Propositional Logic B: Inference, Reasoning, Proof

CS171, Winter Quarter, 2019
Introduction to Artificial Intelligence
Prof. Richard Lathrop

Read Beforehand: R&N 7.1-7.5 (optional: 7.6-7.8)
You will be expected to know

• Basic definitions
  – Inference, derive, sound, complete

• Conjunctive Normal Form (CNF)
  – Convert a Boolean formula to CNF

• Do a short resolution proof

• Horn Clauses

• Do a short forward-chaining proof

• Do a short backward-chaining proof

• Model checking with backtracking search

• Model checking with local search
Review: Inference in Formal Symbol Systems
Ontology, Representation, Inference

• **Formal Symbol Systems**
  – *Symbols* correspond to *things/ideas* in the world
  – *Pattern matching & rewrite* corresponds to *inference*

• **Ontology:** What exists in the world?
  – What must be represented?

• **Representation:** Syntax vs. Semantics
  – What’s Said vs. What’s Meant

• **Inference:** Schema vs. Mechanism
  – Proof Steps vs. Search Strategy
Ontology:
What kind of things exist in the world?
What do we need to describe and reason about?

Review

Reasoning

Representation
A Formal Symbol System

Syntax
What is said

Semantics
What it means

Inference
Formal Pattern Matching

Schema
Rules of Inference

Execution
Search Strategy

Preceding lecture
This lecture
Review

• Definitions:
  – Syntax, Semantics, Sentences, Propositions, Entails, Follows, Derives, Inference, Sound, Complete, Model, Satisfiable, Valid (or Tautology), etc.

• Syntactic Transformations:
  – E.g., \((A \Rightarrow B) \iff (\neg A \lor B)\)

• Semantic Transformations:
  – E.g., \((KB \models \alpha) \equiv (\models (KB \Rightarrow \alpha))\)

• Truth Tables
  – Negation, Conjunction, Disjunction, Implication, Equivalence (Biconditional)
  – Inference by Model Enumeration
If KB is true in the real world, then any sentence $\alpha$ entailed by KB is also true in the real world.

For example: If I tell you (1) Sue is Mary’s sister, and (2) Sue is Amy’s mother, then it necessarily follows in the world that Mary is Amy’s aunt, even though I told you nothing at all about aunts. This sort of reasoning pattern is what we hope to capture.
So --- how do we keep it from “Just making things up.”?

Is this inference correct?
How do you know?
How can you tell?

All cats have four legs.
I have four legs.
Therefore, I am a cat.

How can we **make correct** inferences?
How can we **avoid incorrect** inferences?

“Einstein Simplified: Cartoons on Science”
by Sydney Harris, 1992, Rutgers University Press
So --- how do we keep it from “Just making things up.”? 

• All men are people; 
  Half of all people are women; 
  Therefore, half of all men are women.

• Penguins are black and white; 
  Some old TV shows are black and white; 
  Therefore, some penguins are old TV shows.
If KB is true in the real world, then any sentence α derived from KB by a sound inference procedure is also true in the real world.
Logical inference

• The notion of entailment can be used for logic inference.
  – Model checking (see wumpus example):
    enumerate all possible models and check whether $\alpha$ is true.

• $KB \models_{i} \alpha$ means $KB$ derives a sentence $\alpha$ using inference procedure $i$

• **Sound** (or **truth preserving**):
  The algorithm **only** derives entailed sentences.
  – Otherwise it just makes things up.
    $i$ is sound iff whenever $KB \models_{i} \alpha$ it is also true that $KB \models \alpha$
  – E.g., model-checking is sound
    Refusing to infer any sentence is Sound; so, **Sound** is weak alone.

• **Complete**: The algorithm can derive **every** entailed sentence.
  $i$ is complete iff whenever $KB \models \alpha$ it is also true that $KB \models_{i} \alpha$
  Deriving every sentence is Complete; so, **Complete** is weak alone.
Proof methods

Proof methods divide into (roughly) two kinds:

**Application of inference rules:**
Legitimate (sound) generation of new sentences from old.
– Resolution --- KB is in Conjunctive Normal Form (CNF)
– Forward & Backward chaining

**Model checking:**
Searching through truth assignments.
– Improved backtracking: Davis-Putnam-Logemann-Loveland (DPLL)
– Heuristic search in model space: Walksat.
Examples of Sound Inference Patterns

