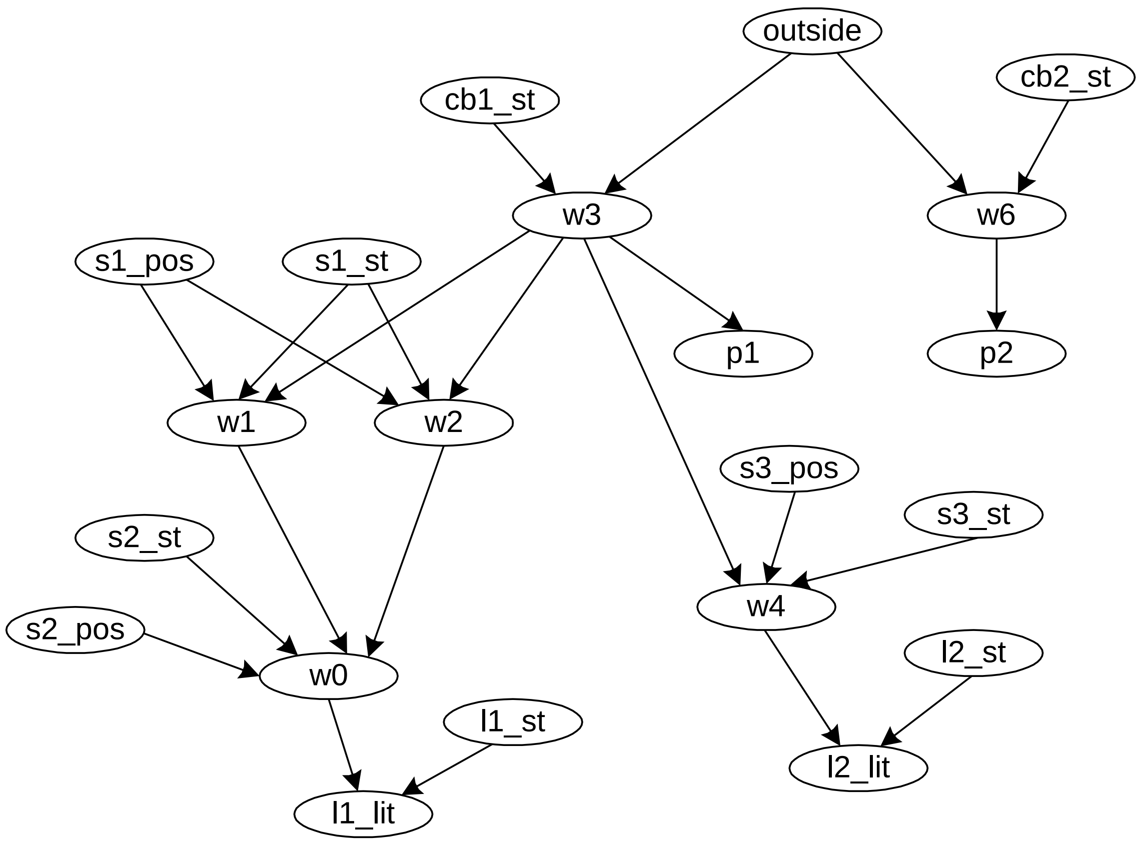
Belief Network Example

- Consider modeling of house wiring to be able to determine things like which lights are on or off based on switch positions, status of circuit breakers, and to be able to diagnose problems with wires, switches, circuit breakers if something unexpected is observed.



Belief Network Example

- Can model the belief network that captures variable dependence (or independence) between random variables.



Belief Network Example

- Can define each random variable and domains (conditional probability tables and prior probabilities for nodes with no parents not shown, but would be required!).
  - The random variable wi denoted whether the i-th wire has power and has domain {live, dead}.
  - The random variable si_pos denotes the position of the –ith switch and has domain {up, down}.
  - The variable si_st denotes the status of the i-th switch and has domain {ok, upside_down, short, intermittent, broken}.
  - The random variable cbi_st denotes the status of the i-th circuit breaker and has domain {on, off}.
  - The random variable outside denotes the status of the main and has domain {live, dead}.
  - The random variable pi denotes the status of the i-th plug and has domain {live, dead}.
  - The random variable li_lit indicates that the i-th light is lit and has domain {no, yes}.
  - The random variable li_st denotes the status of the i-th light and has domain {ok, intermittent, broken}.

