PURDUE CS47100

INTRODUCTION TO AI

ANNOUNCEMENT

- No class on next Tuesday (October 11)
  - Happy fall break!

RECAP: THEOREM PROVING

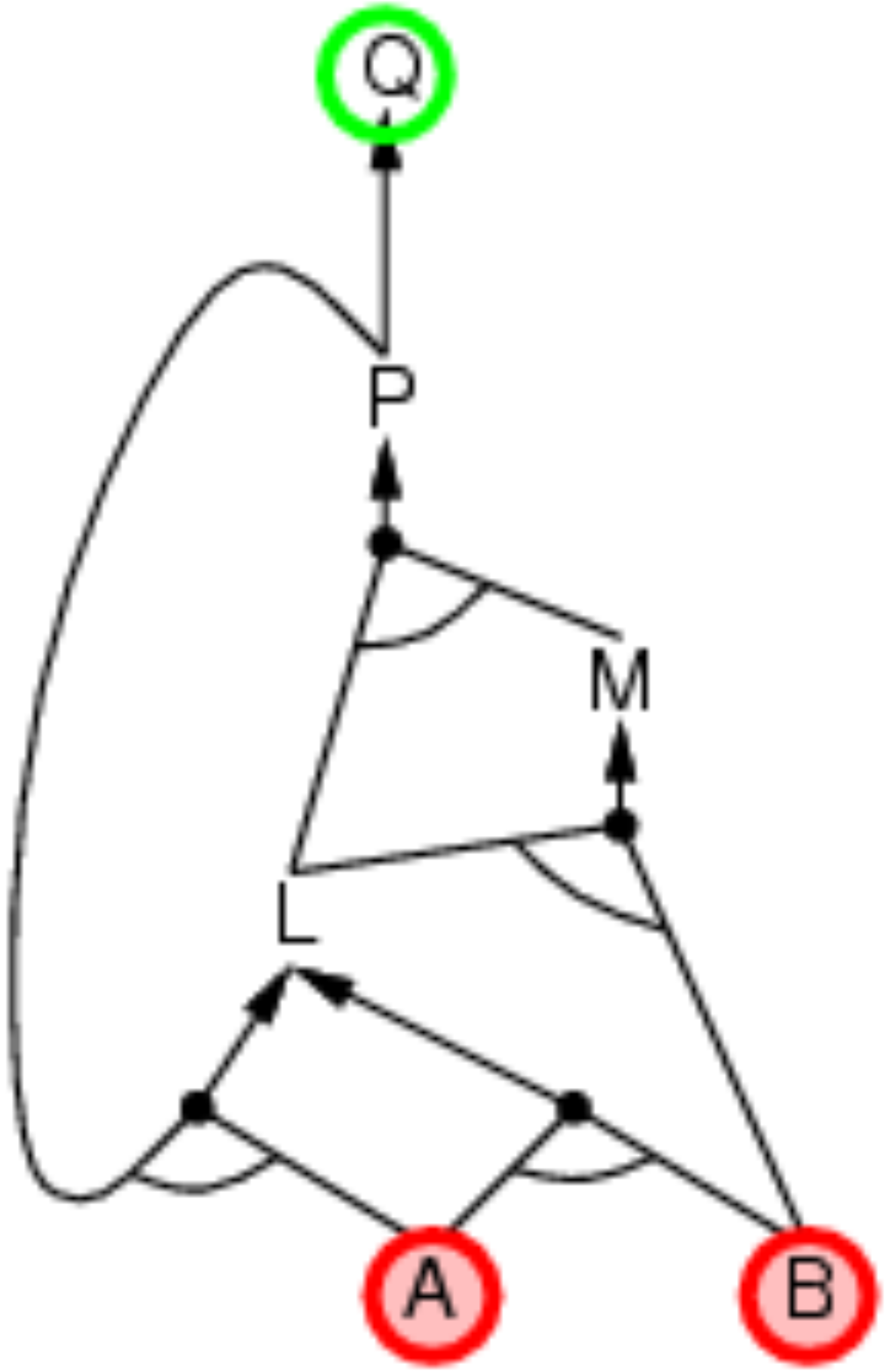
- Proof by inference rules
  - Modus Ponens, And Elimination, all logical equivalence
  - Essentially a search problem
- Proof by resolution
  - Add the negation of the conclusion into KB
  - Convert sentences in KB into conjunctive normal form
  - Repeatedly apply the resolution inference rule to find contradiction
- Forward chaining and backward chaining
  - Applicable when sentences in KB are all horn clauses
  - Repeatedly apply the Modus Ponens rule

BACKWARD CHAINING

- Idea: work backwards from the query q:
  - to prove q by Backward Chaining,
    - check if q is known already, or
    - prove by Backward Chaining 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

