第二章 命题逻辑

戴洪良

计算机科学与技术学院/人工智能学院

hongldai@nuaa.edu.cn

什么是命题（Proposition）？

- 在一般的语境下，命题就是一类满足特定条件的**陈述句所表达的背后的语义。**这个条件是：**能够被一致地赋予真值**（truth value, i.e. true or false）。

- 「雪是白色的」和「雪是白色的」是不是同一个句子?
- 「雪是白色的」和「Snow is white」是不是同一个句子?
- 「雪是白色的」和「Snow is white」的意义是否相同?

例子

- 南京在江苏。
- 2+2=4
- 2+2=5
- 所有的牛都是棕色的。
- 家里有油就会引来美军。

是命题

- 你要跳舞吗?
- 把门关上。
- 3+x=7

不是命题

什么是命题逻辑（Propositional Logic）？

- 命题逻辑（Propositional Logic）是应用一套形式化规则对用符号表示的描述性陈述进行推理的系统。

什么是命题逻辑（Propositional Logic）？

- 命题逻辑（Propositional Logic）是应用一套形式化规则对用符号表示的描述性陈述进行推理的系统。

命题逻辑符号

- 命题符号（Propositional Symbol）：用字母p、q、r、P、Q、R等代表一个命题，也称为命题变元（Propositional Variable）。
- 真值符号：True (T)、False (F)，分别表示命题为真或为假。
- 联结词（Connectives）：可通过联结词对已有命题进行组合，得到新命题。

联结词（Connectives）

联结词	符号	含义
非（not）	$\neg$	命题否定，¬p 读作 “非p”
与（and）	$\wedge$	命题合取，p∧q 读作 “p且q”
或（or）	$\vee$	命题析取，p∨q 读作 “p或q”
条件（conditional）	$\rightarrow$、$\Rightarrow$	命题蕴含，p→q读作 “如果p则q” 或 “p蕴含q”
双向条件（biconditional）	$\leftrightarrow$、$\Leftrightarrow$	命题双向蕴含（bi-implication），p↔q 读作 “p当且仅当q”

联结词（Connectives）

联结词	符号	含义
非（not）	$\neg$	命题否定，¬p 读作 “非p”
与（and）	$\wedge$	命题合取，p∧q 读作 “p且q”
或（or）	$\vee$	命题析取，p∨q 读作 “p或q”
条件（conditional）	$\rightarrow$、$\Rightarrow$	命题蕴含，p→q读作 “如果p则q” 或 “p蕴含q”
双向条件（biconditional）	$\leftrightarrow$、$\Leftrightarrow$	命题双向蕴含（bi-implication），p↔q 读作 “p当且仅当q”

可用联结词和命题变元组成**命题公式**，如：p∧q→r

运算优先顺序：¬、∧、∨、→、↔。可用括号（）控制优先级。

联结词（Connectives）

联结词	符号	含义
非（not）	$\neg$	命题否定，¬p 读作 “非p”
与（and）	$\wedge$	命题合取，p∧q 读作 “p且q”
或（or）	$\vee$	命题析取，p∨q 读作 “p或q”
条件（conditional）	$\rightarrow$、$\Rightarrow$	命题蕴含，p→q读作 “如果p则q” 或 “p蕴含q”
双向条件（biconditional）	$\leftrightarrow$、$\Leftrightarrow$	命题双向蕴含（bi-implication），p↔q 读作 “p当且仅当q”

可用联结词将已有命题组合成新命题，将这种命题称为复合命题（compound proposition）
将不能用联结词和其他命题组合成的命题称为原子命题（atomic proposition）

原子命题、复合命题

- 原子命题例子：
  - 南京在江苏
  - 13能被6整除
  - 存在最大的素数

- 复合命题例子：
  - 天下雨了而且我没带伞

联结词的真值表（Truth Table）

- 真值表：列出所有命题变元的可能取值组合下，命题公式的对应真值

p	q	$\neg p$	$p \land q$	$p \lor q$	$p \Rightarrow q$	$p \Leftrightarrow q$
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

联结词的真值表（Truth Table）

- 真值表：列出所有命题变元的可能取值组合下，命题公式的对应真值

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

解释（interpretation）：对所有命题变元进行一次赋值。

联结词的真值表（Truth Table）

- 真值表：列出所有命题变元的可能取值组合下，命题公式的对应真值

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

P→Q

- 为什么前提P为假时命题总为真?

