Resolution Refutation

Dr. Demetrios Glinos
University of Central Florida

CAP4630 –Artificial Intelligence

Today

- Propositional Logic Review
- Propositional Theorem Proving

Propositional Logic

- The logic of sentences that can be assigned truth values (propositions)
- Not all sentences are propositions (e.g.,"Turn to the right" is not a proposition)

Sentence	$\longrightarrow$	AtomicSentence | ComplexSentence
AtomicSentence	$\longrightarrow$	*True* | *False* | P | Q | R | ...
ComplexSentence	$\longrightarrow$	(Sentence) | [Sentence]
		| $\neg$ Sentence
		| Sentence $\wedge$ Sentence
		| Sentence $\vee$ Sentence
		| Sentence $\Rightarrow$ Sentence
		| Sentence $\Leftrightarrow$ Sentence
OPERATOR PRECEDENCE	:	$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Logic Review

- Logic Concepts
  - Syntax — determines which sentences are well-formed
  - Semantics — uses models to assign truth values

- Models and Truth
  - Satisfiability — model m *satisfies* sentence α if α is true in m
    we denote the set of all models of α by $M(\alpha)$
  - Entailment — if β "follows from" α, we say "α entails β" and write $\alpha \vDash \beta$
    formally: $\alpha \vDash \beta$ if and only if $M(\alpha) \subseteq M(\beta)$

Determining Entailment

- Model checking
  - enumerate all possible worlds (assignments to variables) in which the rules of the KB are true
  - if a particular assertion is true in all of them, then it is entailed by the KB

- Inference algorithms
  - We denote inference algorithm i by $\vdash_i$
  - If we use inference algorithm i to derive α from KB, we write **KB $\vdash_i$ α**
  - Algorithm i is sound (truth-preserving) if it derives only entailed sentences
  - Algorithm i is complete if it can derive *any* sentence that is entailed

Today

- Propositional Logic Review
- Propositional Theorem Proving

Theorem Proving Concepts

- Logical equivalence:
  - $\alpha \equiv \beta$ iff $\alpha \vDash \beta$ and $\beta \vDash \alpha$
- Validity
  - a sentence is valid if it is true in ***all*** possible models
  - also called a "tautology"
  - Every valid sentence is logically equivalent to the sentence *True*
- Deduction Theorem
  - For any α and β: $\alpha \vDash \beta$ if and only if the sentence $(\alpha \Rightarrow \beta)$ is valid
  - Thus, we can decide entailment by proving $(\alpha \Rightarrow \beta) \equiv$ True
    - In particular, we can add θ to our KB if $(KB \Rightarrow \theta) \equiv$ True
- Satisfiability
  - Sentence α is satisfiable if it is true in ("satisfied by") ***some*** model
  - Can decide by enumerating the models and checking
  - *This is the SAT problem: the first problem proved to be NP-complete*

Proof by Refutation

- Also called: Proof by Contradiction (*reductio ad absurdum)*
- Basic idea: Assume what you wish to prove is false
  Show this leads to a contradiction with things known to be true
- What this means for us:

  $\boxed{\alpha \text{ is valid iff } \neg\alpha \text{ is unsatisfiable}}$

  (which is equivalent to: α is not valid iff $\neg\alpha$ is satisfiable)

- A **proof** is a chain of conclusions that leads to the desired goal

Inference Rules

- Modus Ponens: Given α⇒ β and α, then we can infer β

$$\frac{\alpha \Rightarrow \beta,\ \alpha}{\beta}$$

- And-Elimination: ("simplification")

$$\frac{\alpha \wedge \beta}{\alpha}$$

- We can also use logical equivalences as inference rules (see next slide)