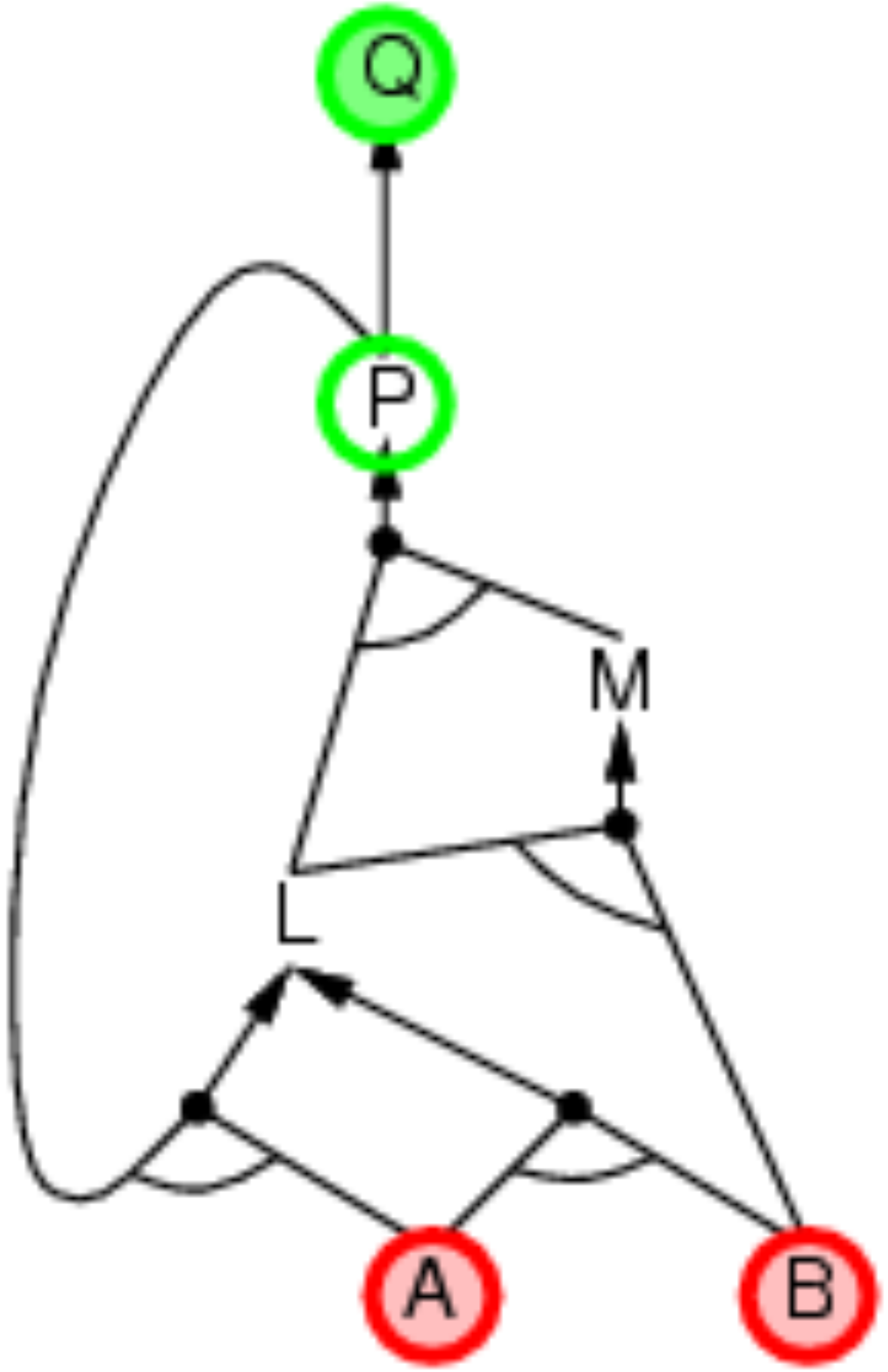
BACKWARD CHAINING EXAMPLE

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$\boxed{\begin{array}{l} A \\ B \end{array}}$$



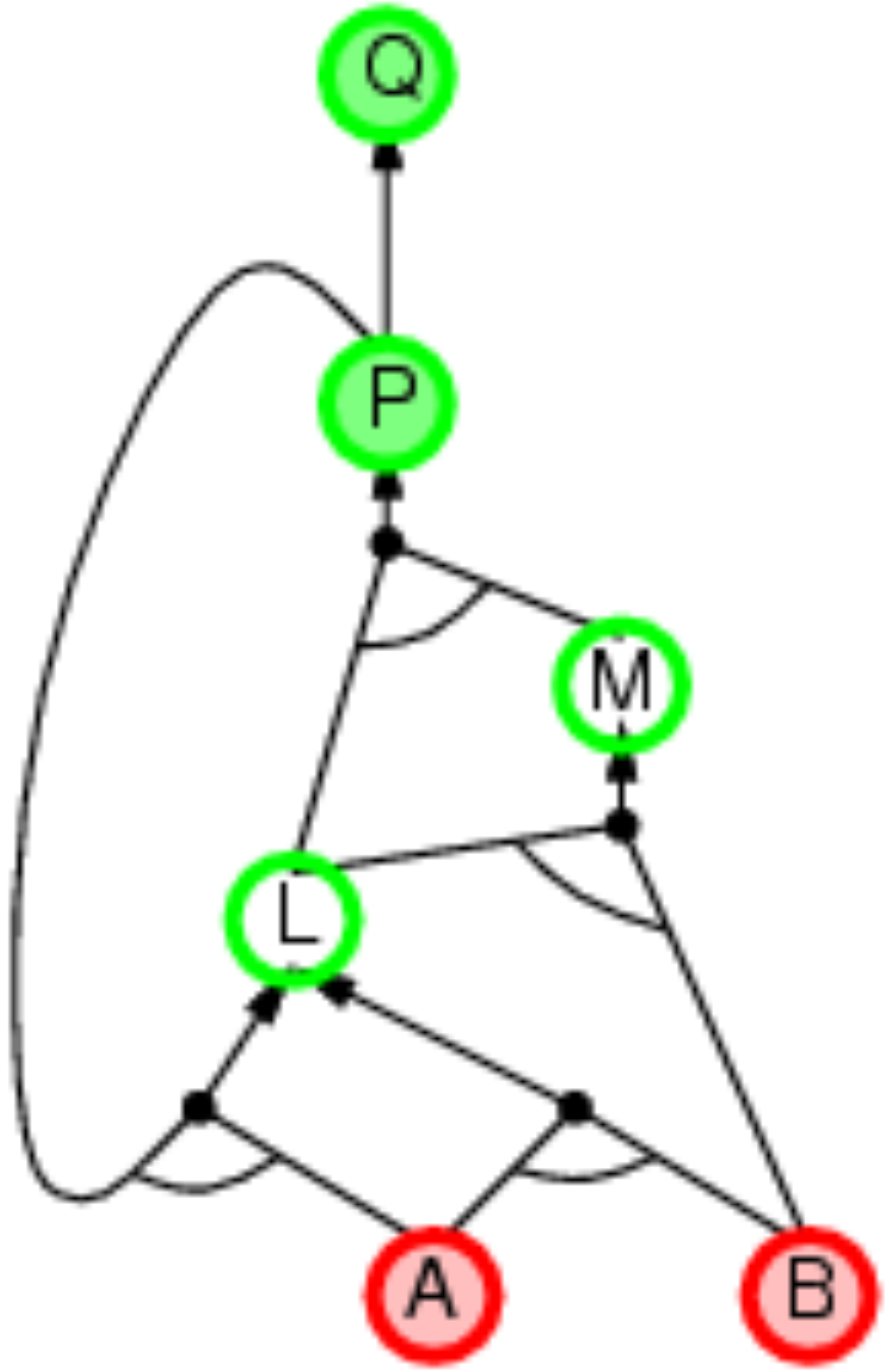
BACKWARD CHAINING EXAMPLE

$$\boxed{P \Rightarrow Q}$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



BACKWARD CHAINING EXAMPLE

$$P \Rightarrow Q$$
$$\boxed{L \wedge M \Rightarrow P}$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



