Artificial Intelligence

Logical Agents

Logical Agents

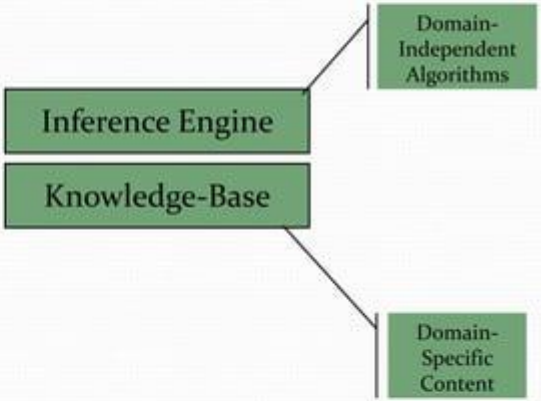
- Humans can know "things" and "reason"
  - Representation: How are the things stored?
  - Reasoning: How is the knowledge used?
    - To solve a problem...
    - To generate more knowledge...
- Knowledge and reasoning are important to artificial agents because they enable successful behaviors difficult to achieve otherwise
  - Useful in partially observable environments
- Can benefit from knowledge in very general forms, combining and recombining information

Knowledge-Based Agents

- Central component of a Knowledge-Based Agent is a *Knowledge-Base*
  - A set of sentences in a formal language
    - Sentences are expressed using a knowledge representation language
- Two generic functions:
  - TELL - add new sentences (facts) to the Knowledge Base
    - "Tell it what it needs to know"
  - ASK - query what is known from the Knowledge Base
    - "Ask what to do next"

Knowledge-Based Agents

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



Knowledge-Based Agents

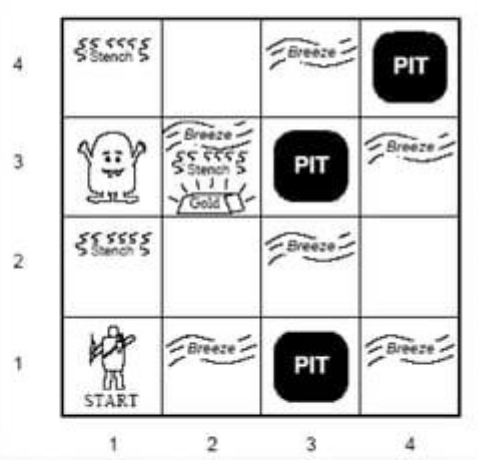
```
function KB-AGENT(percept) returns an action
  static: KB, a knowledge base
          t, a counter, initially 0, indicating time

  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
```

Knowledge-Based Agents

- Declarative
  - You can build a knowledge-based agent simply by "TELLing" it what it needs to know
- Procedural
  - Encode desired behaviors directly as program code
    - Minimizing the role of explicit representation and reasoning can result in a much more efficient system

Wumpus World

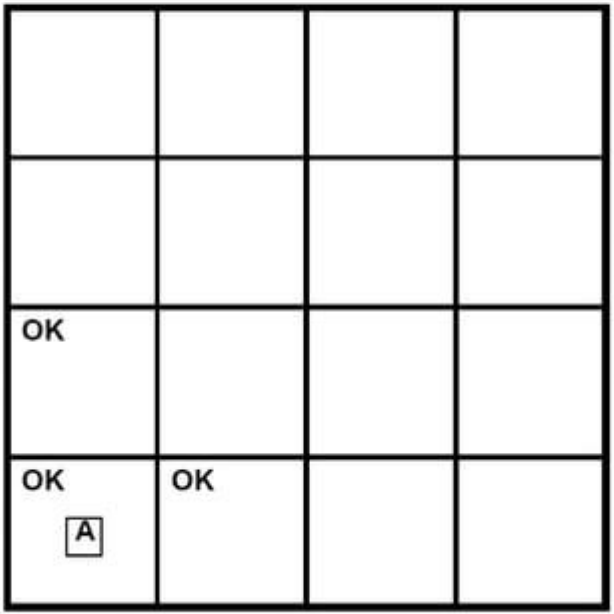
- Performance Measure
  - Gold +1000, Death – 1000
  - Step -1, Use arrow -10
- Environment
  - Square adjacent to the Wumpus are smelly
  - Squares adjacent to the pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills Wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up the gold if in the same square
  - Releasing drops the gold in the same square
- Actuators
  - Left turn, right turn, forward, grab, release, shoot
- Sensors
  - Breeze, glitter, and smell



