CS5100: Foundations of Artificial Intelligence

Propositional Logic

Dr. Rutu Mulkar-Mehta

Lecture 4

Some slides and images used from Cornell CS4700 course notes, with permission

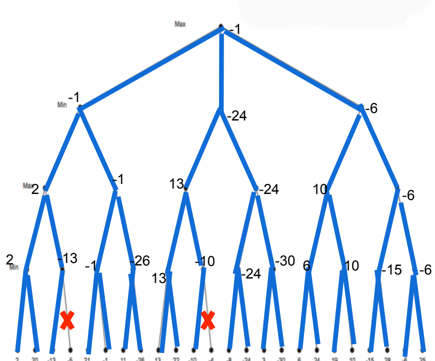
Administrative

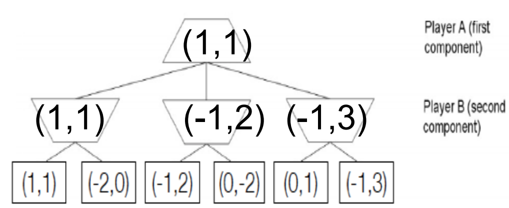
- Tonight at 11:59PM Project 1 is due
- Please email your solutions to:
  - Professor: me@rutumulkar.com
  - TA: k.porterfield7@gmail.com
  - Subject: [PROJECT 1] <FirstName> <LastName>
- Your work will be evaluated in 3 ways:
  - MOSS → general uniqueness of code, and plagiarism detection
  - Autograder → High level running and working of code
  - Readme File → Your understanding of the problem, and partial credit
  - Code Review

Administrative

- Assignment 2 – graded and returned to you
- EC1 – Grading in progress – will be returned next week
- Logic Project: Out next week
  - Enjoy your project free week!
- Next week – In class assignment. You may bring your computer to class and implement it in SWI-Prolog
  - Anyone who cannot bring their laptops to class?

ASSIGNMENT 2 SOLUTIONS





(1pt) What is the best move for Player A? Is this an optimal move for Player B?
(1,1), Not optimal for B

(1pt) Briefly explain why no alpha-beta style pruning is possible in the general non-zero sum case.
Hint : think First about the case where UA (s) = UB (s) for all nodes.
- Dependency between the optimizations
- Possibility of a win-win case

BACK TO LECTURE 5

Outline of the Course

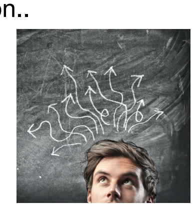
- Search
  - Informed, Uninformed, CSP's, Games
- Logic
  - Today through the next few weeks
  - Very important part of Artificial Intelligence
- Statistics and Learning

Today:

- Knowledge-based agents
- Logic in general - models and entailment
- Propositional (Boolean) logic
- Wumpus world
- Inference rules and theorem proving
  - forward chaining
  - backward chaining
  - resolution
- First Order Logic

Human Agents

- Human Intelligence is achieved by a process of complex reasoning, operating over internal representations of knowledge
- E.g. The light doesn't turn on..
  - Are other lights working?
  - Is there electricity?
  - Is the bulb working?
  - Do the taps have water?
    - (unlikely to consider this)



Logical Agents (Examples)

- Medical diagnosis
  - physician diagnosing a patient infers what disease, based on the knowledge he/she acquired as a student, textbooks, prior cases
- Common sense knowledge / reasoning
  - common everyday assumptions / inferences
  - e.g., "lecture starts at four" infer pm not am
  - when traveling, I assume there is some way to get from the airport to the hotel.
  - e.g. CYC: http://www.cyc.com/ : Commonsense Knowledgebase

Logical Agents

- Agents with some representation of the complex knowledge about the world / its environment, and uses inference to derive new information from that knowledge combined with new inputs (e.g. via perception).

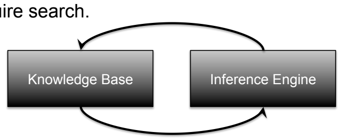
Key issues:
- Knowledge Representation
  How to capture knowledge so that there is least information loss
- Reasoning and inference processes
  How can we use that knowledge and do smart reasoning

Inference Engine

- Why not just run some if-then statements to capture domain logic?
  - Fails quickly when new logic is introduced
  - May not capture all logical relationships
    - e.g. "patient doesn't have cough" (negation)
  - It is domain specific
    - What if we switch to "manufacturing" from "healthcare"

Logical Agents

- Knowledge Base (Domain specific)
  - A set of sentences that describe facts about the world in some formal (representational) language
- Inference Engine (Domain independent)
  - A set of procedures that use the representational language to infer new facts from known ones or answer a variety of KB queries. Inferences typically require search.



