Manara University
Faculty of Engineering
Department of IT

Introduction to Artificial Intelligence

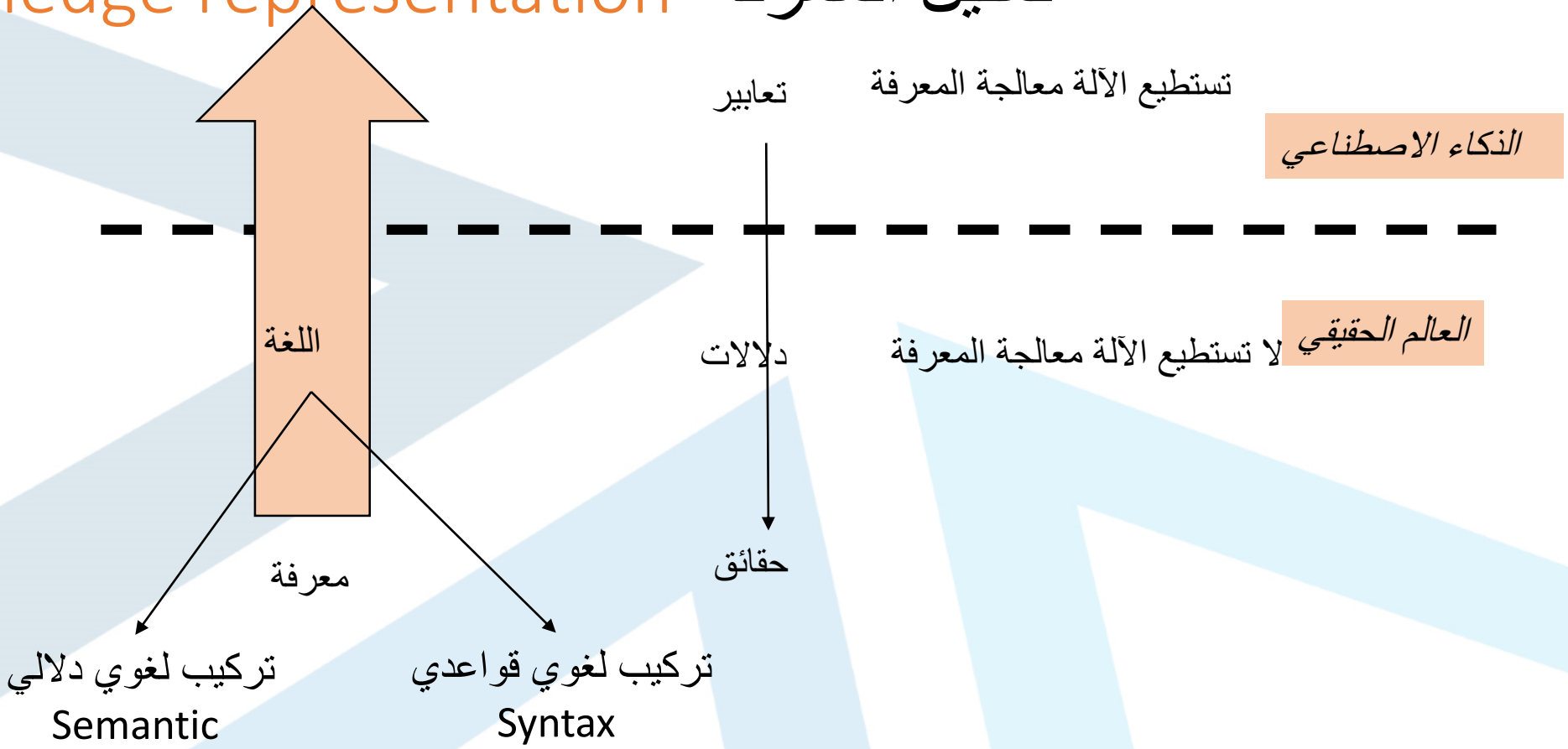
Knowledge representation- Propositional Logic

تمثيل المعرفة – منطق الفرضيات

Lecture 5

Prof Dr. Eng. Mariam M. Saii

Knowledge representation تمثيل المعرفة



Knowledge-Based Agents

- **KB = knowledge base**
  - A set of sentences or facts
  - e.g., a set of statements in a logic language
- **Inference**
  - Deriving new sentences from old
  - e.g., using a set of logical statements to infer new ones
- **A simple model for reasoning**
  - Agent is told or perceives new evidence
    - E.g., A is true
  - Agent then infers new facts to add to the KB
    - E.g., KB = { A -> (B OR C) }, then given A and not C we can infer that B is true
    - B is now added to the KB even though it was not explicitly asserted, i.e., the agent inferred B

- قاعدة المعرفة KB
  - مجموعة من الجمل أو الحقائق
  - مثل: مجموعة من العبارات في لغة منطقية
- الاستدلال
- اشتقاق جمل جديدة من جمل قديمة
- مثل: استخدام مجموعة من العبارات المنطقية لاستنتاج عبارات جديدة
- نموذج بسيط للاستدلال
  - يُخبر الفاعل أو يُدرك أدلة جديدة
  - مثل A :صحيح
  - ثم يستنتج الفاعل حقائق جديدة لإضافتها إلى قاعدة المعرفة
  - مثل{A -> (B OR C)} = KB :، فإذا كانت A وليس C،
- يمكننا استنتاج أن B صحيح يُضاف B الآن إلى قاعدة المعرفة حتى لو لم يُصرّح به صراحةً، أي أن الفاعل استنتج B

knowledge representation schemes

- Logical : use formal logic to represent knowledge e.g., state spaces, Prolog databases
- Procedural : knowledge as a set of instructions for solving a problem
  e.g., production systems, expert systems
- associationist: knowledge as objects/concepts and their associations
  e.g., semantic nets, conceptual dependencies
- structured: extend networks to complex data structures with slots/fillers
  e.g., scripts, frames

مخططات تمثيل المعرفة

- منطقي: استخدام المنطق الرسمي لتمثيل المعرفة (مثل: فضاءات الحالة، قواعد بيانات برولوغ).
- إجرائي: تمثل المعرفة كمجموعة من التعليمات لحل مشكلة. (مثل: أنظمة الإنتاج، النظم الخبيرة).
- ترابطي: تمثل المعرفة بكائنات/مفاهيم وارتباطاتها. (مثل: الشبكات الدلالية، التبعيات المفاهيمية).
- مهيكل: توسيع الشبكات لتشمل هياكل بيانات معقدة ذات فجوات/حشو. (مثل: البرامج النصية، الإطارات) .

Logic

- Logics are formal languages for representing information such that conclusions can be drawn
- Syntax defines the sentences in the language
- Semantics define the "meaning" or interpretation of sentences;
  - connects symbols to real events in the world,
  - i.e., define truth of a sentence in a world

المنطق

- المنطق : لغة تستخدم الاستنتاج الرياضي لاشتقاق معرفة جديدة
- القواعد: تعرّف بناء الجملة في اللغة.
- علم الدلالة: فيُعرّف "معنى" الجمل أو تفسيرها؛
- يربط الرموز بالأحداث الواقعية في العالم،
- أي يُعرّف حقيقة الجملة في عالم ما

Logic

المنطق

- E.g., the language of arithmetic
  - x+2 ≥ y is a sentence; x2+y > {} is not a sentence;
  - 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

Types of Logics

- **Propositional logic** deals with specific objects and concrete statements that are either true or false
  - E.g., John is married to Sue.
- **Predicate logic (also called first order logic, first order predicate calculus)** allows statements to contain variables, functions, and quantifiers
  - For all X, Y: If X is married to Y then Y is married to X.
- **Fuzzy logic** deals with statements that are somewhat vague, such as this paint is grey, or the sky is cloudy.
- **Probability** deals with statements that are possibly true, such as whether I will win the lottery next week.
- **Temporal logic** deals with statements about time, such as John was a student at UC for four years.
- **Modal logic** deals with statements about belief or knowledge, such as Mary believes that John is married to Sue,

أنواع المنطق

- منطق الفرضيات: يتعامل مع أشياء محددة وعبارات ملموسة إما صحيحة أو خاطئة. على سبيل المثال، جون متزوج من سو.
- منطق المسندات: (ويُسمى أيضًا منطق الرتبة الأولى، أو حساب المسند من الرتبة الأولى) يسمح للعبارات بأن تحتوي على متغيرات ودوال ومحددات كمية.