**Classical Syllogism (due to Aristotle)**

- All Ps are Qs
- X is a P
- Therefore, X is a Q

- All Men are Mortal
- Socrates is a Man
- Therefore, Socrates is Mortal

**Implication (Modus Ponens)**

- P implies Q
- P
- Therefore, Q

- Smoke implies Fire
- Smoke
- Therefore, Fire

**Contrapositive (Modus Tollens)**

- P implies Q
- Not Q
- Therefore, Not P

- Smoke implies Fire
- Not Fire
- Therefore, not Smoke

**Law of the Excluded Middle (due to Aristotle)**

- A Or B
- Not A
- Therefore, B

- Alice is a Democrat or a Republican
- Alice is not a Democrat
- Therefore, Alice is a Republican

Why is this different from:

- All men are people
- Half of people are women
- So half of men are women
Inference by Resolution

- KB is represented in CNF
  - KB = AND of all the sentences in KB
  - KB sentence = clause = OR of literals
  - Literal = propositional symbol or its negation

- Find two clauses in KB, one of which contains a literal and the other its negation
  - Cancel the literal and its negation
  - Bundle everything else into a new clause
  - Add the new clause to KB
  - Repeat
Conjunctive Normal Form (CNF)

• **Boolean formulae are central to CS**
  – Boolean logic is the way our discipline works

• **Two canonical Boolean formulae representations:**
  – **CNF = Conjunctive Normal Form**
    • A conjunct of disjuncts = (AND (OR ...) (OR ...))
    • “...” = a list of literals (= a variable or its negation)
    • CNF is used by Resolution Theorem Proving
  – **DNF = Disjunctive Normal Form**
    • A disjunct of conjuncts = (OR (AND ...) (AND ...))
    • DNF is used by Decision Trees in Machine Learning

• **Can convert any Boolean formula to CNF or DNF**
We'd like to prove: \( KB \models \alpha \)
(This is equivalent to \( KB \land \neg \alpha \) is unsatisfiable.)

We first rewrite \( KB \land \neg \alpha \) into **conjunctive normal form (CNF)**.

A “conjunction of disjunctions”
\[
(A \lor \neg B) \land (B \lor \neg C \lor \neg D)
\]

- Any KB can be converted into CNF.
- In fact, any KB can be converted into CNF-3 using clauses with at most 3 literals.
Review: Equivalence & Implication

• Equivalence is a conjoined double implication

\[ (X \iff Y) = [(X \implies Y) \land (Y \implies X)] \]

• Implication is (NOT antecedent OR consequent)

\[ \neg (X \implies Y) = (\neg X \lor Y) \]
Review: de Morgan's rules

• How to bring \( \neg \) inside parentheses
  – (1) Negate everything inside the parentheses
  – (2) Change operators to “the other operator”

• \( \neg(X \land Y \land \ldots \land Z) = (\neg X \lor \neg Y \lor \ldots \lor \neg Z) \)

• \( \neg(X \lor Y \lor \ldots \lor Z) = (\neg X \land \neg Y \land \ldots \land \neg Z) \)
Review: Boolean Distributive Laws

• **Both** of these laws are valid:

• AND distributes over OR
  
  \[- X \land (Y \lor Z) = (X \land Y) \lor (X \land Z)\]
  
  \[- (W \lor X) \land (Y \lor Z) = (W \land Y) \lor (X \land Y) \lor (W \land Z) \lor (X \land Z)\]

• OR distributes over AND
  
  \[- X \lor (Y \land Z) = (X \lor Y) \land (X \lor Z)\]
  
  \[- (W \land X) \lor (Y \land Z) = (W \lor Y) \land (X \lor Y) \land (W \lor Z) \land (X \lor Z)\]
Example: Conversion to CNF

Example: \( B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \)

1. Eliminate \( \iff \) by replacing \( \alpha \iff \beta \) with \( (\alpha \implies \beta) \land (\beta \implies \alpha) \).
   \[
   = (B_{1,1} \implies (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \implies B_{1,1})
   \]

2. Eliminate \( \implies \) by replacing \( \alpha \implies \beta \) with \( \neg \alpha \lor \beta \) and simplify.
   \[
   = (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg (P_{1,2} \lor P_{2,1}) \lor B_{1,1})
   \]

