Next Three-Four Lectures

• Bayesian probability
• Reasoning under uncertainty (Chapter 14)
  – Belief networks (reasoning only) (this lecture)
  – Exact inference
  – Next Lecture
    • Exact inference is NP-hard
    • Approximate inference using Gibbs sampler
• Later modules - reasoning in the presence of no uncertainty (propositional logic)
New Module – Reasoning In the Presence of Uncertainty

• Motivating example (earthquake)
• Note, this is not exhaustive search we are reasoning in the presence of uncertainty
• We assume the network/graph is given so we do not learn.
Ask Queries of Networks

• Two types of queries?
Bayesian Belief Networks

- Combination of probabilistic modeling and DAGs
- Nodes on graph are propositional variables.
- Links represent apriori known causal dependencies.
- Reasoning by merging semantic models and evidence.
- Efficient representation of joint distribution
  - Global semantic
Example: Car diagnosis

Initial evidence: car won’t start
Testable variables (green), “broken, so fix it” variables (orange)
Hidden variables (gray) ensure sparse structure, reduce parameters
Example: Car insurance : Predictive
Primer on Probability

\[ P(\text{Event}) = q \]

Various Interpretations of \( q \)
- Frequentist
- Degree of belief

- Going to learn some methods of manipulating probabilities.
Probability - 1

- Distributions
- Random variables
  - Discrete
    - We’ll cover various methods to manipulate probabilities to get them into a desirable form
  - Continuous
- Background state of information
Probability - II

- Discrete Random Variables
- Continuous Random Variables

Probability distribution (density function) over continuous values

\[ X \in [0, 10] \quad P(x) \geq 0 \]

\[ \int_{0}^{10} P(x) \, dx = 1 \]

\[ P(5 \leq x \leq 7) = \int_{5}^{7} P(x) \, dx \]
Probability - III

• Conditional Probabilities

• Joint Probabilities

• Product Rule

• Marginalization
Bayes Theorem

\[
P(h,D) = P(D|h) \cdot P(h) = P(D|h) \cdot P(h) \]

\[
P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}
\]

- \( P(h) \) = prior probability of hypothesis \( h \)
- \( P(D) \) = prior probability of training data \( D \)
- \( P(h|D) \) = probability of \( h \) given \( D \)
- \( P(D|h) \) = probability of \( D \) given \( h \)
About the Hypothesis Space $P(h)$

- Priors
  - Each $h_i$ should be *Mutually exclusive*
  - Together the hypotheses must be *Totally exhaustive*
  - $\sum P(h_i) = 1$
  - Priors encode knowledge before we see the data
About the Data $P(D)$ and $P(D|H)$

• **Data, $P(D)$**
  – Data is considered to be a sample of all available data.
  – $P(D)$, probability the data will be observed given no knowledge of the hypothesis.
  – Constant for fixed data and if comparing hypotheses, can be ignored

• **Likelihood, $P(D|h)$**
  – Probability a hypothesis generated the observed data or probability of observing data given the hypothesis is true.
  – If the $n$ instances are independent then
    • $P(D|h) = P(D_1|h). P(D_2|h) \ldots P(D_n|h)$
  – Often use the Loglikelihood ($P(D|h)$).
Bayesian Posterior

• $P(h|D)$ is the posterior probability of the hypothesis (given the data).

• Usual aim of Bayesian learning is to find the MAP estimate
  – Most probable model in the model space
  – May be many highly probable models
A Simple Example

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, .008 of the entire population have this cancer.

\[
P(\text{cancer}) = \quad P(\neg \text{cancer}) = \\
P(+) | \text{cancer} = \quad P(-) | \text{cancer} = \\
P(+) | \neg \text{cancer} = \quad P(-) | \neg \text{cancer} =
\]
Basic Rules of Probability

- **Product Rule**: probability $P(A \land B)$ of a conjunction of two events $A$ and $B$:

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

- **Sum Rule**: probability of a disjunction of two events $A$ and $B$:

  $$P(A \lor B) = P(A) + P(B) - P(A \land B)$$

- **Theorem of total probability**: if events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then

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

**Removing uncertainty**
If all conditional probabilities were close to a half …
Bayesian Belief Networks

• Combination of probabilistic modeling and DAGs
• Nodes on graph are propositional variables.
• Links represent apriori known causal dependencies.
• Reasoning by merging semantic models and evidence.
• Efficient representation of joint distribution
  – Global semantic
Direct World Representations