Semantics of Belief Networks

- Need to understand when a belief network is a correct representation of conditional independence.
- Consider the joint probability distribution for n variables using the chain rule:

$$\begin{aligned} P(x_n, \cdots, x_2, x_1) &= P(x_n|x_{n-1}, \cdots, x_1)P(x_{n-1}, \cdots, x_1) \\ &= P(x_n|x_{n-1}, \cdots, x_1)P(x_{n-1}|x_{n-2}, \cdots, x_1) \cdots P(x_2|x_1)P(x_1) \\ &= \Pi_{i=1}^{n} P(x_i|x_{i-1}, \cdots, x_1) \end{aligned}$$

Semantics of Belief Networks

- Order the variables (easy since graph is a DAG). We can re-write the above equation as follows:

$$P(x_n, \cdots, x_2, x_1) = \Pi_{i=1}^{n} P(x_i | Parents(x_i))$$

- Here, **Parents(x_i)** is the set of parent nodes for the node **x_i** and **P(x_i|Parents(x_i)) = P(x_i|x_{i-1},⋯,x_1)**.

- Each **P(x_i|x_{i-1},⋯,x_1)** has the property that you are not conditioning on descendents since **Parents(x_i)** $\subseteq$ **{x_{i-1},⋯,x_1}** (which is true due to ordering).

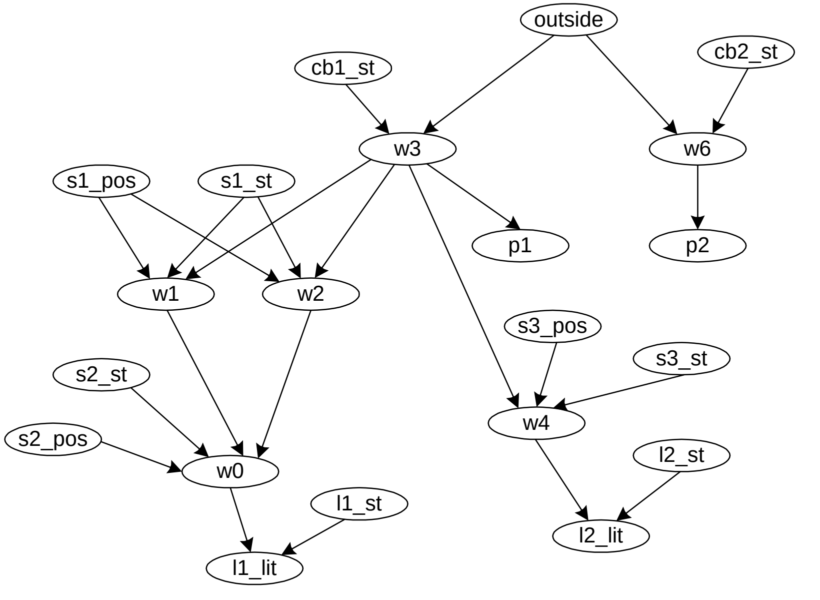
Semantics of Belief Networks

$$P(x_n, \cdots, x_2, x_1) \;=\; \Pi_{i=1}^{n} P(x_i | Parents(x_i))$$

- This equation says that a belief network is a correct representation of the problem only if each node (random variable) is conditionally independent of its predecessors given its parents.
- Alternatively, **a node is conditionally independent of its non-descendents given its parents**.
- This assumption of conditional independence is embedded in the belief network.

Example of Independence

- Knowing s1_pos, s1_st and w3, then knowing l2_lit or cb_2st **does not change** belief in w1.
- Knowing s1_pos, s1_st and w3, then learning l1_lit **does change** belief in w1.



Inference in Belief Networks

- Recall that belief networks specify conditional independence between nodes (random variables).
- It also specifies the full joint distribution of variables:

$$P(x_n, \cdots, x_2, x_1) = \Pi_{i=1}^{n} P(x_i | Parents(x_i))$$

- The basic task of a belief network is as follows:
  - Compute the posterior probability for a query variable given an observed event
  - i.e., an **assignment of values** to a set of **evidence variables** while other variables are **not assigned values** (the so-called **hidden variables**).