من أجل كل X و Y :إذا كان X متزوجاً منY ، فإن Y متزوج منX

- المنطق الضبابي : يتعامل مع العبارات الغامضة نوعاً ما، مثل "هذا الطلاء رمادي"، أو "السماء غائمة".
- المنطق الاحتمالي: يتعامل مع العبارات التي يُحتمل صحتها، مثل "هل سأفوز باليانصيب الأسبوع المقبل؟".
- المنطق الزمني : يتعامل مع العبارات المتعلقة بالزمن، مثل "كان جون طالباً في جامعة كاليفورنيا، لمدة أربع سنوات".
- المنطق النمطي: يتعامل مع العبارات المتعلقة بالاعتقاد أو المعرفة، مثل "تعتقد ماري أن جون متزوج من سو"

Types of Logics

أنواع المنطق

- منطق المسندات Predicate Logic تمثيل قوي للمعارف المنطقية
- منطق الفرضيات أكثر سهولة لكنه أقل فوة

منطق الفرضيات Propositional Logic

- نعبر عن الفرضيات (الحقائق) برموز
- الفرضية إما صحيحة أو خاطئة
- يمكن الربط بين الفرضيات لتشكيل جمل مركبة complex sentences باستخدام الروابط المنطقية Boolean connectives
- الجملة هي تصريح يأخذ قيمة TRUE or FALSE

Propositional logic البنية اللغوية لمنطق الفرضيات

- Symbols
- *True*, *False*
- Implication: =>
- Equivalence: <=>
- And (conjunction): ^
- Or (disjunction): V
- Negation: ¬
- Sentences = combination of the symbols, truth values, and operators
- Literals = Symbols and negated symbols (*A* and ¬*A* are both literals)

- Raining => Wet
- ¬(Busy ^ Sleeping)
- (A ^ B) V ¬C

- الثوابت constant يأخذ قيمة TRUE or FALSE
- الرموز التي تمثل الفرضيات
- الروابط المنطقية

- ^ AND, conjunction
- ˅ OR, disjunction
- → conditional (If then) Implication
- ↔Equivalence , biconditional
- ¬ Negation (unary)
- ( ) parentheses (grouping)

truth tables for connectives جدول الحقيقة

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

OR: P or Q is true or both are true.
XOR: P or Q is true but not both.

Implication is always true
when the premises are False!

Entailment الاستلزام

- Entailment means that 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

KB:

$(A \vee B) \wedge (\neg C \vee A)$

Example KB

A	B	C	KB
F	F	F	F
F	F	T	F
F	T	F	T
F	T	T	F
T	F	F	T
T	F	T	T
T	T	F	T
T	T	T	T

KB ⊨ S

Example KB

KB:

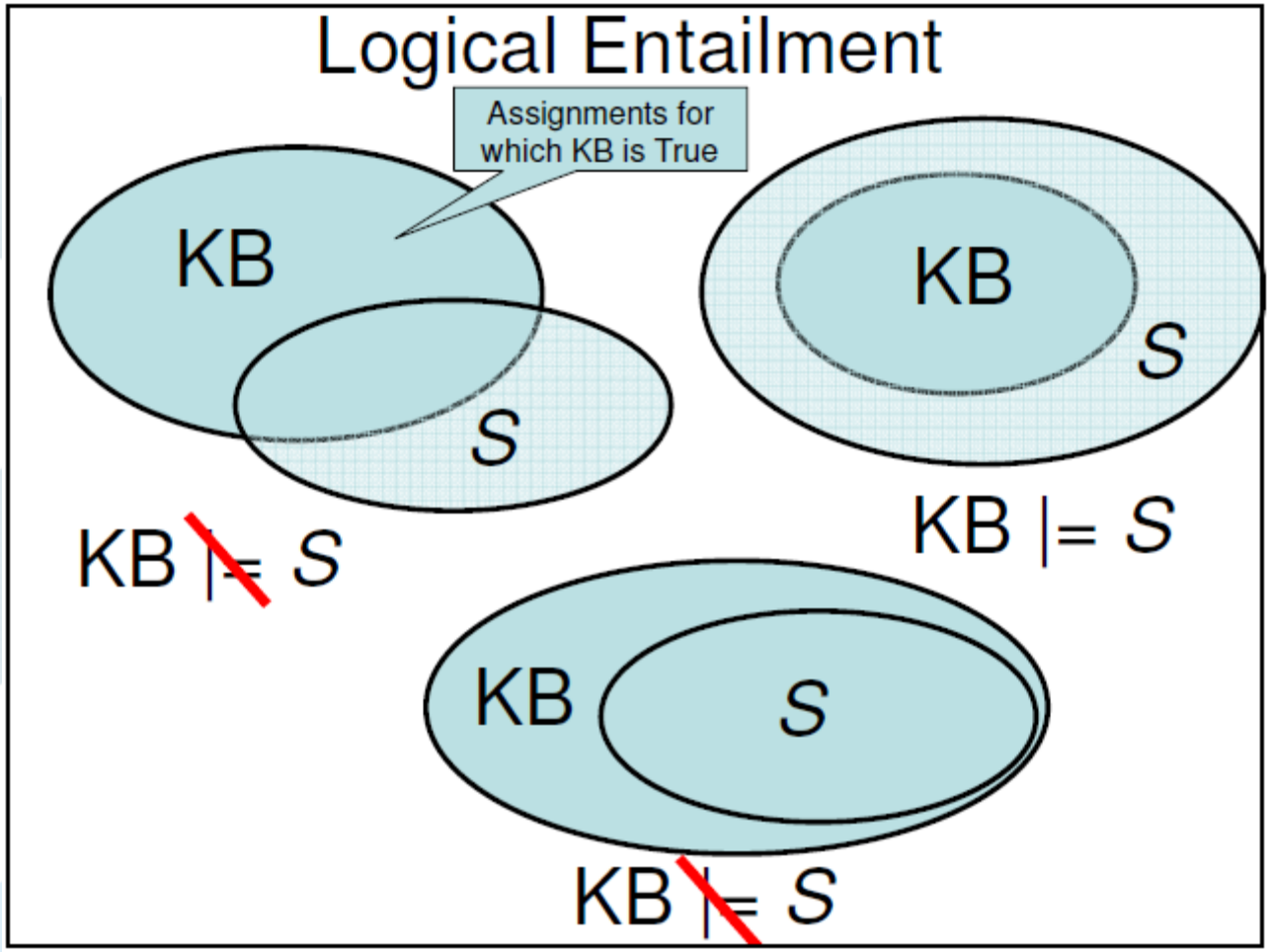
$(A \vee B) \wedge (\neg C \vee A)$

$S: A \wedge C$

A	B	C	KB	S
F	F	F	F	F
F	F	T	F	F
F	T	F	T	F
F	T	T	F	F
T	F	F	T	F
T	F	T	T	T
T	T	F	T	F
T	T	T	T	T

KB ⊭ S because KB is true but *S* is false

Logical entailment



Logical equivalence

Two sentences are logically equivalent if true in same models

$\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$

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

propositional logic: Syntax

- The proposition symbols P_1, P_2 etc 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, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Recursive propositional logic: Semantics

Each model/world specifies true or false for each proposition symbol E.g. $P_{1,2}$ $P_{2,2}$ $P_{3,1}$
false true false

With these symbols, 8 possible models, can be enumerated automatically.

Rules for evaluating truth with respect to a model *m*:

$\neg S$ is true if S is false

$S_1 \wedge S_2$ is true if S_1 is true and S_2 is true

$S_1 \vee S_2$ is true if S_1 is true or S_2 is true

$S_1 \Rightarrow S_2$ is true if S_1 is false or S_2 is true i.e., is false if S_1 is true and S_2 is false

$S_1 \Leftrightarrow S_2$ is true if $S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true

Simple recursive process evaluates an arbitrary sentence, e.g.,

$$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = \mathit{true} \wedge (\mathit{true} \vee \mathit{false}) = \mathit{true} \wedge \mathit{true} = \mathit{true}$$

Validity and satisfiability صحة الجمل وامكانية تحقيقها

A sentence is valid if it is true in all models,

تعد الجملة صحيحة اذا وفقط اذا كانت صحيحة في جميع التفاسير والنماذج

e.g., *True*, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem نظرية الاختصار:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in some model e.g., $A \vee B, C$

A sentence is unsatisfiable if it is false in all models e.g., $A \wedge \neg A$

Satisfiability is connected to inference via the following: $KB \models \alpha$ if and only if $(KB \wedge \neg \alpha)$ is unsatisfiable

(there is no model for which KB=true and α is false)

منطق الفرضيات والـCNF
Propositional Logic and CNF

- يمكننا تحويل أية جملة من جمل منطق الفرضيات إلى CNF. ونحتاج فقط إلى إزالة جميع معاملات الربط باستثناء الرابط OR (دون تعديل معنى الجمل).

Converting to CNF

التحويل إلى 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 ($\vee$ over $\wedge$) and flatten:

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

- حذف التضمين والتكافؤ implications and equivalence
- تضييق مجال النفي إلى عبارة وحيدة.
- استخدام القانونين التوزيعي والتجميعي للتحويل إلى ربط بـAND أو بـOR.
- إنشاء جملة منفصلة لكل عبارة جزئية مرتبطة بـAND.

$(A \vee B) \Leftrightarrow (C \Rightarrow D)$

- حذف $\Leftrightarrow$

$A \vee B) \Rightarrow (C \Rightarrow D)) \wedge ((C \Rightarrow D) \Rightarrow (A \vee B))))$

- حذف $\Rightarrow$

$A \vee B) \vee (\neg C \vee D)) \wedge (\neg (\neg C \vee D) \vee (A \vee B)))\neg)$

- إدخال النفي $\neg$ آخذين الأقواس بعين الاعتبار.