- Can compute any subset of propositions given another subset.
- Perform different types of reasoning
  - Prediction
  - Abduction
  - Explaining away
- Global semantics
- Local semantics exploit conditional independence
Methods of Manipulating Probabilities …
Let’s use them to work out a query
Reasoning with a Bayesian Net

• Reasoning without evidence
• Reasoning with evidence
• Bayesian network reasoning NP-Hard
  – Instance of propositional logic satisfiability problem
• Use Monte Carlo techniques to simulate draws from the joint distribution
Learning Networks

• Four situations
  – Structure known, All variables observed
    • Simple counting exercise!
  – Structure known, some variables unobserved
    • EM
  – Structure unknown, All variables observed
    • Can use BIC
  – Structure unknown, some variables unobserved
    • Structural EM

• Currently focus on finding best model, but will later focus on finding multiple models.
  – How? Why?
Structure Known

• Full Observability
  – Count to work out every conditional probability table stored at a node. Maximum likelihood est.
  – Use Laplace correction to stop zero probabilities

• Partial Observability
  – Postulate a hidden variable
  – E step : calculate expectation of hidden variables
    • How?
  – M step : Maximize likelihood like above.
Decision Making With B Nets

Factoring in **Utility** with belief network computations

Should I have my party inside or outside?

```
in  dry  Regret
    wet  Relieved
  out  dry  Perfect!
      wet  Disaster
```
Minimize Risk

choose the action that maximizes expected utility
Influence Diagrams

Call?
Neighbor Phoned Yes
No Phone Call No

Go Home?

Miss Meeting

Expected Utility of this policy is 100
Influence Diagrams

- Three types of nodes (typical shapes)
  - Chance nodes (oval)
  - Decision nodes (rectangles)
  - Utility nodes (diamonds)
- Belief networks reasoning in the presence of uncertainty
- Influence diagrams decision making in the presence of uncertainty
To Bet or Not Bet

This is a simple example that can be extended to more complex situations. I bet you $15 if your team wins otherwise you give me $15 if my team wins. Of course if my team wins I receive an additional reward beyond the $15 and vice versa. Do you accept the bet?

Weather

\[ P(W=\text{Dry}) = 0.3 \]

\[ P(\text{MTW}=\text{T}|W=\text{Dry})=0.25 \]

\[ P(\text{MTW}=\text{T}|W=\text{Wet})=0.60 \]

MyTeamWins

Util	ity

Bet?

<table>
<thead>
<tr>
<th>MTW</th>
<th>UBet</th>
<th>MUtility</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>40</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>20</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>5</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>15</td>
</tr>
</tbody>
</table>
Putting a Price on Information

- What is it worth to get another piece of information?
- What is the increase in (maximized) expected utility if I make a decision with an additional piece of information?
- Additional information (if free) cannot make you worse off.
- There is no value-of-information if you will not change your decision.
Dynamic Belief Networks (Same Random Variables Changing Over Time)

Previous Networks Snapshot of a Situation

- Markov property: Why?
  - past independent of future given current state;
  - a conditional independence assumption;
  - implied by fact that there are no arcs $t \rightarrow t+2$. 

$State(t) \rightarrow State(t+1) \rightarrow State(t+2)$
DBNet

- State described via random variables.
- Each variable depends only on few others.
Inference by stochastic simulation

Basic idea:
1) Draw $N$ samples from a sampling distribution $S$
2) Compute an approximate posterior probability $\hat{P}$
3) Show this converges to the true probability $P$

Outline:
- Sampling from an empty network
- Rejection sampling: reject samples disagreeing with evidence
- Likelihood weighting: use evidence to weight samples
- Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior
Sampling from an empty network

function PRIOR-SAMPLE(bn) returns an event sampled from bn
    inputs: bn, a belief network specifying joint distribution P(X₁, ..., Xₙ)
    x ← an event with n elements
    for i = 1 to n do
        xᵢ ← a random sample from P(Xᵢ | Parents(Xᵢ))
    return x
Limitations?
Rejection sampling

\( \hat{P}(X|e) \) estimated from samples agreeing with \( e \)

```plaintext
function Rejection-Sampling(X, e, bn, N) returns an estimate of \( P(X|e) \)
    local variables: N, a vector of counts over X, initially zero
    for j = 1 to N do
        x ← Prior-Sample(bn)
        if x is consistent with e then
            N[x] ← N[x] + 1 where x is the value of X in x
    return Normalize(N[X])
```

E.g., estimate \( P(Rain|Sprinkler = true) \) using 100 samples
27 samples have \( Sprinkler = true \)
Of these, 8 have \( Rain = true \) and 19 have \( Rain = false \).