Inference in Belief Networks

In other words…

- Let **E** denote a set of evidence values $\mathbf{E_1, E_2, \cdots, E_m}$. Let the observed event be $\mathbf{e = (e_1, e_2, \cdots, e_m)}$.
- Let **Y** denote a set of non-evidence variables $\mathbf{Y_1, Y_2, \cdots, Y_l}$.
- Let **X** denote the query variable.
- Hence, the belief network is composed of the nodes $\mathbf{X \cup E \cup Y}$.
- The network specifies $\mathbf{P(X \wedge E \wedge Y)}$.
- We want to compute (i.e., answer the query) $\mathbf{P(X \mid E)}$; **i.e., What is the posterior probability of X given the observed evidence E? Is it different than its prior probability?**

Inference in Belief Networks

- We need to note that:

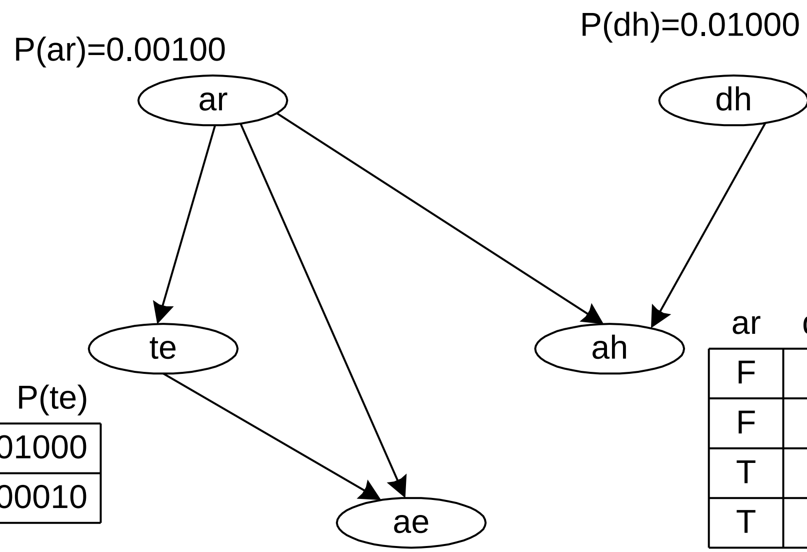
  $P(X|E) = \alpha\, P(X,E)$ where α is a normalization constant (this is just Bayes Theorem…)

- We can write this in terms of the full joint distribution if we sum out the non-evidence variables:

  $P(X|E) = \alpha\, P(X,E) = \alpha \sum_{y} P(X,E,Y)$

Example of Inference

- Consider the following belief network.



ar	P(te)
F	0.01000
T	0.00010

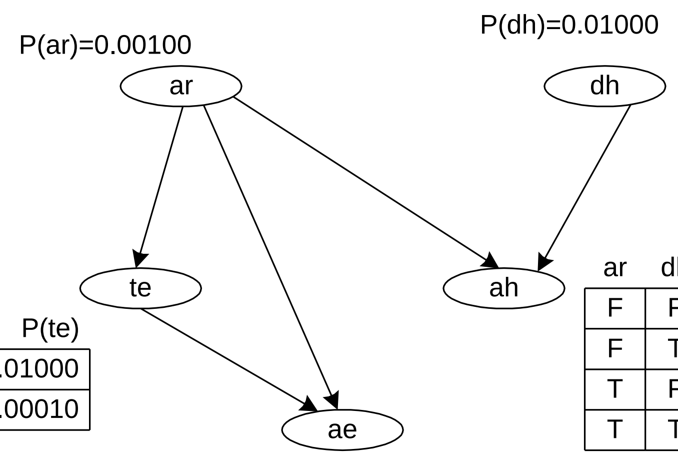
ar	dh	P(ah)
F	F	0.00001
F	T	0.99000
T	F	0.99000
T	T	0.10000

