Announcements

- Coding Homework 2 will be released
  - Due 2/8 at 11:59pm
- Written Homework 2 will be released
  - Due 2/8 at 11:59pm

Propositional Logic

Forest Agostinelli
University of South Carolina

Topics Covered in This Class

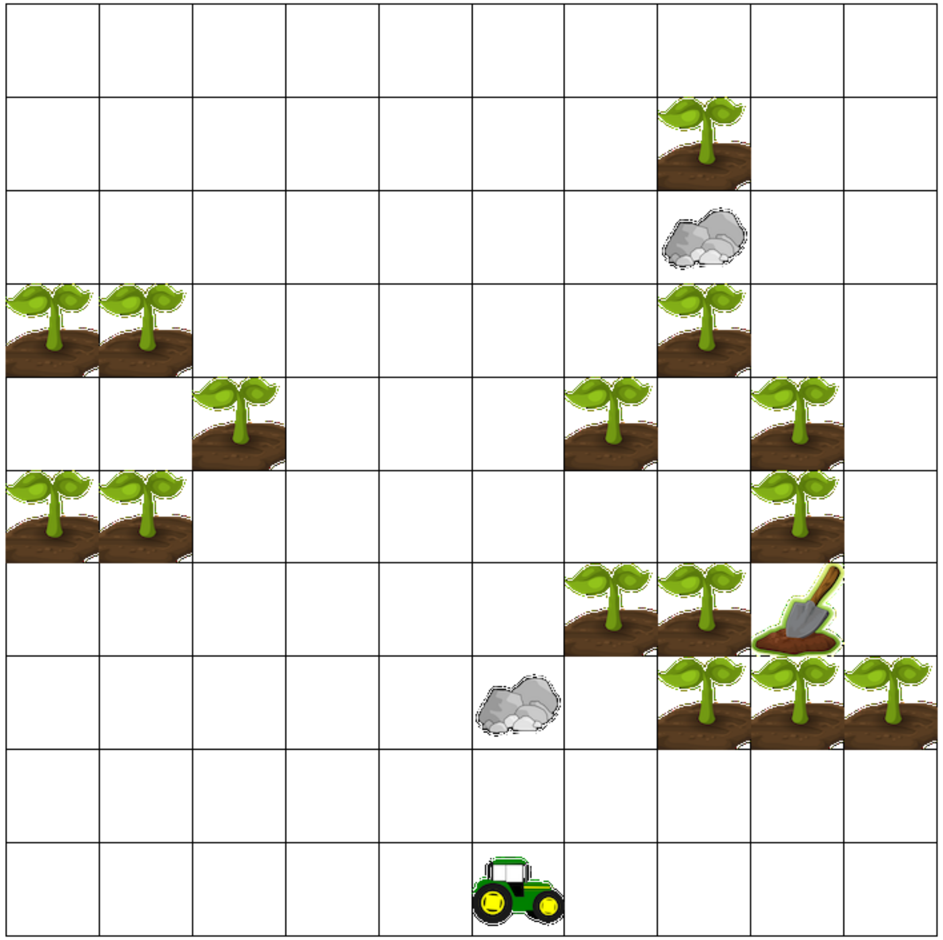
- **Part 1: Search**
  - Pathfinding
    - Uninformed search
    - Informed search
  - Adversarial search
  - Optimization
    - Local search
    - Constraint satisfaction
- **Part 2: Knowledge Representation and Reasoning**
  - Propositional logic
  - First-order logic
  - Prolog
- **Part 3: Knowledge Representation and Reasoning Under Uncertainty**
  - Probability
  - Bayesian networks
- **Part 4: Machine Learning**
  - Supervised learning
    - Inductive logic programming
    - Linear models
    - Deep neural networks
    - PyTorch
  - Reinforcement learning
    - Markov decision processes
    - Dynamic programming
    - Model-free RL
  - Unsupervised learning
    - Clustering
    - Autoencoders

Outline

- Background
- Propositional logic
  - Logical connectives
  - Sentences
- Entailment
  - Model-checking
  - Using sound inference rules
  - Proof by contradiction
    - Resolution
    - Logical equivalences
    - Conjunctive normal form
    - Unicorn proof
- Forward chaining and backward chaining

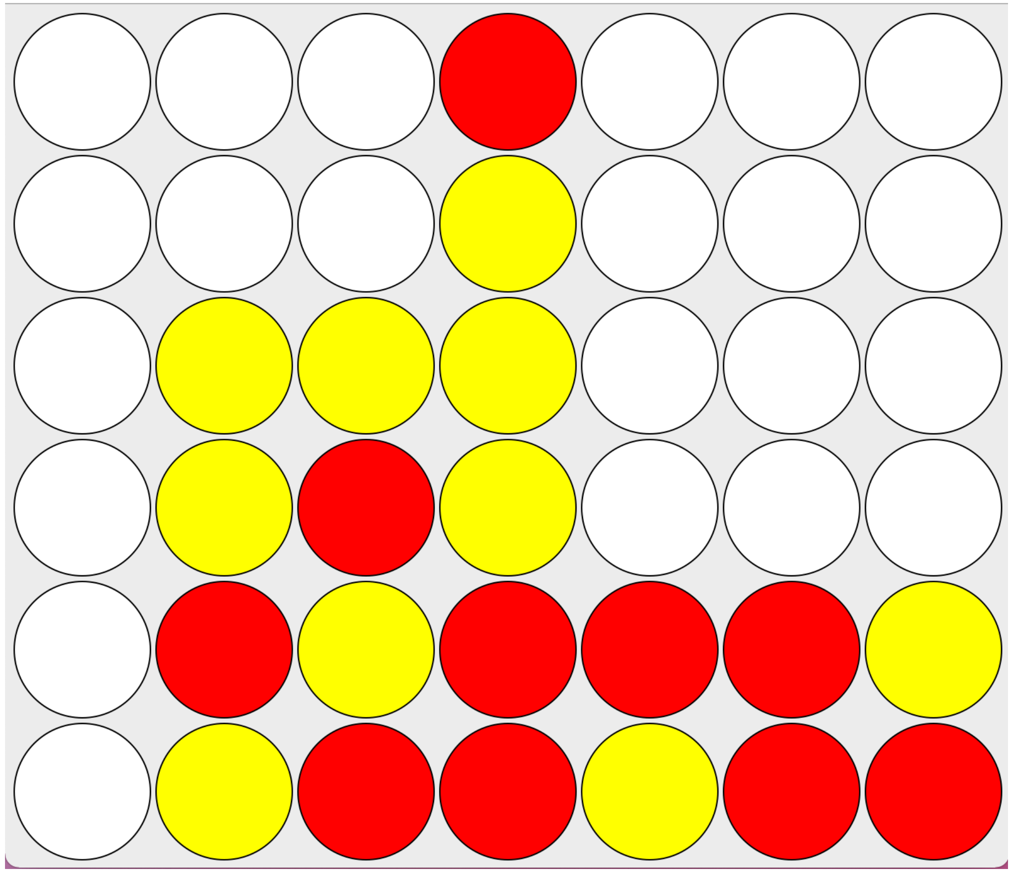
Motivation: Pathfinding

- We can use A* search to find a sequence of actions to go from one state to another
- However, we cannot ask the agent
  - How many plants are there?
  - Will every plant be able to get sun?



