Chapter 7 Logical Agents

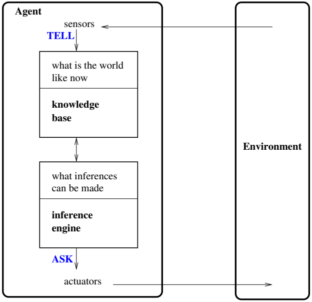
CS4811 - Artificial Intelligence

Nilufer Onder
Department of Computer Science
Michigan Technological University

Outline

- Knowledge-based agents
- Wumpus world
- Logic in general: models and entailment
- Propositional Logic
- Equivalence, validity, satisfiability
- Inference rules and theorem proving
  - Forward chaining
  - Backward chaining
  - Resolution

A knowledge-based agent



The *knowledge base*

- contains domain-specific information
- is a set of sentences in a formal language
  e.g., propositional logic,
  first order logic

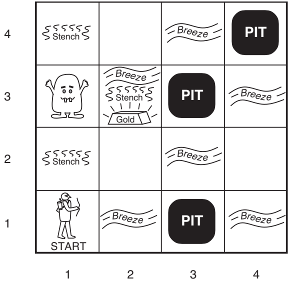
The *inference engine*

- contains domain-independent algorithms
- is a set of inference algorithms
  e.g., model checking,
  forward checking,
  resolution refutation

The Wumpus world

Environment

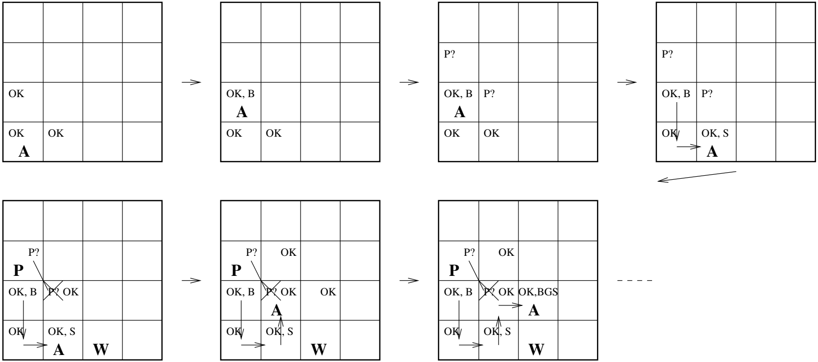
- interconnected caves
- searching for gold
- squares adjacent to Wumpus are smelly
- squares adjacent to pits are breezy
- pits are bottomless
  will trap the agent but not the Wumpus
- glitter is seen if square has gold
- shooting kills wumpus if you are facing it
- shooting uses up the only arrow
- grabbing picks up gold
- releasing drops the gold



Actuators and percepts:

- Actions: left turn, right turn, forward, grab, release, shoot
- Sensors: breeze, glitter, smell

Exploring the Wumpus world



Logic in general

- *Logics* are formal languages for representing information such that conclusions can be drawn
- *Syntax* defines the sentences in the language
- *Semantics* define the “meaning” of sentences, i.e., define *truth* of a sentence in a world
- E.g., the language of arithmetic
  - $x + 2 \geq y$ is a sentence; $x2 + y >$ is not a sentence
  - $x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y
  - $x + 2 \geq y$ is true in a world where $x = 7, y = 1$
  - $x + 2 \geq y$ is false in a world where $x = 0, y = 6$

Logic in general (Wumpus world example)

- Logics are formal languages for representing information such that conclusions can be drawn
- Syntax defines the sentences in the language
- Semantics define the "meaning" of sentences, i.e., define truth of a sentence in a world
- E.g., a language for the Wumpus world
  - $P_{i,j}$ is a proposition
    It's true if there is a pit in [i,j]
  - $B_{i,j}$ a proposition
    It's true if there is a breeze in [i,j]
  - The following are sentences that represent
    "Pits cause breezes in adjacent squares" or
    "A square is breezy if and only if there is an adjacent pit"
    $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
    $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