\( \hat{P}(Rain|Sprinkler = true) = Normalize((8, 19)) = (0.296, 0.704) \)

Similar to a basic real-world empirical estimation procedure
Rejection Sampling as Integration

• Classic polygon problem
• In belief network, what are we integrating?
Analysis of rejection sampling

\[ \hat{P}(X|e) = \alpha N_{PS}(X, e) \quad (\text{algorithm defn.}) \]
\[ = N_{PS}(X, e)/N_{PS}(e) \quad (\text{normalized by } N_{PS}(e)) \]
\[ \approx P(X, e)/P(e) \quad (\text{property of PRIORSAMPLE}) \]
\[ = P(X|e) \quad (\text{defn. of conditional probability}) \]

Hence rejection sampling returns consistent posterior estimates.

Problem: hopelessly expensive if \( P(e) \) is small.

\( P(e) \) drops off exponentially with number of evidence variables!
Likelihood weighting

Idea: fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence

function Likelihood-Weighting(X, e, bn, N) returns an estimate of P(X|e)
local variables: W, a vector of weighted counts over X, initially zero
for j = 1 to N do
    x, w ← Weighted-Sample(bn)
    W[x] ← W[x] + w where x is the value of X in x
return Normalize(W[X])

function Weighted-Sample(bn, e) returns an event and a weight
x ← an event with n elements; w ← 1
for i = 1 to n do
    if X_i has a value x_i in e
    then w ← w × P(X_i = x_i | Parents(X_i))
    else x_i ← a random sample from P(X_i | Parents(X_i))
return x, w
Likelihood weighting example

\[ w = 1.0 \]
Likelihood weighting example

\[
P(C) = 0.50
\]

| C | P(S|C) |
|---|-------|
| T | 0.10  |
| F | 0.50  |

| C | P(R|C) |
|---|-------|
| T | 0.80  |
| F | 0.20  |

| S | R | P(W|S,R) |
|---|---|---------|
| T | T | 0.99    |
| T | F | 0.90    |
| F | T | 0.90    |
| F | F | 0.01    |


**Likelihood weighting example**

\[
P(C) = 0.50
\]

\[
\begin{array}{c|c}
C & P(S|C) \\
\hline
T & 0.10 \\
F & 0.50 \\
\end{array}
\]

\[
\begin{array}{c|c}
C & P(R|C) \\
\hline
T & 0.80 \\
F & 0.20 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
S & R & P(W|S,R) \\
\hline
T & T & 0.99 \\
T & F & 0.90 \\
F & T & 0.90 \\
F & F & 0.01 \\
\end{array}
\]

\[
w = 1.0
\]
Likelihood weighting example

\[ P(C) = 0.50 \]

\[
\begin{array}{c|c}
C & P(S|C) \\
\hline
T & 0.10 \\
F & 0.50 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
S & R & P(W|S,R) \\
\hline
T & T & 0.99 \\
T & F & 0.90 \\
F & T & 0.90 \\
F & F & 0.01 \\
\end{array}
\]

\[ w = 1.0 \times 0.1 \]
Likelihood weighting example

| C | P(S|C) |
|---|-------|
| T | .10   |
| F | .50   |

| C | P(R|C) |
|---|-------|
| T | .80   |
| F | .20   |

\[ w = 1.0 \times 0.1 \times 0.99 = 0.099 \]
Some more examples

- Farming, Microsoft Trouble Shooters etc.
- Most are diagnosis based.
- Useful when we want to reason in the presence of uncertainty. Next lectures. Chapter 7, reasoning in the presence of certainty using propositional logic.
“A Belief Network is A Model of a Real World Situation Car Diagnosis”

What directly recordable pieces of information do we have for car diagnosis?
“A Belief Network is A Model of a Real World Situation Car Diagnosis”

What directly recordable pieces of information do we have for car diagnosis? Battery age, battery dead, lights on, oil light, gas gauge, engine won’t Start

Great. Let's draw the graph! How can we do diagnosis?
“A Belief Network is A Model of a Real World Situation Car Diagnosis”

What directly recordable pieces of information do we have for car diagnosis? Battery age, battery dead, lights on, oil light, gas gauge, engine won’t Start

But is this enough for diagnosis? What else do we need?
“A Belief Network is A Model of a Real World Situation
Car Diagnosis”

What directly recordable pieces of information do we have for car
diagnosis? Battery age, battery dead, lights on, oil light, gas gauge,
engine won’t Start

But is this enough for diagnosis?
What else do we need?