Propositional Logic

- Propositional logic P:
  - defines a language for **symbolic reasoning**
- Proposition:
  - a statement that can hold a *true* or *false* value
    Examples of propositions:
    - This class is great.
    - France is in Europe.
    - It is raining outside.
  - How are you? Not a proposition.

Syntax of Propositional Logic

- Formally propositional logic P:
  - Is defined by Syntax + Semantics + Interpretation of P
- Syntax: Symbols (alphabet) in P:
  - Constants: True, False
  - Propositional symbols

  Examples:
  - P
  - Northeastern University is in South Lake Union.
  - It is raining outside, etc.
- A set of connectives in order of operator precedence
  $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Syntax of Propositional Logic

Sentences in the propositional logic:
- Atomic sentences:
  - Constructed from constants and propositional symbols
  - True, False are (atomic) sentences
  - P, Q or *Light in the room is on*, *It rains outside* are all (atomic) sentences
- Composite sentences:
  - Constructed from valid sentences via connectives
  - If A, B are sentences then
    - $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$, $(A \Leftrightarrow B)$ etc.
    - $(A \vee B) \wedge (A \vee \neg B)$ etc.

    are sentences

Syntax

- *Sentence → Atomic Sentence | Complex Sentence*
- *Atomic Sentence → True | False | P | Q | R …*
- *Complex Sentence → ¬ Sentence |*
  Sentence ∧ Sentence
  Sentence ∨ Sentence
  Sentence ⇒ Sentence

Connective Precedence: ¬, ∧ , ∨ , ⇒ , ⇔

(Backus Naur Form or BNF)

Semantics of Propositional Logic

The semantics are the meaning of the sentences.

The semantics in the propositional logic is defined by:

1. Interpretation of propositional symbols and constants
   – Semantics of atomic sentences
2. Through the meaning of connectives
   – Meaning (semantics) of composite sentences

Semantics of Propositional Logic

- A propositional symbol
  – a statement about the world that is either true or false
  – Examples:
    It is raining outside
    Light in the room is on
- Propositional symbols are mapped to one of the two values:
  – True (T), or False (F), depending on whether the symbol is satisfied in the world
  – Light in the room is on -> ***True***
  – Light in the room is on -> ***False***

Semantic Value of Propositional Logic

- The meaning (**value**) of the propositional symbol for a specific interpretation is given by its interpretation
  – Light in the room is on -> *True*
  – V(Light in the room is on) = *True*

  – It is raining outside -> *False*
  – V(It rains outside) = *False*

Semantics of Composite Sentences

We can compute truth value (V) of Propositional Statements using truth tables

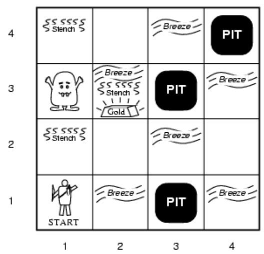
P	Q	¬P	P ∧ Q	P ∨ Q	P ⇒ Q	P ⇔ Q
True	True	False	True	True	True	True
True	False	False	False	True	False	False
False	True	True	False	True	True	False
False	False	True	False	False	True	True

Exercise: 10 minutes

- Consider the following statements:
  – p = It is sunny this afternoon
  – q = it is colder than yesterday
  – r = We will go swimming
  – s = we will take a canoe trip
  – t = We will be home by sunset
- Convert the following to propositional logic:
  1. It is not sunny this afternoon and it is colder than yesterday.
  2. We will go swimming only if it is sunny.
  3. If we do not go swimming then we will take a canoe trip.
  4. If we take a canoe trip, then we will be home by sunset.