BACKWARD CHAINING EXAMPLE

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

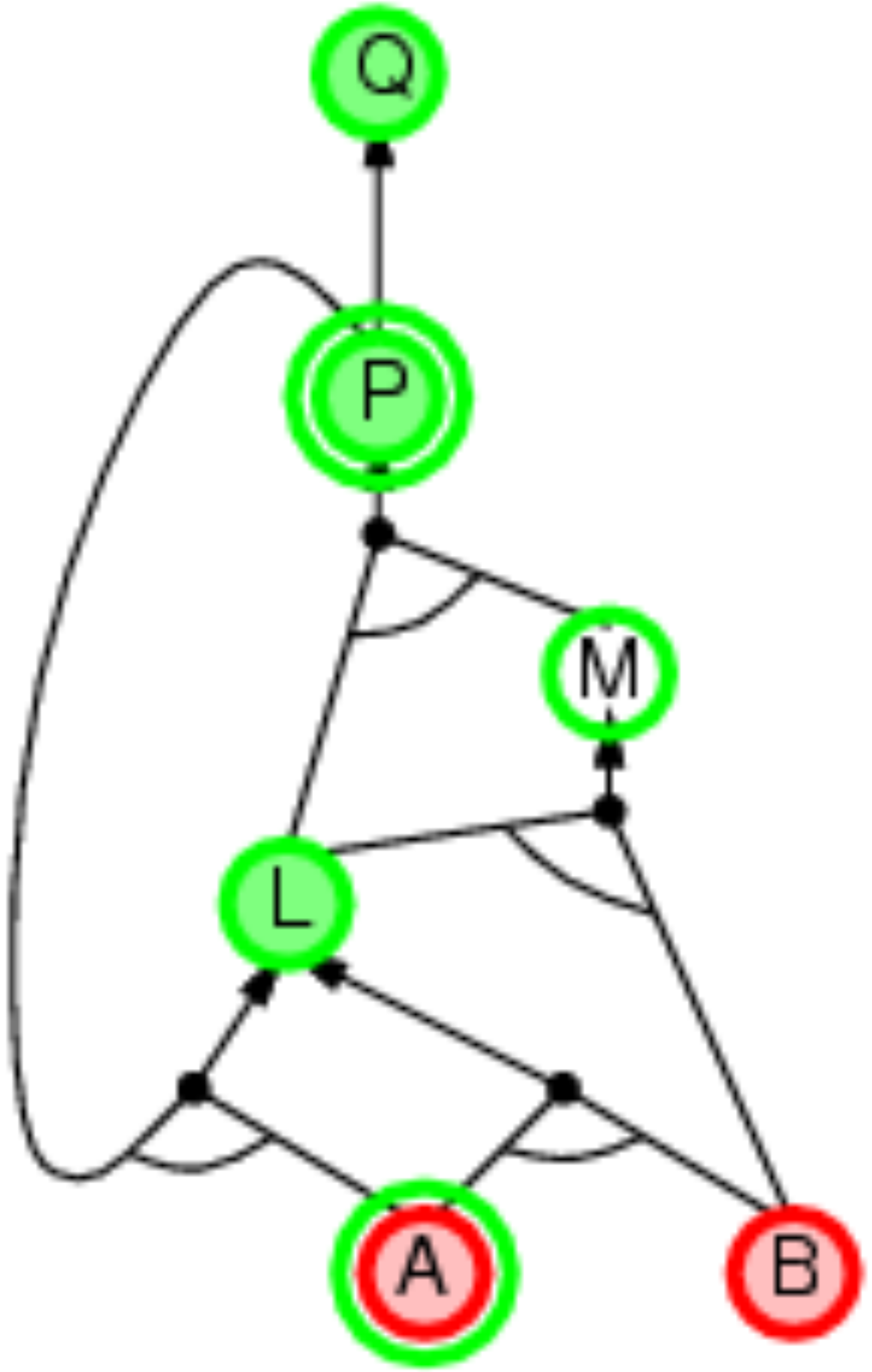
$B \wedge L \Rightarrow M$

$\boxed{A \wedge P \Rightarrow L}$

$A \wedge B \Rightarrow L$

A

B



BACKWARD CHAINING EXAMPLE

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

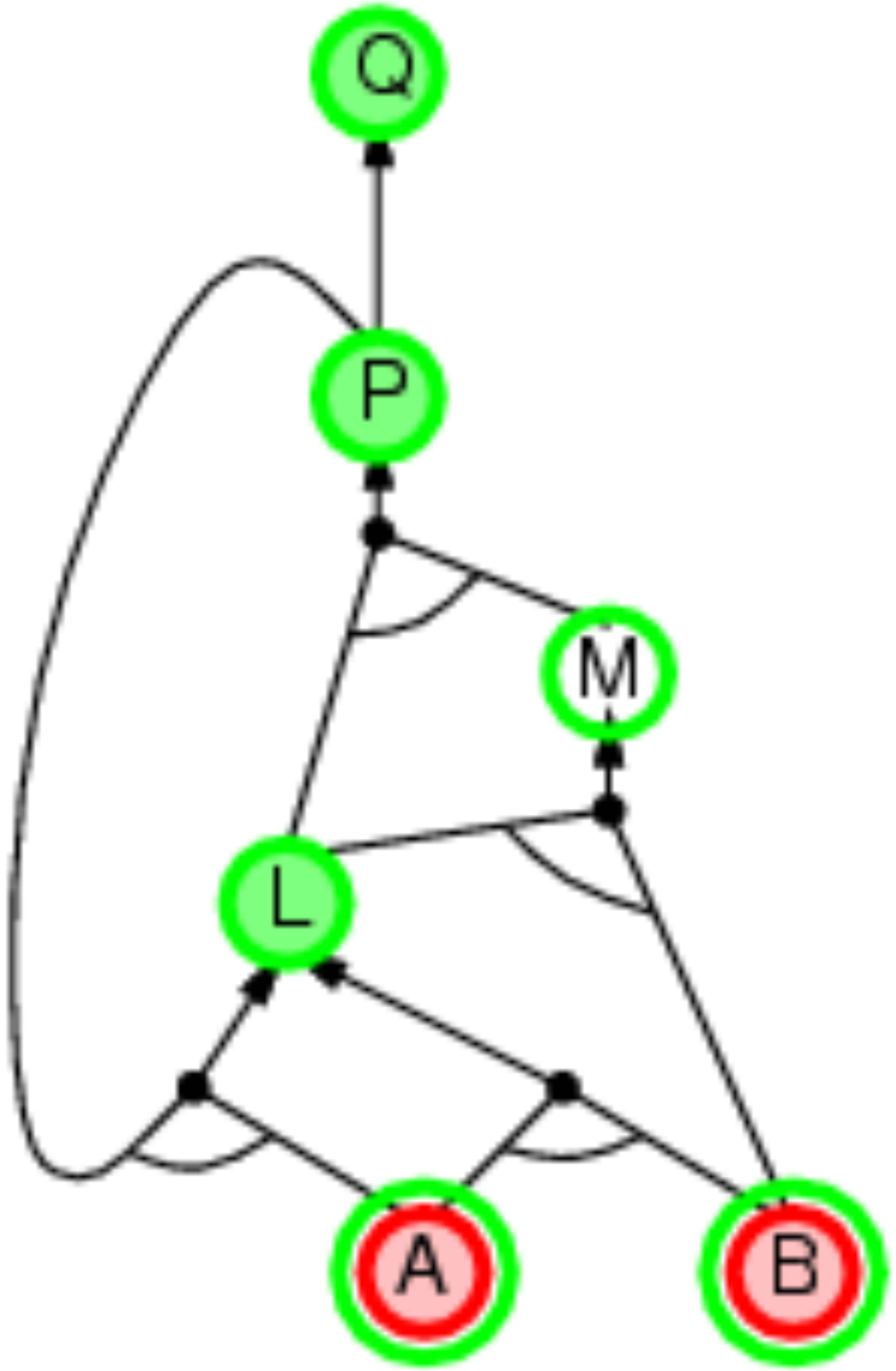
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$\boxed{A \wedge B \Rightarrow L}$