Latent variables!

Latent Causes
Alternator broken, fanbelt broken, no oil, no gas, fuel line blocked,
Starter broken. Is this enough?

Lets draw the graph
“A Belief Network is A Model of a Real World Situation Car Diagnosis”

What directly recordable pieces of information do we have for car diagnosis? Battery age, battery dead, lights on, oil light, gas gauge, engine won’t Start

But is this enough for diagnosis? What else do we need?

Latent variables!

Latent Causes
Alternator broken, fanbelt broken, no oil, no gas, fuel line blocked, Starter broken

Latent recordable pieces of information: no charging, battery flat
Example: Car diagnosis

Initial evidence: car won’t start
Testable variables (green), “broken, so fix it” variables (orange)
Hidden variables (gray) ensure sparse structure, reduce parameters
So What Queries Can We ask

• Typically abduction. Such as …

• Prediction

• Explaining away
So What Queries Can We ask

• Typically abduction. Such as …
• \( P(\text{NoOil} = T \mid \text{Lights} = F, \text{OilLight}=F, \text{Engine Won’t Start}) \). Do this for all causes why…
• Prediction

• Explaining away
• \( P(\text{Alternator Replace} = T \mid \text{Batt Dead} = T, \text{Oil Light} = T) \)
Example: Car insurance

: Predictive
How Do I Work Out The Best Graph Structure

- Covered in 635 … but quickly.
- Use Bayesian Information Criterion. Trades of graph complexity against fit to the data.
- BIC: $-\log(P(D|G) + n\text{Params}/2 + \log(|D|)$
Compact conditional distributions contd.

Noisy-OR distributions model multiple noninteracting causes
1) Parents $U_1 \ldots U_k$ include all causes (can add leak node)
2) Independent failure probability $q_i$ for each cause alone
   \[ P(X | U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = 1 - \prod_{i=1}^{j} q_i \]

<table>
<thead>
<tr>
<th>Cold</th>
<th>Flu</th>
<th>Malaria</th>
<th>$P(\text{Fever})$</th>
<th>$P(\neg \text{Fever})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>0.98</td>
<td>$0.02 = 0.2 \times 0.1$</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>0.94</td>
<td>$0.06 = 0.6 \times 0.1$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>0.88</td>
<td>$0.12 = 0.6 \times 0.2$</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>0.988</td>
<td>$0.012 = 0.6 \times 0.2 \times 0.1$</td>
</tr>
</tbody>
</table>

Number of parameters linear in number of parents
Multiply connected networks:
- can reduce 3SAT to exact inference $\Rightarrow$ NP-hard
- equivalent to counting 3SAT models $\Rightarrow$ #P-complete

1. $A \lor B \lor C$
2. $C \lor D \lor \neg A$
3. $B \lor C \lor \neg D$
HMM Algorithms Applied to DBNET

- HMMs are just very simple DBNs.
- Standard inference & learning algorithms for HMMs are instances of DBN algorithms
  - Forward-backward = polytree
  - Baum-Welch = EM
  - Viterbi = most probable explanation.

*Used extensively as many problems are temporal and have latent states, i.e. voice recognition,*
Word recognition example (1).

- Typed word recognition, assume all characters are separated.

- Character recognizer outputs probability of the image being particular character, $P(\text{image}|\text{character})$.

![Diagram with hidden state and observation]
Two states: ‘Rain’ and ‘Dry’.
Transition probabilities: $P(\text{Rain} | \text{Rain}) = 0.3$, $P(\text{Dry} | \text{Rain}) = 0.7$, $P(\text{Rain} | \text{Dry}) = 0.2$, $P(\text{Dry} | \text{Dry}) = 0.8$
Initial probabilities: say $P(\text{Rain}) = 0.4$, $P(\text{Dry}) = 0.6$

Let’s do a simulation to see the stationary behavior

Stationary distribution over the states ???
Stationary distribution over the states ???