$A \wedge \neg B) \vee (\neg C \vee D)) \wedge ((C \wedge \neg D) \vee (A \vee B))\neg))$

- التوزيع Distribute

$(\neg A \vee \neg C \vee D) \wedge (\neg B \vee \neg C \vee D) \wedge (C \vee A \vee B) \wedge (\neg D \vee A \vee B)$

قواعد الاستدلال Inference Role

- في منطق الفرضيات يوجد العديد من النماذج التي يمكن أن نطلق عليها قواعد الاستدلال
- تصف هذه النماج كيفية استتخراج المعرفة من معارف معروفة مسبقا وممثلة بمنطق الفرضيات

قواعد الاستدلال Inference Role

- في منطق ننطلق من مقدمات Premises ونصل الى نتائج conclusion
-

$$premise \vdash conclusion$$

Modus Ponens:	$x \Rightarrow y,\ x \vdash y$
And Elimination:	$x_1 \wedge x_2 \wedge \ldots \wedge x_n \vdash x_i$
And-Introduction:	$x_1, x_2, \ldots x_n \vdash x_1 \wedge x_2 \wedge \ldots \wedge x_n$
Or-Introduction:	$x \vdash x \vee y \vee z \vee ..$
Double-Negation Elimination:	$\neg \neg x \vdash x$
Unit Resolution:	$x \vee y,\ x \vdash y$

قرار قواعد الاستدلال Inference Role

premise $\vdash$ *conclusion*

$$x \vee y, \neg y \vee z \vdash x \vee z$$

-or-

$$\neg x \Rightarrow y, y \Rightarrow z \vdash \neg x \Rightarrow z$$

المنطق والبرهان

- ليكن لدينا مايلي
  - قاعدة معارف ممثلة كمجموعة من جمل الفرضيات.
  - هدف معلن كجملة مكتوبة بمنطق الفرضيات.
  - قائمة من قواعد الاستنتاج.
- يمكننا كتابة برنامج للتطبيق التكراري لقواعد الاستنتاج على قاعدة المعارف بهدف الوصول إلى الهدف.

مثال

It will snow OR there will be a test.

Dave is Darth Vader OR it will not snow.

Dave is not Darth Vader.

Will there be a test?

Dave is Vader = c *Test = b* Snow = *a*

Knowledge Base (these are all true): $a \vee b$, $c \vee \neg a$, $\neg c$

- استنتاجاً نحن نعلم أن $b \vee c$ **is true**
- ومن خلال القرار وفقاً لقواعد الإنتاج نعلم أن

b is true

تطوير إجرائية للبرهان

- اشتقاق (أو دحض) الهدف من خلال مجموعة من الحقائق المنطقية وفقاً لشجرة بحث كبيرة (جداً).
- يمكن استخدام عدد كبير من قواعد الاستنتاج.
- ولن يكون من السهل اختيار أية القواعد التي يجب أن تطبق.

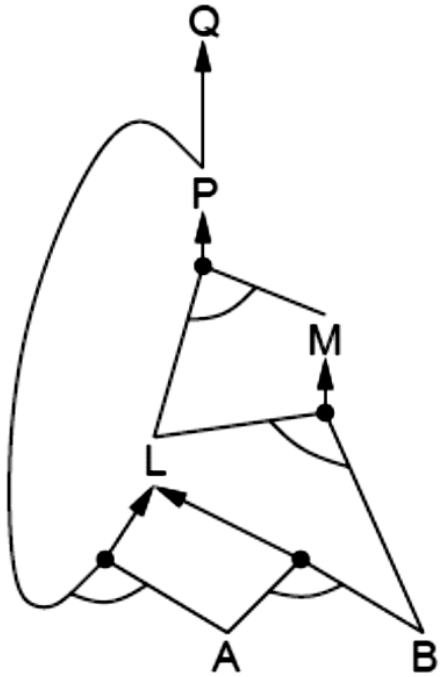
القرار و نمط الربط الشامل Resolution & CNF

- القرار هو قاعدة وحيدة للاستنتاج تعمل بفعالية على شكل خاص من أشكال الجمل.
- يسمى هذا الشكل الخاص صيغة القضية *clause form* أو نمط الربط الشامل *conjunctive normal form (CNF)*
- *خصائص هذا الشكل:*
- *كل جملة هي فصل (OR) بين الرموز.*
- *جميع الجمل مرتبطة ضمناً بـAND.*

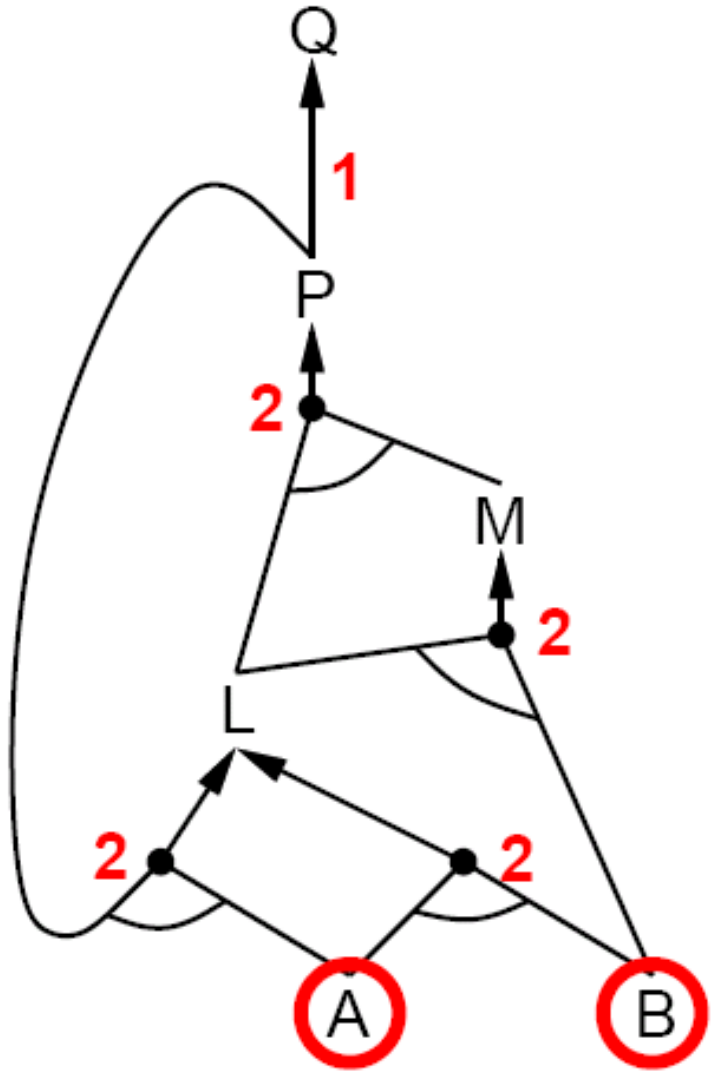
Forward chaining

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

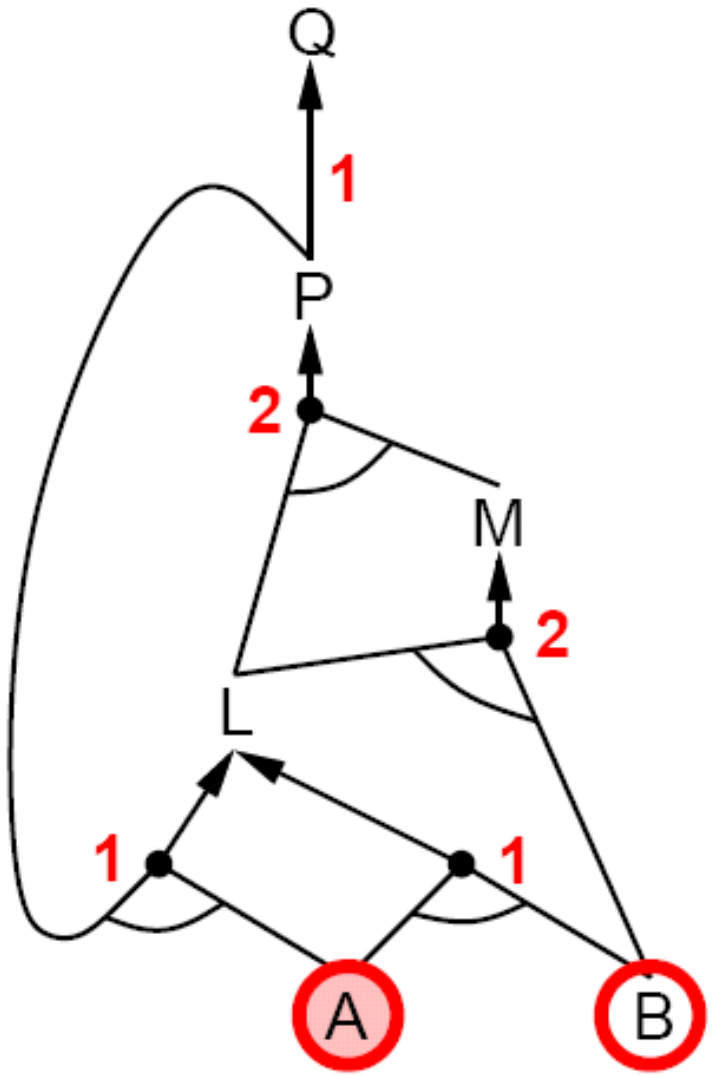
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