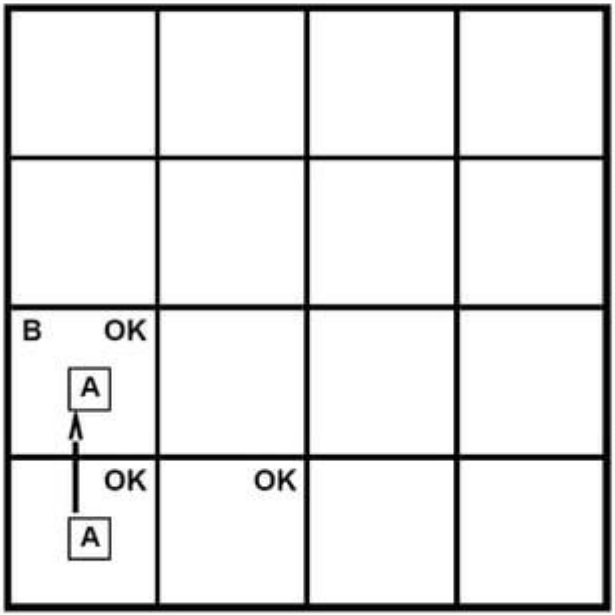
Wumpus World

- Characterization of Wumpus World
  - Observable
    - partial, only local perception
  - Deterministic
    - Yes, outcomes are specified
  - Episodic
    - No, sequential at the level of actions
  - Static
    - Yes, Wumpus and pits do not move
  - Discrete
    - Yes
  - Single Agent
    - Yes