Entailment

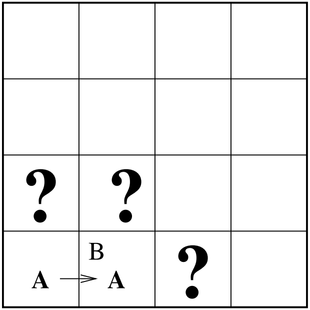
- *Entailment* means one thing follows from another:
  $KB \models \alpha$
- Knowledge base KB entails sentence α
  if and only if
  α is true in all worlds where KB is true
- For example:
  $x + y = 4$ entails $4 = x + y$
- In the Wumpus world,
  "no breeze in [1,1]" entails
  no pit in [1,2] and no pit in [2,1]
- Entailment is a relationship between sentences (i.e., syntax)
  that is based on semantics

Models

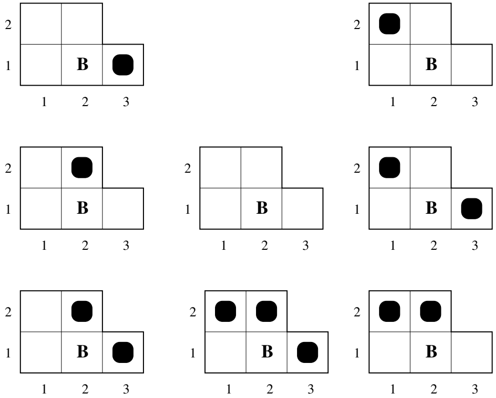
- Logicians typically think in terms of *models*, which are formally structured worlds with respect to which truth can be evaluated
- We say *m* is a model of a sentence α if *alpha* is true in *m*
- $M(\alpha)$ is the set of all models of α,
  i.e., all the worlds in which α is true
- Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$
- For example, in every world where there is no breeze in [1,1], there is no pit in [1,2] and [2,1]

Entailment in the Wumpus world

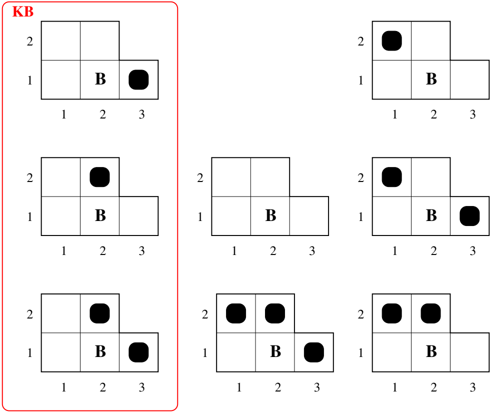
- Consider the situation after detecting nothing in [1,1],
  moving right to [2,1], and
  detecting a breeze in [2,1]
- Write all the possible models for the cells with questions marks
  (only for pits)
- There are 3 cells with Boolean choices (pit or no pit) so
  there are 8 possible models



Wumpus models

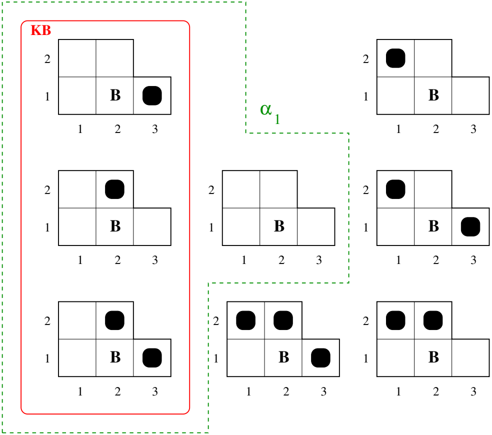


Knowledge base



KB = wumpus-world rules + observations

Model checking α_1

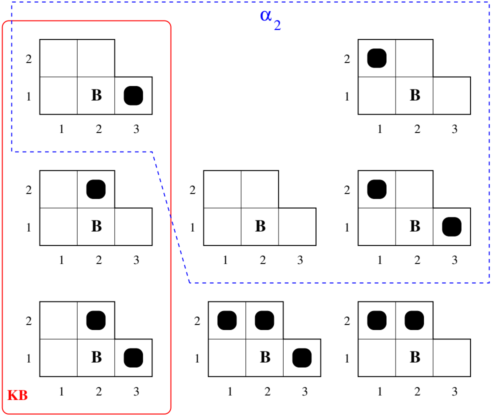


KB = wumpus-world rules + observations

α_1 = "[1,2] is safe"

KB $\models \alpha_1$, proved by model checking

Model checking α_2



KB = wumpus-world rules + observations

α_2 = "[2,2] is safe"