Motivation: Game Playing

- Can play games well
- However, we cannot ask the agent
  - Can one circle have both a red and yellow piece?
  - Which columns have effectively been blocked off for red?



Logic

- It is possible for us to tell an agent facts about the world
  - A plant should be watered every day
  - A plant is considered watered when a sufficient amount water is applied to its roots
  - Rain is water failing from the sky
- It is possible for an agent to perceive facts about the world
  - It is raining
- Using inference, it is possible for an agent to derive new facts about the world from old facts
  - Because it is raining, the plants have been watered for the day
- We can use these new facts to help us make decisions
  - I will check the soil moisture tomorrow to know exactly how much water the plants got

Knowledge-Based Agents

- Knowledge base
  - A set of logical sentences
    - A tomato is a fruit
    - A carrot is a vegetable
  - Sentences that are taken to be true without being inferred from other sentences are called axioms
- One should be able to add new sentences to the knowledge base
  - You can grow fruits and vegetables
- One should be able to ask the knowledge base questions. When the answer to the question is not explicitly contained in the knowledge base, the agent derive new sentences using inference
  - Is a tomato a fruit?
  - Can you grow a tomato?
  - Can you grow a carrot?

Inference

- Deriving new sentences from existing ones
- Types of inference
  - Deduction
  - Induction
  - Abduction

Deduction

- Inferring conclusions from premises that are assumed to be true
- Truth preserving
  - If the premises are assumed to be true, and the inference rules used are valid, inferred sentences are true with absolute certainty
- Socrates is human. All humans are mortal. Therefore, Socrates is mortal.
- **The focus of the upcoming lectures**

Induction

- Inferring general principles from observations
- Not truth preserving
  - What is inferred is **not** true with absolute certainty
- The sun has risen every day of my life. Therefore, the sun rises every day.
- **The focus of the machine learning lectures**

Abduction

- Inferring the most likely explanation from a set of observations
- Not truth preserving
  - What is inferred is **not** true with absolute certainty
- All humans are mortal. All dogs are mortal. Socrates is mortal. Therefore, Socrates is human.
  - Socrates being human is not the only explanation for Socrates being mortal. For example, Socrates could be the name of a dog.

Defining Logics: Ontology

- Concerned with the nature of being or existence
- An agent should know what exists and what could possibly be brought into existence
- Does not say anything about what can be known about what exists

Defining Logics: Epistemology

- Concerned with knowledge
- Given facts about the world, an agent needs to know what can be believed about these facts
  - Perhaps facts can only be true or false
  - Perhaps facts can be probably true
  - Perhaps facts can be kind of true

Formal Logic Languages

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value

- Propositional logic: concrete statements that are either true or false
  - The tomato is red
- First order logic: Allows statements to contain variables, functions, and quantifiers
  - For all X: If X is a tomato, then X is a plant
- Probability: Statements are possibly true
  - Given the color and texture of the tomato, what are the chances it is ripe?
- Fuzzy logic: Statements have some degree of truth
  - The tomato is kind of red
  - The sky is very cloudy

Outline

- Background
- Propositional logic
    - Logical connectives
    - Sentences
- Entailment
    - Model-checking
    - Using sound inference rules
    - Proof by contradiction
        - Resolution
        - Logical equivalences
        - Conjunctive normal form
        - Unicorn proof
- Forward chaining and backward chaining

Propositional Logic

- Symbols are propositions which can be true of false
- Logical sentences are made of symbols connected by logical connectives
- Semantics defines how a model relates to the truth of a sentence

Connectives

- Symbols: P and Q
- Connectives: Negation, conjunction, disjunction, implication, biconditional

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Grammar of Sentences

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & True \mid False \mid P \mid Q \mid R \mid \ldots \\
ComplexSentence & \rightarrow & (\; Sentence \;) \\
& \mid & \neg\, Sentence \\
& \mid & Sentence \land Sentence \\
& \mid & Sentence \lor Sentence \\
& \mid & Sentence \;\Rightarrow\; Sentence \\
& \mid & Sentence \;\Leftrightarrow\; Sentence \\
\\
\text{OPERATOR PRECEDENCE} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Figure 7.7 A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

Implication

- $P \rightarrow Q$
  - "P implies Q"
  - "If P then Q"
  - $\neg P \lor Q$
- In other words: If P is true then I am claiming Q is true, otherwise, I am making no claim
- When the antecedent is false, the entire statement is true. Also known as being vacuously true.
  - If 5 is even, then there is a kangaroo in my room
    - 5 is even -> a kangaroo is in my room

Biconditional

- $P \leftrightarrow Q$
  - “P if and only if Q”
  - “P iff Q”
  - $(P \rightarrow Q) \land (Q \rightarrow P)$

Sentences and Models

- A sentence represents some assertion about the world
  - $(p \wedge q) \rightarrow r$
- A model is an assignment to p, q, and r.
  - The assignment can be either true (T) or false (F)
- If a sentence α is true in model m, then m satisfies α
  - p is T, q is T, r is T
  - p is F, q and r can be assigned to anything
  - q is F, p and r can be assigned to anything
- $M(\alpha)$ is the set of all models that satisfy α

Satisfiability

- A sentence is satisfiable if there exists a model that satisfies it
  - $(p \wedge q) \rightarrow r$
- A sentence is unsatisfiable if there exists no model that satisfies it
  - $p \wedge \neg p$
- The constraint satisfaction problems that we saw are satisfiability (SAT) problems
- Satisfiability is an NP-complete problem

Validity

- A sentence is valid if it is true in all models
  - Also known as a tautology
  - $p \vee \neg p$
- Because they are true in all models, valid sentences are logically equivalent to True

Quick Quiz: Find Models that Satisfy these Sentences

- ~(A ^ B) -> C
- A ^ (B v C)
- A v (B ^ C)
- A v ~A
- A ^ ~A

Outline

- Background
- Propositional logic
  - Logical connectives
  - Sentences
- Entailment
  - Model-checking
  - Using sound inference rules
  - Proof by contradiction
    - Resolution
    - Logical equivalences
    - Conjunctive normal form
    - Unicorn proof
- Forward chaining and backward chaining

Entailment

- The idea that sentence α follows logically from sentence β
  - $\alpha \vDash \beta$
- $\alpha \vDash \beta$ if and only if every model in which α is true β is also true
- In other words, $M(\alpha) \subseteq M(\beta)$
- p entails $p \vee q$
  - However, $p \vee q$ **does not** entail p
- We use entailment to do deductive inference

The Deduction Theorem

- $\alpha \vDash \beta$ if and only if every model in which α is true β is also true
  - $M(\alpha) \subseteq M(\beta)$
- **Deduction theorem**
  - For any sentences α and β $\alpha \vDash \beta$ if and only if the sentence $\alpha \rightarrow \beta$ is valid
- Therefore, we can decide if $\alpha \vDash \beta$ by checking if $\alpha \rightarrow \beta$ in every model