te	ar	P(ae)
F	F	0.00010
F	T	0.99000
T	F	0.99000
T	T	0.10000

Example of Inference

- Without making any observations, we can compute P(te):

$$
\begin{aligned}
P(te) &= P(te \wedge ar) + P(te \wedge \neg ar) \\
&= P(te|ar)P(ar) + P(te|\neg ar)P(\neg ar) \\
&= P(te|ar)P(ar) + P(te|\neg ar)(1 - P(ar)) \\
&= 0.0001(0.001) + 0.01(1 - 0.001) \\
&= 0.00999
\end{aligned}
$$



ar	P(te)
F	0.01000
T	0.00010

te	ar	P(ae)
F	F	0.00010
F	T	0.99000
T	F	0.99000
T	T	0.10000

ar	dh	P(ah)
F	F	0.00001
F	T	0.99000
T	F	0.99000
T	T	0.10000

Example of Inference

- However, say we have observed **ae** to be **TRUE**. We would like to find if **te** becomes more likely given this observation; i.e., we want to find **P(te|ae)**.

- Network tells us that:

$$\mathbf{P(ae,te,ah,ar,dh) = P(dh)P(ar)P(ah|ar,dh)P(te|ar)P(ae|te,ar)}$$

- So we want to find:

$$\mathbf{P(te|ae)} = \alpha \sum_{ar} \sum_{ah} \sum_{dh} \mathbf{P(dh)P(ar)P(ah|ar,dh)P(te|ar)P(ae|te,ar)}$$

- Note that we need to find **P(te|ae) and P(¬ te|ae)** in order to **normalize properly**.

- To find one of these values, we need to sum 8 terms, each involving 5 multiplications. All the information required is in the network.

Example of Inference

- We can compute P(te|ae):

ah	dh	ar	P(dh)	P(ar)	P(ah|ar,dh)	P(te|ar)	P(ae|te,ar)	
T	F	F	0.99000	0.99900	0.00001	0.01000	0.99000	0.00000009791199
T	F	T	0.99000	0.00100	0.99000	0.00010	0.10000	0.00000000980100
T	T	F	0.01000	0.99900	0.99000	0.01000	0.99000	0.00009791199000
T	T	T	0.01000	0.00100	0.10000	0.00010	0.10000	0.00000000001000
F	F	F	0.99000	0.99900	0.99999	0.01000	0.99000	0.00979110108801
F	F	T	0.99000	0.00100	0.01000	0.00010	0.10000	0.00000000009900
F	T	F	0.01000	0.99900	0.01000	0.01000	0.99000	0.00000098901000
F	T	T	0.01000	0.00100	0.90000	0.00010	0.10000	0.00000000009000

P(te|ae) = 0.00989011000000

- Similarly, we can compute P($\neg$te|ae) = 0.000999693198000001.

- Hence, P(te|ae) = α < 0.00989011, 0.000999693198000001> = <0.9082,0.0918> and our belief that te=TRUE given observation of ae has increased to 0.9082 from its prior probability of 0.00999.

Example of Inference

- We could have also calculated **P(ar|ae) = 0.0909** and **P(dh|ae) = 0.01** in a similar fashion.
- Consider now that we observe both **ae** = **TRUE and ah = TRUE.**
- A similar calculation would give that **P(te|ae,ah) = 0.0917** (and the summation would have been over only four terms), **P(ar|ae,ah)=0.908** and **P(dh|ae,ah)=0.0926**.
- Note that the observation of additional evidence **ah** tends to "**explain away**" things.
    - Once **both ae** and **ah** are **observed**, a **single cause ar** that can **explain both the observations becomes very likely**.

Constructing Belief Networks

- Only stated briefly…
- To construct a belief network, you need to consider several things:
  - What are the relevant variables, and what should the domains be? (Defines the nodes).
  - What is the relationship between the variables? (Defines the arcs).
  - How does the value of one variable depend on the variables that locally influence it (i.e., its parents)? (Defines the CPTs).
- Basically, should form an order to adding variables (and therefore edges) to the network. Begin with the root causes, and continue from there.

More on Understanding Conditional Independence

- Can understand dependence in a network by considering how evidence is transmitted in a belief network.