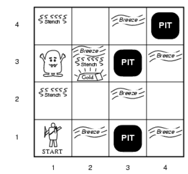
Wumpus World PEAS description

- The **Wumpus World** is a cave consisting of rooms connected by passageways
- Goal
  - Get the gold
  - Avoid the Wumpus (beast)
  - Avoid the bottomless pits
- Sensors:
  - If there is a breeze → you are close to a pit
  - If there is a Stench → you are close to the Wumpus
  - If you see glitter, you are in the same square as the gold

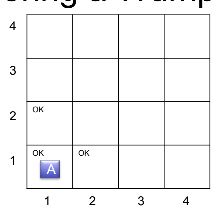


Wumpus World PEAS description

- **Performance measure**
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- **Environment**
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square
- **Actuators**: Left turn, Right turn, Forward, Grab, Release, Shoot
- **Sensors**: Stench, Breeze, Glitter, Bump, Scream



Exploring a Wumpus world



Agent Position: (1,1)

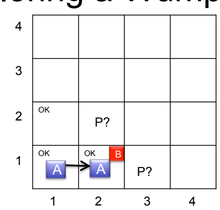
No stench ($\neg S_{11}$), No breeze ($\neg B_{11}$) →

no pit ($\neg P_{11}$) and no Wumpus in (2,1) or (1,2) ($\neg W_{21}$, $\neg W_{12}$)

Agent still alive, so no pit or Wumpus in (1,1).

Agent moves to (2,1)

Exploring a Wumpus world



Agent Position: (2,1)

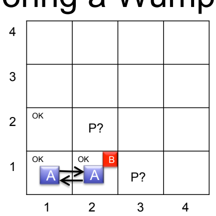
Breeze in (2,1) B_{21} →

there must be a pit in (1,1), (2,2) or (3,1) $P_{11} \vee P_{22} \vee P_{31}$

Player already ruled out (1,1) ($\neg P_{11}$)

Since there is danger in moving to (3,1) and (2,2), player moves back to (1,1)

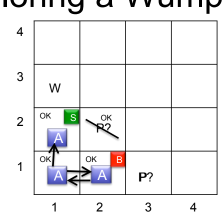
Exploring a Wumpus world



Agent Position: (1,1)

Move to Position (1,2) as that is the safest next place to explore

Exploring a Wumpus world



Agent Position: (1,2)

Stench in this position S_{12} → $W_{13} \vee W_{22} \vee W_{11}$

We know that $\neg W_{11}$ and $\neg W_{22}$

Therefore W_{13}

Next available safe spot → (2,2,) …. and so on

Takeaways

- Agent draws conclusions based on available information
- Conclusion is guaranteed to be correct if the information is correct
- The agent keeps adding new information to its model based on its percepts
- The agent updates its incomplete model based on new percepts

A simple knowledge-based agent

```
function KB-AGENT( percept) returns an action
   static: KB, a knowledge base
           t, a counter, initially 0, indicating time
   TELL(KB, MAKE-PERCEPT-SENTENCE( percept, t))
   action ← ASK(KB, MAKE-ACTION-QUERY(t))
   TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   t ← t + 1
   return action
```

- The agent must be able to:
  - Represent states, actions, etc.
  - Incorporate new percepts
  - Update internal representations of the world
  - Deduce hidden properties of the world
  - Deduce appropriate actions

Wumpus world characterization

- Fully Observable:
  - No – only local perception
- Deterministic:
  - Yes – outcomes exactly specified
- Episodic:
  - No – sequential at the level of actions
- Static:
  - Yes – Wumpus and Pits do not move
- Discrete:
  - Yes
- Single-agent:
  - Yes – Wumpus is essentially a natural feature

Logic in general

- Logic is formal language that is used for representing information such that conclusions can be drawn from it
- 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 ≥ y is a sentence; x2+y > {} is not a sentence
  - x+2 ≥ y is true iff the number x+2 is no less than the number y
  - x+2 ≥ y is true in a world where x = 7, y = 1
  - x+2 ≥ y is false in a world where x = 0, y = 6

Contradiction and Tautology

- Contradiction and Tautology
  - Contradiction is Always ***False***
    - e.g. $P \land \neg P$
  - Tautology is Always ***True***
    - e.g. $P \lor \neg P$

Sound and Complete Inference

- Soundness:
  - An inference algorithm that derives only entailed sentences is called a sound or truth preserving algorithm
- Completeness
  - An inference algorithm is complete if it can derive any sentence that is entailed.

