IF440: Artificial Intelligence

Topic 03: Propositional Logic

Moeljono Widjaja, Ph.D.
Arrival Dwi Sentosa, B.Sc., M.T.

Universitas Multimedia Nusantara
Jl. Scientia Boulevard, Gading Serpong
Tangerang, Banten-15811 Indonesia

Even Semester - 2022/2023

Capaian Pembelajaran Mingguan Mata Kuliah (Sub-CPMK):

Sub-CPMK-03 - Mahasiswa dapat menjelaskan konsep dasar dari model kecerdasan buatan berbasis representasi pengetahuan dan logika proposisi serta mengidentifikasi ketepatgunaan model logika proposisi untuk suatu permasalahan – (C2);

Review

1. Intelligent agent environment
2. Types of agents
3. Search agents

Outline

Concept of Knowledge representation

Introduction

- Success in solving problems with AI depends naturally on our ability to:
  - **Represent** the knowledge about the world.
  - **Reason** with the knowledge to obtain meaningful answers.

Knowledge Representation

- Knowledge representation (KR) is the study of
  - how **knowledge and facts** about the world can be represented, and
  - what **kinds of reasoning** can be done with that knowledge.

Knowledge-Based Agents

Knowledge-based agents
Knowledge base
Inference engine

- **Knowledge base (KB)**: a set of sentences that describe the world and its behavior in some formal (representational) language.
- **Inference engine**: 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.

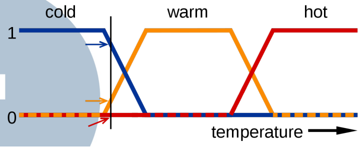
Knowledge Representation Language

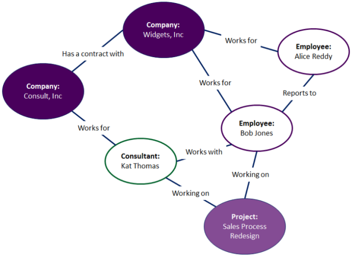
- Key aspects of knowledge representation languages:
  - **Syntax**: describes how sentences are formed in the language.
  - **Semantics**: describes the meaning of sentences, what is it the sentence refers to in the real world.

How can knowledge be represented?

How can knowledge be represented?

- Symbolic
  - Logic: propositional logic, fuzzy logic, first order logic
  - Ontology
  - Prolog





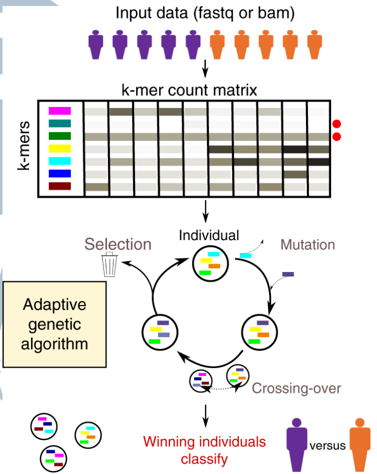
PROLOG Paradigm

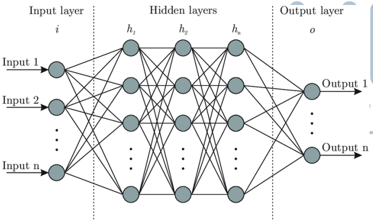
Examples (Queries)

English	PROLOG
"is a sparrow an animal?"	?- animal(sparrow).
answer: "yes"	yes
"is a table an animal?"	?- animal(table).
answer: "no"	no
"what is a dog?"	?- isa(dog, X).
answer: "a mammal"	X = mammal

How can knowledge be represented?

- Non-Symbolic
  - Neural Networks
  - Genetic Algorithms





Logic

LOGIC

- Language for expressing reasoning steps.
- Syntax:
  The **atomic symbols** of the logical language, and the rules for constructing well-formed, non-atomic expressions (symbol structures) of the logic.
- Semantics:
  Define the **truth of a sentence**.

Propositional logic

Proposition

- An expression in some language
  - that is true or false
  - whose negation makes sense
  - that can be believed or not
  - whose negation can be believed or not
- Examples:
  - Propositions:
    - Betty is the driver of the car.
    - Barack Obama is sitting down or standing up.
    - If Opus is a penguin, then Opus doesn't fly.
  - Non-Propositions:
    - Barack Obama
    - how to ride a bicycle
    - If the fire alarm rings, leave the building.