如果你赢了比赛，我给你100块钱。

从含义上看，“如果P则Q”只对P为真的情况做了限定

1）你赢了比赛，我给了100块钱
2）你赢了比赛，我没给100块钱
3）你输了比赛，我给了100块钱
4）你输了比赛，我没给100块钱

P是充分条件

p	q	$p \Rightarrow q$
False	False	True
False	True	True
True	False	False
True	True	True

P↔Q

我给你100块钱 当且仅当 你赢了比赛。

1）你赢了比赛，我给了100块钱
2）你赢了比赛，我没给100块钱
3）你输了比赛，我给了100块钱
4）你输了比赛，我没给100块钱

p	q	$p \Leftrightarrow q$
False	False	True
False	True	False
True	False	False
True	True	True

翻译

- 有如下句子：
  - It is not sunny today and it is colder than yesterday.
  - We will go swimming only if it is sunny.
  - If we do not go swimming then we will take a canoe trip.
  - If we take a canoe trip, then we will be home by sunset.
- 令：
  - p = It is sunny today
  - q = It is colder than yesterday
  - r = We will go swimming
  - s = We will take a canoe trip
  - t = We will be home by sunset

翻译

- 有如下句子：
  - It is not sunny today and it is colder than yesterday. $\neg p \wedge q$
  - We will go swimming only if it is sunny. $r \rightarrow p$
  - If we do not go swimming then we will take a canoe trip. $\neg r \rightarrow s$
  - If we take a canoe trip, then we will be home by sunset. $s \rightarrow t$

- 令：
  - p = It is sunny today
  - q = It is colder than yesterday
  - r = We will go swimming
  - s = We will take a canoe trip
  - t = We will be home by sunset

矛盾式和重言式

- 一些命题公式在任何解释（interpretation）下都得到相同的真值（总为真或总为假）。
- 总为假：矛盾式（Contradiction）

$$P \wedge \neg P$$

- 总为真：重言式（Tautology）

$$P \vee \neg P$$

只有P、Q都为假才为真

$$\neg(P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q)$$

$$\neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$$

DeMorgan's Laws

只有P、Q都为真才为假

p	q	$p \Leftrightarrow q$
False	False	True
False	True	False
True	False	False
True	True	True

模型、有效性和可满足性

- 如果一个解释（interpretation）令一个句子集合（a set of sentences）中的所有句子为真，那么称它为该句子集合的一个模型（model）
- 句子（sentence）：在命题逻辑中可理解为用符号表示的命题

例子：

句子：P∨Q

P	Q	$P \vee Q$
True	*True*	*True*
True	*False*	*True*
False	*True*	*True*
False	*False*	*False*

模型、有效性和可满足性

- 如果一个解释（interpretation）令一个句子集合（a set of sentences）中的所有句子为真，那么称它为该句子集合的一个模型（model）
- 句子（sentence）：在命题逻辑中可理解为用符号表示的命题

例子：

句子：P∨Q

P	Q	$P \vee Q$
True	*True*	*True*
True	*False*	*True*
False	*True*	*True*
False	*False*	*False*

模型、有效性和可满足性

- 如果一个解释（interpretation）令一个句子集合（a set of sentences）中的所有句子为真，那么称它为该句子集合的一个模型（model）
- 句子（sentence）：在命题逻辑中可理解为用符号表示的命题

例子：

句子： $P \vee Q$

$(P \vee Q) \wedge \neg Q$

$((P \vee Q) \wedge \neg Q) \Rightarrow P$

P	Q	$P \vee Q$	$(P \vee Q) \wedge \neg Q$	$((P \vee Q) \wedge \neg Q) \Rightarrow P$
True	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
False	*True*	*True*	*False*	*True*
False	*False*	*False*	*False*	*True*

模型、有效性和可满足性

- 如果一个解释（interpretation）令一个句子集合（a set of sentences）中的所有句子为真，那么称它为该句子集合的一个模型（model）
- 句子（sentence）：在命题逻辑中可理解为用符号表示的命题

例子：

句子：$P \vee Q$

$(P \vee Q) \wedge \neg Q$

$((P \vee Q) \wedge \neg Q) \Rightarrow P$

P	Q	$P \vee Q$	$(P \vee Q) \wedge \neg Q$	$((P \vee Q) \wedge \neg Q) \Rightarrow P$
True	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
False	*True*	*True*	*False*	*True*
False	*False*	*False*	*False*	*True*

模型、有效性和可满足性

- 如果一个解释（interpretation）令一个句子集合（a set of sentences）中的所有句子为真，那么称它为该句子集合的一个模型（model）
- 如果一个句子有模型，那么称它是可满足的（satisfiable）
  - 等同于：至少有一个解释使它为真