Logical Equivalences

$(A \wedge B) \equiv (B \wedge A)$	$\wedge$ is commutative
$(A \vee B) \equiv (B \vee A)$	$\vee$ is commutative
$((A \wedge B) \wedge C) \equiv (A \wedge (B \wedge C))$	$\wedge$ is associative
$((A \vee B) \vee C) \equiv (A \vee (B \vee C))$	$\vee$ is associative
$\neg(\neg A) \equiv A$	Double-negation elimination
$(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A)$	Contraposition
$(A \Rightarrow B) \equiv (\neg A \vee B)$	Implication elimination
$(A \Leftrightarrow B) \equiv ((A \Rightarrow B) \wedge (B \Rightarrow A))$	Biconditional elimination
$\neg(A \wedge B) \equiv (\neg A \vee \neg B)$	"De Morgan"
$\neg(A \vee B) \equiv (\neg A \wedge \neg B)$	"De Morgan"
$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee (A \wedge C))$	Distributivity of $\wedge$ over $\vee$
$(A \vee (B \wedge C)) \equiv ((A \vee B) \wedge (A \vee C))$	Distributivity of $\vee$ over $\wedge$

Example: KB for the Wumpus World

- Let P = pit, W = Wumpus, B = breeze, and S = stench
- Let us use subscripts to represent the location, e.g., $P_{3,2}$

- Consider this KB:

R_1:	$\neg P_{1,1}$	no pit in 1,1
R_2:	$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$	breeze iff pit in adjacent square
...	...	...
R_3:	$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$	breeze iff pit in adjacent square
R_4:	$\neg B_{1,1}$	no breeze in 1,1
R_5:	$B_{2,1}$	breeze in 2,1

The above is sufficient to derive: $\neg P_{1,2}$

Example: Proof of $\neg P_{1,2}$

Rule	Proposition	Reasoning
R_6:	$B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1}) \wedge (P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}$	R_2, biconditional elimination
R_7:	$(P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}$	R_6, and-elimination
R_8:	$\neg B_{1,1} \Rightarrow \neg (P_{1,2} \vee P_{2,1})$	R_7, contrapositive
R_9:	$\neg (P_{1,2} \vee P_{2,1})$	R_4, R_8, modus ponens
R_{10}:	$\neg P_{1,2} \wedge \neg P_{2,1}$	R_9, De Morgan's rule
R_{11}:	$\neg P_{1,2}$	R_{10}, and-elimination

Finding a Proof

- This is a search problem

Initial state:	the initial KB
Actions:	all inference rules applied to all sentences to which they can be applied
Result:	add sentence produced to the KB
Goal:	a state containing the sentence we are trying to prove

- We can use any of our search algorithms (BFS, DFS, UCS, IDS, A*) to find the solution

Thus, we can substitute search for model enumeration (model checking)

Monotonicity

- A property of logical systems

  The set of entailed sentences can only increase as information is added to the KB

- This is a consistency requirement

  if $KB \vDash \alpha$ then $KB \wedge \beta \vDash \alpha$

  - i.e., As we add validly entailed sentences to the KB, this cannot invalidate previously entailed sentences
  - Example: Adding a rule that there are 8 pits in the Wumpus World does not invalidate our conclusion about $P_{1,2}$

Proof by Resolution

- Reconsider our Wumpus World proof
  - If biconditional elimination did not exist, the proof would not work
  - So, we must have assurance that the inference rules we use have the completeness property

- **Theorem (without proof):** The resolution rule + any complete search algorithm, is a complete inference algorithm

- Recall the resolution rule:

$$\frac{\alpha \lor \beta, \ \neg \alpha}{\beta}$$

← *these are the premises*
← *this is the resolvent*

- So, resolution is all we need !

Conjunctive Normal Form (CNF)

- Resolution applies only to disjunctions of literals (called "clauses")
- That's OK, because every sentence of propositional logic is equivalent to a conjunction of clauses

$$\frac{\alpha \vee \beta, \quad \neg\alpha}{\beta}$$

- How to convert a proposition to CNF:

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$
2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$
3. Move $\neg$ inwards to apply only to individual literals, using
   a. $\neg(\neg\alpha) \equiv \alpha$ (double negation elimination)
   b. $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ (De Morgan)
   c. $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$ (De Morgan)
4. Now use distributivity to distribute $\vee$ over $\wedge$ whenever possible:

$$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

Example: Resolution Refutation

Given: $KB = (A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$

Prove: G using resolution refutation

Example: Resolution Refutation

Given: $KB = (A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Example: Resolution Refutation