SENTENCES vs. PROPOSITIONS

- A sentence is an expression of a (written) language that begins with a capital letter and ends with a period, question mark, or exclamation point.
- Some sentences do not contain a proposition:
  "Hi!", "Why?", "Pass the salt!"
- Some sentences do not express a proposition, but contain one:
  "Is Betty driving the car?"
- Some sentences contain more than one proposition:
  If Opus is a penguin, then Opus doesn't fly.

PROPOSITIONAL LOGIC

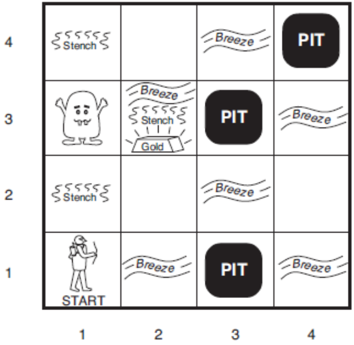
- Syntax and Semantic
  - $\neg P$ is true if P is false (**negation**).
  - $P \wedge Q$ is true iff both P and Q are true (**conjunction**).
  - $P \vee Q$ is true iff either P or Q are true (**disjunction**).
  - $P \implies Q$ is true unless P is true and Q is false (**implication**).
  - $P \iff Q$ is true iff P and Q are both true or both false (**biconditional**).

PROPOSITIONAL LOGIC

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{True} \mid \textit{False} \mid P \mid Q \mid R \mid \ldots \\
\textit{ComplexSentence} & \rightarrow & (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
& \mid & \neg\ \textit{Sentence} \\
& \mid & \textit{Sentence} \land \textit{Sentence} \\
& \mid & \textit{Sentence} \lor \textit{Sentence} \\
& \mid & \textit{Sentence} \Rightarrow \textit{Sentence} \\
& \mid & \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
 & & \\
\text{OPERATOR PRECEDENCE} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

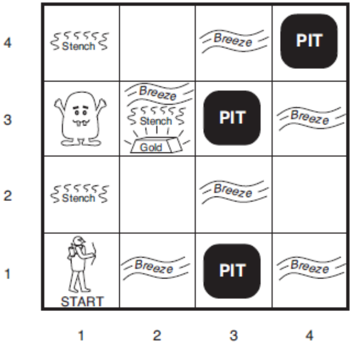
Figure 7.7 A BNF (Backus–Naur Form) grammar of sentences in propositional logic, along with operator precedences, from highest to lowest.

The Wumpus World



- Adjacent rooms are connected (horizontally or vertically)
- Lurking in the cave is the Wumpus
- Agent can smell the Wumpus (**stench**)
- Agent feels a **breeze** if pit nearby
- Agent can shoot ONE arrow at (and kill) the Wumpus (**scream**)
- Some rooms contain pits that will trap agent
- One room contains a pot of gold (**glitter**)
- When agent walks into a wall, it will perceive a **bump**

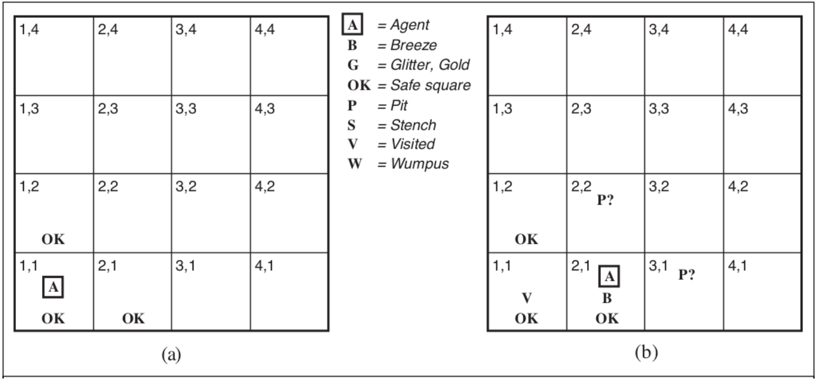
The Wumpus World



- PEAS:
  - **Performance measure:** +1000 for walk out w/ gold; -1000 for dying; -1 for each action; -10 for arrow
  - **Environment:** a 4x4 grid. Agent starts at [1,1]; gold and pits randomly distributed.
  - **Actuators:** agent can move forward, backward, left, or right.
  - **Sensors:** [Stench, Breeze, Glitter, Bump, Scream]