P	Q	$P \vee Q$	$(P \vee Q) \wedge \neg Q$	$((P \vee Q) \wedge \neg Q) \Rightarrow P$
True	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
False	*True*	*True*	*False*	*True*
False	*False*	*False*	*False*	*True*

模型、有效性和可满足性

- 如果一个解释（interpretation）令一个句子集合（a set of sentences）中的所有句子为真，那么称它为该句子集合的一个模型（model）
- 如果一个句子有模型，那么称它是可满足的（satisfiable）
  - 等同于：至少有一个解释使它为真
- 如果一个句子在所有解释下都为真，则称它是有效的（valid）
  - 等同于：重言式

P	Q	$P \vee Q$	$(P \vee Q) \wedge \neg Q$	$((P \vee Q) \wedge \neg Q) \Rightarrow P$
True	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
False	*True*	*True*	*False*	*True*
False	*False*	*False*	*False*	*True*

知识库（Knowledge Base）和语义蕴含（Entailment）

知识库（Knowledge Base）

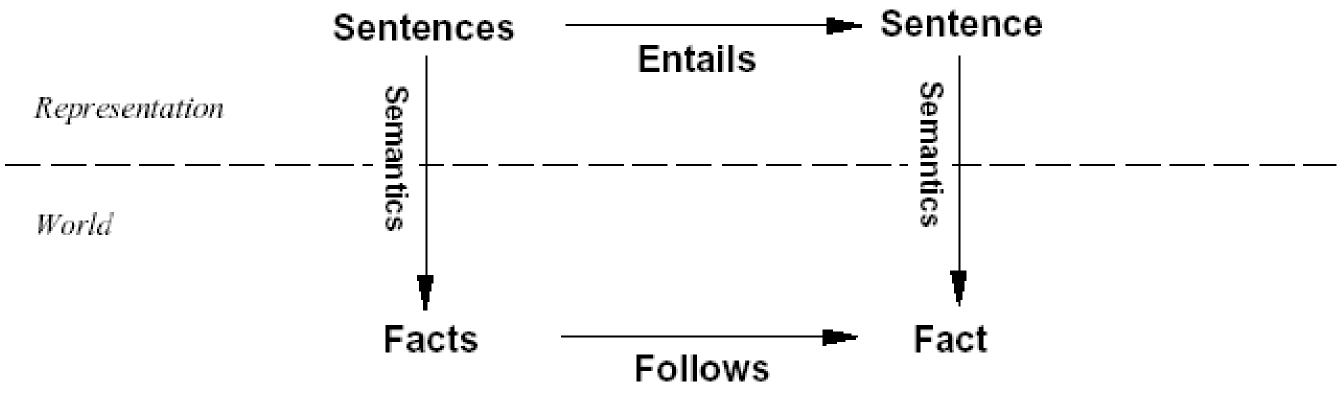
- A knowledge base (KB) is a set of representations of facts about the world.
- 知识库是一个句子的集合（a set of sentences）。

例: KB: $P \vee Q$

$P \Leftrightarrow Q$

语义蕴含（Entailment)

- Entailment (语义蕴含) reflects the relation of one fact in the world following from the others.



- 知识库KB蕴含（entails）句子α 当且仅当 α在所有KB为真的情况下都为真。表示为

$$KB \models \alpha$$

语义蕴含（Entailment）

- 知识库KB蕴含（entails）句子 α 当且仅当 α 在所有KB为真的情况下都为真。表示为

KB: $P \vee Q$ $\qquad\qquad$ α: $(P \vee \neg Q) \wedge Q$

$P \Leftrightarrow Q$

$$KB \models \alpha \quad ?$$

		KB		α
P	Q	$P \vee Q$	$P \Leftrightarrow Q$	$(P \vee \neg Q) \wedge Q$
True	*True*	*True*	*True*	*True*
True	*False*	*True*	*False*	*False*
False	*True*	*True*	*False*	*False*
False	*False*	*False*	*True*	*False*

语义蕴含（Entailment）

- 知识库KB**蕴含（entails）**句子α 当且仅当 α在所有KB为真的情况下都为真。表示为

KB: $P \vee Q$
$P \Leftrightarrow Q$

α: $(P \vee \neg Q) \wedge Q$

$$KB \models \alpha \quad ?$$

		KB		α	
P	Q	$P \vee Q$	$P \Leftrightarrow Q$	$(P \vee \neg Q) \wedge Q$	
True	*True*	*True*	*True*	*True*	✓
True	*False*	*True*	*False*	*False*	
False	*True*	*True*	*False*	*False*	
False	*False*	*False*	*True*	*False*	