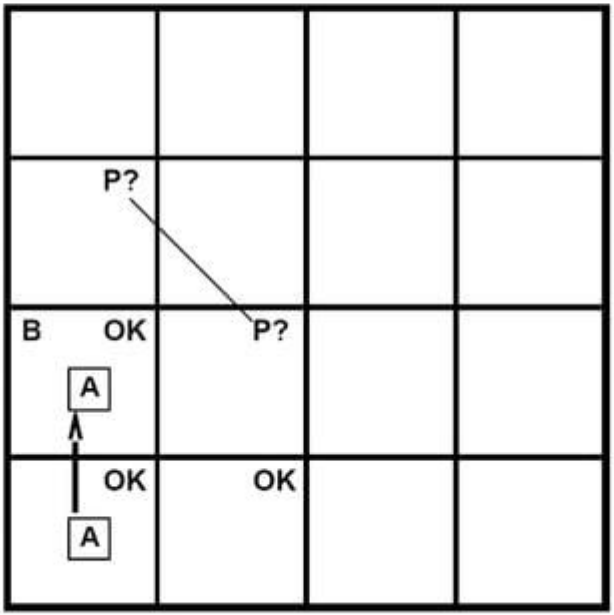
Wumpus World



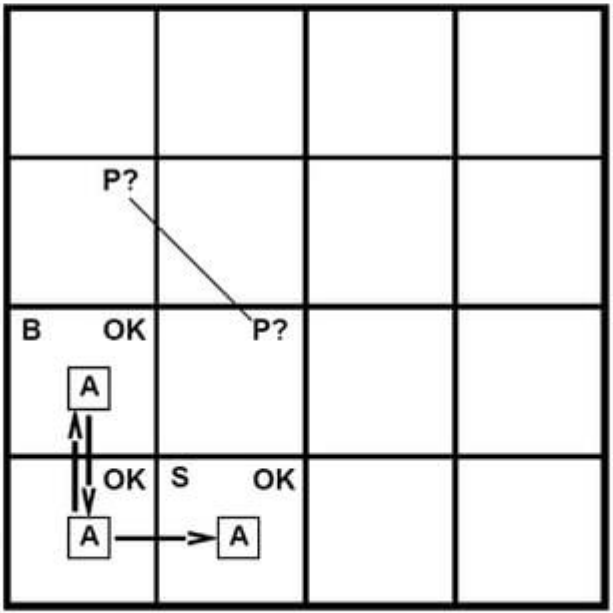
Wumpus World



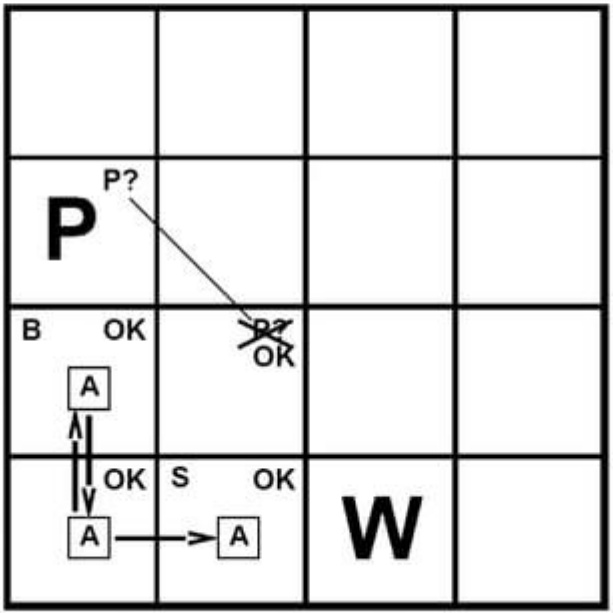
Wumpus World



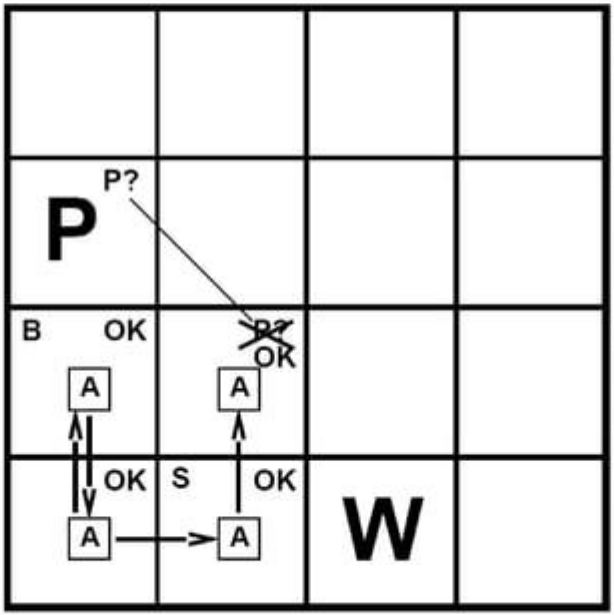
Wumpus World



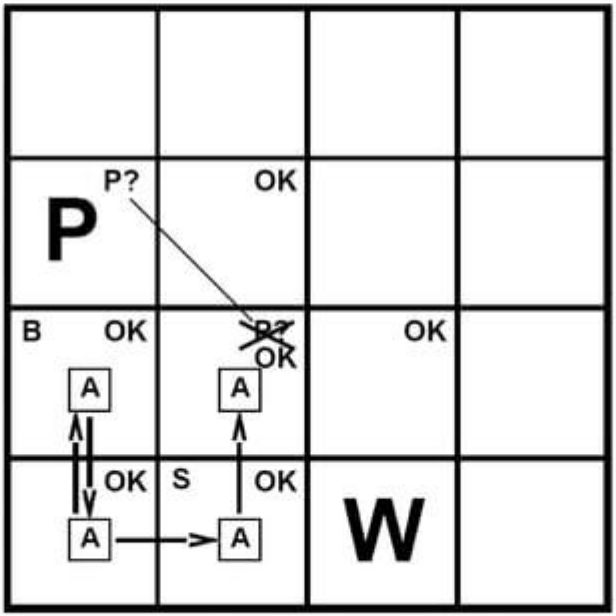
Wumpus World



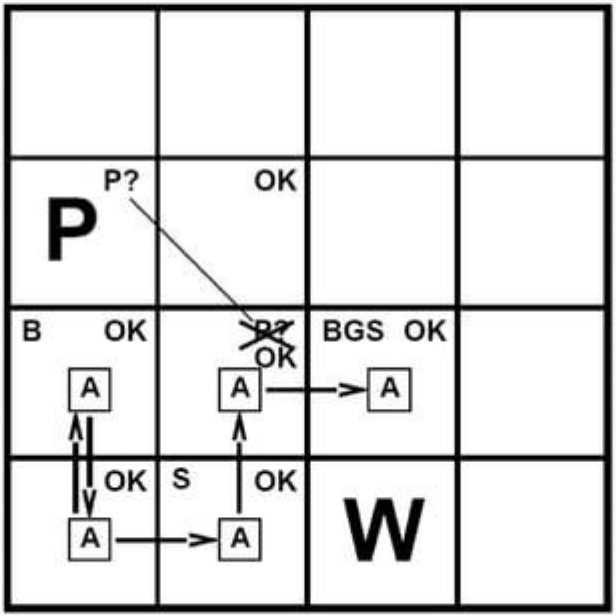
Wumpus World



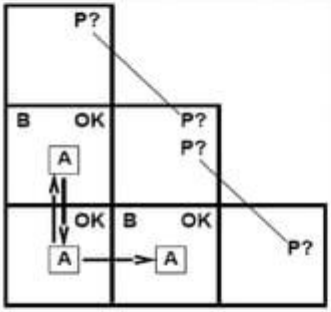
Wumpus World



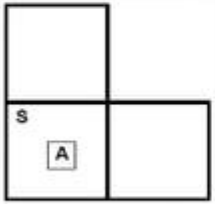
Wumpus World



Other Sticky Situations



- Breeze in (1,2) and (2,1)
  - No safe actions



- Smell in (1,1)
  - Cannot move

Logic

- Knowledge bases consist of sentences in a formal language
  - Syntax
    - Sentences are well formed
  - Semantics
    - The "meaning" of the sentence
    - ***The truth of each sentence with respect to each possible world (model)***