The Wumpus World

- Step 1

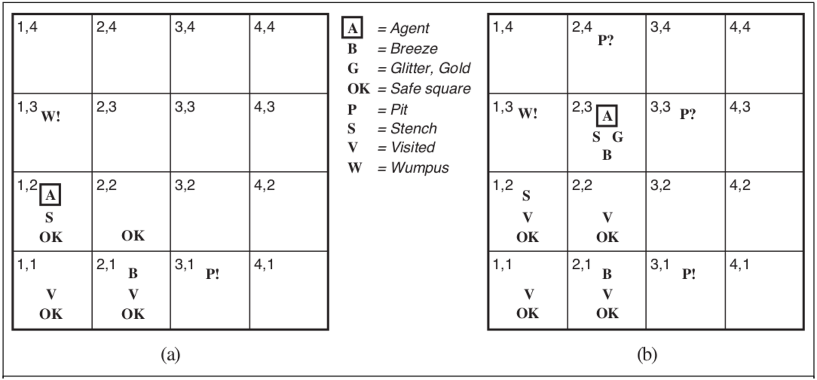


[None, None, None, None, None]

[None, Breeze, None, None, None]

The Wumpus World

- Step 3 and 5



[Stench, None, None, None, None]

[Stench, Breeze, Glitter, None, None]

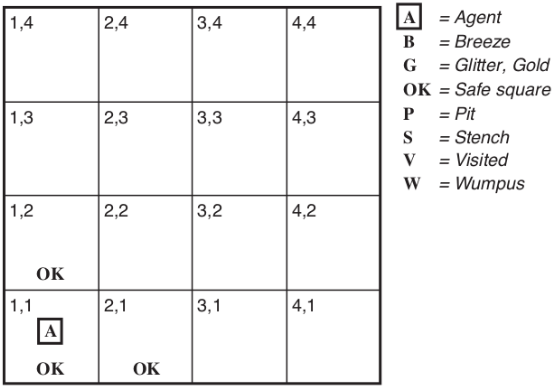
PROPOSITIONAL LOGIC

The Wumpus World:

- $P_{x,y}$ is true if there is a pit in [x,y]
- $W_{x,y}$ is true if there is a Wumpus in [x,y]
- $B_{x,y}$ is true if the agent perceives a breeze in [x,y]
- $S_{x,y}$ is true if the agent perceives a stench in [x,y]
- Example:
  - a square is breezy if a neighbouring square has a pit,
  - $B_{1,1} \iff (P_{1,2} \vee P_{2,1})$, where $B_{1,1}$ means that there is a breeze in [1,1].

The Wumpus World

- Starting state

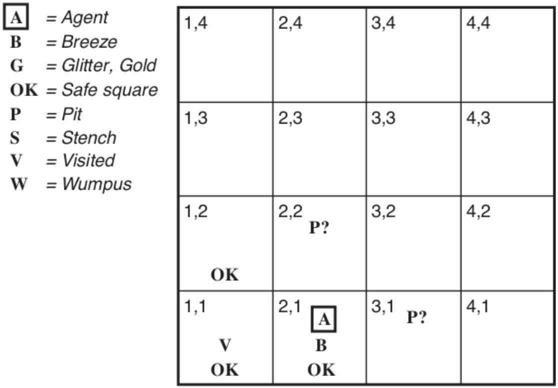


[None, None, None, None, None]

PROPOSITIONAL LOGIC

- Sentences: suffice to derive $\neg P_{1,2}$.
  - There is no pit in [1,1]:
    $R_1 : \neg P_{1,1}$
  - A square is breezy if and only if there is a pit in a neighboring square:
    $R_2 : B_{1,1} \iff (P_{1,2} \vee P_{2,1})$
    $R_3 : B_{2,1} \iff (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
  - After visiting [1,1], [1,2], [2,1]. The breeze percepts for the first two squares visited in the specific world the agent is in:
    $R_4 : \neg B_{1,1}$
    $R_5 : B_{2,1}$
- $KB = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

Proof by Enumeration



- 7 relevant symbols: $B_{1,1}, B_{2,1}, P_{1,1}, P_{1,2}, P_{2,1}, P_{2,2}, P_{3,1}$.
- $2^7 = 128$ models.