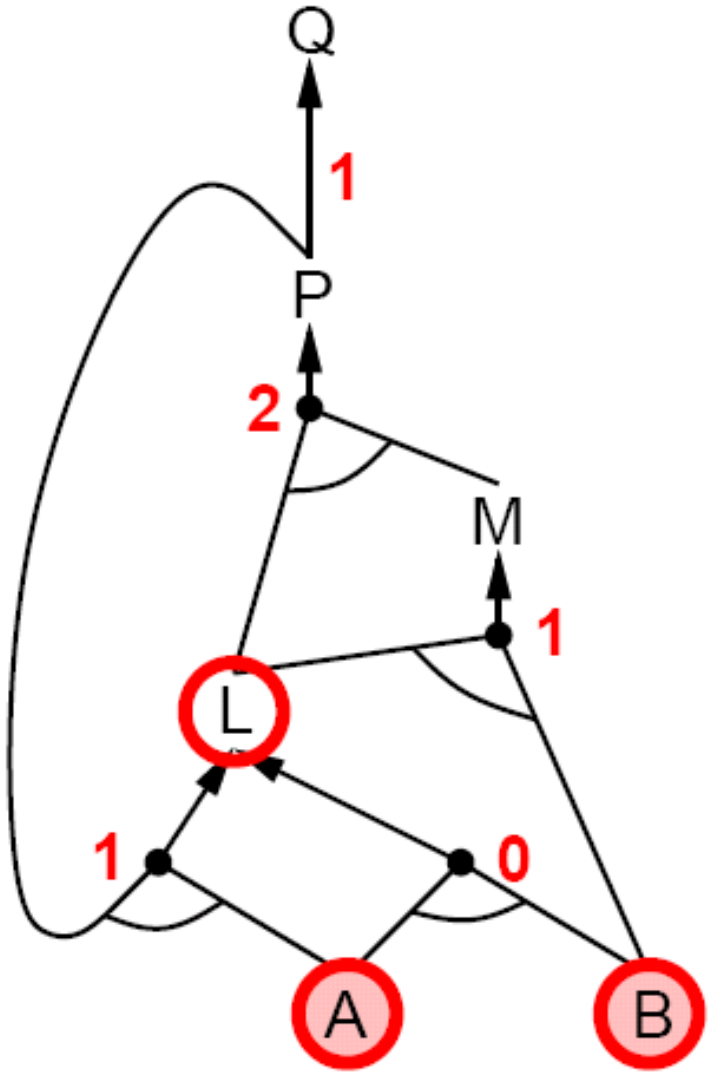
Forward chaining example



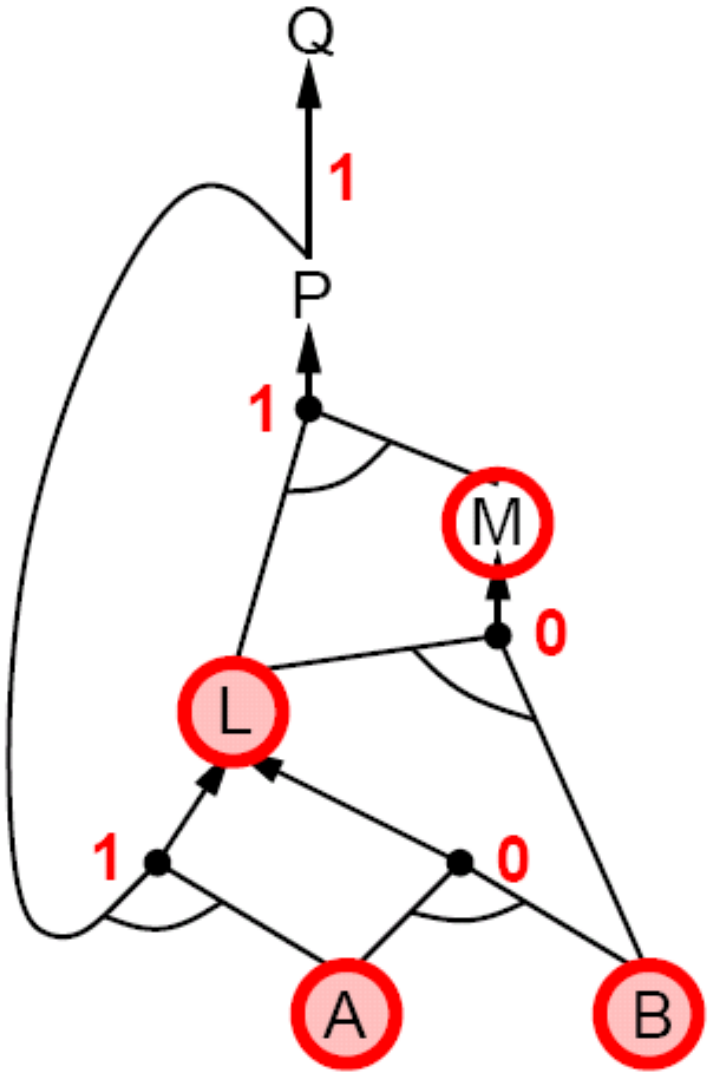
Forward chaining example



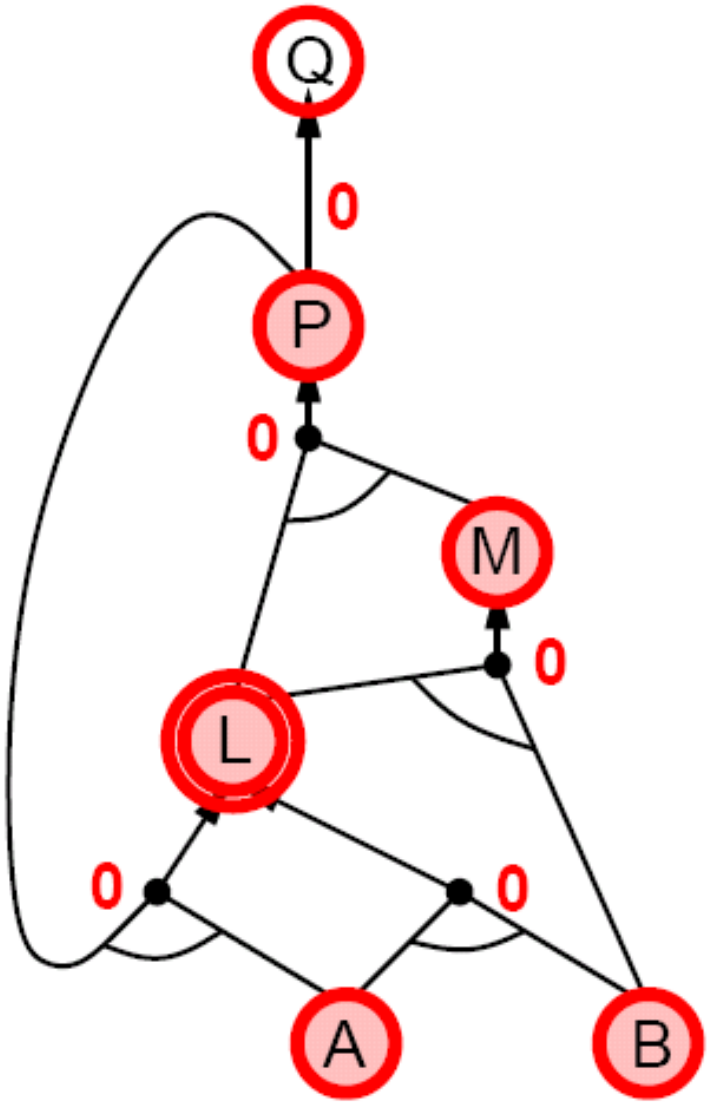
Forward chaining example



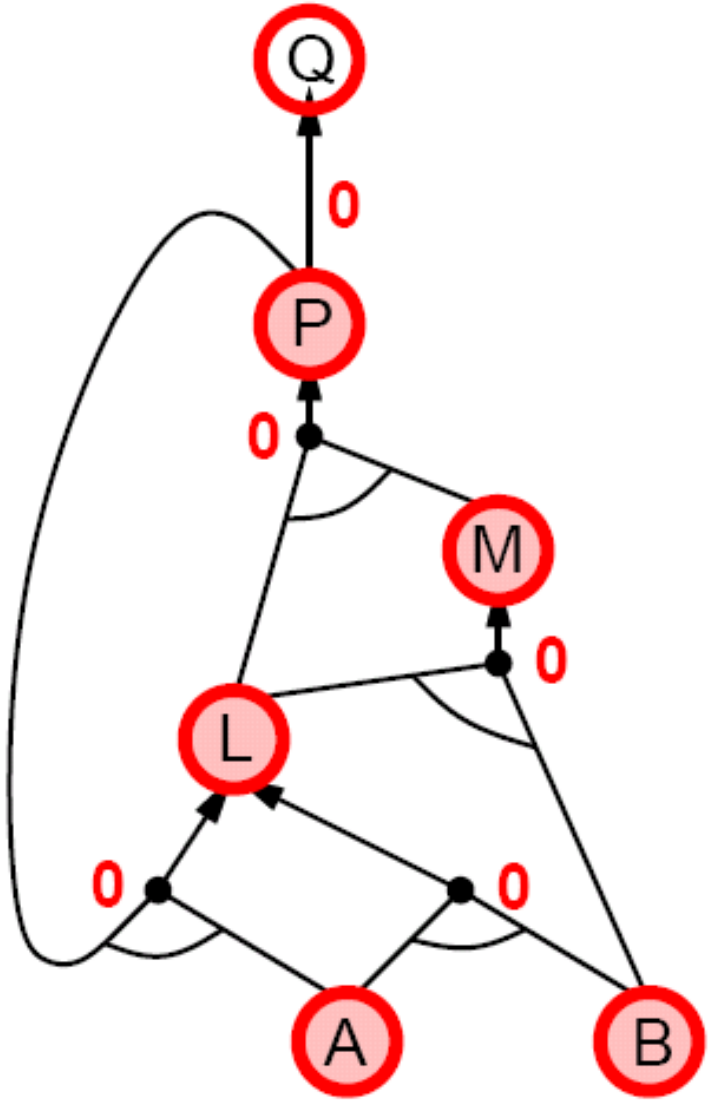
Forward chaining example



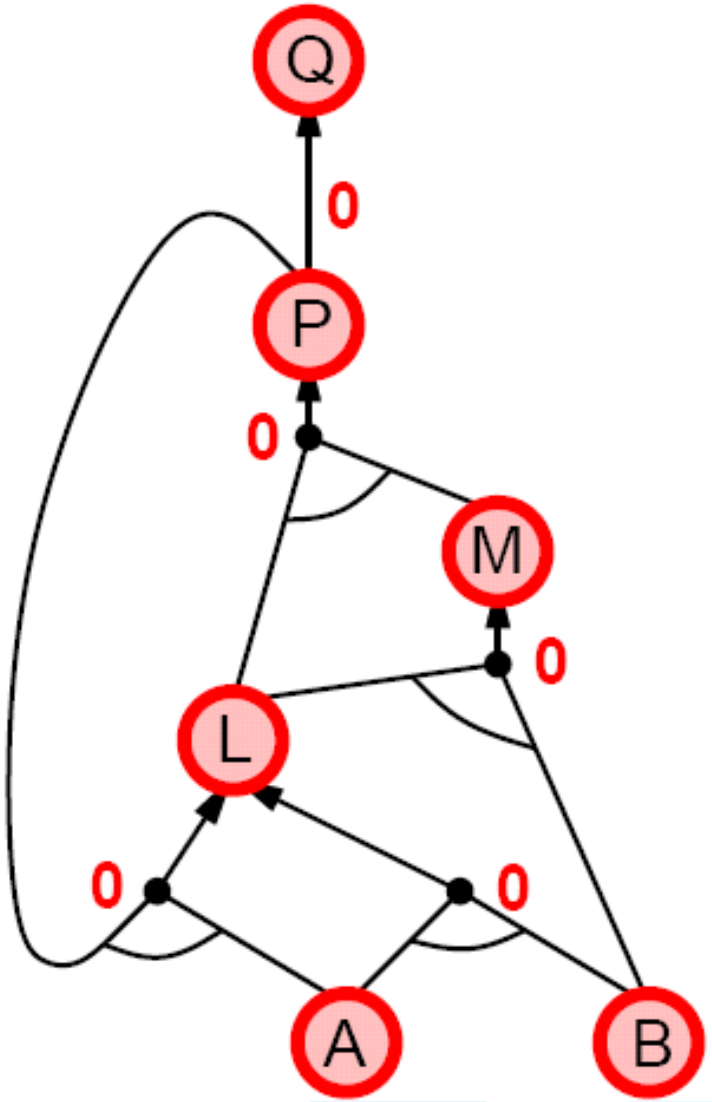
Forward chaining example



Forward chaining example