Satisfiability

- A sentence (or set of sentences) is satisfiable if there exists **some interpretation** that makes it true
- An interpretation satisfies a set of sentences if it makes them true

Validity

- A sentence (or set of sentences) is valid if
  - it is true under all interpretations
  - Example: P v ~P

Model, Validity and Satisfiability

- Model: "Possible World" or Mathematical Abstraction of the real world
- A sentence is **satisfiable** if there is at least one interpretation under which the sentence can evaluate to *True*
- A sentence is **valid** if it is *True* in all interpretations

			Satisfiable Sentence	Valid Sentence
P	Q	P ∨ Q	(P ∨ Q) ∧¬Q	((P ∨ Q) ∧¬Q) ⇒ Q
True	True	True	False	True
True	False	True	True	True
False	True	True	False	True
False	False	False	False	True

Entailment

- A \|= B
- All interpretations that satisfy A also satisfy B

Logical Equivalences

- Sentences A and B are logically equivalent if:
  - they are true under exactly the same interpretations
  - A \|= B and B \|= A

Logical equivalence

- Two sentences are logically equivalent iff true in same models: α ≡ ß iff α ⊨ β and β ⊨ α

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

Entailment

- **Entailment** reflects the relation of one fact in the world following from others
  - A |= B
  - Under all interpretations in which A is true, B is true as well
  - All models of A are models of B
  - Whenever A is true, B is true as well
  - A entails B
  - B logically follows from A

43

Models

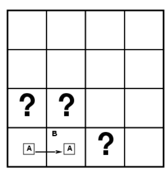
- Logicians typically think in terms of models, which are formally structured worlds with respect to which truth can be evaluated
- We say *m* is a model of a sentence α if α is true in *m*
- *M(α)* is the set of all models of α
- Then KB ⊨ α iff *M(KB)* ⊆ *M*(α)
  - E.g. *KB* = Giants won and Reds won
  - α = Giants won
- If KB is true then α must be true r When KB is false α can be anything

Entailment in the wumpus world

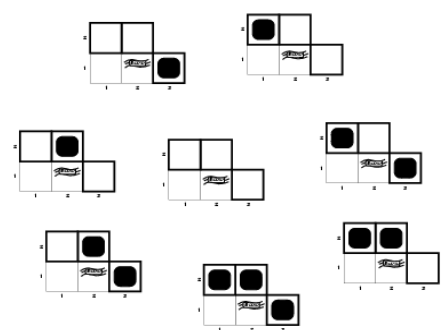
Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

Consider possible models for *KB* assuming only pits

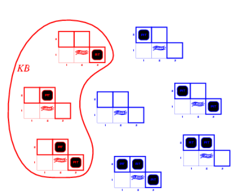
3 Boolean choices ⇒ 8 possible models



Wumpus models

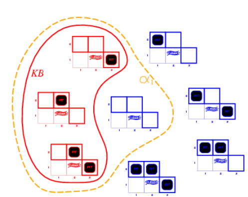


Wumpus models



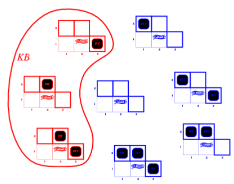
- *KB* = wumpus-world rules + observations

Wumpus models



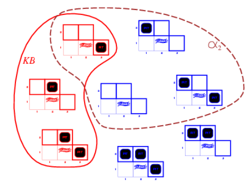
- *KB* = wumpus-world rules + observations
- α_1 = "[1,2] is safe", *KB* ⊨ α_1, proved by model checking

Wumpus models



- *KB* = wumpus-world rules + observations

Wumpus models



- *KB* = wumpus-world rules + observations
- α_2 = "[2,2] is safe", $KB \not\models \alpha_2$

Solving Logical Inference Problems

- Three Approaches:
  - Truth Table Approach
  - Inference Rules Approach
  - Resolution – Refutation Approach

Truth Table Approach

- Problem: KB |= α?
  - We need to check all the possible interpretations for which the KB is true (models of KB) whether α is true for each of them
- Truth Table
  - Enumerates the truth values of sentences for all possible interpretations

52

Truth Table Approach

KB		α
A	B	A ∧ B
F	F	F
F	F	F
F	T	F
F	T	F
T	F	F
T	F	F
T	T	T
T	T	T

- Problem: KB |= α?
  - A, B entails A∧B?
- Solution
  - Generate table for all possible interpretations
  - Check whether α is true whenever KB is true