Proof by Enumeration

- Inference procedure: enumerate all models -> time consuming

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Figure 7.9 A truth table constructed for the knowledge base given in the text. KB is true if R_1 through R_5 are true, which occurs in just 3 of the 128 rows (the ones underlined in the right-hand column). In all 3 rows, $P_{1,2}$ is false, so there is no pit in [1,2]. On the other hand, there might (or might not) be a pit in [2,2].

PROPOSITIONAL THEOREM PROVING

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

PROPOSITIONAL THEOREM PROVING

- Inference and Proofs:
  - Inference rules is applied to derive a proof – a chain of conclusions that leads to goal.
  - Modus Ponens:

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

if (WumpusAhead ∧ WumpusAlive) ⇒ Shoot

(*WumpusAhead* ∧ WumpusAlive)

Shoot

  - AND-elimination: from a conjunction, any of the conjuncts can be inferred.

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

(WumpusAhead ∧ WumpusAlive)

WumpusAlive

  - Logical equivalence as inference rules:

$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)} \quad \text{and} \quad \frac{(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

PROOF BY DEDUCTION

- Example: **Wumpus World**. Start with the knowledge base containing R_1 through R_5 and show how to prove $\neg P_{1,2}$ (there is no pit in [1,2]).
  1. Apply biconditional elimination to R_2
     $R_6 : (B_{1,1} \implies (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \implies B_{1,1})$
  2. And-elimination to R_6
     $R_7 : ((P_{1,2} \vee P_{2,1}) \implies B_{1,1})$
  3. Logical equivalence for contra-positives gives
     $R_8 : (\neg B_{1,1} \implies \neg(P_{1,2} \vee P_{2,1}))$
  4. Modus ponens with R_8 and the precept R_4
     $R_9 : \neg(P_{1,2} \vee P_{2,1})$
  5. De Morgan's Rules
     $R_{10} : \neg P_{1,2} \wedge \neg P_{2,1}$

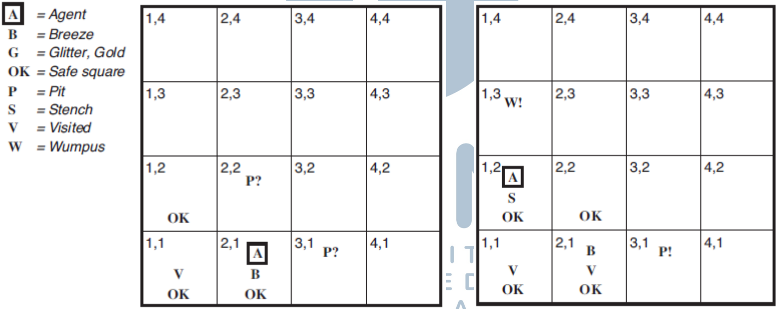
Conclusion: neither [1,2] nor [2,1] contains a pit.

Proof by Resolution

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

Proof by Resolution

- **Wumpus World.** The agent returns from [2,1] to [1,1] and then goes to [1,2], where it perceives a stench, but no breeze.



Proof by Resolution

1. Add the following facts to KB.
   $R_{11} : \neg B_{1,2}$
   $R_{12} : B_{1,2} \iff (P_{1,1} \vee P_{2,2} \vee P_{1,3})$
2. Derive the absence of pits in [2,2] and [1,3].
   $R_{13} : \neg P_{2,2}$
   $R_{14} : \neg P_{1,3}$
3. Biconditional elimination to R_3 followed by Modus Ponens with R_5, to obtain the fact that there is a pit in [1,1], [2,2], or [3,1].
   $R_{15} : P_{1,1} \vee P_{2,2} \vee P_{3,1}$
4. $\neg P_{2,2}$ in R_{13} resolves with the literal $P_{2,2}$ in R_{15} to give the **resolvent**.
   $R_{16} : P_{1,1} \vee P_{3,1}$
5. The literal $\neg P_{1,1}$ in R_1 resolves with the literal $P_{1,1}$ in R_{16}.
   $R_{17} : P_{3,1}$