Given: KB = $(A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Next, we negate the conclusion and add it to the KB

R6: $\neg G$

Example: Resolution Refutation

Given: $KB = (A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Next, we negate the conclusion and add it to the KB

R6: $\neg G$

Then, we use the resolution rule repeatedly until a contradiction is found (or not)

R7: $\neg D$, (R5, R6, resolution)

Example: Resolution Refutation

Given: KB = $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Next, we negate the conclusion and add it to the KB

R6: $\neg G$

Then, we use the resolution rule repeatedly until a contradiction is found (or not)

R7: $\neg D$, (R5, R6, resolution)

R8: $\neg C$, (R4, R6, resolution)

Example: Resolution Refutation

Given: KB = $(A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Next, we negate the conclusion and add it to the KB

R6: $\neg G$

Then, we use the resolution rule repeatedly until a contradiction is found (or not)

R7: $\neg D$, (R5, R6, resolution)

R8: $\neg C$, (R4, R6, resolution)

R9: $\neg B$, (R3, R7, resolution)

Example: Resolution Refutation

Given: KB = $(A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Next, we negate the conclusion and add it to the KB

R6: $\neg G$

Then, we use the resolution rule repeatedly until a contradiction is found (or not)

R7: $\neg D$, (R5, R6, resolution)

R8: $\neg C$, (R4, R6, resolution)

R9: $\neg B$, (R3, R7, resolution)

R10: $\neg A$, (R2, R8, resolution)

Example: Resolution Refutation

Given: KB = $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Next, we negate the conclusion and add it to the KB

R6: $\neg G$

Then, we use the resolution rule repeatedly until a contradiction is found (or not)

R7: $\neg D$, (R5, R6, resolution)

R8: $\neg C$, (R4, R6, resolution)

R9: $\neg B$, (R3, R7, resolution)

R10: $\neg A$, (R2, R8, resolution)

R11: A , (R1, R9, resolution)

Example: Resolution Refutation

Given: $KB = (A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$

R1 R2 R3 R4 R5

Prove: G using resolution refutation

Proof:

First, we confirm the clauses are CNF and we number them

Next, we negate the conclusion and add it to the KB

R6: $\neg G$

Then, we use the resolution rule repeatedly until a contradiction is found (or not)

R7: $\neg D$, (R5, R6, resolution)

R8: $\neg C$, (R4, R6, resolution)

R9: $\neg B$, (R3, R7, resolution)

R10: $\neg A$, (R2, R8, resolution)

R11: A , (R1, R9, resolution)

R12: { }, (R10, R11, resolution) ← *empty set here proves the contradiction*

Completeness of Resolution

- **Resolution Closure**
  - for a set S of clauses
  - denoted by **RC(S)**
  - is the set of all clauses derivable by repeated application of the resolution rule to clauses in S or their derivatives
    - RC(S) must be finite for finite S, provided we eliminate duplicate literals within clauses
    - therefore, the procedure to generate the set will eventually terminate

- **Ground Resolution Theorem** (Completeness theorem for resolution in prop. logic):
  - If a set of clauses is unsatisfiable, then the resolution closure of those clauses contains the empty clause.
  - i.e., P and ¬ P cannot both be satisfied unless {} is in the set

Propositional Logic Word Problem

Given the KB:

R1: If Sarah was drunk then either James is the murderer or Sarah lies

R2: Either James is the murderer or Sarah was not drunk and the crime took place after midnight

R3: If the crime took place after midnight then either James is the murderer or Sarah lies

R4: Sarah does not lie when sober

Prove that James is the murderer

Propositional Logic Word Problem

Given the KB:

R1: If Sarah was drunk then either James is the murderer or Sarah lies

R2: Either James is the murderer or Sarah was not drunk and the crime took place after midnight

R3: If the crime took place after midnight then either James is the murderer or Sarah lies

R4: Sarah does not lie when sober

Prove that James is the murderer

How to solve:

1. Express rules as propositions (identify concepts and logical connectives)
2. Negate the desired conclusion and add it to the KB
3. Convert all propositions to conjunctive normal form
4. Use resolution refutation to prove a contradiction, if possible