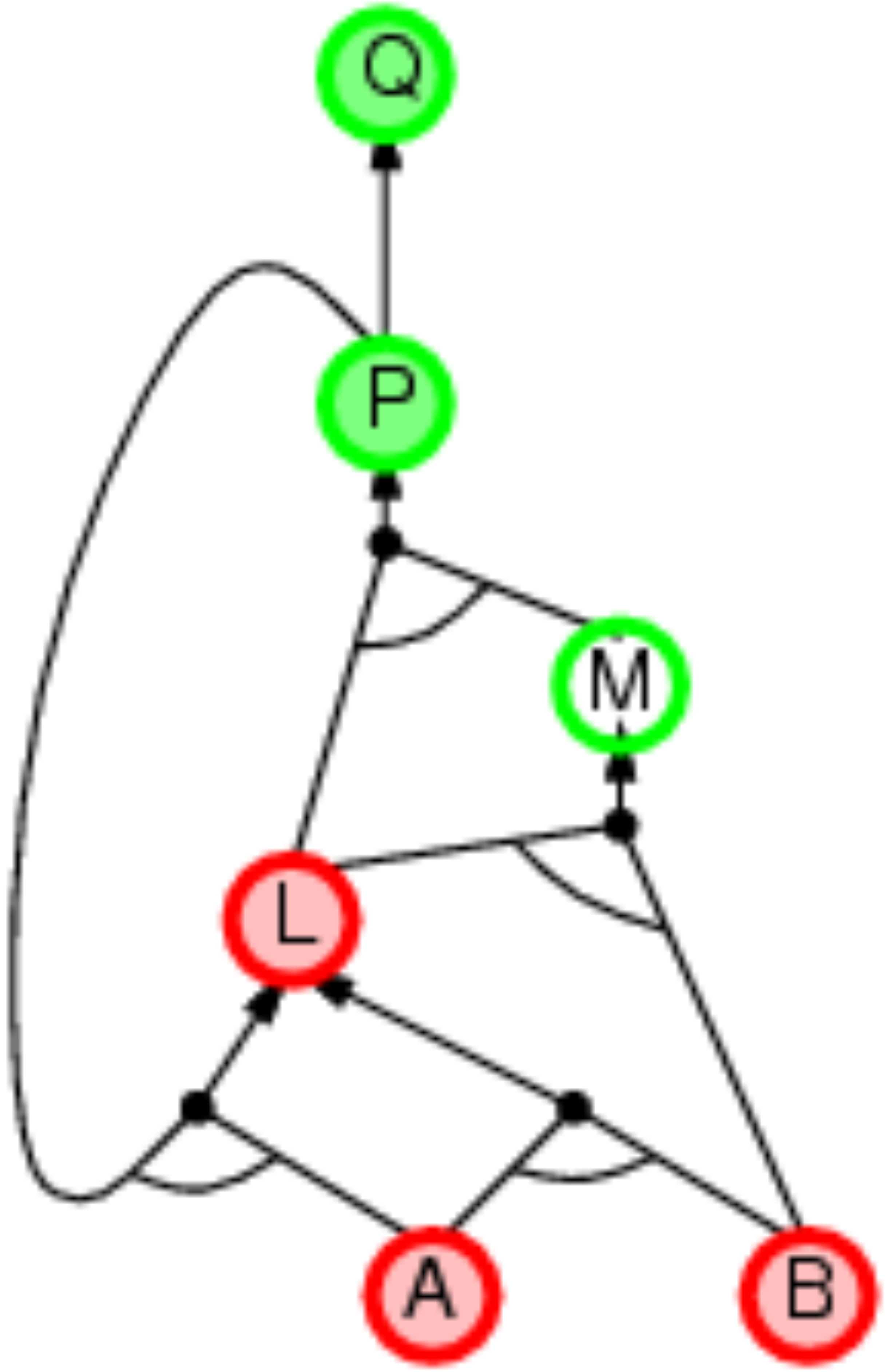
A

B



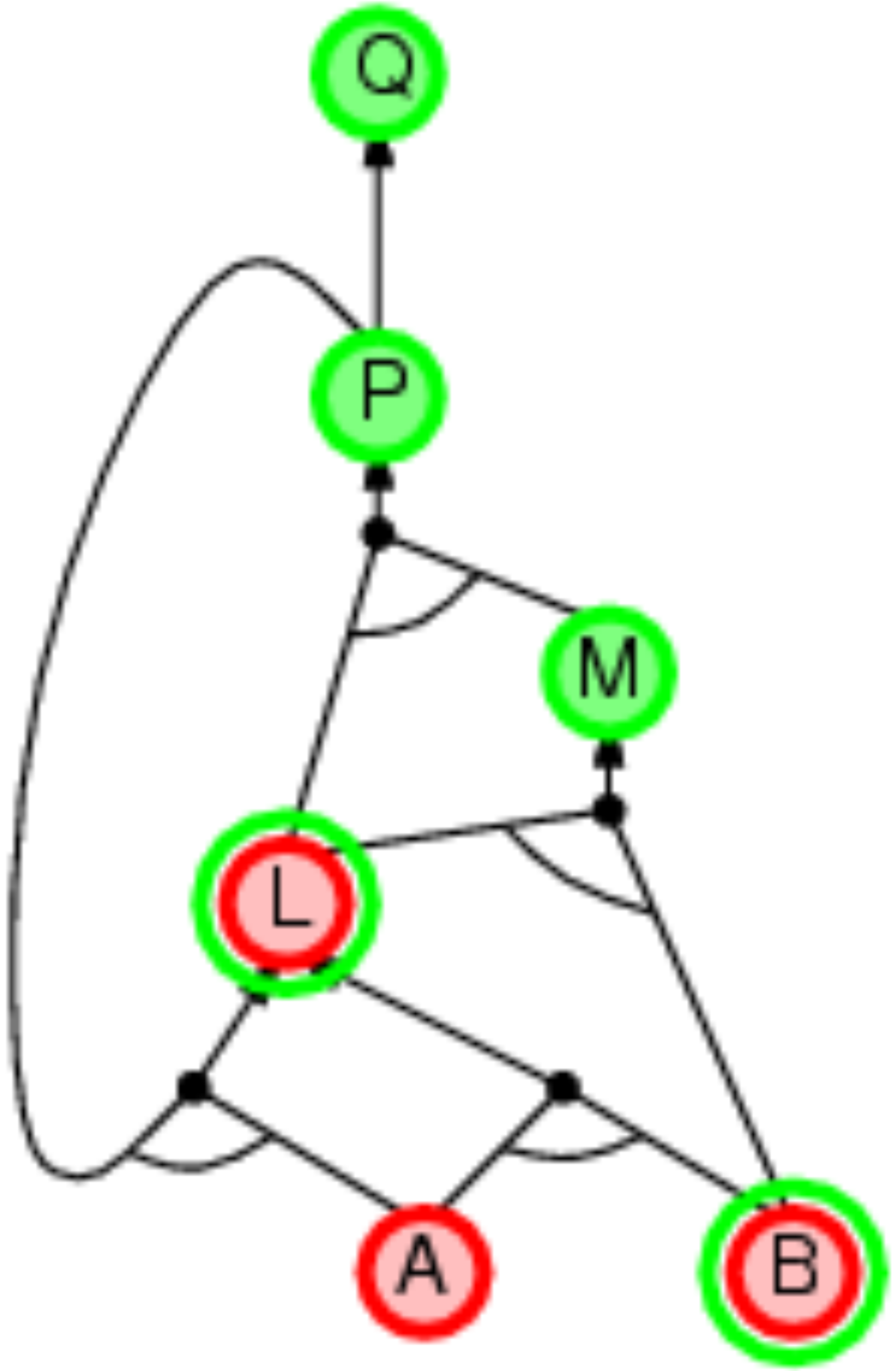
BACKWARD CHAINING EXAMPLE

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