Horn Clauses

PROPOSITIONAL THEOREM PROVING

- **Horn Clauses:**
  A disjunction of literals of which at most one is positive.
  Ex.: $(\neg P_{1,2} \vee \neg P_{2,1} \vee B_{1,1}), (\neg P_{1,2} \vee \neg P_{2,1} \vee \neg B_{1,1})$
- **Definite Clauses:**
  Restricted inference algorithm.
  A disjunction of literals of which exactly one is positive.
  Ex.: $(\neg P_{1,2} \vee \neg P_{2,1} \vee B_{1,1})$

PROPOSITIONAL THEOREM PROVING

- Every definite clause can be re-written as an implications.
  Proposition symbol (fact)
  Conjunction of symbols (body or premise) $\implies$ symbol (head)
  $(\neg P_{1,2} \vee \neg P_{2,1} \vee B_{1,1}) \equiv (P_{1,2} \wedge P_{2,1}) \implies B_{1,1}$
- Inference with Horn clauses can be done through the **forward-chaining** and **backward-chaining**: natural and easy for humans to follow.

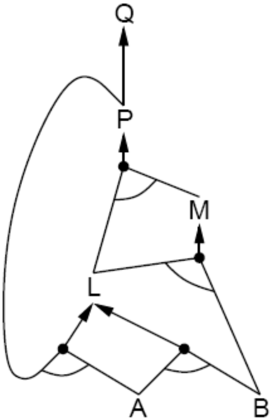
REASONING WITH HORN CLAUSES

- **Forward Chaining**
  - For each new piece of data, generate all new facts, until the desired fact is generated.
  - Data-directed reasoning - reasoning in which the focus of attention starts with the known data.
  - E.g. decision support system, medical diagnosis system.
- **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.
  - E.g. Computer forensic investigation.

FORWARD CHAINING

- Fire any rule whose premises are satisfied in the KB.
- Add its conclusion to the KB until the query is found.
- Ex.: Given *A* and *B* are true, prove that *Q* is true.

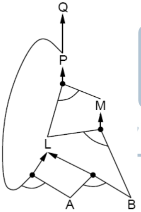
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

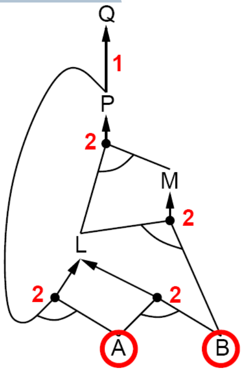


FORWARD CHAINING

- Counting how many literals in each premise.

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

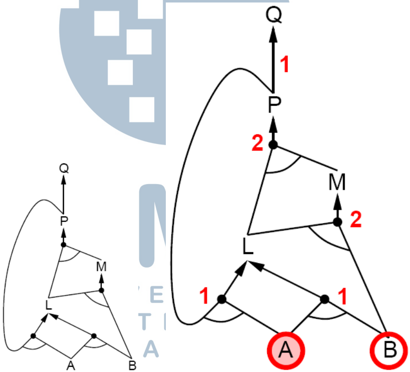




FORWARD CHAINING

- Counting how many literals in each premise.

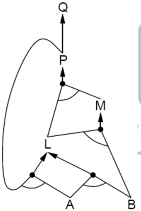
$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$

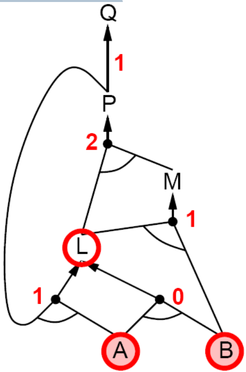


FORWARD CHAINING

- Counting how many literals in each premise.

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

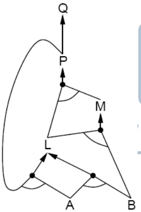


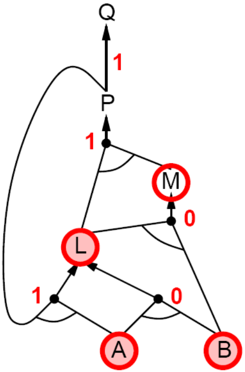


FORWARD CHAINING

- Counting how many literals in each premise.