语义蕴含（Entailment）

- 用$M(\alpha)$表示α的所有模型的集合。那么$KB \models \alpha$当且仅当$M(KB) \subseteq M(\alpha)$。

说明:$M(\alpha)$ 也就是所有使 α 为真的解释。$KB \models \alpha$ 直观理解就是在使 KB 为真的世界里面 α 也全都要为真。

证: 先证 $\Rightarrow$
$\forall m \in M(KB)$，因为 $KB \models \alpha$，可推出 m 也使 α 为真，所以 $m \in M(\alpha)$，所以 $M(KB) \subseteq M(\alpha)$。

再证 $\Leftarrow$
$\forall m \in M(KB)$，$m \in M(\alpha)$，因此对于所有使得 KB 为真的解释，α 也为真，根据定义即 $KB \models \alpha$。

证毕。

可靠和完备的推理（Inference）

- 推理（Inference）就是根据KB中的已有句子获得新句子的过程
  - 我们希望在计算机上进行推理

- 如果句子α可以从KB通过推理过程i得到，则记为: $KB \vdash_i \alpha$

Derives(推导出)

- 可靠性（Soundness） 推理过程是可靠的（Sound）

If $KB \vdash_i \alpha$ then it is true that $KB \models \alpha$

不包含错的结果

- 完备性（Completeness） 推理过程是完备的（Complete）

If $KB \models \alpha$ then it is true that $KB \vdash_i \alpha$

能得出所有对的结果

逻辑推理问题（Logical Inference Problem）

- 逻辑推理问题

给定：
- 一个KB
- 一个句子α（称为一个定理（theorem））

问： KB是否语义蕴含α? $KB \models \alpha$?

求解逻辑推理问题

- 如何设计流程来回答

$$KB \models \alpha \ ?$$

- 方法：
  - 真值表法（Truth-table approach）
  - 推理规则法（Inference rules）
    - 转换为SAT问题

真值表法

- 检查所有让KB为真的解释，看它们每个是否都令α为真。
- 真值表（Truth table）：列出所有解释下句子的真值。

		KB		α	
P	Q	$P \vee Q$	$P \Leftrightarrow Q$	$(P \vee \neg Q) \wedge Q$	
True	*True*	*True*	*True*	*True*	✓
True	*False*	*True*	*False*	*False*	
False	*True*	*True*	*False*	*False*	
False	*False*	*False*	*True*	*False*	

真值表法

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	*True*	*True*				
True	*True*	*False*				
True	*False*	*True*				
True	*False*	*False*				
False	*True*	*True*				
False	*True*	*False*				
False	*False*	*True*				
False	*False*	*False*				

真值表法

Example: $KB = (A \vee C) \wedge (B \vee \neg C) \qquad \alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	*True*	*True*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*True*	*True*	*True*
True	*False*	*True*	*True*	*False*	*False*	*True*
True	*False*	*False*	*True*	*True*	*True*	*True*
False	*True*	*True*	*True*	*True*	*True*	*True*
False	*True*	*False*	*False*	*True*	*False*	*True*
False	*False*	*True*	*True*	*False*	*False*	*False*
False	*False*	*False*	*False*	*True*	*False*	*False*

真值表法

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α	
True	*True*	*True*	*True*	*True*	*True*	*True*	✓
True	*True*	*False*	*True*	*True*	*True*	*True*	✓
True	*False*	*True*	*True*	*False*	*False*	*True*	
True	*False*	*False*	*True*	*True*	*True*	*True*	✓
False	*True*	*True*	*True*	*True*	*True*	*True*	✓
False	*True*	*False*	*False*	*True*	*False*	*True*	
False	*False*	*True*	*True*	*False*	*False*	*False*	
False	*False*	*False*	*False*	*True*	*False*	*False*	

真值表法

Example: $KB = (A \vee C) \wedge (B \vee \neg C)$ $\alpha = (A \vee B)$

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α
True	True	True	True	True	True	True
True	True	False	True	True	True	True
True	False	True	True	False	False	True
True	False	False	True	True	True	True
False	True	True	True	True	True	True
False	True	False	False	True	False	True
False	False	True	True	False	False	False
False	False	False	False	True	False	False

$KB \models \alpha$ 成立

真值表法对命题逻辑是可靠且完备的。

真值表法的缺点

真值表法的计算复杂度是多少?

有n个命题符号，表就要有2^n行!

随命题符号数增多指数级增长

真值表法的缺点

真值表法的计算复杂度是多少?