- Example:

  x + 2 >= y is a sentence

  x2 + y > is not a sentence

  x + 2 >= y is true iff x + 2 is no less than y

  x + 2 >= y is true in a world where x = 7, y=1

  x + 2 >= y is false in world where x = 0, y =6

Logic

- ***Entailment*** means that one thing follows logically from another

  $\alpha \models \beta$

- $\alpha \models \beta$ iff in every model in which α is true, β is also true

- if α is true, then β must be true

- the truth of β is "contained" in the truth of α

Logic

- Example:
  - A Knowledge Base containing
    - "Cleveland won"
    - "Dallas won"
    - Entails...
      - "Either Cleveland won or Dallas won"

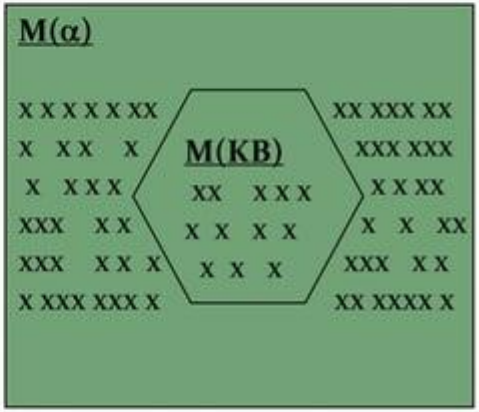
- Example:

  $x + y = 4$ entails $4 = x + y$

Logic

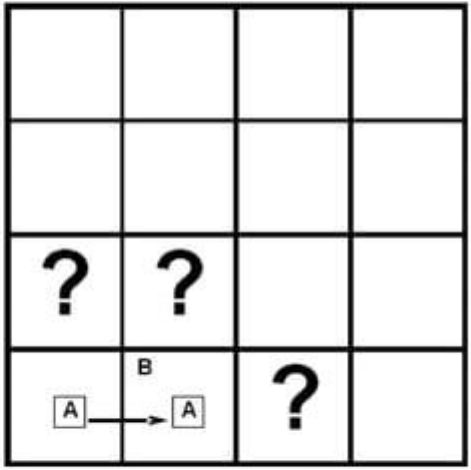
- A model is a formally structured world with respect to which truth can be evaluated
  - M is a model of sentence α if α is true in m

- Then KB $\models \alpha$ if $M(KB) \subseteq M(\alpha)$

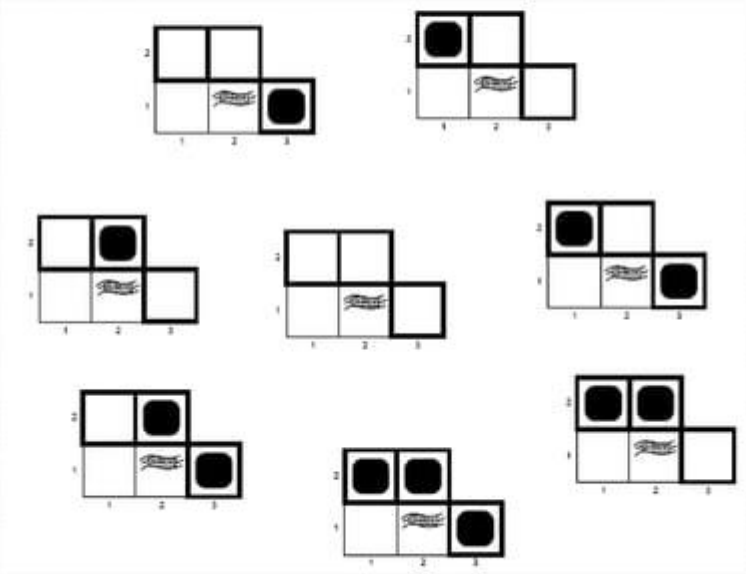


Logic

- Entailment in Wumpus World
- Situation after detecting nothing in [1,1], moving right, breeze in [2,1]
- Consider possible models for ? assuming only pits
- 3 Boolean choices => 8 possible models



Logic



Logic

- ***Inference*** is the process of deriving a specific sentence from a KB (where the sentence must be entailed by the KB)
  - KB $|-_i \alpha$ = sentence α can be derived from KB by procedure I
- "KB's are a haystack"
  - Entailment = needle in haystack
  - Inference = finding it