KB $\not\models \alpha_2$, the knowledge base does not entail α_2

Inference

- $KB \vdash_i \alpha$ means
  sentence α can be derived from KB by procedure i
- Consequences of KB are a haystack; α is a needle
  Entailment is a needle in haystack; inference is finding it
- *Soundness*: i is sound if
  whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$
- *Completeness*: i is complete if
  whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$
- Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure
  i.e., the procedure will answer any question whose answer follows from what is known by the KB

Propositional logic: Syntax

Propositional logic is the simplest logic–illustrates basic ideas

The proposition symbols such as P_1, P_2 are sentences

If S is a sentence, $\neg S$ is a sentence (*negation*)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (*conjunction*)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (*disjunction*)

If S_1 and S_2 are sentences, $S_1 \rightarrow S_2$ is a sentence (*implication*)

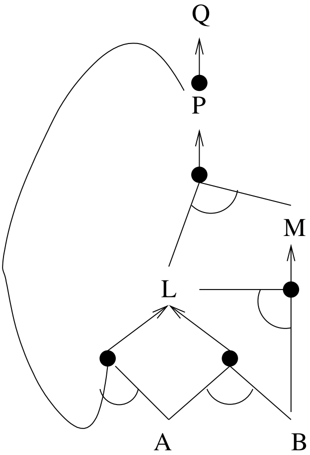
If S_1 and S_2 are sentences, $S_1 \rightarrow S_2$ is a sentence (*biconditional*)