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

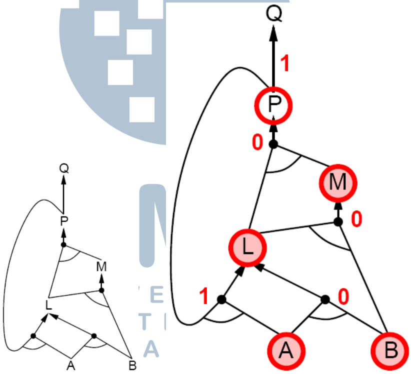




FORWARD CHAINING

- Counting how many literals in each premise.

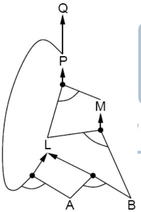
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B

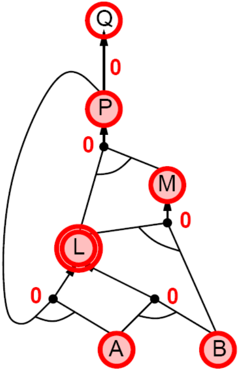


FORWARD CHAINING

- Counting how many literals in each premise.

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

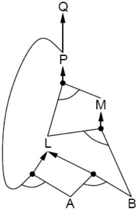


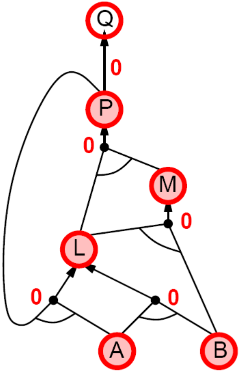


FORWARD CHAINING

- Counting how many literals in each premise.

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

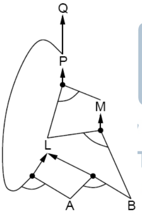


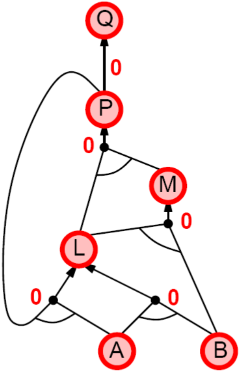


FORWARD CHAINING

- Counting how many literals in each premise.

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



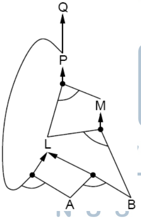


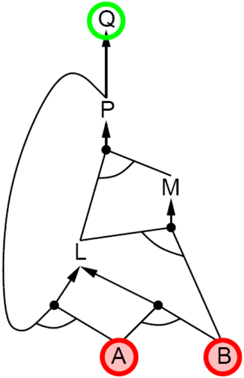
BACKWARD CHAINING

- Idea: work backwards from the query q:
  - To prove q by B.C.,
    - Check if q is known already, or
    - Prove by BC all premises of some rule concluding q.

BACKWARD CHAINING

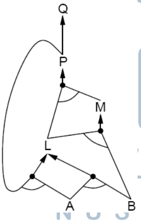
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B

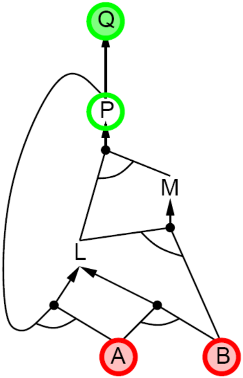




BACKWARD CHAINING

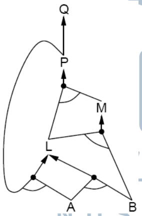
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

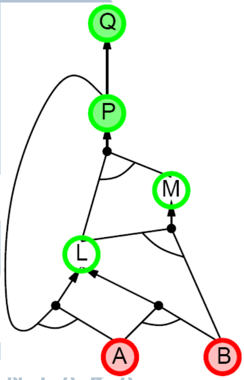




BACKWARD CHAINING

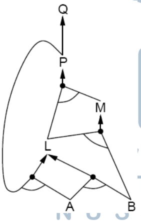
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