Logic

- Soundness
  - i is sound if...
  - whenever KB $|-_i \alpha$ is true, KB $|= \alpha$ is true
- Completeness
  - i is complete if
  - whenever KB $|= \alpha$ is true, KB $|-_i \alpha$ is true
- 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

Propositional Logic

- AKA Boolean Logic
- False and True
- Proposition symbols P_1, P_2, etc are sentences
- NOT: If S_1 is a sentence, then $\neg S_1$ is a sentence (negation)
- AND: If S_1, S_2 are sentences, then $S_1 \wedge S_2$ is a sentence (conjunction)
- OR: If S_1, S_2 are sentences, then $S_1 \vee S_2$ is a sentence (disjunction)
- IMPLIES: If S_1, S_2 are sentences, then $S_1 \Rightarrow S_2$ is a sentence (implication)
- IFF: If S_1, S_2 are sentences, then $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Propositional Logic

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

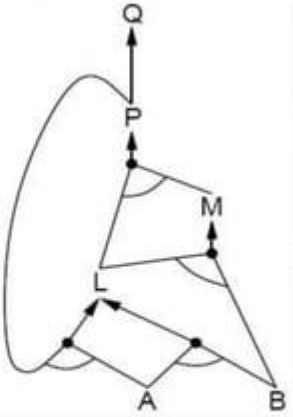
Reasoning with Horn Clauses

- Forward Chaining
  - For each new piece of data, generate all new facts, until the desired fact is generated
  - Data-directed reasoning
- Backward Chaining
  - To prove the goal, find a clause that contains the goal as its head, and prove the body recursively
  - (Backtrack when you chose the wrong clause)
  - Goal-directed reasoning

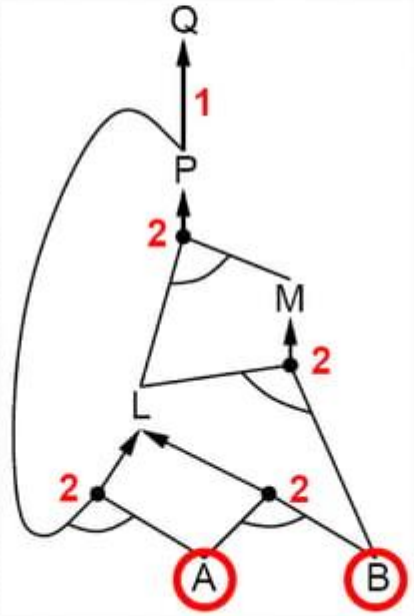
Forward Chaining

- AND-OR Graph
  - multiple links joined by an arc indicate conjunction – every link must be proved
  - multiple links without an arc indicate disjunction – any link can be proved

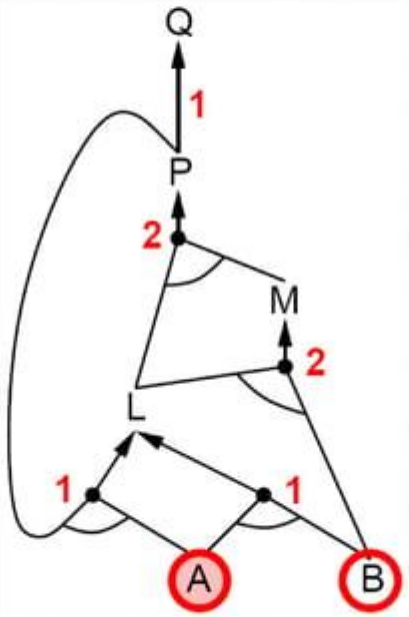
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