有n个命题符号，表就要有2^n行!
随命题符号数增多指数级增长

观察： 一般只有少数的行KB是True
思考： 能否让过程更高效?
方法： 推理规则法（Inference rules）

- 只考虑KB为True的那些情况
- 用推理规则基于KB中的句子进行 *可靠*推理

求解逻辑推理问题

- 如何设计流程来回答

$$KB \models \alpha \ ?$$

- 方法：
  - 真值表法（Truth-table approach）
  - 推理规则法（Inference rules）
    - 转换为SAT问题

逻辑等价

逻辑等价： 给定命题p和命题q，如果p和q在所有情况下都具有同样真假结果，那么p和q在逻辑上等价，一般用≡来表示，即$p \equiv q$。

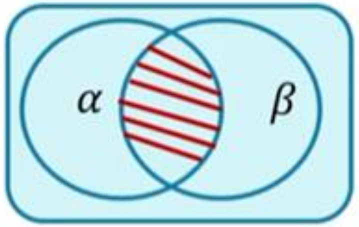
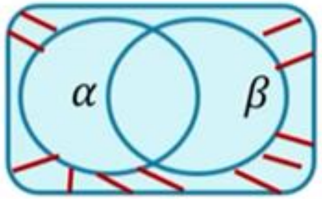
逻辑等价为命题进行形式转换带来了可能，基于这些转换不再需要逐一列出p和q的真值表来判断两者是否在逻辑上等价，而是可直接根据已有逻辑等价公式来判断p和q在逻辑上是否等价。

逻辑等价

逻辑等价的例子

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ ($\wedge$的交互律)	$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)
$\alpha \vee \beta \equiv \beta \vee \alpha$ ($\vee$的交互律)	$(\alpha \Leftrightarrow \beta) \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$(双向消除)
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ ($\wedge$的结合律)	$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)
$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ ($\vee$的结合律)	$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)
$\neg(\neg\alpha) \equiv \alpha$ (双重否定)	$(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ ($\wedge$ 对 $\vee$的分配律)
$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	$(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ ($\vee$ 对 $\wedge$的分配律)

逻辑等价

$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	秋天天气变凉⇒大雁南飞越冬≡大雁没有南飞越冬⇒秋天天气没有变凉
$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)	α为假、则命题恒为真；α为真、则β须为真
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)	
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)	

逻辑等价与双向蕴含

- $X \equiv Y$当且仅当$X \leftrightarrow Y$是重言式

p	q	$p \Leftrightarrow q$
False	False	True
False	True	False
True	False	False
True	True	True

逻辑推理规则

- Modus ponens（肯定前件）

$$\frac{A \Rightarrow B, \quad A}{B}$$

前提（premise）

结论（conclusion）

- 如果以上前提中的两个句子为真，那么结论也为真
- 肯定前件推理规则是**可靠**的：

A	B	$A \Rightarrow B$
False	*False*	*True*
False	*True*	*True*
True	*False*	*False*
True	$\boxed{\textit{True}}$	*True*

逻辑推理规则

- 与消除（And-elimination）

$$\frac{A_1 \wedge A_2 \wedge \ldots \wedge A_n}{A_i}$$

- 与导入（And-introduction）

$$\frac{A_1, A_2, \ldots, A_n}{A_1 \wedge A_2 \wedge \ldots \wedge A_n}$$

- 或导入（Or-introduction）

$$\frac{A_i}{A_1 \vee A_2 \vee \ldots \vee A_i \vee \ldots \vee A_n}$$

逻辑推理规则

- 双重否定消除（Elimination of double negation）

$$\frac{\neg\neg A}{A}$$

- 单项归结（Unit resolution）

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

特殊情况

- 归结（Resolution）

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

以上推理规则都是**可靠**的，可通过真值表证明

逻辑推理规则

肯定前件（Modus ponens)	$\dfrac{A \Rightarrow B, A}{B}$
与消除（And-elimination)	$\dfrac{A_1 \wedge A_2 \wedge \ldots \wedge A_n}{A_i}$
与导入（And-introduction)	$\dfrac{A_1, A_2, \ldots, A_n}{A_1 \wedge A_2 \wedge \ldots \wedge A_n}$
或导入（Or-introduction)	$\dfrac{A_i}{A_1 \vee A_2 \vee \ldots \vee A_i \vee \ldots \vee A_n}$
双重否定消除（Elimination of double negation)	$\dfrac{\neg\neg A}{A}$
单项归结（Unit resolution)	$\dfrac{A \vee B, \neg A}{B}$
归结（Resolution)	$\dfrac{A \vee B, \neg B \vee C}{A \vee C}$