Forward chaining example



Backward chaining

Idea: work backwards from the query q:

to prove q by BC,

check if q is known already, or

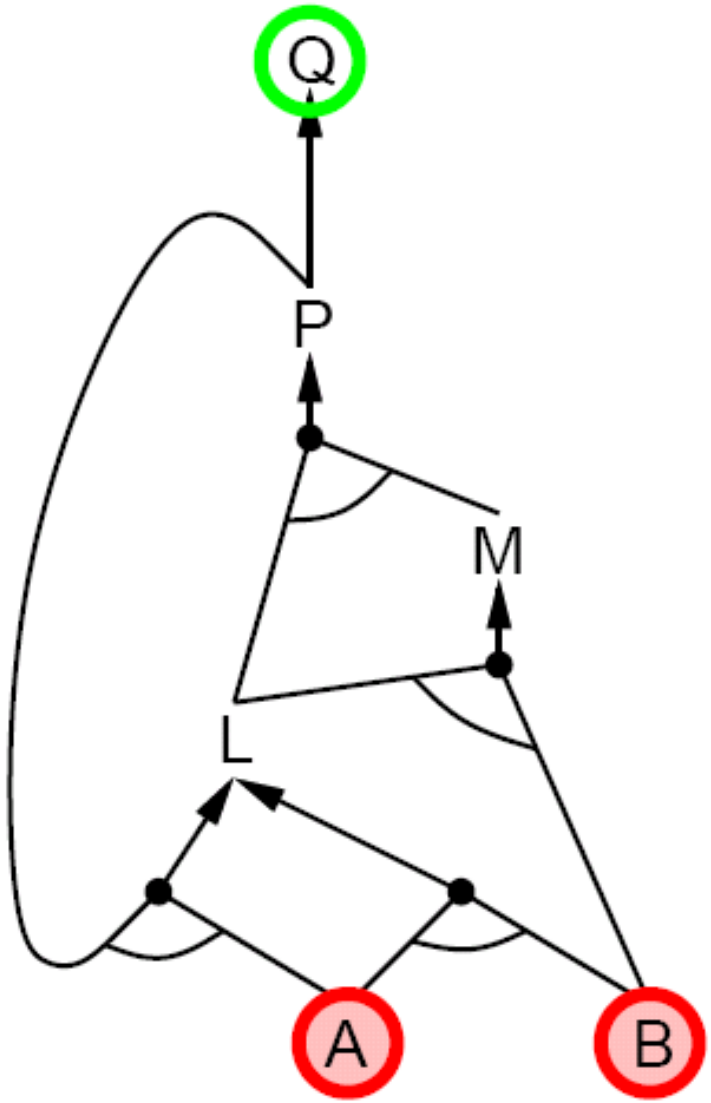
prove by BC all premises of some rule concluding q

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

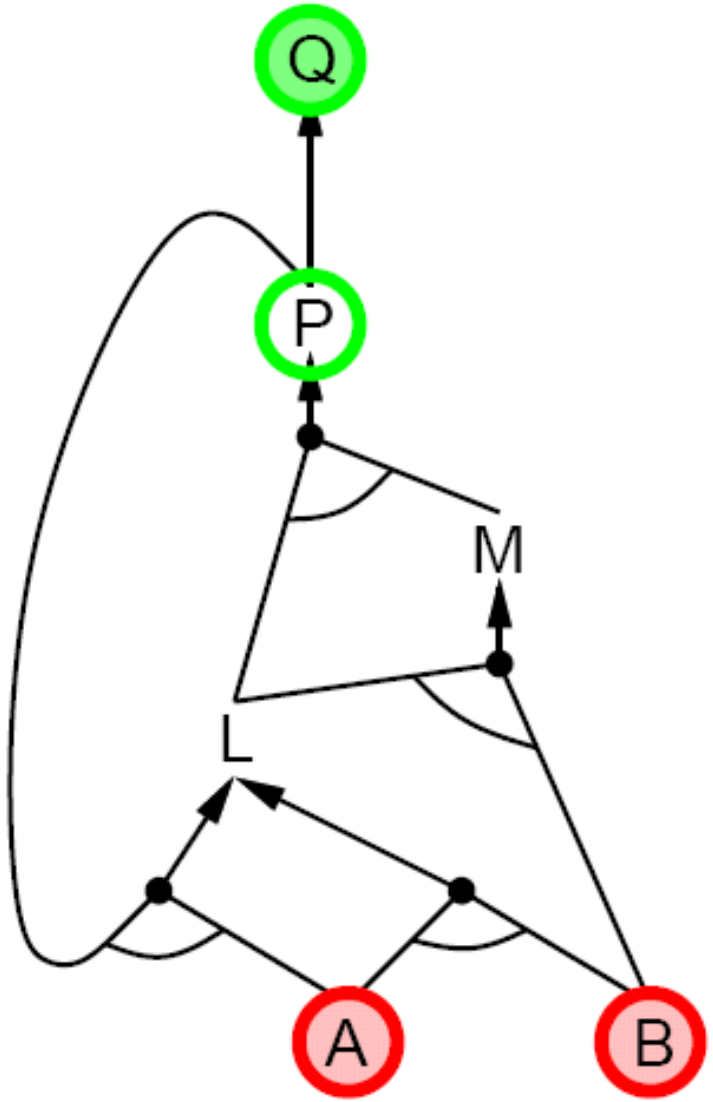
Avoid repeated work: check if new subgoal