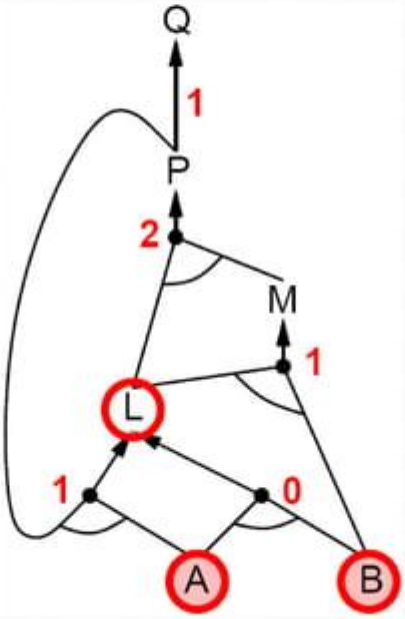
Forward Chaining



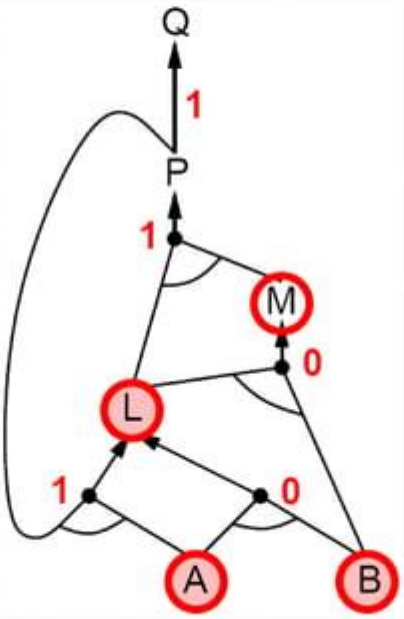
Forward Chaining



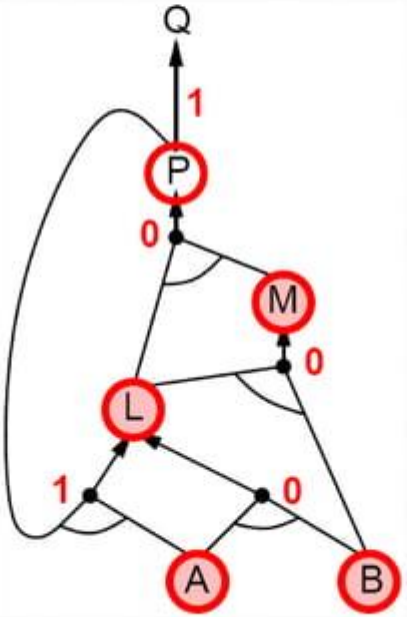
Forward Chaining



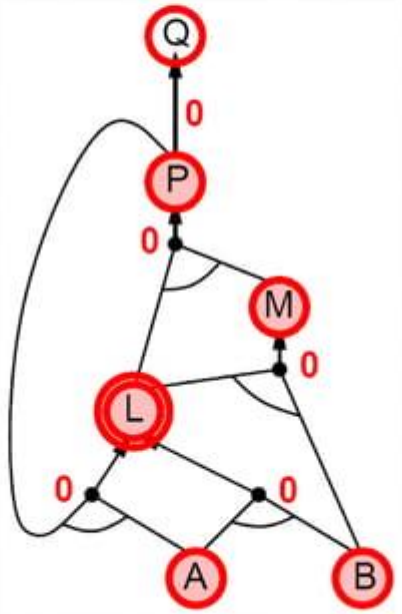
Forward Chaining



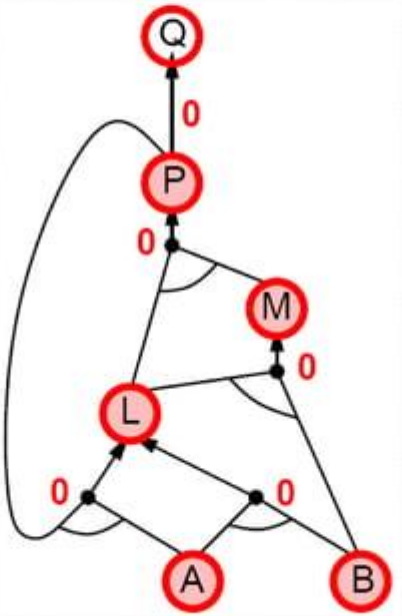
Forward Chaining



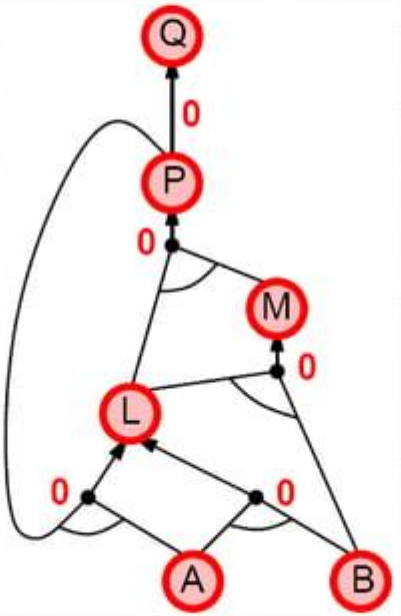
Forward Chaining



Forward Chaining



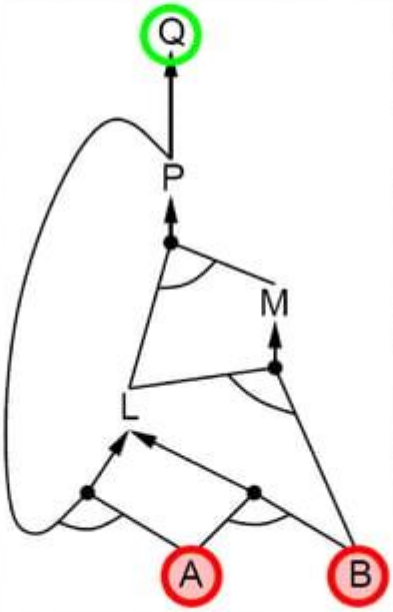
Forward Chaining



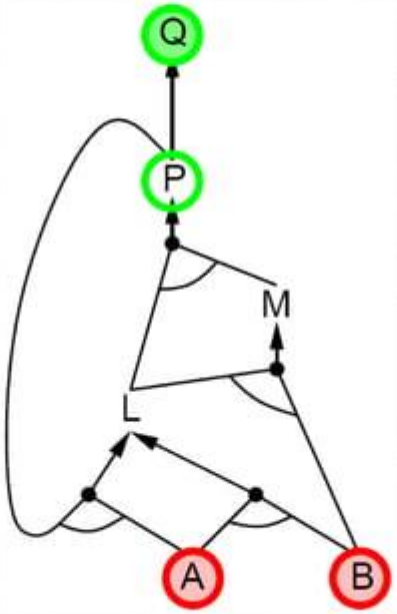
Backward Chaining

- 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
  - Has already been proved true, or
  - Has already failed