推理规则法例子

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$

推理规则法例子

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$
4. P

根据1和与消除

$$\frac{A_1 \wedge A_2 \wedge \ldots \wedge A_n}{A_i}$$

推理规则法例子

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$
4. P
5. R

根据2、4和与肯定前件

$$\frac{A \Rightarrow B, A}{B}$$

推理规则法例子

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$
4. P
5. R
6. Q

根据1和与消除

$$\frac{A_1 \wedge A_2 \wedge \ldots \wedge A_n}{A_i}$$

推理规则法例子

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$
4. P
5. R
6. Q
7. $(Q \wedge R)$

根据5、6和与导入

$$\frac{A_1, A_2, \ldots, A_n}{A_1 \wedge A_2 \wedge \ldots \wedge A_n}$$

推理规则法例子

KB: $P \wedge Q \quad P \Rightarrow R \quad (Q \wedge R) \Rightarrow S$ **Theorem:** S

1. $P \wedge Q$
2. $P \Rightarrow R$
3. $(Q \wedge R) \Rightarrow S$
4. P
5. R
6. Q
7. $(Q \wedge R)$
8. S

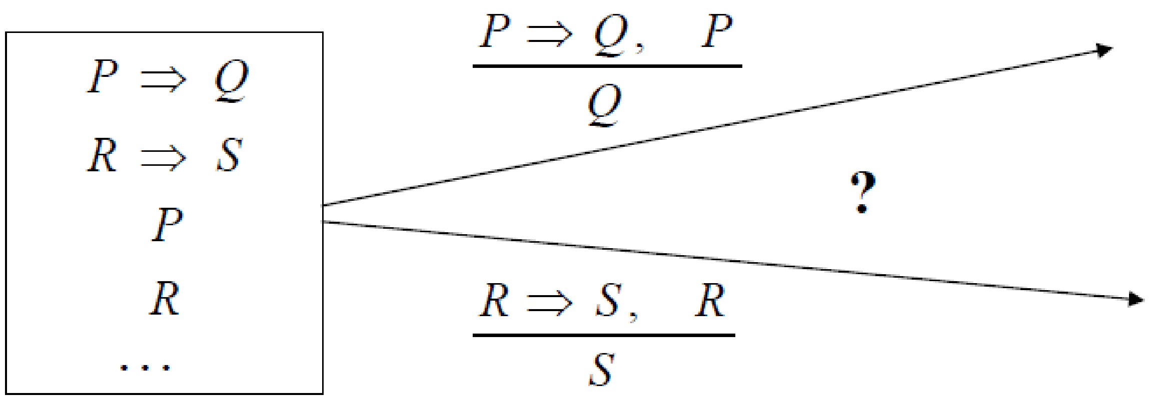
根据3、7和肯定前件

$$\frac{A \Rightarrow B, A}{B}$$

已得到S

逻辑推理与搜索

- 如何实现?
- 为证明α对KB成立，可能需应用若干个可靠的推理规则
- 可以通过搜索来实现



从搜索角度看，真值表法是直接枚举所有可能并检查结果

再看翻译

- 有如下句子：
  - It is not sunny today and it is colder than yesterday. $\neg p \wedge q$
  - We will go swimming only if it is sunny. $r \rightarrow p$
  - If we do not go swimming then we will take a canoe trip. $\neg r \rightarrow s$
  - If we take a canoe trip, then we will be home by sunset. $s \rightarrow t$
- 令：
  - p = It is sunny today
  - q = It is colder than yesterday
  - r = We will go swimming
  - s = We will take a canoe trip
  - t = We will be home by sunset

再看翻译

- 命题逻辑的句子可能包含¬、∧、∨、→、↔这些联结词
- 一个句子中可能包含嵌套的句子

所有的联结词都是必要的吗？

可以限制句子结构的深度吗？

再看翻译

- 命题逻辑的句子可能包含¬、∧、∨、→、↔这些联结词
- 一个句子中可能包含嵌套的句子

所有的联结词都是必要的吗？
不是，可以只用¬、∧、∨将句子重写为等价的句子

可以限制句子结构的深度吗？
可以

如：转化为范式（Normal Form）

范式（Normal Forms）

- 命题逻辑中的句子可以被转换成范式，从而简化推理。

合取范式(Conjunctive Normal Form, CNF)

- 有限个简单析取式构成的合取式

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

析取范式(Disjunctive Normal Form, DNF)

- 有限个简单合取式构成的析取式

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

析取范式与合取范式统称为**范式(normal form)**

范式（Normal Forms）

合取范式(Conjunctive Normal Form, CNF)

- Conjunction of clauses, where a clause is a disjunction of literals

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

析取范式(Disjunctive Normal Form, DNF)

- Disjunction of clauses, where a clause is a conjunction of literals

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

Literal: a propositional symbol or its negation.