3. Move \( \neg \) inwards using de Morgan's rules and simplify.
   \[
   \neg(\alpha \lor \beta) \equiv (\neg \alpha \land \neg \beta), \quad \neg(\alpha \land \beta) \equiv (\neg \alpha \lor \neg \beta)
   
   = (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1})
   \]

4. Apply distributive law (\( \land \) over \( \lor \)) and simplify.
   \[
   = (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1})
   \]
Example: Conversion to CNF

Example: \( B_{1,1} \iff (P_{1,2} \lor P_{2,1}) \)

From the previous slide we had:

\[
= (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1})
\]

5. KB is the conjunction of all of its sentences (all are true), so write each clause (disjunct) as a sentence in KB:

\[
\text{Often, Won’t Write “∨” or “∧”}
\]

(we know they are there)

\[
\text{KB =} \quad \rightarrow \quad (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \quad \rightarrow \quad (\neg B_{1,1} \quad P_{1,2} \quad P_{2,1})
\]

\[
\rightarrow \quad (\neg P_{1,2} \lor B_{1,1}) \quad \rightarrow \quad (\neg P_{1,2} \quad B_{1,1})
\]

\[
\rightarrow \quad (\neg P_{2,1} \lor B_{1,1}) \quad \rightarrow \quad (\neg P_{2,1} \quad B_{1,1})
\]

\[
\text{(same)}
\]

\[
\text{...}
\]

\[
\text{...}
\]

\[
\text{...}
\]
Inference by Resolution

- **KB is represented in CNF**
  - KB = AND of all the sentences in KB
  - KB sentence = clause = OR of literals
  - Literal = propositional symbol or its negation

- **Find two clauses in KB, one of which contains a literal and the other its negation**
  - Cancel the literal and its negation
  - Bundle everything else into a new clause
  - Add the new clause to KB
  - Repeat
Resolution = Efficient Implication

Recall that \((A \implies B) = (\neg A \lor B)\)
and so:
\[
(Y \lor X) = (\neg X \implies Y)
\]
\[
(\neg Y \lor Z) = (Y \implies Z)
\]
which yields:
\[
((Y \lor X) \land (\neg Y \lor Z)) \models (\neg X \implies Z) = (X \lor Z)
\]

Recall: All clauses in KB are conjoined by an implicit AND (= CNF representation).
Resolution Examples

• Resolution: inference rule for CNF: *sound and complete*! *

\[(A \lor B \lor C)\]
\[(\neg A)\]

\[\therefore (B \lor C)\]

“If A or B or C is true, but not A, then B or C must be true.”

\[(A \lor B \lor C)\]
\[(\neg A \lor D \lor E)\]

\[\therefore (B \lor C \lor D \lor E)\]

“If A is false then B or C must be true, or if A is true then D or E must be true, hence since A is either true or false, B or C or D or E must be true.”

\[(A \lor B)\]
\[(\neg A \lor B)\]

\[\therefore (B \lor B) \equiv B\]

Simplification is done always.

* Resolution is “refutation complete” in that it can prove the truth of any entailed sentence by refutation.

* You can start two resolution proofs in parallel, one for the sentence and one for its negation, and see which branch returns a correct proof.
More Resolution Examples

- \((P \land Q \land \neg R \land S) \) with \((P \land \neg Q \land W \land X)\) yields \((P \land \neg R \land S \land W \land X)\)
  - Order of literals within clauses does not matter.
- \((P \land Q \land \neg R \land S)\) with \((\neg P)\) yields \((Q \land \neg R \land S)\)
- \((\neg R)\) with \((R)\) yields \((\) or FALSE
- \((P \land Q \land \neg R \land S)\) with \((P \land R \land \neg S \land W \land X)\) yields \((P \land Q \land \neg R \land R \land W \land X)\) or \((P \land Q \land S \land \neg S \land W \land X)\) or TRUE
- \((P \land \neg Q \land R \land \neg S)\) with \((P \land \neg Q \land R \land \neg S)\) yields None possible
- \((P \land \neg Q \land \neg S \land W)\) with \((P \land R \land \neg S \land X)\) yields None possible
- \((\neg \ A) \land (\neg \ B) \land (\neg \ C) \land (\neg \ D)\) with \((\neg \ C) \land D)\) yields \((\neg \ A) \land (\neg \ B) \land (\neg \ C)\)
- \((\neg \ A) \land (\neg \ B) \land (\neg \ C)\) with \((\neg \ A) \land C)\) yields \((\neg \ A) \land (\neg \ B)\)
- \((\neg \ A) \land (\neg \ B)\) with \((B)\) yields \((\neg \ A)\)
- \((A \land C)\) with \((A \land \neg \ C)\) yields \((A)\)
- \((\neg \ A)\) with \((A \land \neg \ C)\) yields \((\) or FALSE
Only Resolve **ONE** Literal Pair!
If more than one pair, result always = TRUE. **Useless!!** Always simplifies to TRUE!!