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

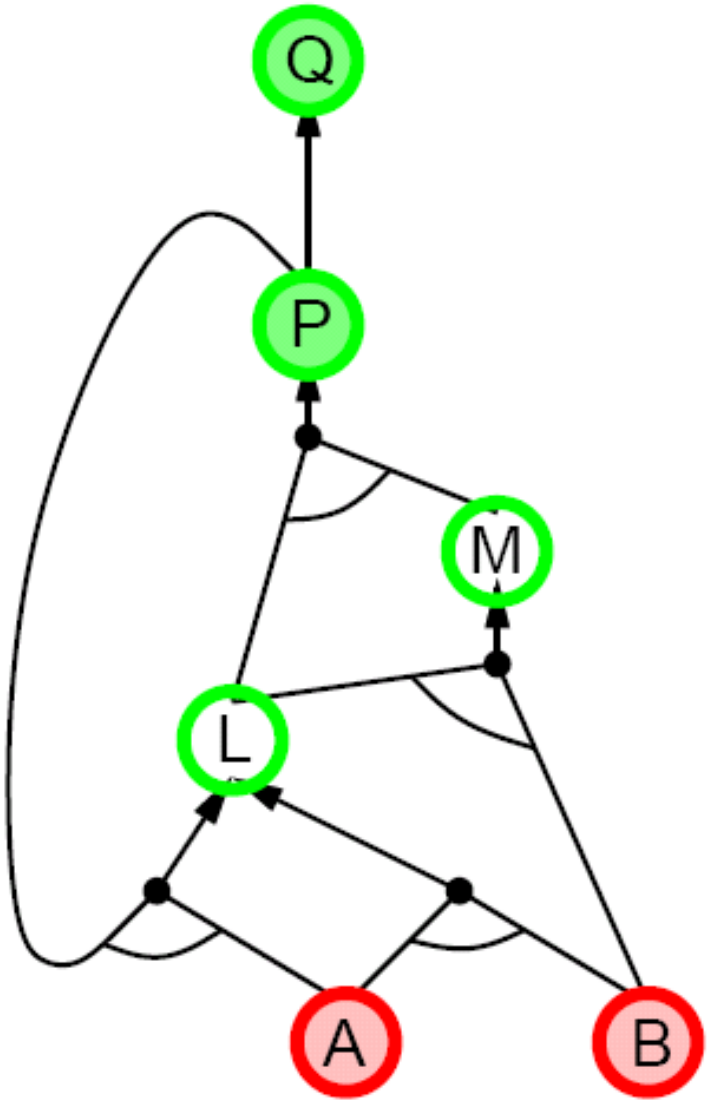
Backward chaining example



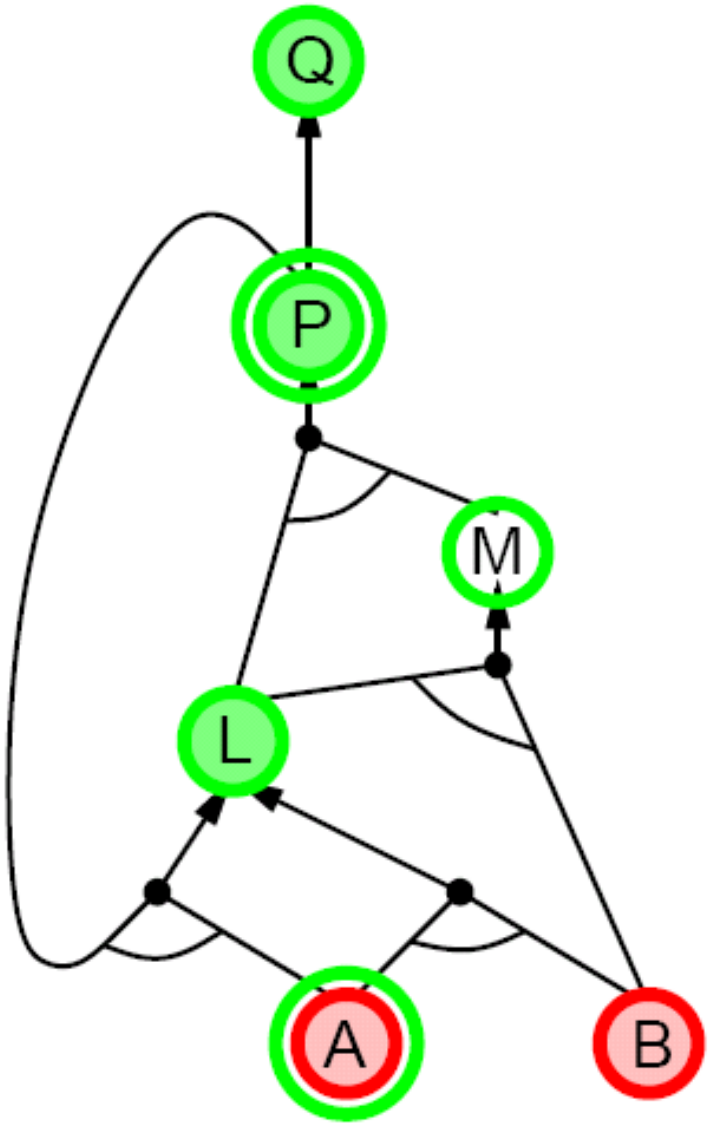
Backward chaining example



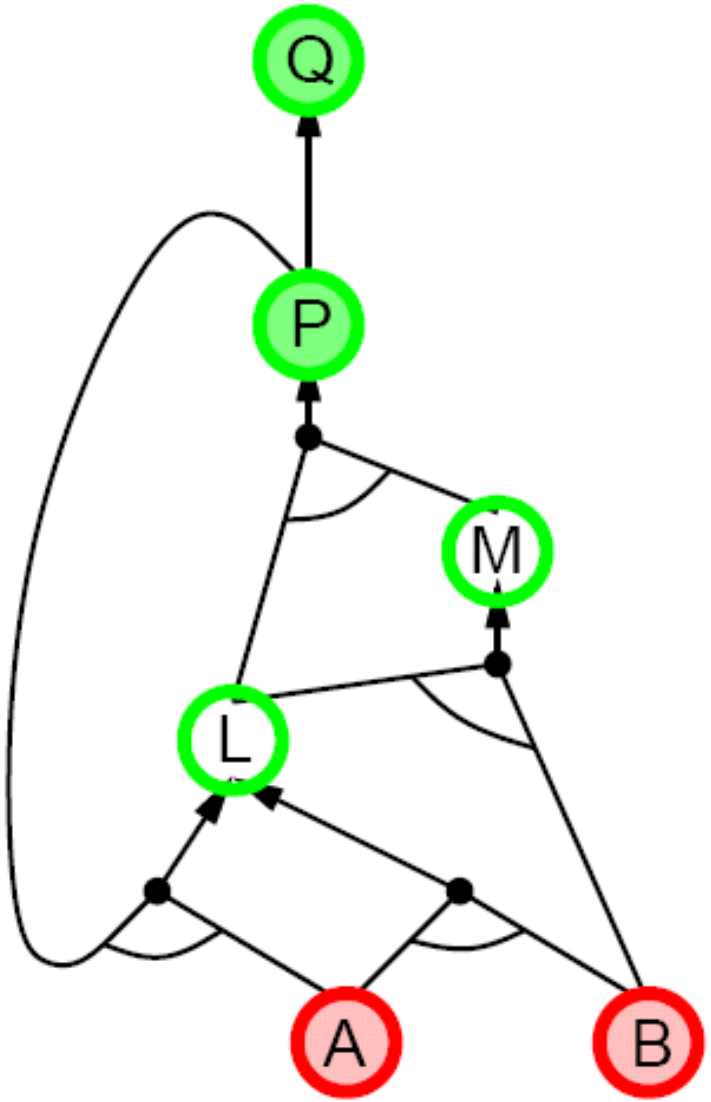
Backward chaining example



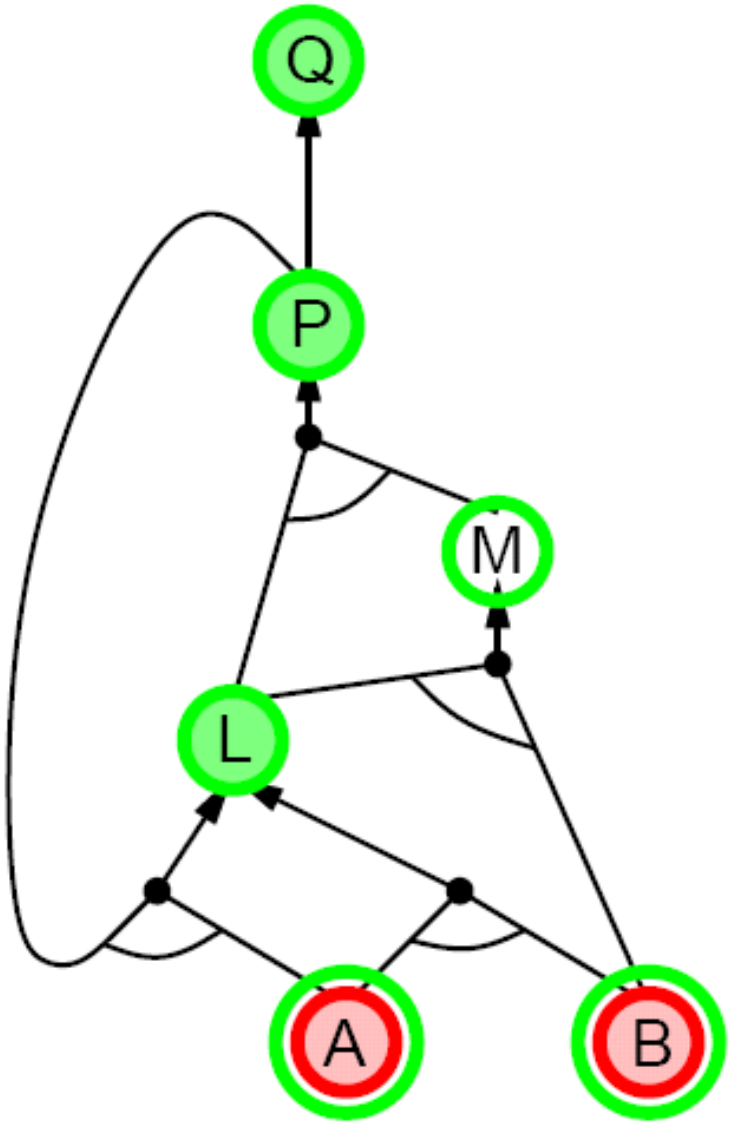
Backward chaining example



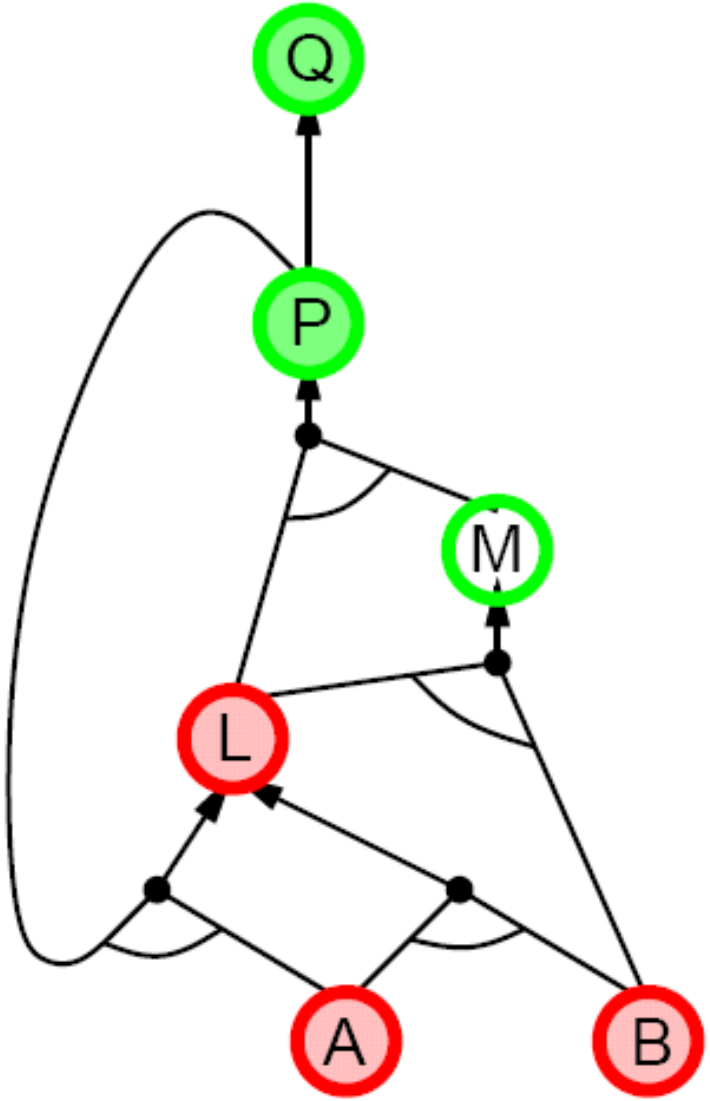
Backward chaining example



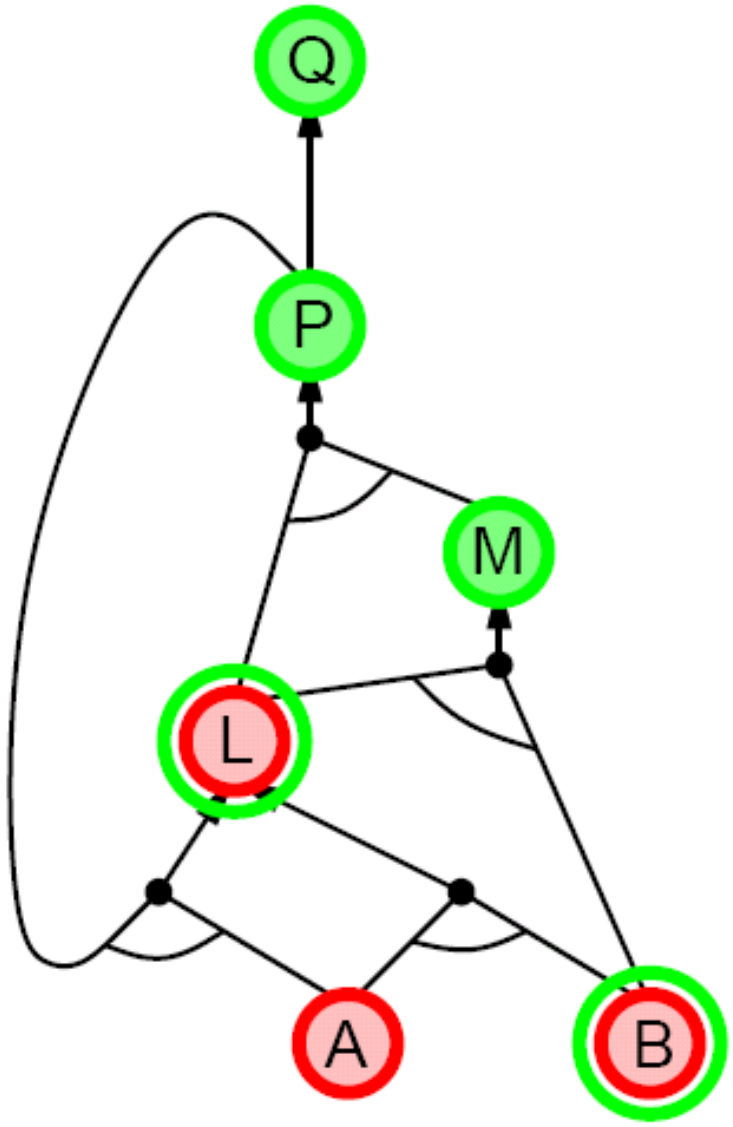
Backward chaining example



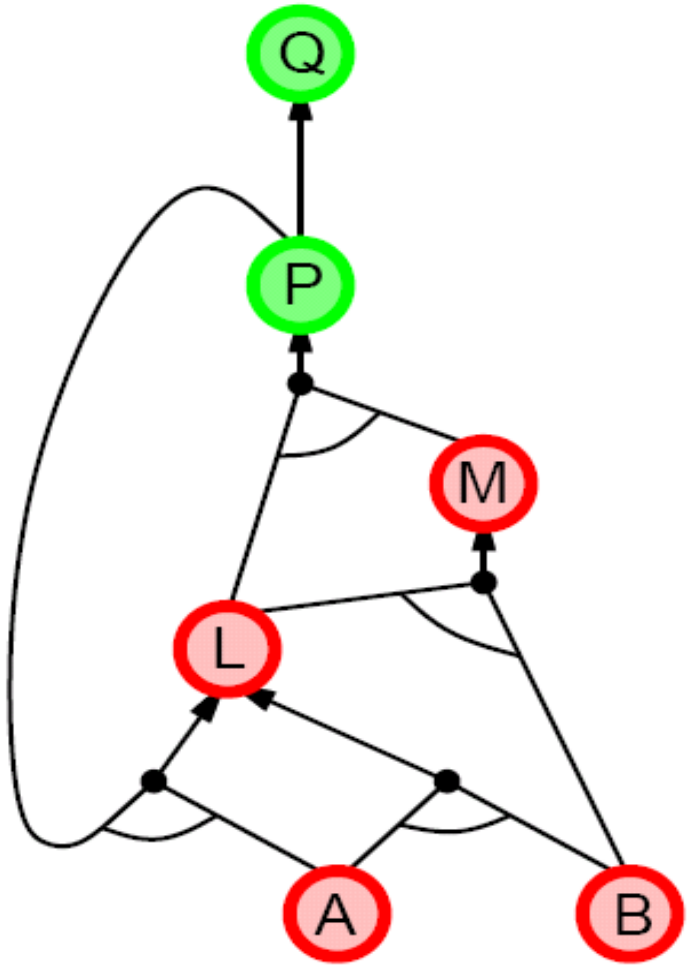
Backward chaining example



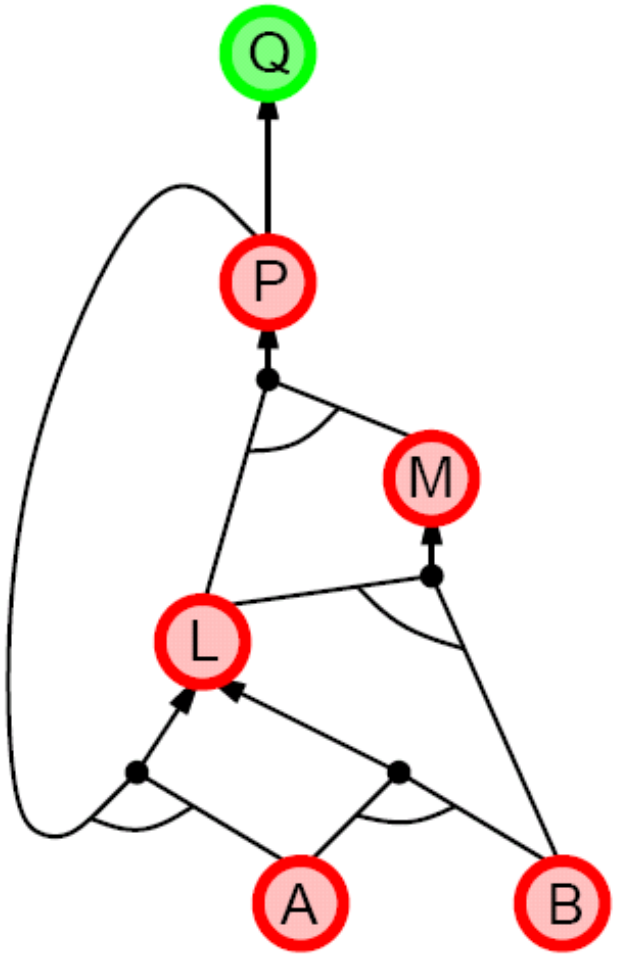