A,B, Entails A∧B

53

Truth Table Approach

			KB	α
A	B	C	A ∧ C	B ∧ C
F	F	F	F	F
F	F	T	F	F
F	T	F	F	F
F	T	T	F	T
T	F	F	F	F
T	F	T	T	F
T	T	F	F	F
T	T	T	T	T

- Problem: KB |= α ?
  - KB = A ^ C, C
  - α = B ^ C

A^C, C does not entail B∧C

54

Truth Table Approach

- The Truth Table approach is sound and complete for Propositional Logic

Limitations of Truth Table Approach

- Search Space for truth tables is exponential.
  - 3 variables – 2^3 = 8
  - 10 variables – 2^ 10 = 1024
  - KB is true only in a small set of interpretations
- We need to make the process more efficient if we want to use this on a larger scale

Inference Rules Approach

- …
- We need to make the process more efficient if we want to use this on a larger scale
- Solution:
  - Check only the entries where KB = *True*
  - This is the idea behind the *inference rules approach*

Inference Rules for Logic

- **Modus Ponens**

$$\frac{A \Rightarrow B,\ A}{B}$$

Premise

Conclusion

- If the premise is true then the conclusion is also true
- Exercise – Can you prove this using the truth table?

A	B	A => B
True	True	
True	False	
False	True	
False	False	

$$\frac{A \Rightarrow B,\ A}{B}$$

Inference Rules for Logic

- And Elimination

$$\frac{A \wedge B}{A}$$

From the conjunction any of the conjuncts can be inferred

- Bi-Conditional Elimination

$$\frac{A \Leftrightarrow B}{(A \Rightarrow B) \wedge (B \Rightarrow A)}$$

- Double Negation Elimination

$$\frac{\neg\neg A}{A}$$

Inference Rules for Logic: Resolution

- Resolution takes two clauses and produces a new clause containing all the literals of the two original clauses except the two complementary literals

$$\frac{A \vee B,\ \neg A}{B}$$

Solve using Inference

- KB: $(B_{11} \Leftrightarrow (P_{12} \vee P_{21}))$; $\neg B_{11}$
- Prove: $\neg P_{12}$.

KB:
1. $\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
2. $\neg P_{1,2} \vee B_{1,1}$
3. $\neg P_{2,1} \vee B_{1,1}$
4. $\neg B_{11}$
5. $\neg P_{2,1}$ (From (3) and (4) And-elimination)
6. $\neg P_{1,2}$ (From (2) and (4 And Elimination)

Proved!

Resolution Example

winter V summer
¬winter V cold

- Either winter or ¬winter is true, so we know that summer or cold is true
- Resolution rule:
  - Given: $P_1 \vee P_2 \vee P_3 \ldots \vee P_n$, and $\neg P_1 \vee Q_1 \ldots \vee Q_m$
  - Conclude: $P_2 \vee P_3 \ldots \vee P_n \vee Q_1 \ldots \vee Q_m$
    Complementary literals P_1 and $\neg P_1$ "cancel out"

Resolution in Wumpus World

- There is a pit at 2,1 or 2,3 or 1,2 or 3,2
  - $P_{21} \vee P_{23} \vee P_{12} \vee P_{32}$
- There is no pit at 2,1
  - $\neg P_{21}$
- Therefore (by resolution) the pit must be at 2,3 or 1,2 or 3,2
  - $P_{23} \vee P_{12} \vee P_{32}$

$$\frac{P_{21} \vee P_{23} \vee P_{12} \vee P_{32},\ \neg P_{21}}{P_{23} \vee P_{12} \vee P_{32}}$$

Resolution

- Any complete search algorithm, applying only the resolution rule, can derive any conclusion entailed by any KB in propositional logic.

Proof using Resolution

- To try to prove P from KB:
  - Convert KB and P into CNF
    (Conjunctive Normal Form – Conjunction of clauses, clauses joined by 'and' relation)
  - To prove P, prove KB ∧ ¬P is contradictory (empty clause)
  - Specifically, apply resolution using a complete search algorithm until:
    - No new clauses can be added, (KB does not entail P)
    - The empty clause is derived (KB does entail P).