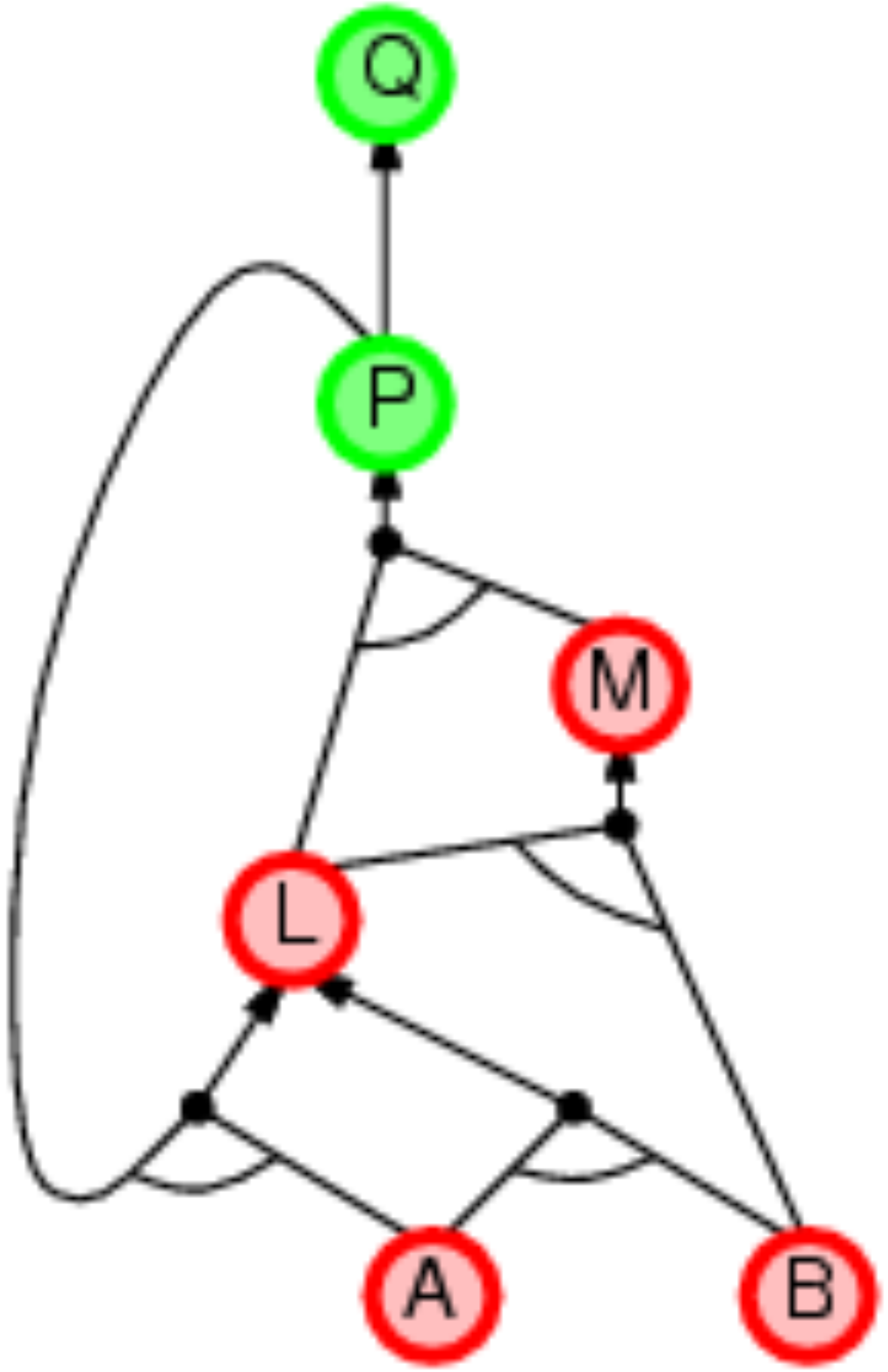
BACKWARD CHAINING EXAMPLE

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$\boxed{B \wedge L \Rightarrow M}$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