No!

\[(\text{OR} \ A \ B \ C \ D)\]
\[(\text{OR} \ \neg A \ \neg B \ F \ G)\]

\[-----------------------------\]
\[(\text{OR} \ C \ D \ F \ G)\]

No! This is wrong!

Yes! (but = TRUE)

\[(\text{OR} \ A \ B \ C \ D)\]
\[(\text{OR} \ \neg A \ \neg B \ F \ G)\]

\[-----------------------------\]
\[(\text{OR} \ B \ \neg B \ C \ D \ F \ G)\]

Yes! (but = TRUE)

(Resolution theorem provers routinely pre-scan the two clauses for two complementary literals, and if they are found won’t resolve those clauses.)
Resolution Algorithm

- The resolution algorithm tries to prove: \( KB \models \alpha \equiv KB \land \neg \alpha \text{ unsatisfiable} \)

- Generate all new sentences from KB and the (negated) query.
- One of two things can happen:

1. We find \( P \land \neg P \) which is unsatisfiable. I.e.* we can entail the query.

2. We find no contradiction: there is a model that satisfies the sentence \( KB \land \neg \alpha \) (non-trivial) and hence we cannot entail the query.

* I.e. = \textit{id est} = that is
Resolution example
Stated in English

• “Laws of Physics” in the Wumpus World:
  – “A breeze in B11 is equivalent to a pit in P12 or a pit in P21.”

• Particular facts about a specific instance:
  – “There is no breeze in B11.”

• Goal or query sentence:
  – “Is it true that P12 does not have a pit?”
Resolution example
Stated in Propositional Logic

• “Laws of Physics” in the Wumpus World:
  – “A breeze in B11 is equivalent to a pit in P12 or a pit in P21.”
    \[(B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1}))\]
    We converted this sentence to CNF in the CNF example we worked above.

• Particular facts about a specific instance:
  – “There is no breeze in B11.”
    \[(\neg B_{1,1})\]

• Goal or query sentence:
  – “Is it true that P12 does not have a pit?”
    \[(\neg P_{1,2})\]
Resolution example

Resulting Knowledge Base stated in CNF

• “Laws of Physics” in the Wumpus World:
  \((\neg B_{1,1} \quad P_{1,2} \quad P_{2,1})\)
  \((\neg P_{1,2} \quad B_{1,1})\)
  \((\neg P_{2,1} \quad B_{1,1})\)

• Particular facts about a specific instance:
  \((\neg B_{1,1})\)

• Negated goal or query sentence:
  \((P_{1,2})\)
Resolution example
A Resolution proof ending in ( )

• Knowledge Base at start of proof:

\[
\begin{align*}
\neg B_{1,1} & \quad P_{1,2} & \quad P_{2,1} \\
\neg P_{1,2} & \quad B_{1,1} \\
\neg P_{2,1} & \quad B_{1,1} \\
\neg B_{1,1} & \\
\neg B_{1,1} & \\
P_{1,2} &
\end{align*}
\]

A resolution proof ending in ( ):

• Resolve \((\neg P_{1,2} \quad B_{1,1})\) and \((\neg B_{1,1})\) to give \((\neg P_{1,2})\)
• Resolve \((\neg P_{1,2})\) and \((P_{1,2})\) to give ( )

• Consequently, the goal or query sentence is entailed by KB.
• Of course, there are many other proofs, which are OK iff correct.
Resolution example

Graphical view of the proof

• $KB = (B_{1,1} \iff (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1}$

• $\alpha = \neg P_{1,2}$

A sentence in KB is not “used up” when it is used in a resolution step. It is true, remains true, and is still in KB.
Detailed Resolution Proof Example

• **In words:** *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.*

Problem 7.2, R&N page 280. (Adapted from Barwise and Etchemendy, 1993.)

**Note for non-native-English speakers:** immortal = not mortal
Detailed Resolution Proof Example

• **In words:** *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.*

• **First, Ontology:** What do we need to describe and reason about?