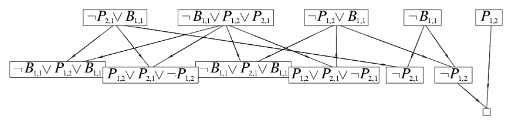
Resolution algorithm

- Proof by contradiction, i.e., show $KB \wedge \neg\alpha$ unsatisfiable

```
function PL-RESOLUTION(KB, α) returns true or false
    clauses ← the set of clauses in the CNF representation of KB ∧ ¬α
    new ← { }
    loop do
        for each Ci, Cj in clauses do
            resolvents ← PL-RESOLVE(Ci, Cj)
            if resolvents contains the empty clause then return true
            new ← new ∪ resolvents
        if new ⊆ clauses then return false
        clauses ← clauses ∪ new
```

Resolution example

- $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$
- $\alpha = \neg P_{1,2}$



Simple Resolution EG

- When the agent is in 1,1, there is no breeze, so there can be no pits in neighboring squares
  - KB: $(B_{11} \Leftrightarrow (P_{12} \vee P_{21}))$; $\neg B_{11}$
  - Prove: $\neg P_{12}$.

69

Conversion to CNF

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$
3. Move $\neg$ inwards using de Morgan's rules and double-negation:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$
4. Apply distributivity law ($\wedge$ over $\vee$) and flatten:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Forward and backward chaining

- Horn Form (restricted)
  KB = conjunction of Horn clauses
  Horn Clauss is a clause with at most one positive literal
  - Horn clause =
    - proposition symbol; or
    - (conjunction of symbols) $\Rightarrow$ symbol
  - E.g., $C \wedge (B \Rightarrow A) \wedge (C \wedge D \Rightarrow B)$
- Modus Ponens (for Horn Form): complete for Horn KBs

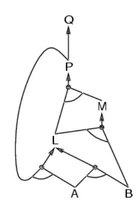
$$\frac{\alpha_1, \ldots, \alpha_n, \qquad \alpha_1 \wedge \ldots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

- Can be used with forward chaining or backward chaining.
- These algorithms are very natural and run in linear time

Forward chaining

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

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



Forward chaining algorithm

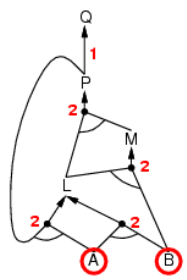
```
function PL-FC-ENTAILS?(KB, q) returns true or false
  local variables: count, a table, indexed by clause, initially the number of premises
                   inferred, a table, indexed by symbol, each entry initially false
                   agenda, a list of symbols, initially the symbols known to be true

   while agenda is not empty do
      p ← POP(agenda)
      unless inferred[p] do
         inferred[p] ← true
         for each Horn clause c in whose premise p appears do
            decrement count[c]
            if count[c] = 0 then do
               if HEAD[c] = q then return true
               PUSH(HEAD[c], agenda)
   return false
```