Propositional Logic Word Problem

Given the KB:

R1: If Sarah was drunk then either James is the murderer or Sarah lies

R2: Either James is the murderer or Sarah was not drunk and the crime took place after midnight

R3: If the crime took place after midnight then either James is the murderer or Sarah lies

R4: Sarah does not lie when sober

Prove that James is the murderer

Step 1: Express rules as propositions (identify concepts and logical connectives)

let A = James is murderer; B = Sarah was drunk; C = Sarah lies; D = murder after midnight

R1: $B \Rightarrow (A \lor C)$

R2: $A \lor (\neg B \land D)$

R3: $D \Rightarrow (A \lor C)$

R4: $\neg B \Rightarrow \neg C$

Propositional Logic Word Problem

Given the KB:

R1: $B \Rightarrow (A \lor C)$

R2: $A \lor (\neg B \land D)$

R3: $D \Rightarrow (A \lor C)$

R4: $\neg B \Rightarrow \neg C$

Prove that James is the murderer

Step 2: Negate the desired conclusion and add it to the KB

let A = James is murderer; B = Sarah was drunk; C = Sarah lies; D = murder after midnight

R5: $\neg A$

Propositional Logic Word Problem

Given the KB:

R1:	$B \Rightarrow (A \lor C)$
R2:	$A \lor (\neg B \land D)$
R3:	$D \Rightarrow (A \lor C)$
R4:	$\neg B \Rightarrow \neg C$
R5:	$\neg A$

Step 3: Convert all propositions to conjunctive normal form

final CNF form

R1:	$\neg B \lor (A \lor C)$	R1:	$\neg B \lor A \lor C$
R2:	$A \lor (\neg B \land D)$	R21:	$A \lor \neg B$
		R22:	$A \lor D$
R3:	$\neg D \lor (A \lor C)$	R3:	$\neg D \lor A \lor C$
R4:	$\neg (\neg B) \lor \neg C$	R4:	$B \lor \neg C$
R5:	$\neg A$	R5:	$\neg A$

Propositional Logic Word Problem

Given the KB:

R1:	$\neg B \lor A \lor C$
R21:	$A \lor \neg B$
R22:	$A \lor D$
R3:	$\neg D \lor A \lor C$
R4:	$B \lor \neg C$
R5:	$\neg A$

Step 4: Use resolution refutation to prove a contradiction, if possible

[Take a few minutes to work this out]

Propositional Logic Word Problem

Given the KB:

R1:	$\neg B \lor A \lor C$
R21:	$A \lor \neg B$
R22:	$A \lor D$
R3:	$\neg D \lor A \lor C$
R4:	$B \lor \neg C$
R5:	$\neg A$

Step 4: Use resolution refutation to prove a contradiction, if possible

R6:	D	R5, R22, resolution

Propositional Logic Word Problem

Given the KB:

R1:	$\neg B \lor A \lor C$
R21:	$A \lor \neg B$
R22:	$A \lor D$
R3:	$\neg D \lor A \lor C$
R4:	$B \lor \neg C$
R5:	$\neg A$

Step 4: Use resolution refutation to prove a contradiction, if possible

R6:	D	R5, R22, resolution
R7:	$A \lor C$	R6, R3, resolution

Propositional Logic Word Problem

Given the KB:

R1:	$\neg B \lor A \lor C$
R21:	$A \lor \neg B$
R22:	$A \lor D$
R3:	$\neg D \lor A \lor C$
R4:	$B \lor \neg C$
R5:	$\neg A$

Step 4: Use resolution refutation to prove a contradiction, if possible

R6:	D	R5, R22, resolution
R7:	$A \lor C$	R6, R3, resolution
R8:	$\neg B$	R5, R1, resolution

Propositional Logic Word Problem

Given the KB:

R1:	$\neg B \lor A \lor C$
R21:	$A \lor \neg B$
R22:	$A \lor D$
R3:	$\neg D \lor A \lor C$
R4:	$B \lor \neg C$
R5:	$\neg A$