Deductive Inference

- We are only interested in sound (truth preserving) inference algorithms
- We are also interested in complete inference algorithms
- **Sound**
  - An inference algorithm is sound if it **only** derives entailed sentences
- **Complete**
  - An inference algorithm is complete if it can derive **every** entailed sentence

Entailment using Model Checking

- By the deduction theorem
  - For any sentences α and β $\alpha \vDash \beta$ if and only if the sentence $\alpha \rightarrow \beta$ is valid
  - Therefore, we can decide if $\alpha \vDash \beta$ by checking if $\alpha \rightarrow \beta$ in every model
- If we want to know if $\mathrm{KB} \vDash \alpha$, then we can enumerate all possible models of the world and check that α is true in every model where KB is true
  - Also called the truth table method

Model Checking

- KB={p} logically entails α = p v q

p	q	p	$p \lor q$
1	1	1	1
1	0	1	1
0	1	0	1
0	0	0	0

Model Checking

- KB={p} does not logically entail α = p ^ q

p	q	p	$p \wedge q$
1	1	1	1
1	0	1	0
0	1	0	0
0	0	0	0

Model Checking

- KB={p, q} logically entails α = p ^ q

p	q	p	q	$p \wedge q$
1	1	1	1	1
1	0	1	0	0
0	1	0	1	0
0	0	0	0	0

Quick Quiz: Model Checking

- Does KB={$m \rightarrow p$ v q, $p \rightarrow q$} entail $\alpha = m \rightarrow q$?
- If it is Monday, then we are eating pizza or quesadillas
- If we are eating pizza, then we are eating quesadillas
- Does this entail that if it is Monday then we are eating quesadillas?

m	p	q	$m \Rightarrow p \vee q$	$p \Rightarrow q$	$m \Rightarrow q$
1	1	1	1	1	1
1	1	0	1	0	0
1	0	1	1	1	1
1	0	0	0	1	0
0	1	1	1	1	1
0	1	0	1	0	1
0	0	1	1	1	1
0	0	0	1	1	1

Model Checking

- Complete
- Sound
- If there are n symbols, then the time complexity is $O(2^n)$

Outline

- Background
- Propositional logic
  - Logical connectives
  - Sentences
- Entailment
  - Model-checking
  - Using sound inference rules
  - Proof by contradiction
    - Resolution
    - Logical equivalences
    - Conjunctive normal form
    - Unicorn proof
- Forward chaining and backward chaining

Entailment as a Search Problem

- State: The knowledge base
- Actions: Sound inference rules
- Transition function: Adding inferred sentence to knowledge base
- Goal state: When the knowledge base contains the sentence we would like to prove

Sound Inference

- We are only interested in sound (truth preserving) inference rules
- We are also interested in inferences rules yield complete search algorithms
- **Sound**
  - An inference algorithm is sound if it **only** derives entailed sentences
  - Model checking is sound
- **Complete**
  - An inference algorithm is complete if it can derive **every** entailed sentence
  - Model checking is complete

Sound Inference Rules

- Implication: $\frac{\alpha \to \beta,\ \alpha}{\beta}$
  - If the cat is hungry then the cat meows. The cat is hungry. Therefore, the cat is meowing.
- Contrapositive: $\frac{\alpha \to \beta,\ \neg\beta}{\neg\alpha}$
  - If the cat is hungry then the cat meows. The cat is not meowing. Therefore, the cat is not hungry.
- And elimination
  - $\frac{\alpha \land \beta}{\beta}$
  - We cooked rice and beans. Therefore, we cooked rice.
- We can use these inference rules as actions to search for a sentence we would like to infer

Prove Using Inference Rules

- Inference rules: $\frac{\alpha\rightarrow\beta,\ \alpha}{\beta}$, $\frac{\alpha\rightarrow\beta,\ \neg\beta}{\neg\alpha}$, $\frac{\alpha\wedge\beta}{\beta}$
- Knowledge Base
  - $Y \rightarrow \neg R$
  - $(\neg Y \rightarrow Q) \wedge B$
  - $\neg Q$
- Prove
  - $\neg R$

- $\frac{(\neg Y\rightarrow Q)\wedge B}{\neg Y\rightarrow Q}$ by and elimination
- $\frac{\neg Y\rightarrow Q, \neg Q}{Y}$ by contraposition
- $\frac{Y\rightarrow\neg R,\ Y}{\neg R}$ by implication

Prove Using Inference Rules

- Inference rules: $\frac{\alpha \rightarrow \beta,\, \alpha}{\beta}$, $\frac{\alpha \rightarrow \beta,\, \neg\beta}{\neg\alpha}$, $\frac{\alpha \land \beta}{\beta}$
- Knowledge Base
  - $M \rightarrow (\mathrm{P} \lor \mathrm{Q})$
  - $\mathrm{P} \rightarrow \mathrm{Q}$
- Prove
  - $M \rightarrow Q$
- Cannot be proven!
  - But, we know that, given the knowledge base, $M \rightarrow Q$
  - What can we say about our search algorithm that only uses these inference rules?

Complete Inference?

- We can prove entailment using the aforementioned inference rules for a limited number of cases
- What we can prove is limited by the inference rules that we have available to us
- Using a single inference rule, resolution and proof by contradiction, we will have a complete inference algorithm when coupled with any search algorithm

Outline

- Background
- Propositional logic
  - Logical connectives
  - Sentences
- Entailment
  - Model-checking
  - Using sound inference rules
  - Proof by contradiction
    - Resolution
    - Logical equivalences
    - Conjunctive normal form
    - Unicorn proof
- Forward chaining and backward chaining

Proof By Contradiction

- $KB \vDash \alpha$ iff the sentence $\mathrm{KB} \rightarrow \alpha$ is valid

- A sentence β is valid iff $\neg\beta$ is unsatisfiable
- $KB \vDash \alpha$ iff $KB \land \neg\alpha$ is unsatisfiable
  - Reductio ad absurdum (reduction to an absurd thing)
  - Proof by refutation or proof by contradiction
  - Resolution is both sound and refutation complete

Resolution

- Given a conjunction of two clauses, if a literal in one clause is the complement of a literal in another, we remove these literals and combine all the other literals with disjunctions
- Clause
  - A disjunction of literals
  - A
  - $A \vee \neg B$
  - $B \vee \neg C \vee \neg D$
- Literal
  - In propositional logic: A True, False, a symbol, or a negated symbol
- Example
  - Clause 1: $A \vee B$
  - Clause 2: $\neg B \vee C$
  - Resolution: $A \vee C$
- Note:
  - We can only resolve one pair of literals!
  - If more than one pair of complementary literals, then it is always equal to true!