• Use these propositional variables (“immortal” = “not mortal”):
  
  - $Y = \text{unicorn is mythical}$
  - $R = \text{unicorn is mortal}$
  - $M = \text{unicorn is a mammal}$
  - $H = \text{unicorn is horned}$
  - $G = \text{unicorn is magical}$
Detailed Resolution Proof Example

• In words: 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.**

\[ Y = \text{unicorn is mythical} \quad R = \text{unicorn is mortal} \]
\[ M = \text{unicorn is a mammal} \quad H = \text{unicorn is horned} \]
\[ G = \text{unicorn is magical} \]

• **Second, translate to Propositional Logic, then to CNF:**
  • Propositional logic (prefix form, aka Polish notation):
    – \( (\Rightarrow Y \ (\neg R) ) \); same as \( (Y \Rightarrow (\neg R) ) \) in infix form
  • CNF (clausal form); recall \( (A \Rightarrow B) = (\neg A \lor B) \)
    – \( (\neg Y \ (\neg R) ) \)

Prefix form is often a better representation for a parser, since it looks at the first element of the list and dispatches to a handler for that operator token.
Detailed Resolution Proof Example

• **In words:** *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
  R = unicorn is mortal
  M = unicorn is a mammal
  H = unicorn is horned
  G = unicorn is magical

• **Second, translate to Propositional Logic, then to CNF:**

  • Propositional logic (prefix form):
    – (⇒ (NOT Y) (AND R M)) ; same as ( (NOT Y) ⇒ (R AND M) ) in infix form

  • CNF (clausal form)
    – (M Y)
    – (R Y)

  If you ever have to do this “for real” you will likely invent a new domain language that allows you to state important properties of the domain --- then parse that into propositional logic, and then CNF.
In words: 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
R = unicorn is mortal
M = unicorn is a mammal
H = unicorn is horned
G = unicorn is magical

Second, translate to Propositional Logic, then to CNF:
- Propositional logic (prefix form):
  - (=> (OR (NOT R) M) H) ; same as (Not R) OR M => H in infix form
- CNF (clausal form)
  - (H (NOT M))
  - (H R)
In words: 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
R = unicorn is mortal
M = unicorn is a mammal
H = unicorn is horned
G = unicorn is magical

Second, translate to Propositional Logic, then to CNF:

Propositional logic (prefix form)

– (=> H G) ; same as H => G in infix form

CNF (clausal form)

– ( (NOT H) G)
Detailed Resolution Proof Example

- **In words:** 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**
- **R = unicorn is mortal**
- **M = unicorn is a mammal**
- **H = unicorn is horned**
- **G = unicorn is magical**

- **Current KB (in CNF clausal form) =**

\[
\begin{align*}
&M \vee \neg Y \\
&\neg R \vee \neg Y \\
&M \vee Y \\
&R \vee Y \\
&H \vee \neg M \\
&H \vee R \\
&\neg H \vee G
\end{align*}
\]
Detailed Resolution Proof Example

• **In words:** *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.*

  
  \[
  \begin{align*}
  Y &= \text{unicorn is mythical} & R &= \text{unicorn is mortal} \\
  M &= \text{unicorn is a mammal} & H &= \text{unicorn is horned} \\
  G &= \text{unicorn is magical}
  \end{align*}
  \]

• **Third, negated goal to Propositional Logic, then to CNF:**

  • Goal sentence in propositional logic (prefix form)
    – \((\text{AND } H \text{ G})\) ; same as \(H \text{ AND } G\) in infix form
  
  • Negated goal sentence in propositional logic (prefix form)
    – \((\text{NOT } (\text{AND } H \text{ G}) ) = (\text{OR } (\text{NOT } H) (\text{NOT } G) )\)

• **CNF (clausal form)**
  – \(( (\text{NOT } G) (\text{NOT } H) )\)
Detailed Resolution Proof Example

• **In words:** *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.*

  \[ Y = \text{unicorn is mythical} \quad \quad R = \text{unicorn is mortal} \]

  \[ M = \text{unicorn is a mammal} \quad \quad H = \text{unicorn is horned} \]

  \[ G = \text{unicorn is magical} \]