范式（Normal Forms）

- 一个合取范式是成立的，当且仅当它的每个简单析取式都是成立的。
- 一个析取范式是不成立的，当且仅当它的每个简单合取式都不成立。

- 任一命题公式都存在着与之等价的析取范式与合取范式（注意:命题公式的析取范式与合取范式都不是唯一的）

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

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

范式（Normal Forms）

- 例子

问题：求$\neg(\alpha \rightarrow \beta) \vee \neg\gamma$的析取范式与合取范式

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ ($\wedge$的交互律)	$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)
$\alpha \vee \beta \equiv \beta \vee \alpha$ ($\vee$的交互律)	$(\alpha \Leftrightarrow \beta) \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$(双向消除)
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ ($\wedge$的结合律)	$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)
$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ ($\vee$的结合律)	$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)
$\neg(\neg\alpha) \equiv \alpha$ (双重否定)	$(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ ($\wedge$ 对 $\vee$的分配律)
$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	$(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ ($\vee$ 对 $\wedge$的分配律)

范式（Normal Forms）

- 例子

问题：求$\neg(\alpha \rightarrow \beta) \vee \neg\gamma$的析取范式与合取范式

$$\neg(\alpha \rightarrow \beta) \vee \neg\gamma$$

$\Rightarrow$ $\neg(\neg\alpha \vee \beta) \vee \neg\gamma$

$\Rightarrow$ $(\alpha \wedge \neg\beta) \vee \neg\gamma$ (析取范式)

$\Rightarrow$ $(\alpha \vee \neg\gamma) \wedge (\neg\beta \vee \neg\gamma)$ (合取范式)

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ ($\wedge$的交互律)	$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)
$\alpha \vee \beta \equiv \beta \vee \alpha$ ($\vee$的交互律)	$(\alpha \Leftrightarrow \beta) \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$(双向消除)
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ ($\wedge$的结合律)	$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)
$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ ($\vee$的结合律)	$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)
$\neg(\neg\alpha) \equiv \alpha$ (双重否定)	$(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ ($\wedge$ 对 $\vee$的分配律)
$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	$(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ ($\vee$ 对 $\wedge$的分配律)

转化成合取范式CNF

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

1. 消除 $\Leftarrow, \Rightarrow$

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

2. 用DeMorgan's Law和双重否定消除

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

3. 用分配和结合律转化成CNF

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

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

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ (∧的交互律)	$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)
$\alpha \vee \beta \equiv \beta \vee \alpha$ (∨的交互律)	$(\alpha \Leftrightarrow \beta) \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$(双向消除)
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ (∧的结合律)	$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)
$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ (∨的结合律)	$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)
$\neg(\neg\alpha) \equiv \alpha$ (双重否定)	$(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ (∧对 ∨的分配律)
$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	$(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ (∨对 ∧的分配律)

考虑可满足性问题

既Satisfiability (SAT) problem，SAT问题

- 判断一个句子是否是可满足的（存在一个解释使其为真）

- 判断一个合取范式(CNF)的句子是否是可满足的（CNF-SAT）

$$(P \vee Q \vee \neg R) \wedge (\neg P \vee \neg R \vee S) \wedge (\neg P \vee Q \vee \neg T)$$

求解SAT问题

- 这是一个NP-complete问题

- 深度优先搜索

基本思路：

1. 选一个变元，并选一个真值（T或F）赋给它
2. 继续选未被赋值的变元，直到所有变元都被赋值，或已有的部分赋值已令句子为假。最后句子为真则可返回，为假则回溯。

- 局部搜索

基本思路：

1. 对所有变元随机赋值
2. 每次改变其中一个变元的真值，直到令句子为真或达到最大尝试次数

> 每次改变变元真值的尝试可更倾向于选择使更多子句（clause）为真的

参考GSAT、WalkSAT算法

推理问题和可满足性问题

- 逻辑推理问题
  - KB是否蕴含α
  - 所有令KB为真的解释是否也令α为真

- 可满足性（SAT）问题
  - 是否存在一个解释使句子为真

两个问题间能否建立联系?