Resolution: English Examples

- R = I am going to read
- P = I am going to play the piano
- English example 1
  - R or P
  - not R
  - Therefore, P
- English example 2
  - R or P
  - not R or not P
  - Therefore, true (tautology)

Resolution Examples

- $\frac{A \vee B, \neg B}{A}$
- $\frac{A \vee B \vee C, \neg B}{A \vee C}$
- $\frac{A \vee B, \neg B \vee C}{A \vee C}$
- $\frac{A \vee B \vee C, \neg A \vee D \vee E}{B \vee C \vee D \vee E}$
- $\frac{A \vee B, A \vee \neg B}{A \vee A \equiv A}$
  - Note the simplification
- $\frac{A \vee B \vee C, \neg A \vee \neg B \vee C}{A \vee \neg A \vee C} \equiv True$

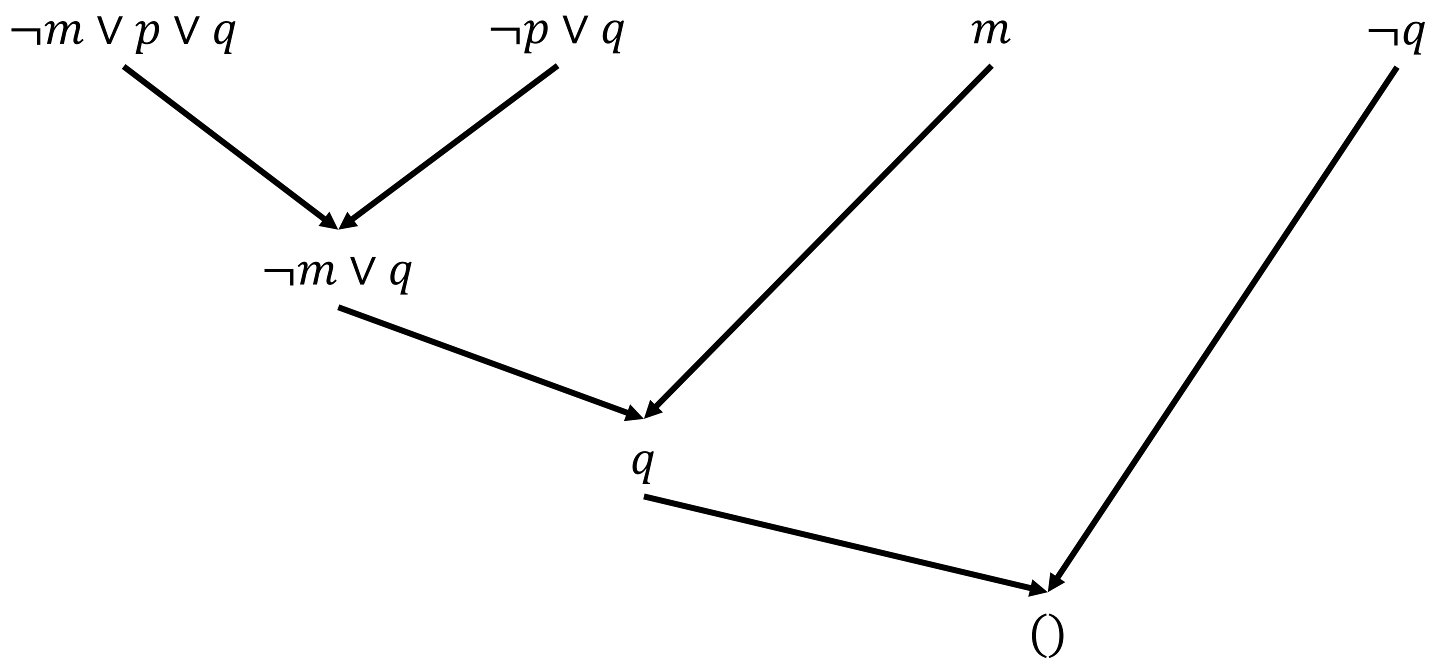
Proof by Contradiction with Resolution

- Problem
  - KB: $\left(m \rightarrow (p \vee q)\right) \wedge (p \rightarrow q)$
  - Prove: $m \rightarrow q$
- Convert into logically equivalent conjunction of clauses so that we can do resolution
  - KB: $(\neg m \vee p \vee q) \wedge (\neg p \vee q)$
  - Prove: $\neg m \vee q$
- Negate $\neg m \vee q$ (what we want to prove)
  - $m \wedge \neg q$

Proof By Resolution

- Show that $KB \land \neg\alpha$ is unsatisfiable
  - KB = $(\neg m \lor p \lor q) \land (\neg p \lor q)$
  - $\alpha = \neg m \lor q$
  - $\neg\alpha = m \land \neg q$

- Implicit conjunction over all clauses
  - Including ones derived using resolution
- Therefore, one contradiction means it is unsatisfiable



Logical Equivalence

- How did we get these logical equivalences?
  - $\big(m \rightarrow (p \vee q)\big) \wedge (p \rightarrow q) \equiv (\neg m \vee p \vee q) \wedge (\neg p \vee q)$
  - $m \rightarrow q \equiv \neg m \vee q$
  - $\neg(\neg m \vee q) \equiv m \wedge \neg q$

Logical Equivalence

- Two sentences are logically equivalent if they are true in the same set of models
- $\alpha \equiv \beta$ iff $\alpha \vDash \beta$ and $\beta \vDash \alpha$
- We can use this definition to rewrite sentences
- $\alpha \land \beta \equiv \beta \land \alpha$
- $\alpha \rightarrow \beta \equiv \neg\alpha \lor \beta$
- $\alpha \rightarrow \beta \equiv \neg\beta \rightarrow \neg\alpha$

- How can we be sure of these equivalences?
  - Truth tables!
  - Using the definition of equivalence, we can check if they are true in all models
  - We can then use this to move past truth tables

Logical Equivalence

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) \quad \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) \quad \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

Conjunctive Normal Form (CNF)

- A conjunction of clauses
  - $(A \lor \neg B) \land (B \lor \neg C \lor \neg D)$
- Every sentence of propositional logic is logically equivalent to a conjunction of clauses (aka a conjunction of disjunctions)
  - Therefore, any KB can be converted into CNF
  - We will use logical equivalences to help us do this

Steps to Convert to CNF

- Eliminate Biconditionals
  - Biconditional elimination
- Eliminate implications
  - Implication elimination
- Push negations inward
  - De Morgan
  - Double negation elimination
- Distribute disjunctions over conjunctions
  - Distributivity

CNF: Examples

$$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha) \quad \text{commutativity of } \wedge$$
$$(\alpha \vee \beta) \equiv (\beta \vee \alpha) \quad \text{commutativity of } \vee$$
$$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{associativity of } \wedge$$
$$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{associativity of } \vee$$
$$\neg(\neg\alpha) \equiv \alpha \quad \text{double-negation elimination}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta) \quad \text{implication elimination}$$
$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination}$$
$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \quad \text{De Morgan}$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \quad \text{De Morgan}$$
$$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{distributivity of } \wedge \text{ over } \vee$$
$$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge$$