- Forward chaining is sound and complete for Horn KB

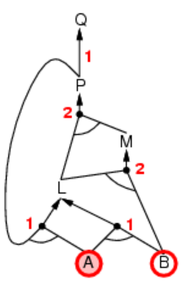
Forward chaining example

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



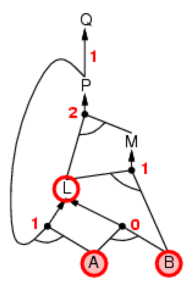
Forward chaining example

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



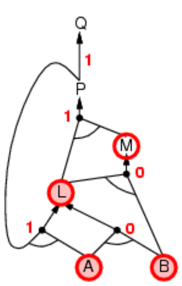
Forward chaining example

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



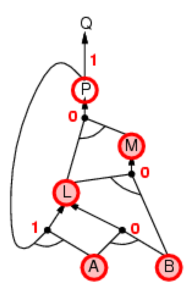
Forward chaining example

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



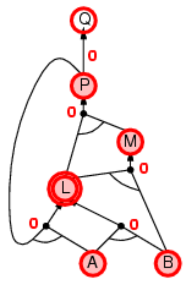
Forward chaining example

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



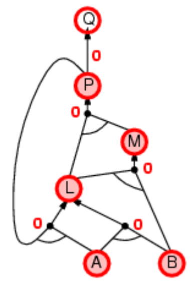
Forward chaining example

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

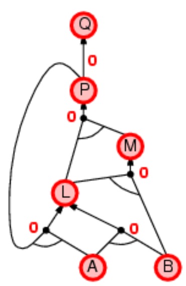


Forward chaining example

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



Forward chaining example



Proof of completeness

- FC derives every atomic sentence that is entailed by *KB*
  1. FC reaches a fixed point where no new atomic sentences are derived
  2. Consider the final state as a model *m*, assigning true/false to symbols
  3. Every clause in the original *KB* is true in *m*
     $a_1 \wedge \ldots \wedge a_k \Rightarrow b$
  4. Hence *m* is a model of *KB*
  5. If $KB \models q$, *q* is true in every model of *KB*, including *m*

Backward chaining

Idea: work backwards from the query *q*:

- to prove *q* by BC,
  - check if *q* is known already, or
  - prove by BC all premises of some rule concluding *q*

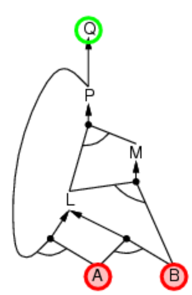
Avoid loops: check if new subgoal is already on the goal stack

Avoid repeated work: check if new subgoal

1. has already been proved true, or
2. has already failed

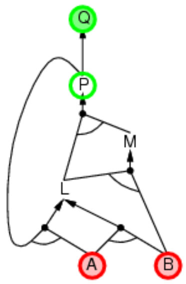
Backward chaining example

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



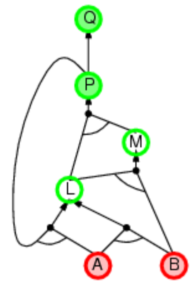
Backward chaining example

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



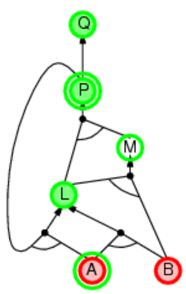
Backward chaining example

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



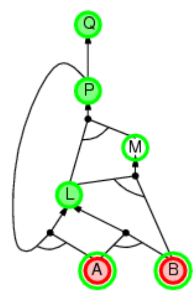
Backward chaining example

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



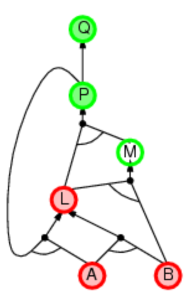
Backward chaining example

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



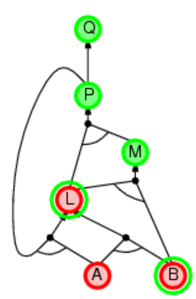
Backward chaining example

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



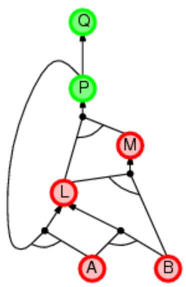
Backward chaining example

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



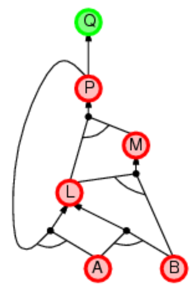
Backward chaining example

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



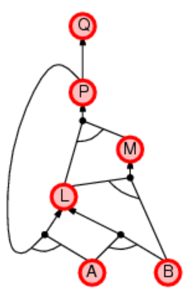
Backward chaining example

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



Backward chaining example

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



Forward 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

Expressiveness limitation of propositional logic

- KB contains "physics" sentences for every single square
- For every time t and every location $[x,y]$,

$L^t_{x,y} \wedge FacingRight^t \wedge Forward^t \Rightarrow L^t_{x+1,y}$

- Rapid proliferation of clauses

Proposition Logic 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
- Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.
- Resolution is complete for propositional logic
  Forward, backward chaining are linear-time, complete for Horn clauses
- Propositional logic lacks expressive power

Exercises

1. According to some political pundits, a person who is radical (R) is electable (E) if he/she is conservative (C), but otherwise is not electable.

   Which of the following represent this assertion?

   a) $(R \wedge E) \Leftrightarrow C$
   b) $R \Rightarrow (E \Leftrightarrow C)$
   c) $R \Rightarrow ((C \Rightarrow E) \vee \sim E)$

Next Week

- We will cover concepts on First Order Logic
- Get a head start by installing: SWI-Prolog
  - http://www.swi-prolog.org/Download.html
- Quick SWI-Prolog Demo