- Two states: ‘Rain’ and ‘Dry’.
- Transition probabilities: \( P(\text{‘Rain’|‘Rain’})=0.3 \), \( P(\text{‘Dry’|‘Rain’})=0.7 \), \( P(\text{‘Rain’|‘Dry’})=0.2 \), \( P(\text{‘Dry’|‘Dry’})=0.8 \).
- Initial probabilities: say \( P(\text{‘Rain’})=0.4 \), \( P(\text{‘Dry’})=0.6 \).

\[
0.3 \cdot P(\text{Rain}) + 0.2 \cdot P(\text{Dry}) = P(\text{Rain})
\]
\[
0.8 \cdot P(\text{Dry}) + 0.7 \cdot P(\text{Rain}) = P(\text{Dry})
\]
\[
P(\text{Rain}) = \frac{2}{7} \cdot P(\text{Dry}) \quad \text{and} \quad P(\text{Rain}) + P(\text{Dry}) = 1
\]
\[
P(\text{Dry}) = \frac{7}{9} \quad \text{and} \quad P(\text{Rain}) = \frac{2}{9}
\]
Example of Hidden Markov Model

Evaluation Problem: \( P(“RDRDD”) \)
Use marginal probabilities?
Decoding Problem

Trellis representation of an HMM

\[ \begin{align*}
O_1 & \quad \ldots \quad O_k & \quad O_{k+1} & \quad O_K = \text{Observations} \\
S_1 & \quad \ldots \quad S_1 & \quad S_1 & \quad S_1 \\
S_2 & \quad \ldots \quad S_2 & \quad S_2 & \quad S_2 \\
S_i & \quad \ldots \quad S_i & \quad S_j & \quad S_i \\
S_N & \quad \ldots \quad S_N & \quad S_N & \quad S_N \\
\end{align*} \]

Time = 1 \quad k \quad k+1 \quad K

CSI 535 - Lecture 12
The Markov chain

With $Sprinkler = true, WetGrass = true$, there are four states:

Wander about for a while, average what you see
MCMC

• Stochastic process that creates a stationary probability distribution over the states.
• What distribution are we interested in?
• What computation are we performing???
• Metropolis Hastings Algorithm
• Gibbs Sampler
• Issues with MCMC
  – Warm up period
  – Visually verify convergence
Gibbs Sampling

• Update “component” of a system randomly:
  \[ P(X^t_i \mid X^{t-1}_{i-1}, X^{t-1}_{i+1}, \ldots, X^{t-1}_n) \]
• For belief networks due to conditional independence reduces to:
  \[ = P(X^t_i \mid PA^{t-1}_i) \]
• Simulation produces state combinations of the real-world accordingly to how likely they are.
Behavior Over Time

Initial State: TFT

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.6</td>
<td></td>
</tr>
</tbody>
</table>
Two Chains that Will Converge but with Differing Efficiencies

\[ \pi(A) = 0.5 \]

\[ \pi(B) = 0.2 \]

\[ \pi(C) = 0.3 \]

Figure 6-1: Simple three state system with posterior probabilities.

Let us suppose two Markov chains have been constructed which have converged to the stationary (posterior) distribution. If we were to obtain the sequence of states visited by the chains they could be:


and

A, B, A, B, C, A, C, A, C, A
Figuratively What is Going On?
Dynamics of Gibbs and Metropolis Samplers

- Motivating simple two dimensional case.
- Dynamics of movement
- Mixing, poor mixing, Multi-modality, stickiness, sensitivity to initial conditions
Measuring Convergence (1)

FFF

TTT

FTF

CSI 535 - Lecture 12
Measuring Convergence (3)

Start State

<table>
<thead>
<tr>
<th>Transition</th>
<th>FFF</th>
<th>TTT</th>
<th>TFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>0.451537</td>
<td>0.439024</td>
<td>0.453196</td>
</tr>
<tr>
<td>FT</td>
<td>0.548463</td>
<td>0.560976</td>
<td>0.546804</td>
</tr>
<tr>
<td>TF</td>
<td>0.46875</td>
<td>0.431421</td>
<td>0.476684</td>
</tr>
<tr>
<td>TT</td>
<td>0.53125</td>
<td>0.568579</td>
<td>0.523316</td>
</tr>
</tbody>
</table>