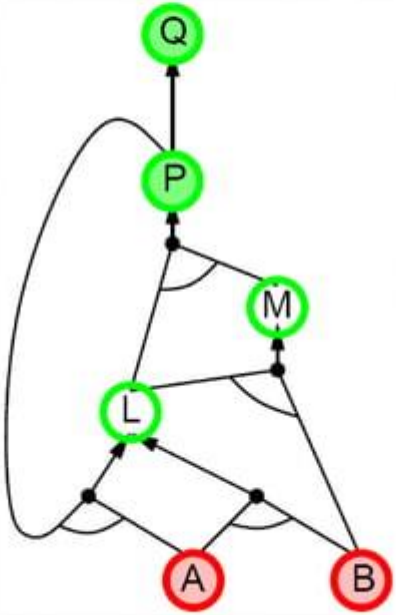
Backward Chaining



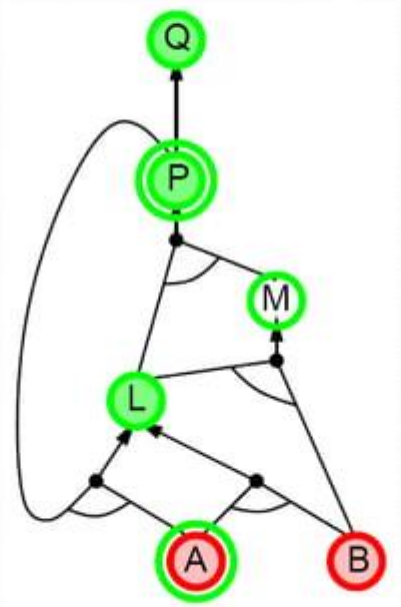
Backward Chaining



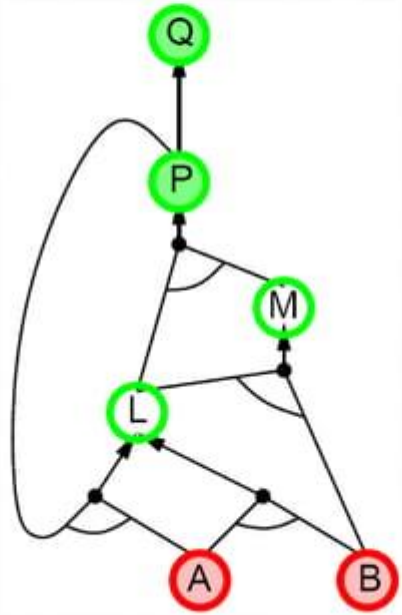
Backward Chaining



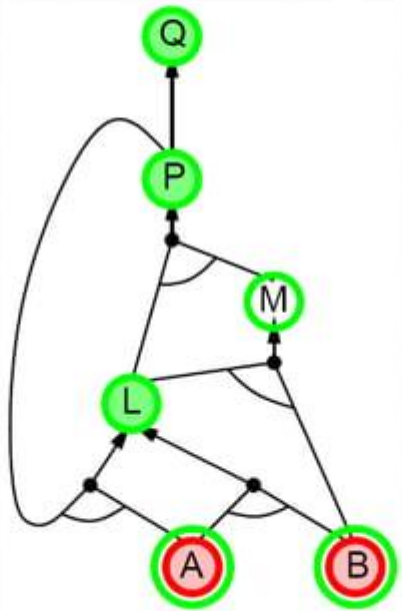
Backward Chaining



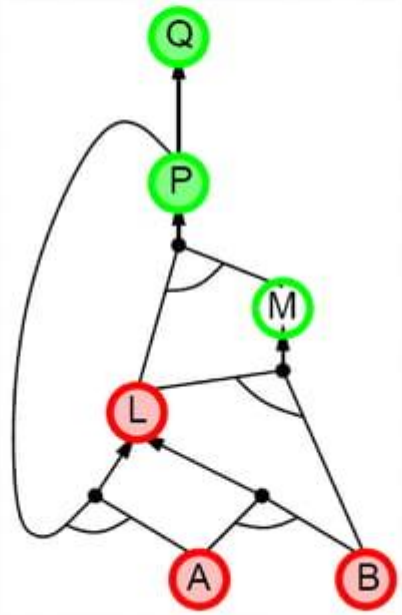
Backward Chaining



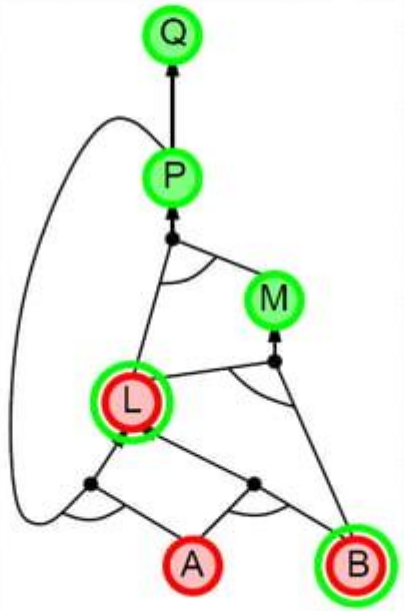
Backward Chaining



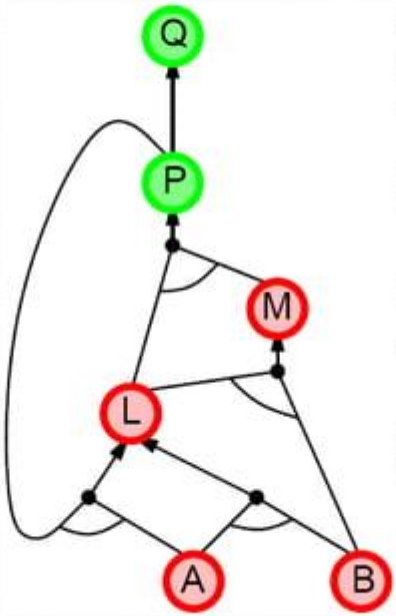
Backward Chaining



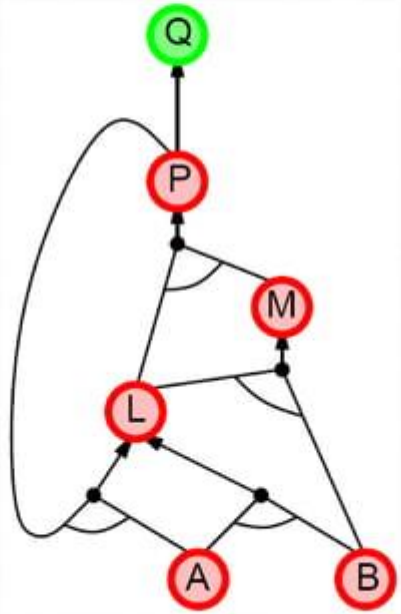
Backward Chaining



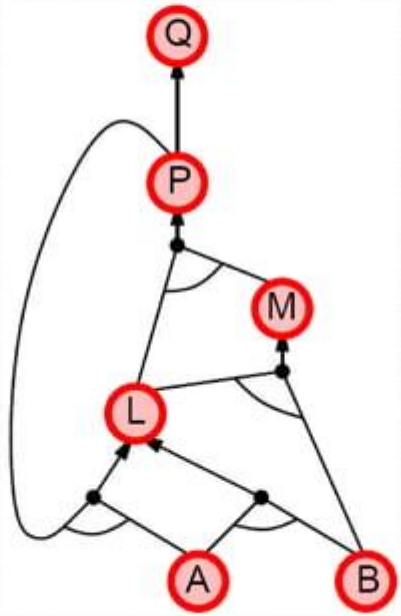
Backward Chaining



Backward Chaining



Backward Chaining



Forward Chaining vs. Backward Chaining

- Forward Chaining is data driven
  - Automatic, unconscious processing
  - E.g. object recognition, routine decisions
  - May do lots of work that is irrelevant to the goal
- Backward Chaining is goal driven
  - Appropriate for problem solving
  - E.g. "Where are my keys?", "How do I start the car?"
- The complexity of BC can be much less than linear in size of the KB

Thanks...