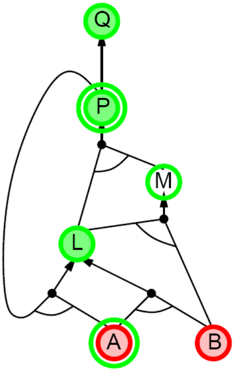




BACKWARD CHAINING

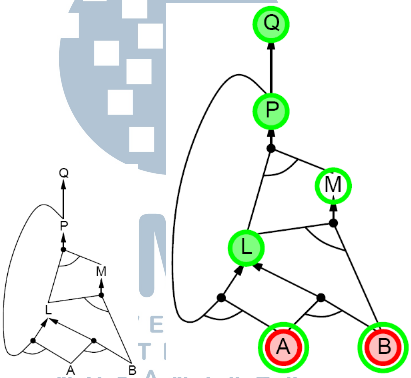
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B





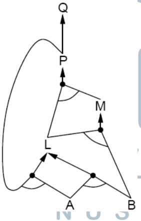
BACKWARD CHAINING

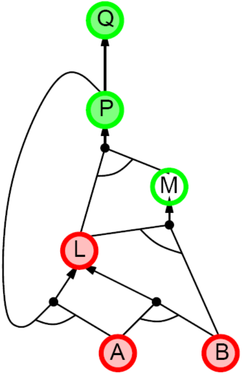
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



BACKWARD CHAINING

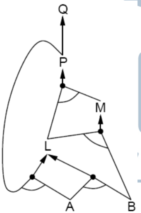
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B

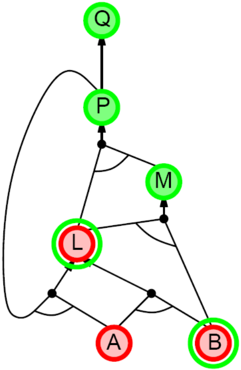




BACKWARD CHAINING

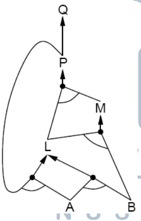
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B

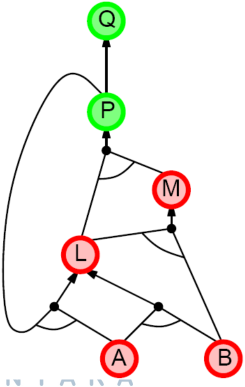




BACKWARD CHAINING

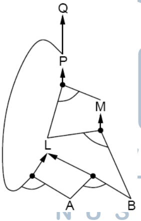
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

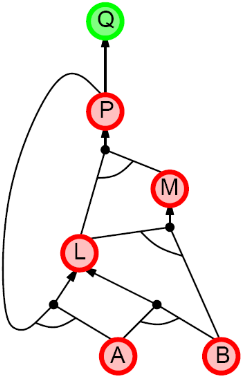




BACKWARD CHAINING

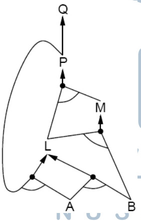
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

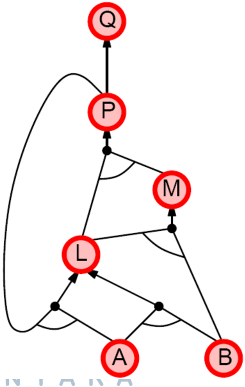




BACKWARD CHAINING

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





Next Week

- First-Order Logic
- Syntax of FOL
- Quantifiers
- Assertions & Queries
- Inference in FOL

Visi dan Misi

Visi

Menjadi Program Studi Strata Satu Informatika **unggulan** yang menghasilkan lulusan **berwawasan internasional** yang **kompeten** di bidang Ilmu Komputer (*Computer Science*), **berjiwa wirausaha** dan **berbudi pekerti luhur**.

Misi

1. Menyelenggarakan pembelajaran dengan teknologi dan kurikulum terbaik serta didukung tenaga pengajar profesional.
2. Melaksanakan kegiatan penelitian di bidang Informatika untuk memajukan ilmu dan teknologi Informatika.
3. Melaksanakan kegiatan pengabdian kepada masyarakat berbasis ilmu dan teknologi Informatika dalam rangka mengamalkan ilmu dan teknologi Informatika.