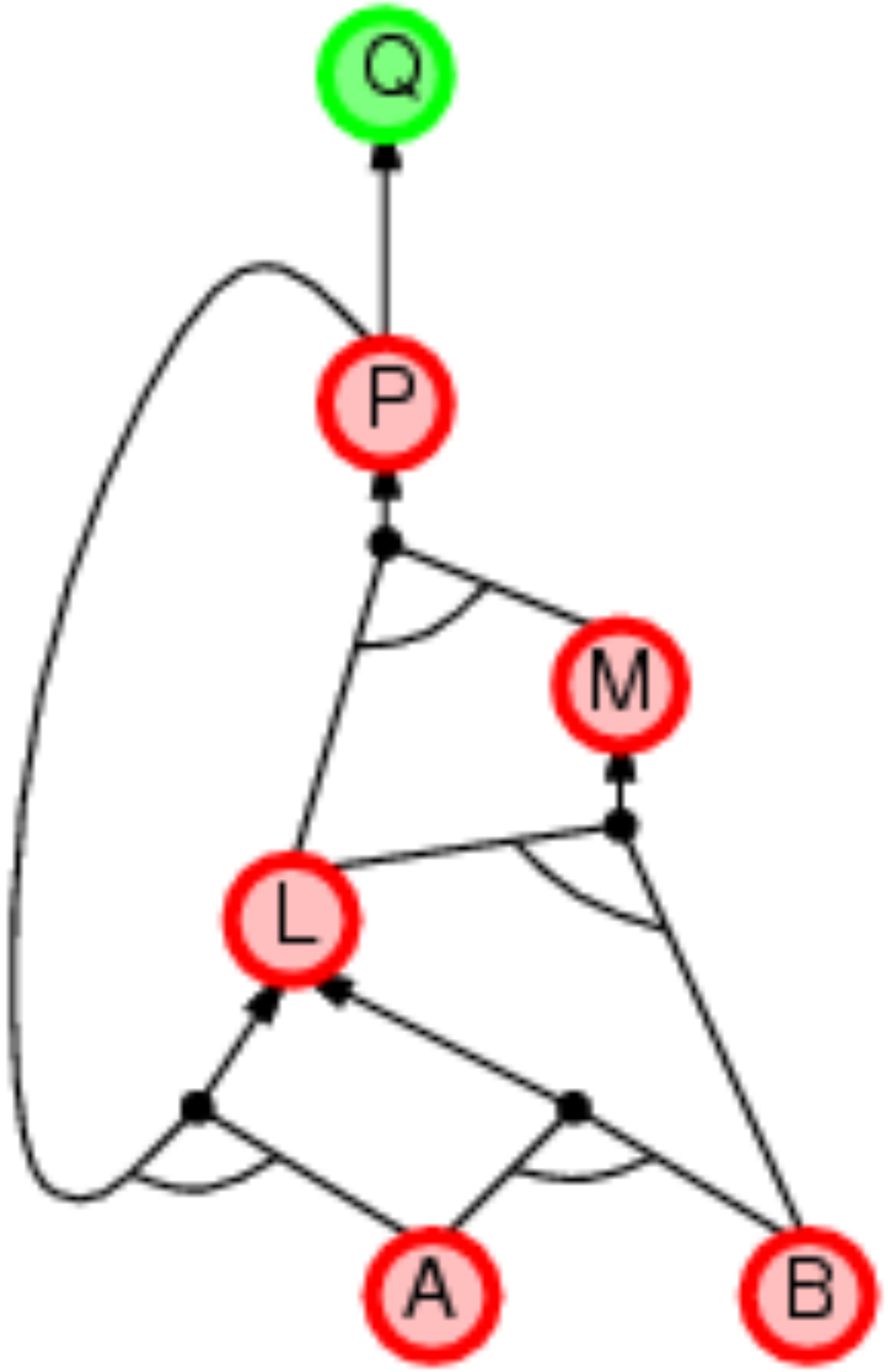
BACKWARD CHAINING EXAMPLE

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



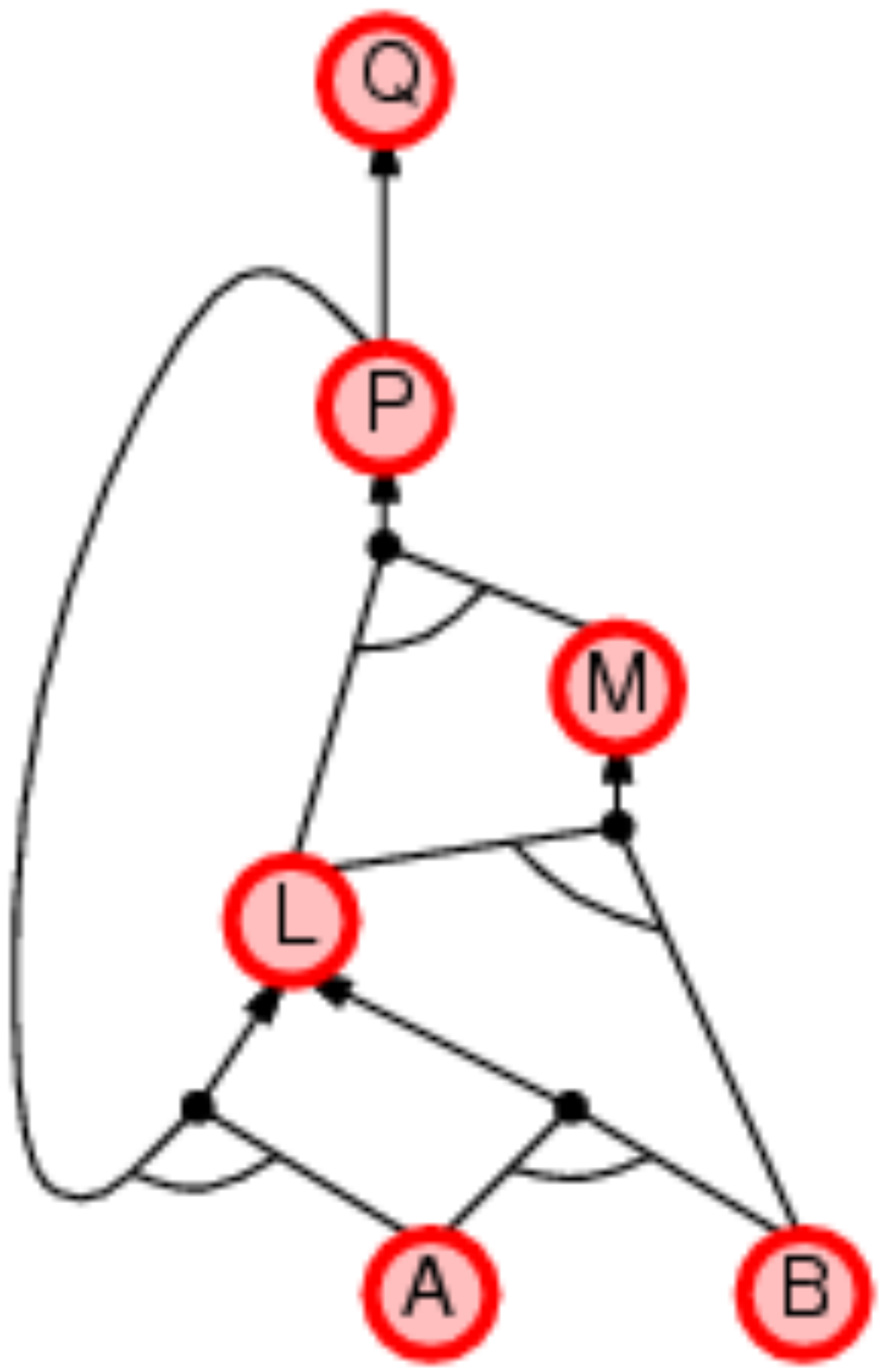
BACKWARD CHAINING EXAMPLE

$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



BACKWARD CHAINING EXAMPLE

$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



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

HOW TO DETERMINE ENTAILMENT?

- Methods can be divided into (roughly) two kinds:
  - **Theorem proving: Application of inference rules**
    - Legitimate (sound) generation of new sentences from old
    - Proof = a sequence of inference rule applications
      Can use inference rules as operators in a standard search algorithm
    - Typically require transformation of sentences into a normal form
  - **Model checking**
    - Truth table enumeration (always exponential in n)
    - Improved backtracking, e.g., Davis-Putnam-Logemann-Loveland (DPLL)
    - Heuristic search in model space (sound but incomplete), e.g., min-conflicts-like hill-climbing algorithms

MODEL CHECKING

- Approach 1: Enumeration
  - See if for any model of KB, it is also a model of α
- Approach 2:
  - KB $\vDash \alpha$ if and only if (KB $\wedge \neg \alpha$) is unsatisfiable
  - So let's see if we can find a model for KB $\wedge \neg \alpha$, i.e., whether there is a truth value assignment of symbols that makes KB $\wedge \neg \alpha$ to be true?
  - This can be modeled as a Constraint Satisfaction Problem!

HOW TO CONFIGURE LOGICAL INFERENCE AS A CSP?

- Variables? Symbols
- Domains? True/False
- Constraints? Each clause in the CNF must be true

BACKTRACKING SEARCH FOR LOGICAL INFERENCE: DPLL

- DPLL (Davis, Putnam, Logemann, and Loveland, 1962)
- Effectively a backtracking search algorithm with domain specific heuristics:
  - Early determination: a clause is true if any literal in it is true; a sentence is false if any clause is false
  - Unit clause heuristic: the literal in a unit clause (i.e., a clause with one single literal) must set to be true
  - Pure symbol heuristic: a pure symbol (always show with the same "sign" in all clauses) should be assigned a truth value that makes its literal true

$$A \lor \neg B \qquad \neg B \lor \neg C \qquad A \lor C$$

LOCAL SEARCH FOR LOGICAL INFERENCE: THE WALKSAT ALGORITHM

- Local search algorithm
  - Start with full assignment of truth values for symbols
  - Evaluation function: number of unsatisfied clauses
  - "Walk" through space by randomly selecting a unsatisfied clause, and choosing a symbol in it to flip:
    - Using min-conflicts heuristic
    - Random selection of symbol
    - Balance between greediness and randomness

PROS AND CONS OF PROPOSITIONAL LOGIC

- Propositional logic is **declarative**
- Propositional logic allows partial/disjunctive/negated information (unlike most data structures and databases)
- Propositional logic is **compositional**: meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- Meaning in propositional logic is **context-independent** (unlike natural language, where meaning depends on context)
- Propositional logic has limited **expressive power** (unlike natural language)
  - E.g., cannot say "pits cause breezes in adjacent squares" except by writing one sentence for each square

EXPLORING A WUMPUS WORLD

A 4X4 wumpus-world agent using propositional logic:

$\neg P_{1,1}$

$\neg W_{1,1}$

$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$

$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$

$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$

$\neg W_{1,1} \vee \neg W_{1,2}$

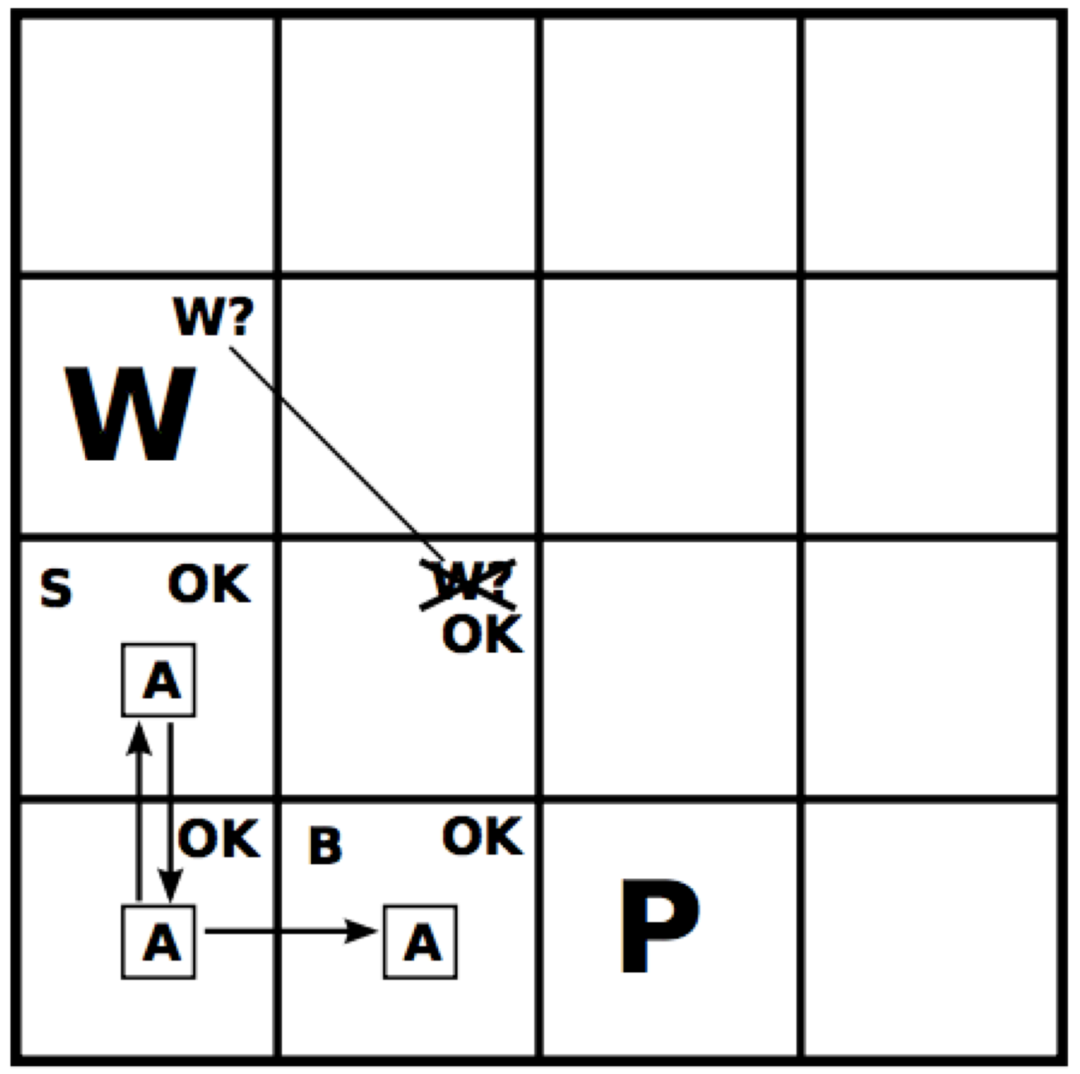
$\neg W_{1,1} \vee \neg W_{1,3}$

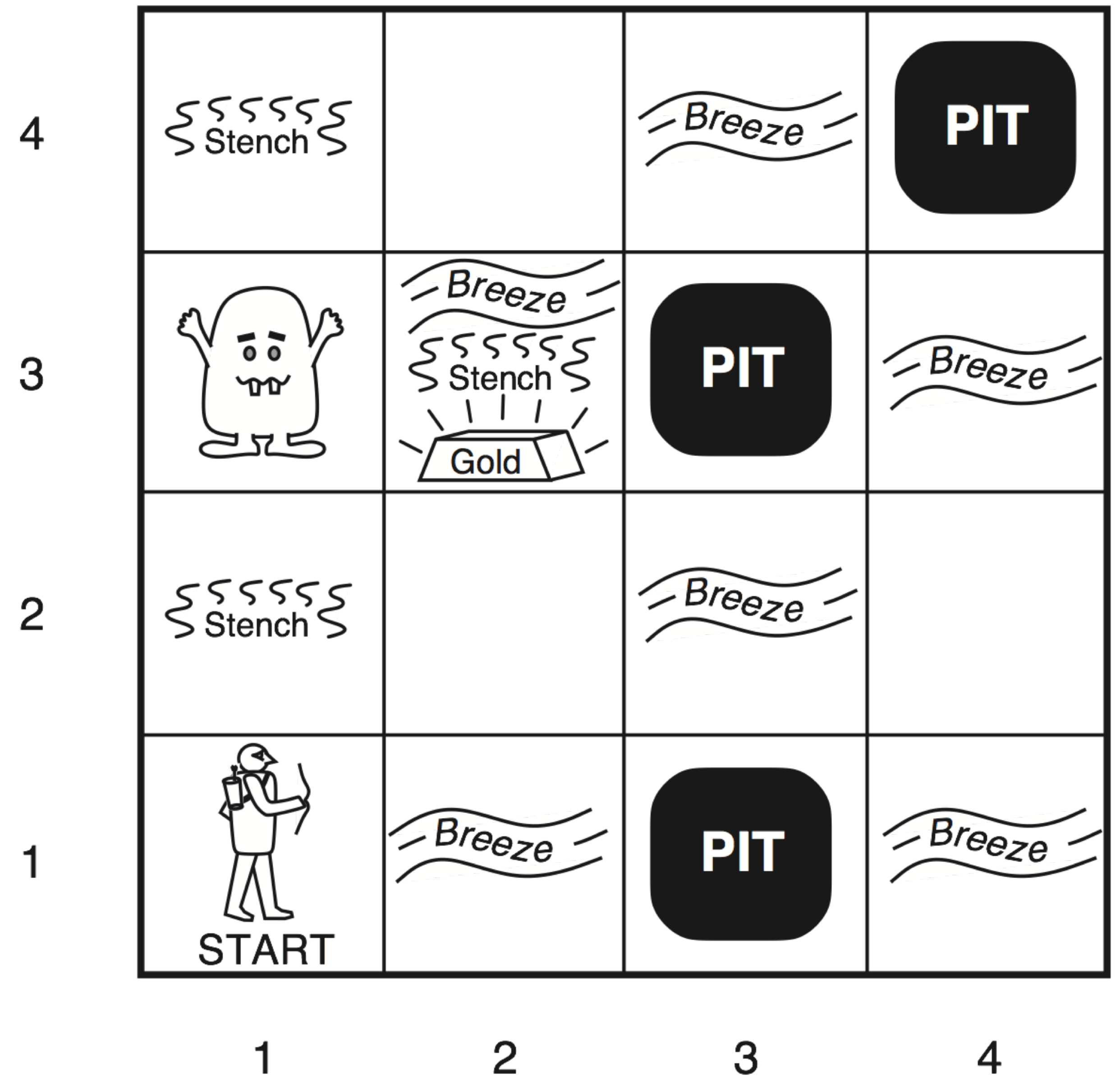
…

Holds for **EVERY** (x,y) pairs!