• **Current KB + negated goal** (in CNF clausal form) =

\[
\begin{align*}
& (\neg Y) (\neg R) \quad (M Y) \quad (R Y) \quad (H (\neg M)) \\
& (H R) \quad (\neg H) G \quad (\neg G) (\neg H)
\end{align*}
\]
Detailed Resolution Proof Example

• **In words:** *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.*

  \[
  (\neg Y \neg R) \quad (M Y) \quad (R Y) \quad (H \neg M) \\
  (H R) \quad (\neg H G) \quad (\neg G \neg H)
  \]

• **Fourth, produce a resolution proof ending in ( ):**
  • Resolve \((\neg H \neg G)\) and \((\neg H G)\) to give \((\neg H)\)
  • Resolve \((\neg Y \neg R)\) and \((Y M)\) to give \((\neg R M)\)
  • Resolve \((\neg R M)\) and \((R H)\) to give \((M H)\)
  • Resolve \((M H)\) and \((\neg M H)\) to give \((H)\)
  • Resolve \((\neg H)\) and \((H)\) to give \(( )\)

• Of course, there are many other proofs, which are OK iff correct.
Detailed Resolution Proof Example

Graph view of proof

\[(\neg Y \neg R)(YR)(YM)(RH)(\neg MH)(\neg HG)(\neg G \neg H)\]
Detailed Resolution Proof Example

Graph view of a different proof

• $\neg Y \neg R (Y R) (Y M) (R H) (\neg M H) (\neg H G) (\neg G \neg H)$
Horn Clauses

- Resolution can be exponential in space and time.
- If we can reduce all clauses to “Horn clauses” inference is linear in space and time.

A clause with at most 1 positive literal.

- Every Horn clause can be rewritten as an implication with a conjunction of positive literals in the premises and at most a single positive literal as a conclusion.

\[ A \lor \neg B \lor \neg C \]

- 1 positive literal and \( \geq 1 \) negative literal: definite clause (e.g., above)
- 0 positive literals: integrity constraint or goal clause

\[ (\neg A \lor \neg B) \equiv (A \land B \Rightarrow False) \] states that \((A \land B)\) must be false

- 0 negative literals: fact

\[ (A) \equiv (\text{True} \Rightarrow A) \] states that \(A\) must be true.

- Forward Chaining and Backward chaining are sound and complete with Horn clauses and run linear in space and time.
Forward chaining (FC)

- Idea: fire any rule whose premises are satisfied in the $KB$, add its conclusion to the $KB$, until $Query$ is found.

- This proves that $KB \Rightarrow Query$ is true in all possible worlds (i.e. trivial), and hence it proves entailment.

- Forward chaining is sound and complete for Horn $KB$
Forward chaining example

Numbers at each AND node indicate the number of outstanding preconditions yet to be satisfied before **all** of that AND node input preconditions have been satisfied. It is an efficient book-keeping mechanism for determining when an AND node is satisfied. The AND node is satisfied when its number of outstanding preconditions yet to be satisfied is zero.
Forward chaining example
Forward chaining example
Forward chaining example
Forward chaining example
Forward chaining example
Forward chaining example
Backward chaining (BC)

Idea: work backwards from the query $q$

- check if $q$ is known already, or
- prove by BC all premises of some rule concluding $q$
- Hence BC maintains a stack of sub-goals that need to be proved to get to $q$.

Avoid loops: check if new sub-goal is already on the goal stack

Avoid repeated work: check if new sub-goal

1. has already been proved true, or
2. has already failed
Backward chaining example
Backward chaining example
Backward chaining example
Backward chaining example

we need $P$ to prove $L$ and $L$ to prove $P$. 
Backward chaining example

As soon as you can move forward, do so.
Backward chaining example
Backward chaining example
Backward chaining example
Backward chaining example
Backward chaining example
Forward vs. backward chaining

• FC is data-driven, automatic, unconscious processing,
  – e.g., object recognition, routine decisions

• May do lots of work that is irrelevant to the goal

• BC is goal-driven, appropriate for problem-solving,
  – e.g., Where are my keys? How do I get into a PhD program?

• Complexity of BC can be much less than linear in size of KB
Model Checking

Two families of efficient algorithms:

• Complete backtracking search algorithms:
  – E.g., DPLL algorithm

• Incomplete local search algorithms
  – E.g., WalkSAT algorithm
The DPLL algorithm

Determine if an input propositional logic sentence (in CNF) is satisfiable. This is just backtracking search for a CSP.