- Can consider several types of connections that exist in a network:

  - **Serial connection** – information entered at the beginning of the connection can be propagated down the connection provided there is no intermediate node has evidence (in which case we stop the propagation).

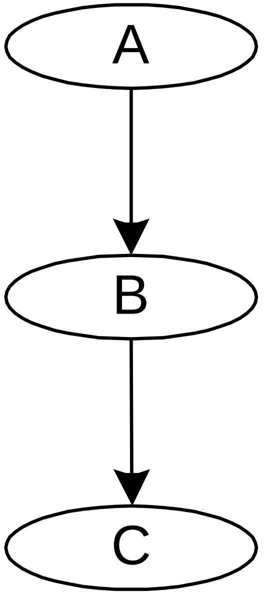
  - Diverging connection – information can be propagated between two child nodes provided the parent node has no evidence.

  - **Converging connection** – information can be propagated between parent nodes when the child node has evidence.

- Rules for these connections are sufficient to describe a complete procedure for determining if two nodes in a belief network are dependent or not. This is formally known as d-separation.

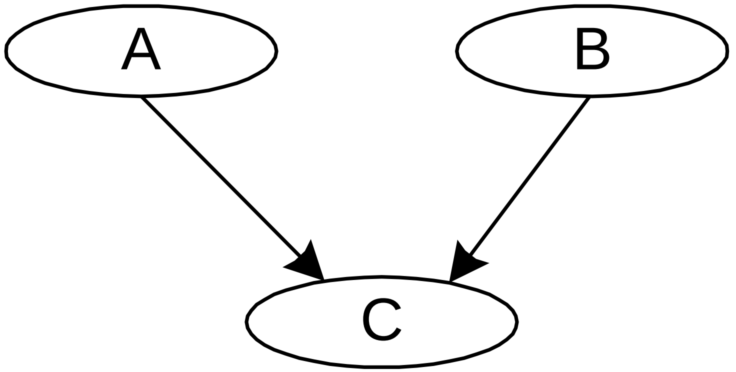
Serial Connection

- In the example below:
  - A and C are dependent.
  - A and C are independent given B (evidence for B).
  - Intuitively, the only way for A to affect C is by affecting B.



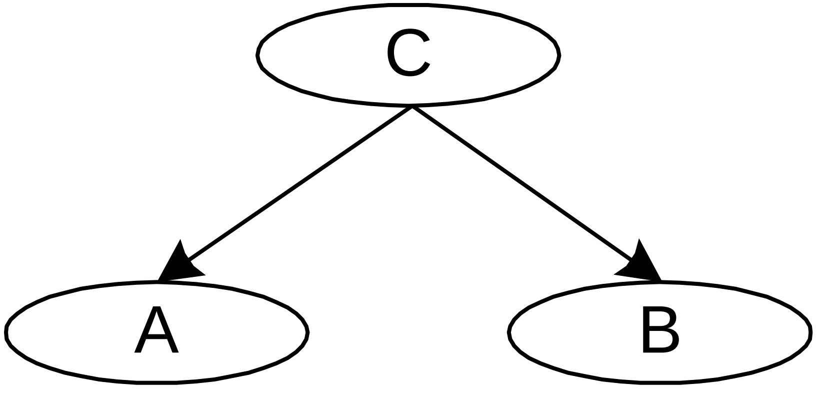
Converging Connection

- In the example below:
  - A and B are independent.
  - A and B are dependent given C.
  - Intuitively, A can be used to explain away B.



Diverging Connection

- In the example below:
  - A and B are dependent.
  - A and B are independent given C.
  - Intuitively, C can explain A and B, meaning one can only affect the other by changing the belief in C.



D-Separation

- X is d-separated from Y if, **for all paths** between X and Y **there exists** an intermediate node Z for which:
  - The connection is serial or diverging and there is evidence for Z.
  - The connection is converging and Z (nor any of its descendants) have received any evidence.

- X is independent of Y given Z for some conditional probabilities if and only if X is d-separated from Y given Z.

Example of Independence Questions

- If there was evidence for B, which probabilities would change?
- If there was evidence for N, which probabilities would change?
- If there was evidence for M and N, which variables probabilities would change?
- Etc…