Marginal Probability of $X_2$ Over Various Time Periods

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.668182</td>
<td>0.467614</td>
<td>0.617045</td>
</tr>
</tbody>
</table>

![Graph](image-url)
Now onto reasoning in the presence of certainty using propositional and first order logic

Note, not a course on logic,
Just how logic is used in A.I.
Knowledge Based Agents

Knowledge base = set of sentences in a **formal** language

**Declarative** approach to building an agent (or other system):

*TELL* it what it needs to know

Then it can *ASK* itself what to do—answers should follow from the KB

Agents can be viewed at the **knowledge level**

i.e., what they know, regardless of how implemented

Or at the **implementation level**

i.e., data structures in KB and algorithms that manipulate them
Lets Compare Reasoning with Uncertainty and Reasoning with Certainty

Representation of knowledge,
What we know
Explore other issues later on …
Logics at High Level

Logics are formal languages for representing information such that conclusions can be drawn.

Syntax defines the sentences in the language.

Semantics define the “meaning” of sentences; i.e., define truth of a sentence in a world.

E.g., the language of arithmetic:

\[ x + 2 \geq y \] is a sentence; \[ x2 + y > \] is not a sentence.

\[ x + 2 \geq y \] is true iff the number \( x + 2 \) is no less than the number \( y \).

\[ x + 2 \geq y \] is true in a world where \( x = 7, \ y = 1 \).

\[ x + 2 \geq y \] is false in a world where \( x = 0, \ y = 6 \).
Notion of a Model

Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated.

We say $m$ is a model of a sentence $\alpha$ if $\alpha$ is true in $m$.

$M(\alpha)$ is the set of all models of $\alpha$.

Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$.

E.g. $KB = $ Giants won and Reds won

$\alpha = $ Giants won
Inference

$KB \vdash_i \alpha = $ sentence $\alpha$ can be derived from $KB$ by procedure $i$

**Soundness:** $i$ is sound if whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$

**Completeness:** $i$ is complete if whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$
Logic Types

Logics are characterized by what they commit to as “primitives”


Epistemological commitment: what states of knowledge?

<table>
<thead>
<tr>
<th>Language</th>
<th>Ontological Commitment</th>
<th>Epistemological Commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional logic</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>First-order logic</td>
<td>facts, objects, relations</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Temporal logic</td>
<td>facts, objects, relations, times</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Probability theory</td>
<td>facts</td>
<td>degree of belief 0…1</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>degree of truth</td>
<td>degree of belief 0…1</td>
</tr>
</tbody>
</table>
Propositional Logic: Syntax

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols $P_1$, $P_2$ etc are sentences

If $S$ is a sentence, $\neg S$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \land S_2$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \lor S_2$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \Rightarrow S_2$ is a sentence

If $S_1$ and $S_2$ is a sentence, $S_1 \Leftrightarrow S_2$ is a sentence
Propositional Logic: Semantics

Each model specifies true/false for each proposition symbol

E.g. \( A \quad B \quad C \)
\( \text{True True False} \)

Rules for evaluating truth with respect to a model \( m \):

\[
\begin{align*}
\neg S & \quad \text{is true iff} & S & \quad \text{is false} \\
S_1 \land S_2 & \quad \text{is true iff} & S_1 & \quad \text{is true and} & S_2 & \quad \text{is true} \\
S_1 \lor S_2 & \quad \text{is true iff} & S_1 & \quad \text{is true or} & S_2 & \quad \text{is true} \\
S_1 \Rightarrow S_2 & \quad \text{is true iff} & S_1 & \quad \text{is false or} & S_2 & \quad \text{is true} \\
\text{i.e., is false iff} & & S_1 & \quad \text{is true and} & S_2 & \quad \text{is false} \\
S_1 \Leftrightarrow S_2 & \quad \text{is true iff} & S_1 \Rightarrow S_2 & \quad \text{is true and} & S_2 \Rightarrow S_1 & \quad \text{is true}
\end{align*}
\]
Inference via Enumeration

Let $\alpha = A \lor B$ and $KB = (A \lor C) \land (B \lor \neg C)$

Is it the case that $KB \models \alpha$?
Check all possible models—$\alpha$ must be true wherever $KB$ is true

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$A \lor C$</th>
<th>$B \lor \neg C$</th>
<th>$KB$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>