Any better representation??

- $\Rightarrow$ 64 distinct proposition symbols, many many sentences…

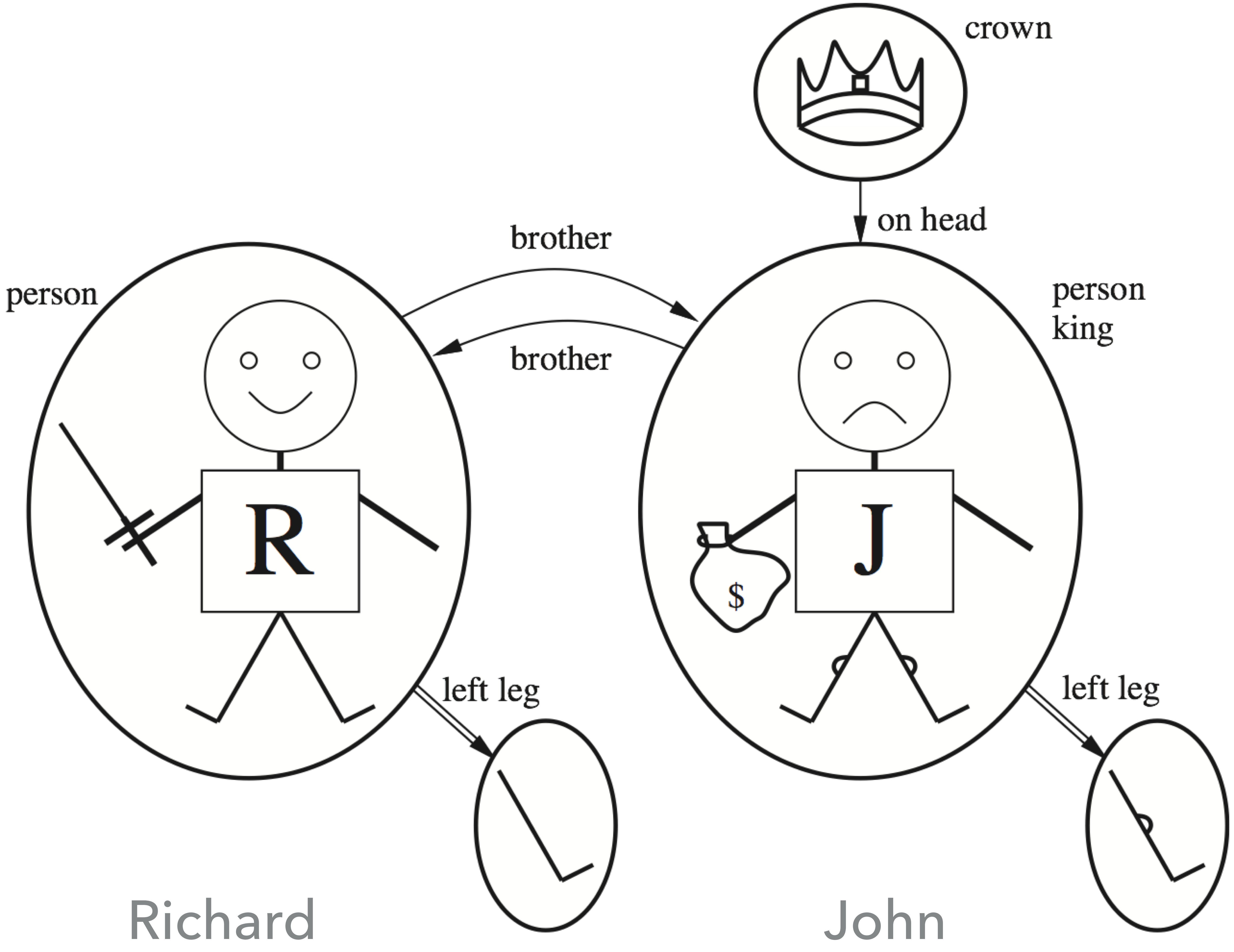




FIRST-ORDER LOGIC

- Propositional logic assumes the world contains:
  - **Facts** – there is no breeze, it is snowing, …
- First-order logic assumes the world contains:
  - **Facts** – there is no breeze, it is snowing, …
  - **Objects** – people, houses, numbers, colors, …
  - **Relations** – red, round, prime, brother of, bigger than, part of, comes between, …
  - **Functions** – father-of, best-friend, more-than, plus, …

EXAMPLE


crown
on head
brother
brother
person
person
king
R
J
$
left leg
left leg
Richard
John

BASIC SYNTACTIC ELEMENTS OF FOL

- Constants John, Richard, 2,…
- Predicates Brother, >,…
- Functions Sqrt, LeftLegOf,…
- Variables x, y, a, b,…
- Connectives $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
- Equality =
- Quantifiers $\forall, \exists$

Constants: objects

Relation: set of tuples of objects that are related e.g., {<Richard,John>, <John, Richard>}

Function: relations where a given set of objects are related to exactly one object e.g., {<John>→John's left leg, <Richard>→Richard's left leg}

ATOMIC SENTENCES

Term = function ($term_1$,…,$term_n$) or constant or variable

Atomic sentence = predicate ($term_1$,…,$term_n$) or $term_1$ = $term_2$

- Examples
  - brother(KingJohn, RichardTheLionheart)
  - height(Richard) = height(KingJohn)
  - KingJohn
  - x

COMPLEX SENTENCES

- Complex sentences are made from atomic sentences using connectives:
  $\neg S$ $\quad S_1 \wedge S_2$ $\quad S_1 \vee S_2$ $\quad S_1 \Rightarrow S_2$ $\quad S_1 \Leftrightarrow S_2$
- Examples
  - $Sibling(John,Richard) \Rightarrow Sibling(Richard,John)$
  - $>(1,2) \vee \leq(1,2)$
  - $>(1,2) \wedge \neg >(1,2)$

TRUTH IN FIRST-ORDER LOGIC

- Model contains objects (**domain elements**) and relations among them
- *Interpretation* specifies referents for

  constant symbols → **objects**
  predicate symbols → **relations**
  function symbols → **functions**

- Sentences are true with respect to a **model** and an **interpretation**
  - An atomic sentence predicate($term_1$,...,$term_n$) is true iff the **objects** referred to by $term_1$,...,$term_n$ are in the **relation** referred to by predicate
    - E.g., brother(John, Richard); sibling(Tom, sonOf(Henry))

UNIVERSAL QUANTIFICATION

- ∀<variables> <sentence>
- Everyone in CS471 is smart: $\forall x \; in(x, CS471) \Rightarrow Smart(x)$
- $\forall x \; P$ is true in a model m iff P is true with x interpreted as *each* possible object in the model
- Roughly speaking, equivalent to the **conjunction** of **instantiations** of P

$$
\begin{aligned}
& In(John, CS471) \Rightarrow Smart(John) \\
\wedge \quad & In(Jane, CS471) \Rightarrow Smart(Jane) \\
\wedge \quad & In(CS471, CS471) \Rightarrow Smart(CS471) \\
\wedge \quad & \ldots
\end{aligned}
$$

A COMMON MISTAKE TO AVOID

- Typically, $\Rightarrow$ is the main connective with $\forall$

- Common mistake: using $\wedge$ as the main connective with $\forall$
  - $\forall x\ In(x, CS471) \wedge Smart(x)$
  - means “Everyone is in CS471 and everyone is smart”