Backward chaining example



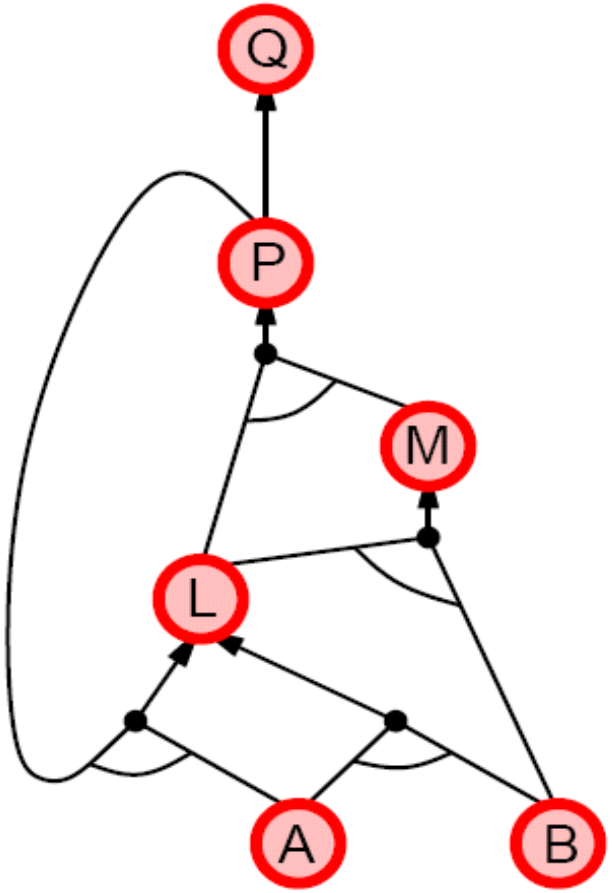
Backward chaining example



Backward chaining example



Backward chaining example



Forward vs. backward chaining

FC is data-driven, cf. 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

ليكن لدينا مجموعة القواعد الآتية:

Rule 1: P ^ Q ⟶ Goal

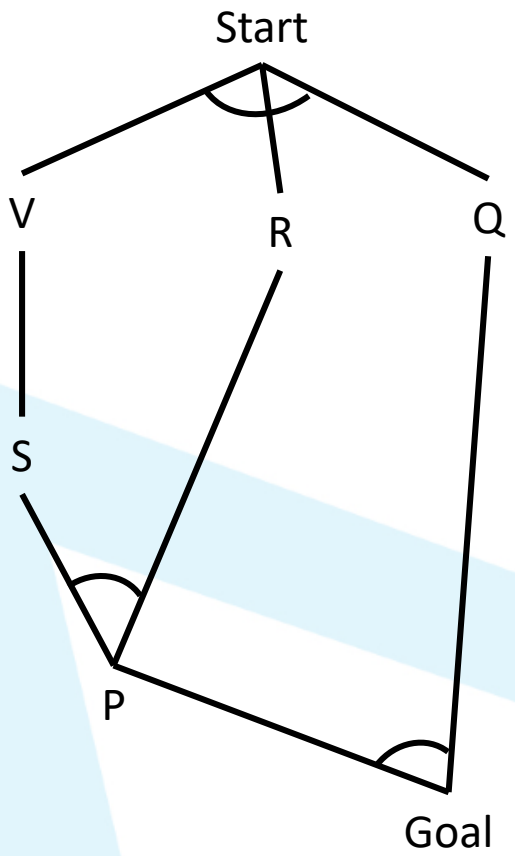
Rule 2: R^S ⟶ P

Rule 3: W^R ⟶ Q

Rule 4: T^U ⟶ Q

Rule 5: V ⟶ S

Rule 6: Start ⟶ V^R^Q

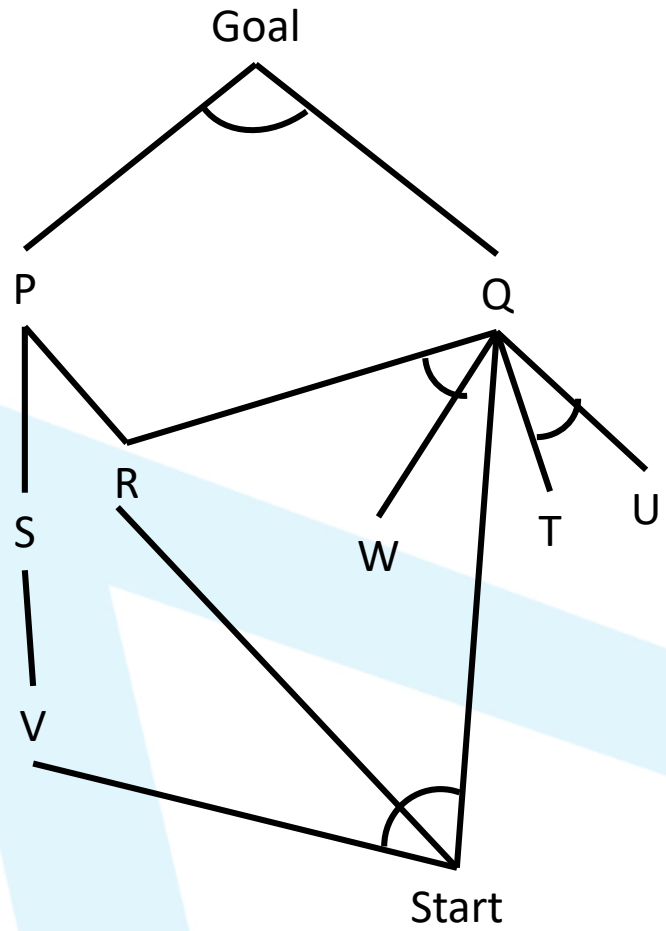


Seq.	Working memory	Conflict rules	Rule fire
0	start	6	6
1	Start, V, R, Q	6, 5	5
2	Start, V, R, Q, S	6,5,2	2
3	Start, V, R, Q,S,P	6,5,2,1	1
4	Start, V, R, Q,S,P, goal	6,5,2,1	done

ليكن لدينا مجموعة القواعد الآتية:

Rule 1: P ^ Q → Goal
Rule 2: R^S → P
Rule 3: W^R → Q
Rule 4: T^U → Q
Rule 5: V → S
Rule 6: Start → V^R^Q

Seq.	Working memory	Conflict rules	Rule fire
0	Goal	1	1
1	Goal, P , Q	1,2,3,4	2
2	Goal, P , Q, R, S	1,2,3,4,5	3
3	Goal, P , Q, R, S, W	1,2,3,4,5	4
4	Goal, P , Q, R, S, W, T, U	1,2,3,4,5	5
5	Goal, P , Q, R, S, W, T, U, V	1,2,3,4,5,6	6
6	Goal, P , Q, R, S, W, T, U, V, Start	1,2,3,4,5,6	Done



And Or Tree