Improvements:

1. Early termination
   A clause is true if any literal is true.
   A sentence is false if any clause is false.

2. Pure symbol heuristic
   Pure symbol: always appears with the same "sign" in all clauses.
   e.g., In the three clauses \((A \lor \neg B)\), \((\neg B \lor \neg C)\), \((C \lor A)\), A and B are pure, C is impure.
   Make a pure symbol literal true. (if there is a model for \(S\), then making a pure symbol true is also a model).

3. Unit clause heuristic
   Unit clause: only one literal in the clause
   The only literal in a unit clause must be true.

Note: literals can become a pure symbol or a unit clause when other literals obtain truth values. e.g.

\[(A \lor True) \land (\neg A \lor B)\]
\[A = pure\]
The WalkSAT algorithm

- Incomplete, local search algorithm

- Evaluation function: The min-conflict heuristic of minimizing the number of unsatisfied clauses

- Balance between greediness and randomness

**Walksat Procedure**

Start with random initial assignment.
Pick a random unsatisfied clause.
Select and flip a variable from that clause:
  - With probability $p$, pick a random variable.
  - With probability $1-p$, pick greedily a variable that minimizes the number of unsatisfied clauses.
Repeat to predefined maximum number flips; if no solution found, restart.
Hard satisfiability problems

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

\[(\neg D \lor \neg B \lor C) \land (B \lor \neg A \lor \neg C) \land (\neg C \lor \neg B \lor E) \land (E \lor \neg D \lor B) \land (B \lor E \lor \neg C)\]

\[m = \text{number of clauses (5)}\]

\[n = \text{number of symbols (5)}\]

– Hard problems seem to cluster near \(m/n = 4.3\) (critical point)
Hard satisfiability problems
Hard satisfiability problems

- Median runtime for 100 satisfiable random 3-CNF sentences, $n = 50$
Hardness of CSPs

• $x_1 \ldots x_n$ discrete, domain size $d$: $O(d^n)$ configurations

• “SAT”: Boolean satisfiability: $d=2$
  – The first known NP-complete problem

• “3-SAT”
  – Conjunctive normal form (CNF)
  – At most 3 variables in each clause:
    $$(x_1 \lor \neg x_7 \lor x_{12}) \land (\neg x_3 \lor x_2 \lor x_7) \land \ldots$$
  – Still NP-complete

• How hard are “typical” problems?
Hardness of random CSPs

• Random 3-SAT problems:
  – n variables, p clauses in CNF: \((x_1 \lor \neg x_7 \lor x_{12}) \land (\neg x_3 \lor x_2 \lor x_7) \land \ldots\)
  – Choose any 3 variables, signs uniformly at random
  – What’s the probability there is no solution to the CSP?
  – Phase transition at \((p/n) \approx 4.25\)
  – “Hard” instances fall in a very narrow regime around this point!
Hardness of random CSPs

- Random 3-SAT problems:
  - \( n \) variables, \( p \) clauses in CNF: \((x_1 \lor \neg x_7 \lor x_{12}) \land (\neg x_3 \lor x_2 \lor x_7) \land \ldots\)
  - Choose any 3 variables, signs uniformly at random
  - What’s the probability there is **no** solution to the CSP?
  - Phase transition at \((p/n) \approx 4.25\)
  - “Hard” instances fall in a very narrow regime around this point!
Common Sense Reasoning
Example, adapted from Lenat

You are told: John drove to the grocery store and bought a pound of noodles, a pound of ground beef, and two pounds of tomatoes.

- Is John 3 years old?
- Is John a child?
- What will John do with the purchases?
- Did John have any money?
- Does John have less money after going to the store?
- Did John buy at least two tomatoes?
- Were the tomatoes made in the supermarket?
- Did John buy any meat?
- Is John a vegetarian?
- Will the tomatoes fit in John’s car?

- Can Propositional Logic support these inferences?
Summary

• Logical agents apply inference to a knowledge base to derive new information and make decisions

• Basic concepts of logic:
  – syntax: formal structure of sentences
  – semantics: truth of sentences wrt models
  – entailment: necessary truth of one sentence given another
  – inference: deriving sentences from other sentences
  – soundness: derivations produce only entailed sentences
  – completeness: derivations can produce all entailed sentences

• Resolution is complete for propositional logic. Forward and backward chaining are linear-time, complete for Horn clauses

• Propositional logic lacks expressive power