- $(C \wedge D) \vee E$
  - $(C \vee E) \wedge (D \vee E)$
- $A \wedge B \wedge (C \vee D)$
  - Already in CNF
- $\neg(A \leftrightarrow B)$
  - $\neg\big((A \rightarrow B) \wedge (B \rightarrow A)\big) \equiv \neg(\neg A \vee B) \vee \neg(\neg B \vee A)$
  - $\equiv (A \wedge \neg B) \vee (B \wedge \neg A) \equiv \big(A \vee (B \wedge \neg A)\big) \wedge \big(\neg B \vee (B \wedge \neg A)\big)$
  - $\equiv (A \vee B) \wedge (A \vee \neg A) \wedge (\neg B \vee B) \wedge (\neg B \vee \neg A)$
- $(A \leftrightarrow B) \vee C$
  - $\big((\neg A \vee B) \wedge (\neg B \vee A)\big) \vee C \equiv ((\neg A \vee B) \vee C) \wedge ((\neg B \vee A) \vee C)$
  - $\equiv (\neg A \vee B \vee C) \wedge (\neg B \vee A \vee C)$

Unicorn Example

- *If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.*
  - Prove that the unicorn is both magical and horned
- Y = Unicorn is mythical
- M = Unicorn is a mammal
- G = Unicorn is magical
- R = Unicorn is mortal
- H = Unicorn is horned

- Knowledge Base
  - $Y \rightarrow \neg R$
  - $\neg Y \rightarrow (R \wedge M)$
  - $(\neg R \vee M) \rightarrow H$
  - $H \rightarrow G$
- Prove
  - $G \wedge H$

Convert to Conjunctive Normal Form (CNF)

- $Y \rightarrow \neg R$
- $\neg Y \rightarrow (R \land M)$
- $(\neg R \lor M) \rightarrow H$
- $H \rightarrow G$

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) \quad \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) \quad \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) \quad \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) \quad \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) \quad \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) \quad \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) \quad \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) \quad \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

- $Y \rightarrow \neg R$
  - $\neg Y \lor \neg R$ (implication elimination)
- $\neg Y \rightarrow (R \land M)$
  - $Y \lor (R \land M)$ (implication elimination)
  - $(Y \lor R) \land (Y \lor M)$(distribution)
- $(\neg R \lor M) \rightarrow H$
  - $\neg(\neg R \lor M) \lor H$ (implication elimination)
  - $(R \land \neg M) \lor H$ (De Morgan's)
  - $(R \lor H) \land (\neg M \lor H)$ (distribution)
- $H \rightarrow G$
  - $\neg H \lor G$ (implication elimination)

Clauses

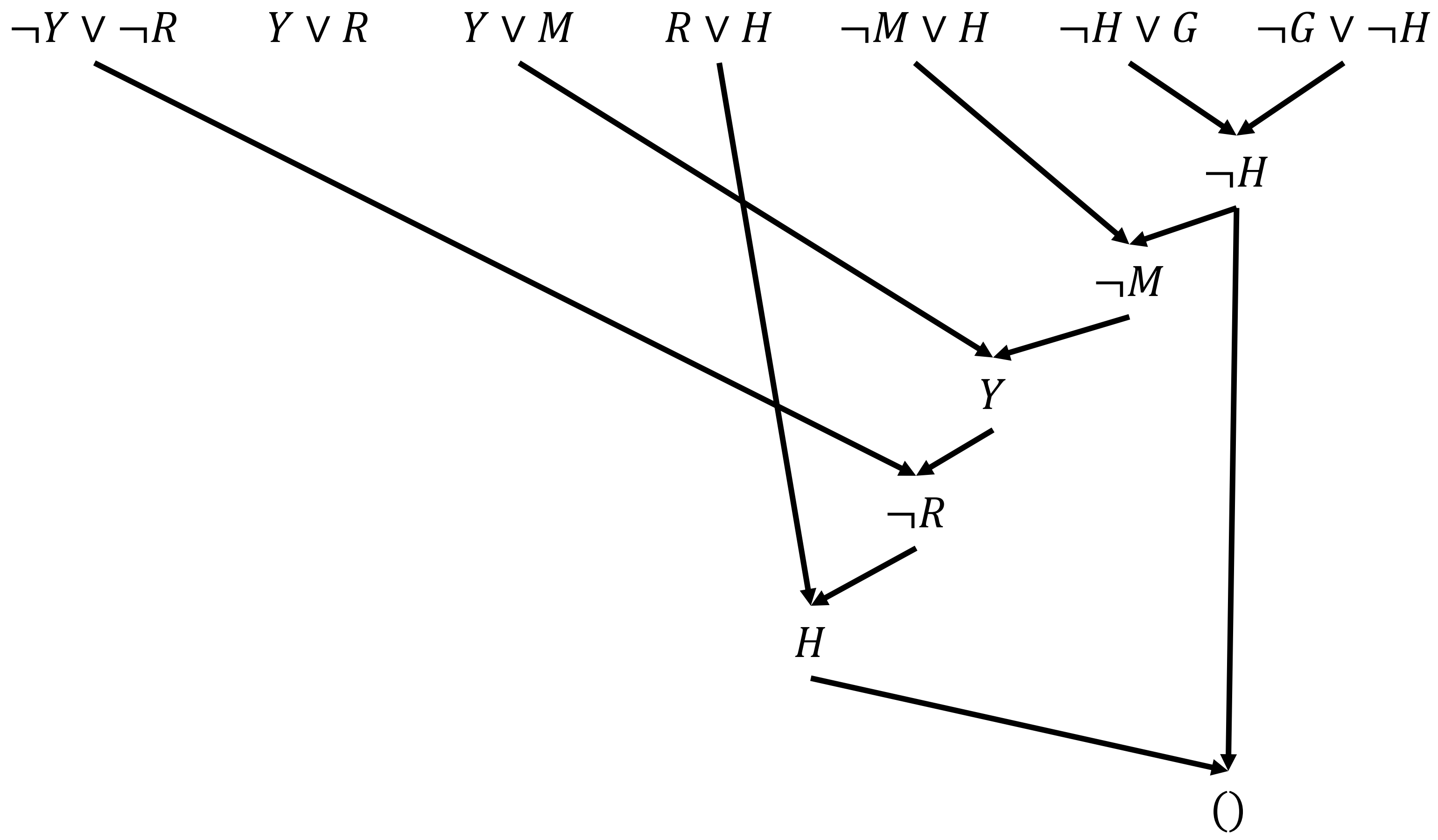
- $Y \rightarrow \neg R$
  - $\neg Y \vee \neg R$ (implication elimination)
- $\neg Y \rightarrow (R \wedge M)$
  - $Y \vee (R \wedge M)$ (implication elimination)
  - $(Y \vee R) \wedge (Y \vee M)$(distribution)
- $(\neg R \vee M) \rightarrow H$
  - $\neg(\neg R \vee M) \vee H$ (implication elimination)
  - $(R \wedge \neg M) \vee H$ (De Morgan)
  - $(R \vee H) \wedge (\neg M \vee H)$ (distribution)
- $H \rightarrow G$
  - $\neg H \vee G$ (implication elimination)

- Our clauses are
  - $\neg Y \vee \neg R$
  - $Y \vee R$
  - $Y \vee M$
  - $R \vee H$
  - $\neg M \vee H$
  - $\neg H \vee G$
  - $\neg G \vee \neg H$ (negation of what we want to prove)