The forward chaining algorithm

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

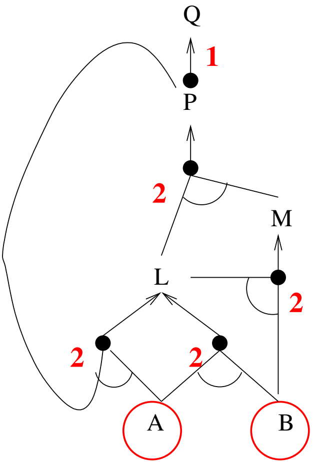
```
function PL-FC-ENTAILS?(KB, q) returns true or false
   inputs: KB, the knowledge base, a set of propositional Horn clauses
           q, the query, a proposition symbol
   local variables: count, a table, indexed by clause, initially the number of premises
                    inferred, a table, indexed by symbol, each entry initially false
                    agenda, a list of symbols, initially the symbols known in KB

   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 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

Initially

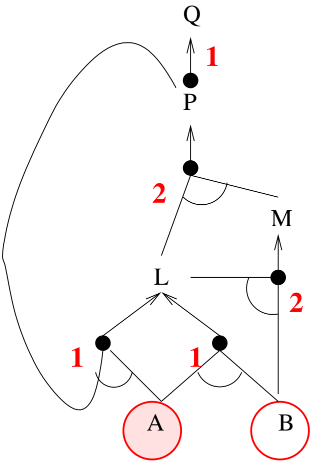


Agenda: A, B

Inferred: $\emptyset$

premise	conclusion	count
P	Q	1
$L \wedge M$	P	2
$B \wedge L$	M	2
$A \wedge P$	L	2
$A \wedge B$	L	2

Take A

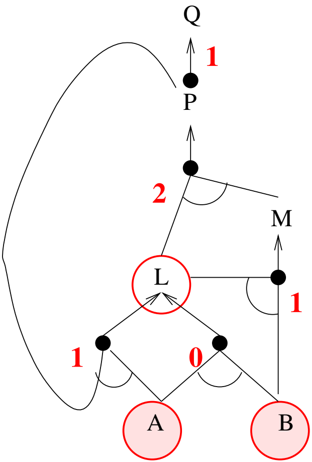


Agenda: B

Inferred: A

premise	conclusion	count
P	Q	1
$L \wedge M$	P	2
$B \wedge L$	M	2
$A \wedge P$	L	1
$A \wedge B$	L	1

Take B

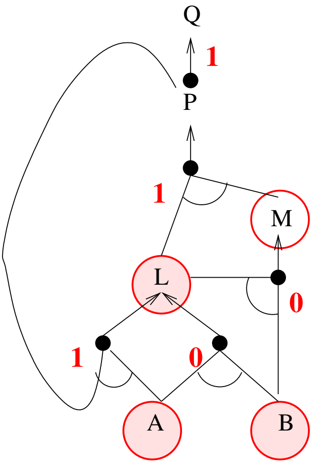


Agenda: L

Inferred: A, B

premise	conclusion	count
P	Q	1
$L \wedge M$	P	2
$B \wedge L$	M	1
$A \wedge P$	L	1
$A \wedge B$	L	0

Take L

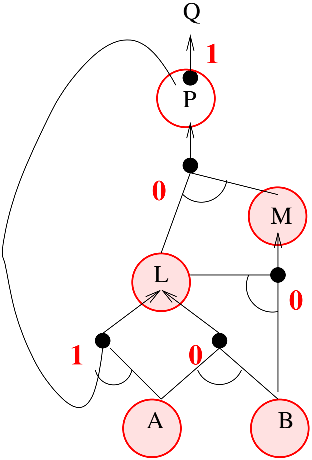


Agenda: M

Inferred: A, B, L

premise	conclusion	count
P	Q	1
$L \wedge M$	P	1
$B \wedge L$	M	0
$A \wedge P$	L	1
$A \wedge B$	L	0

Take M

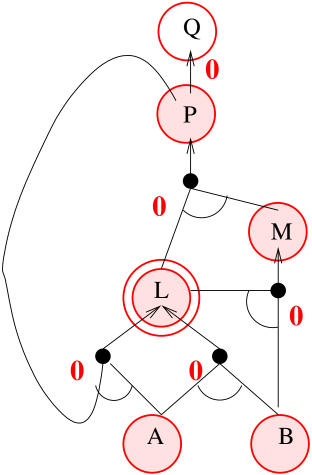


Agenda: P

Inferred: A, B, L, M

premise	conclusion	count
P	Q	1
$L \wedge M$	P	0
$B \wedge L$	M	0
$A \wedge P$	L	1
$A \wedge B$	L	0

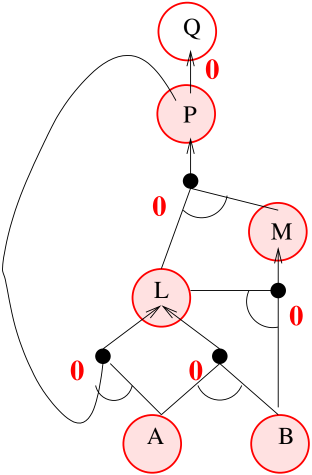
Take P



Agenda: Q
Inferred: A, B, L, M, P

premise	conclusion	count
P	Q	0
$L \wedge M$	P	0
$B \wedge L$	M	0
$A \wedge P$	L	0
$A \wedge B$	L	0

L will not be inferred the second time

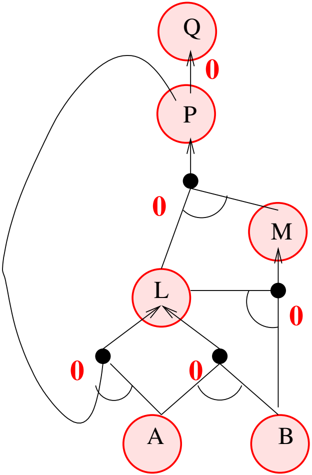


Agenda: Q

Inferred: A, B, L, M, P

premise	conclusion	count
P	Q	0
$L \wedge M$	P	0
$B \wedge L$	M	0
$A \wedge P$	L	0
$A \wedge B$	L	0

Take Q



Agenda: $\emptyset$
Inferred: A, B, L, M, P, Q

premise	conclusion	count
P	Q	0
$L \wedge M$	P	0
$B \wedge L$	M	0
$A \wedge P$	L	0
$A \wedge B$	L	0

Party time! Scenario 1

- If Chris goes to a party, Pat also does
- If Pat goes to party, Pat can't study
- If Pat can't study, Pat fails
- Chris went to a party

Can we prove Pat will fail using forward chaining?

Party time! Scenario 2

- If Chris goes to a party, Pat also does
- If Chris doesn't go to the party, Pat will
- If Pat goes to party, Pat can't study
- If Pat can't study, Pat fails

Can we prove Pat will fail using forward chaining?

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 with respect to models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundess: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
  - optimal decisions depend on information state, not real state
- Forward and backward chaining are linear-time, complete for Horn clauses
- Resolution is complete for propositional logic
- Propositional logic lacks expressive power

Sources for the slides

- AIMA textbook (3^{rd} edition)
- AIMA slides (http://aima.cs.berkeley.edu/)