Step 4: Use resolution refutation to prove a contradiction, if possible

R6:	D	R5, R22, resolution
R7:	$A \lor C$	R6, R3, resolution
R8:	$\neg B$	R5, R1, resolution
R9:	$\neg C$	R8, R4, resolution

Propositional Logic Word Problem

Given the KB:

R1:	$\neg B \lor A \lor C$
R21:	$A \lor \neg B$
R22:	$A \lor D$
R3:	$\neg D \lor A \lor C$
R4:	$B \lor \neg C$
R5:	$\neg A$

Step 4: Use resolution refutation to prove a contradiction, if possible

R6:	D	R5, R22, resolution
R7:	$A \lor C$	R6, R3, resolution
R8:	$\neg B$	R5, R21, resolution
R9:	$\neg C$	R8, R4, resolution
R10:	A	R9, R7, resolution

Propositional Logic Word Problem

Given the KB:

R1:	$\neg B \lor A \lor C$
R21:	$A \lor \neg B$
R22:	$A \lor D$
R3:	$\neg D \lor A \lor C$
R4:	$B \lor \neg C$
R5:	$\neg A$

Step 4: Use resolution refutation to prove a contradiction, if possible

R6:	D	R5, R22, resolution	
R7:	$A \lor C$	R6, R3, resolution	
R8:	$\neg B$	R5, R1, resolution	
R9:	$\neg C$	R8, R4, resolution	
R10:	A	R9, R7, resolution	
R11:	{ }	R10, R5, resolution.	Contradiction. So, James is the murderer.

Horn Clauses

- Resolution refutation is complete, but it is also NP-complete
- We can get better performance by restricting our CNF clauses somewhat

- **Horn clause**
  - a *disjunction of literals* of which *at most one is positive*
  - includes definite clauses, in which *exactly one is positive*
    - Ex., $\neg P \vee \neg Q \vee R$, which we can write as $\underbrace{(\overbrace{P \wedge Q}^{\textit{body}}) \Rightarrow \overbrace{R}^{\textit{head}}}_{\textit{rule}}$
    - Ex., $\underbrace{S}_{\textit{fact}}$
  - also includes goal clauses, which have *no positive literals*
    - Ex., $\neg P \vee \neg Q$
    - Ex., $\neg S$

Q: Why do we call these "goal" clauses?

Inferencing with Horn Clauses

- We can perform inferencing (determine entailment) with Horn clauses in time O (|KB|), i.e., linear in the size of the KB using

forward chaining

to prove a proposition Q that's not already in KB

start with KB

loop through every rule in the KB

if its premises (body) is/are satisfied, then "fire" the rule and add its conclusion to the KB (if not already in it)

if just added Q, then done (success)

else, if added at least 1 conclusion to KB in last pass

then loop again

else

exit (failure)

Inferencing with Horn Clauses (2)

- We can also perform inferencing with Horn clauses in time O (|KB|) using

backward chaining

to prove a proposition Q that's not already in KB

start with KB

find all rules that have Q as their conclusion (root rules)

for each such rule

attempt to satisfy each premise

if all premises are satisfied, then fire the rule and add the conclusion to the KB

else, replace each unsatisfied premise with the rules for which such premise is the conclusion and recur to attempt to satisfy such rule

if one root rule cannot ultimately be satisfied, keep trying others until success or no more root rules to satisfy

Q: Which do you think is more efficient, forward or backward chaining?

Inferencing with Horn Clauses (3)

- We can also perform inferencing with Horn clauses in time O (|KB|) using

backward chaining

to prove a proposition Q that's not already in KB

start with KB

find all rules that have Q as their conclusion (root

for each such rule

attempt to satisfy each premise

if all premises are satisfied, then fire the rule ar

to the KB

else, replace each unsatisfied premise with the rules for which such premise is the conclusion and recur to attempt to satisfy such rule

if one root rule cannot ultimately be satisfied, keep trying others until success or no more root rules to satisfy

Q: Which do you think is more efficient, forward or backward chaining?

A: Backtracking search can often succeed in much less than linear time, because it only looks at "relevant" rules, not all rules in the KB