Proof By Contradiction with Resolution

$\neg Y \vee \neg R$ $\quad Y \vee R$ $\quad Y \vee M$ $\quad R \vee H$ $\quad \neg M \vee H$ $\quad \neg H \vee G$ $\quad \neg G \vee \neg H$

Proof By Contradiction with Resolution



Outline

- Background
- Propositional logic
  - Logical connectives
  - Sentences
- Entailment
  - Model-checking
  - Using sound inference rules
  - Proof by contradiction
    - Resolution
    - Logical equivalences
    - Conjunctive normal form
    - Unicorn proof
- Forward chaining and backward chaining

Horn Clauses

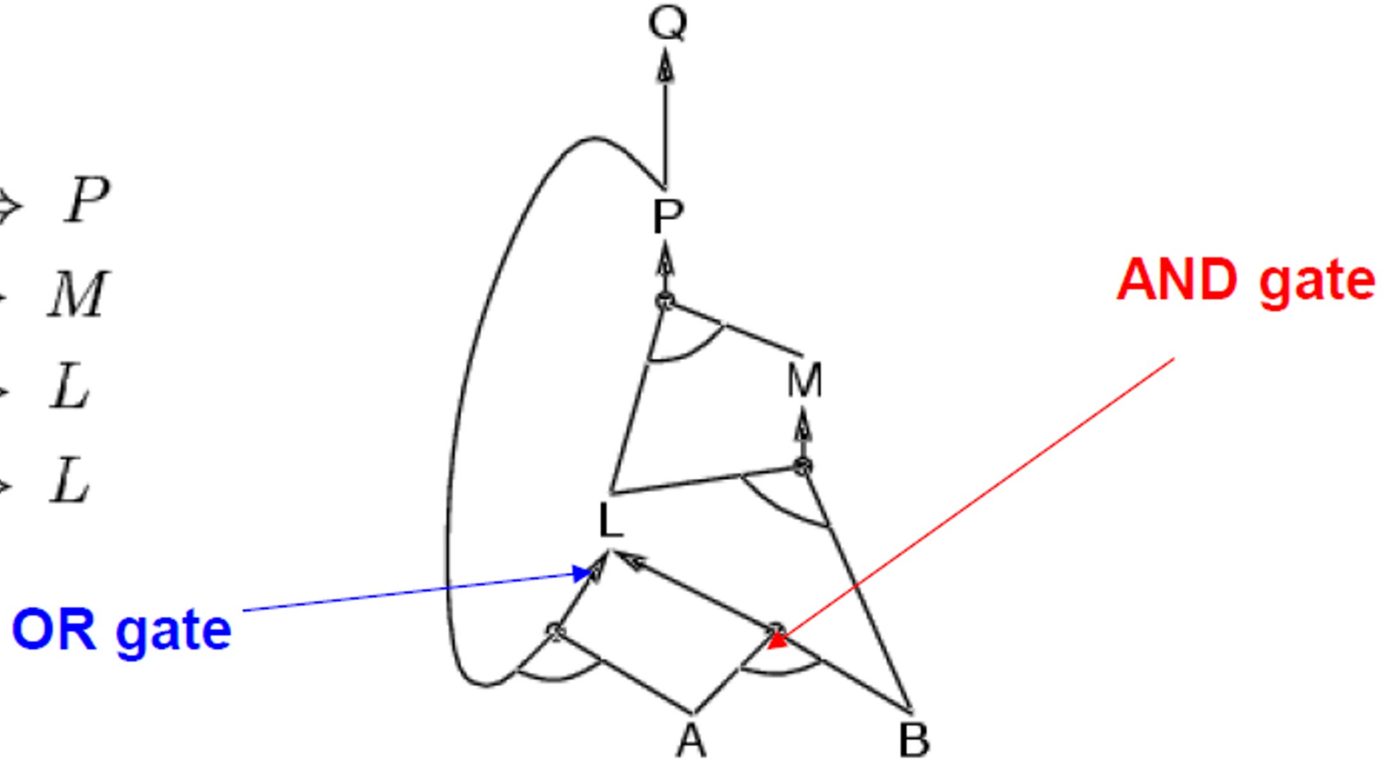
- A disjunction of literals in which **at most** one is positive
- $\neg P \vee Q$
- $\neg L \vee \neg M \vee P$
- $\neg B \vee \neg L \vee M$
- $\neg A \vee \neg P \vee L$
- $\neg A \vee \neg B \vee L$
- A
- B

Forward Chaining

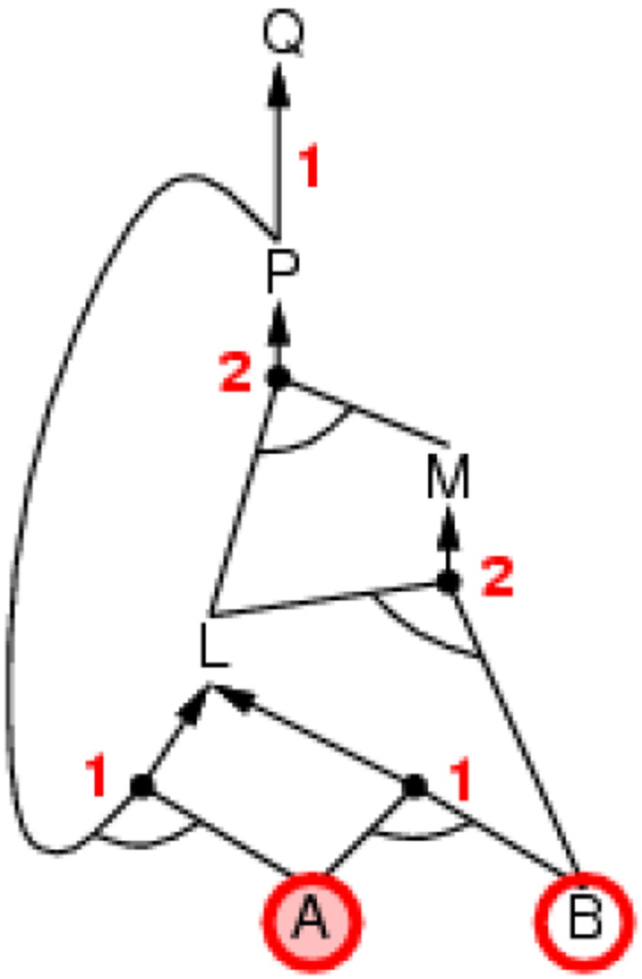
- "Activate" every literal whose premises are satisfied until the sentence you want to prove also has all its premises satisfied

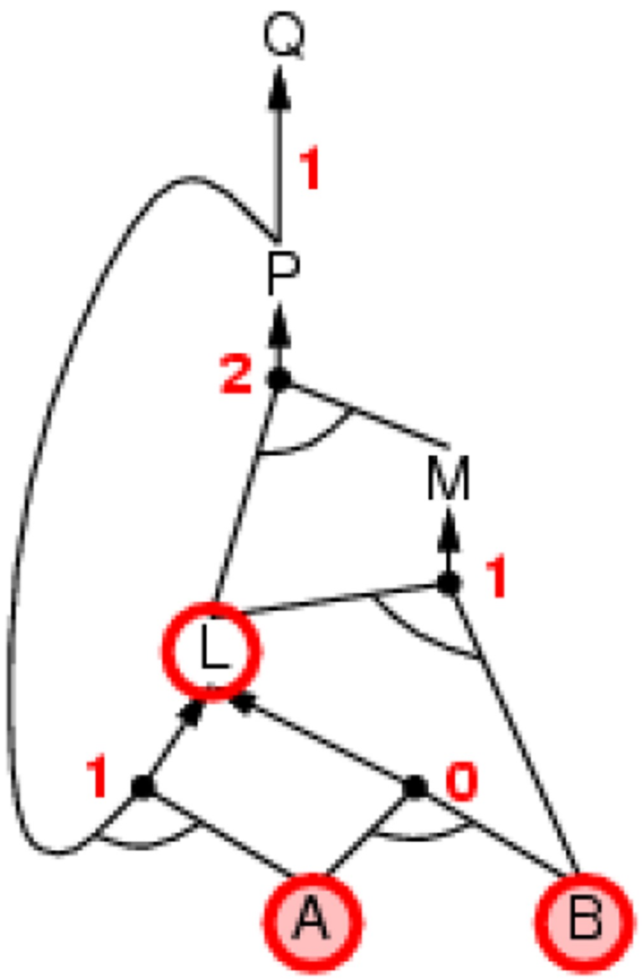
- $\neg P \vee Q$
- $\neg L \vee \neg M \vee P$
- $\neg B \vee \neg L \vee M$
- $\neg A \vee \neg P \vee L$
- $\neg A \vee \neg B \vee L$
- A
- B

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

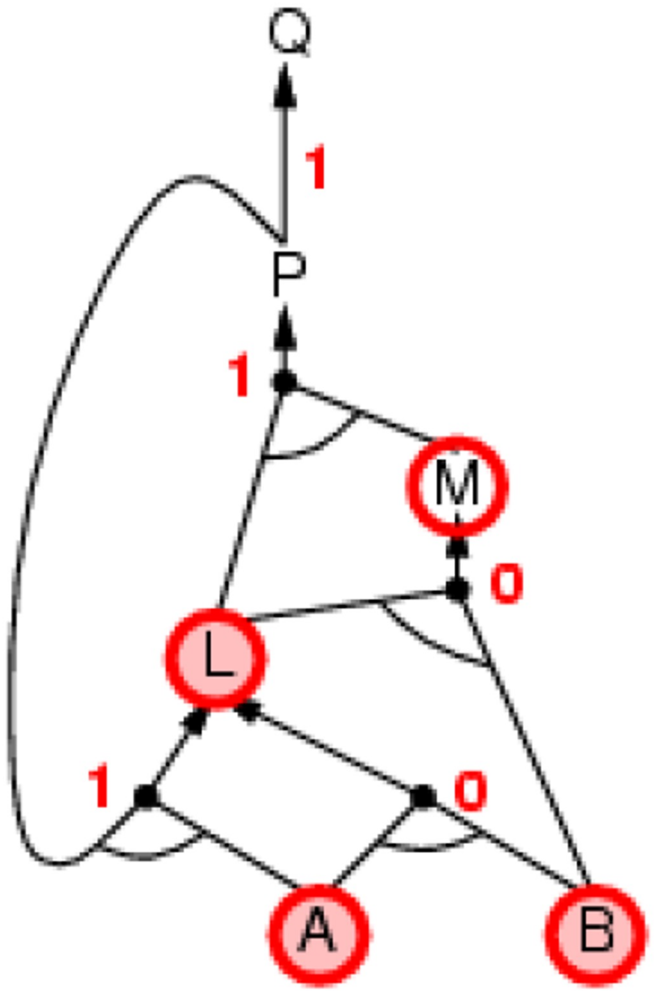


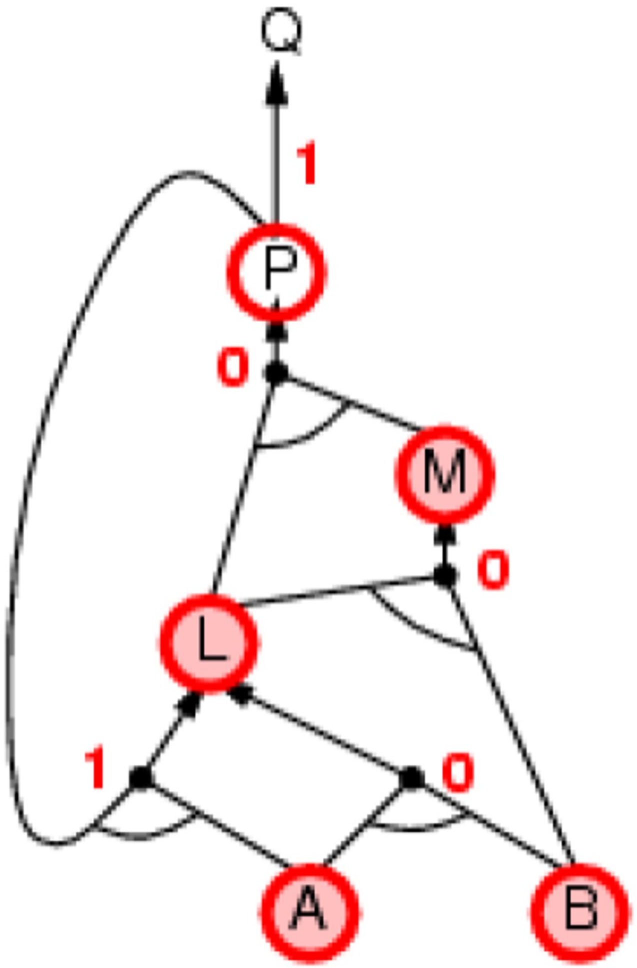
Forward Chaining



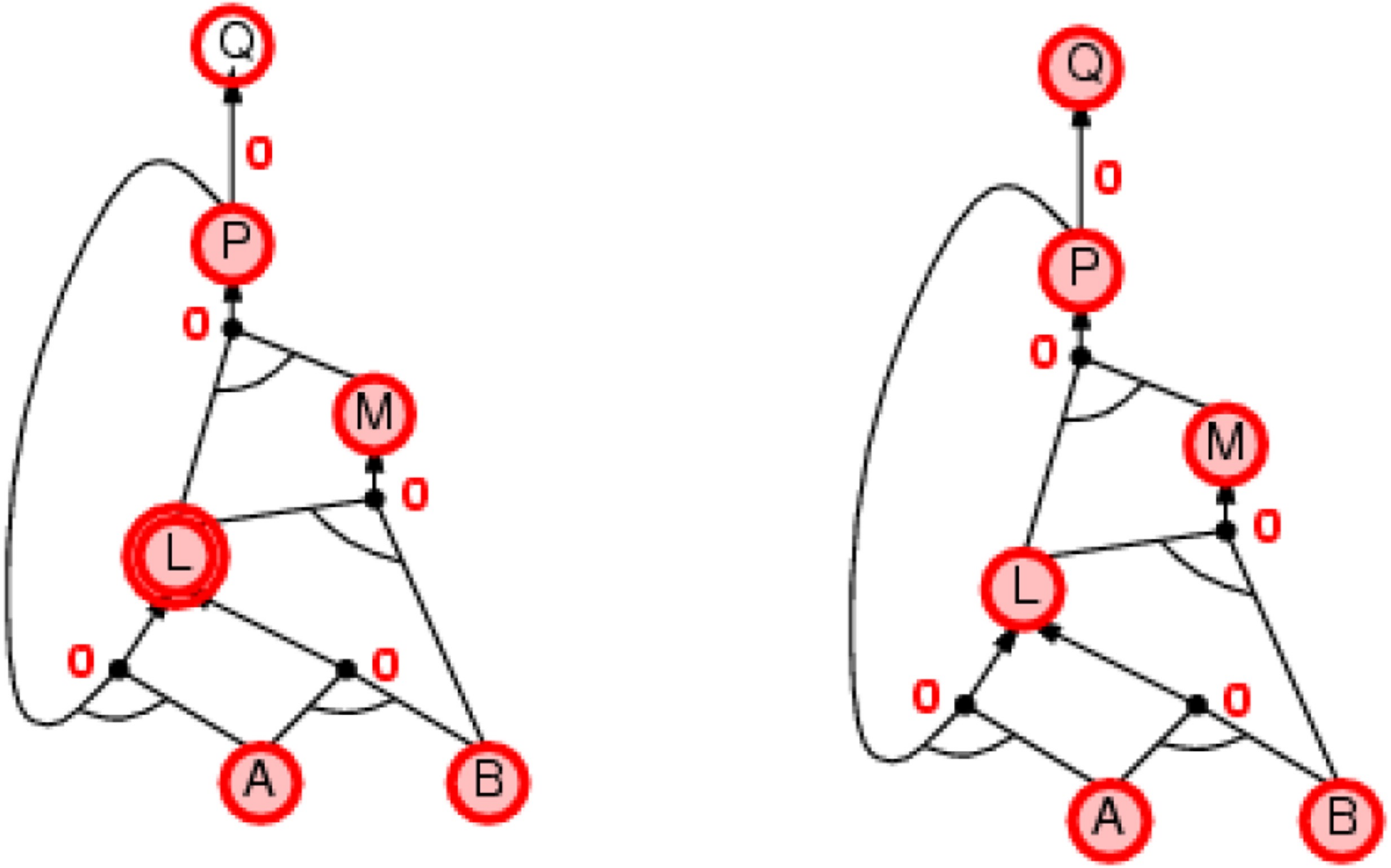


Forward Chaining



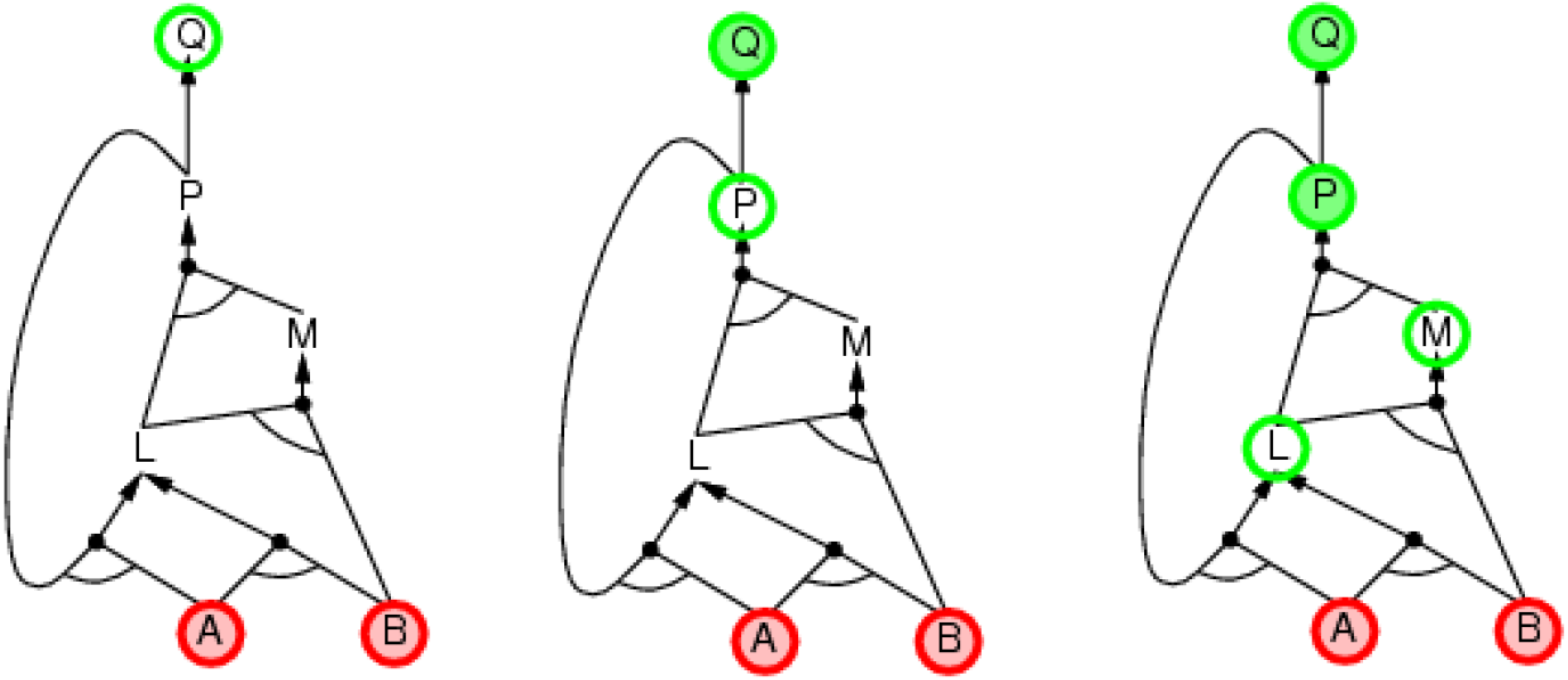


Forward Chaining



Backward Chaining

- Work backwards from what you want to prove until reaching given literals



Forward Chaining vs Backward Chaining

- Forward Chaining
  - Data driven
  - May do work that is irrelevant to the goal
  - Time complexity is linear in the number of literals
- Backward chaining
  - Goal driven
  - Often can be much faster than forward chaining

Limitations of Propositional Logic

- "All humans are mortal"
- Propositional logic
  - $h_1 \wedge m_1$
  - $h_2 \wedge m_2$
  - $h_3 \wedge m_3$
  - …

Summary

- Logical connectives
- Sentences
  - Satisfiability
  - Validity
- Entailment
  - Model checking
  - Using sound inference rules
  - Proof by contradiction
    - To show that $KB \vDash \alpha$ we show $KB \land \neg\alpha$ is unsatisfiable
    - Resolution
    - Conjunctive normal form
- Horn Clauses
  - Forward chaining
  - Backward chaining

Next Time

- First order logic