关系：

$KB \models \alpha$ 当且仅当
$(KB \wedge \neg\alpha)$ 是不可满足的

- 推理问题也是NP-complete
- 用于求解SAT问题的方法也可用于求解推理问题

利用合取范式（CNF）和SAT推理

- 推理规则法能否从CNF获益?
- 考虑只用一条规则为写成CNF形式的KB进行推理

归结（Resolution）规则似乎跟CNF很配

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

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

前提和结论都是简单析取式

归结规则（Resolution Rule）

- 可为表示为CNF形式的KB进行**可靠**推理

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

A	B	C	$A \vee B$	$\neg B \vee C$	$A \vee C$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
False	*True*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*	*True*
True	*False*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*False*	*True*
True	*True*	*True*	*True*	*True*	*True*

使用归结规则推理

- 归结规则是可靠（sound）的
- 但它是完备（complete）的吗?

- 对一个KB直接重复应用归结规则可能无法得到某些*有效*的句子

例： 有 $(A \wedge B)$，希望得到 $(A \vee B)$

用归结规则无法得到

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

使用归结规则推理

- 归结规则是可靠（sound）的
- 但它是完备（complete）的吗?

- 但是，使用归结规则可以正确地判断一个CNF句子是否可满足
  - 不断使用归结规则，直到得出矛盾（不可满足）或无法再继续得到新句子（可满足）

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

使用归结规则推理

- 转化为SAT问题

证明 $(KB \wedge \neg\alpha)$ 不可满足
则证明 $KB \models \alpha$

$KB \models \alpha$ 当且仅当
$(KB \wedge \neg\alpha)$ 是不可满足的

对于CNF形式的任何KB和α，可采用归结规则判断

$$(KB \wedge \neg\alpha)$$

的可满足性

因此归结规则是完备的

归结算法

算法流程:

- 将 KB 和 $\neg\alpha$ 转为 CNF 形式
- 从 $(KB \wedge \neg\alpha)$ 开始，迭代使用归结规则
- 满足以下条件时停止:
  - 得到矛盾（证明 $KB \models \alpha$ 成立）
  - 无法再得到新的句子（证明不成立）

归结算法例子

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

归结算法例子

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

第一步. 将KB转化为CNF

$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)

$$P \Rightarrow Q \qquad\longrightarrow\qquad (\neg P \vee R)$$
$$(P \wedge Q) \Rightarrow S \qquad\longrightarrow\qquad (\neg Q \vee \neg R \vee S)$$

$$\mathbf{KB}: \quad P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S)$$

第二步. 对定理（theorem）取非

$$S \longrightarrow \neg S$$

第三步. 对得到的子句（clause）使用归结

$$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$$

归结算法例子

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$

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

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

归结算法例子

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

$$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$$

$$R$$

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

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

归结算法例子

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

$$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$$

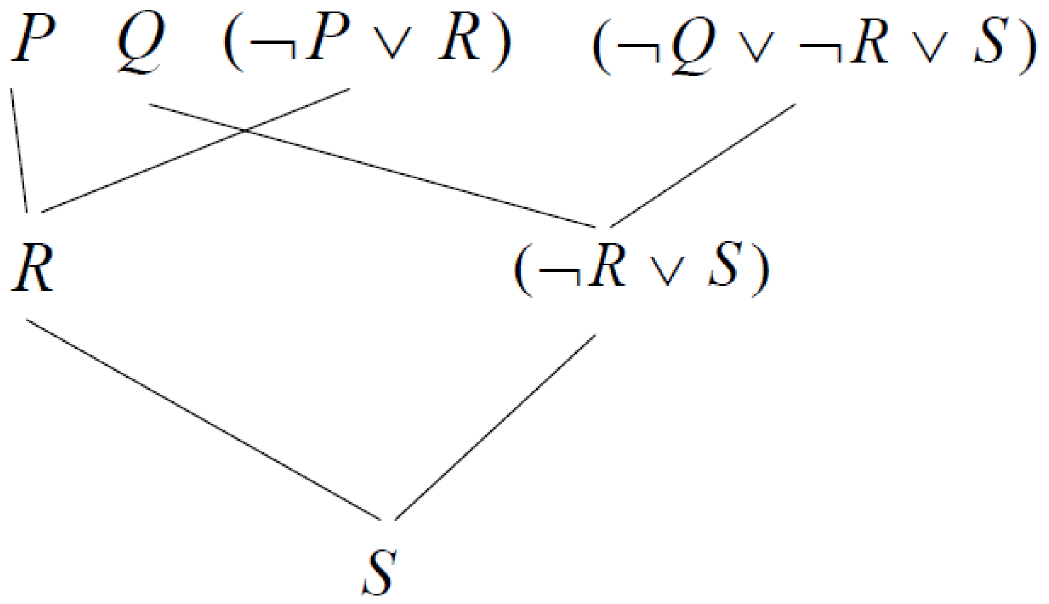
$$R \qquad\qquad (\neg R \vee S)$$

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

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

归结算法例子

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

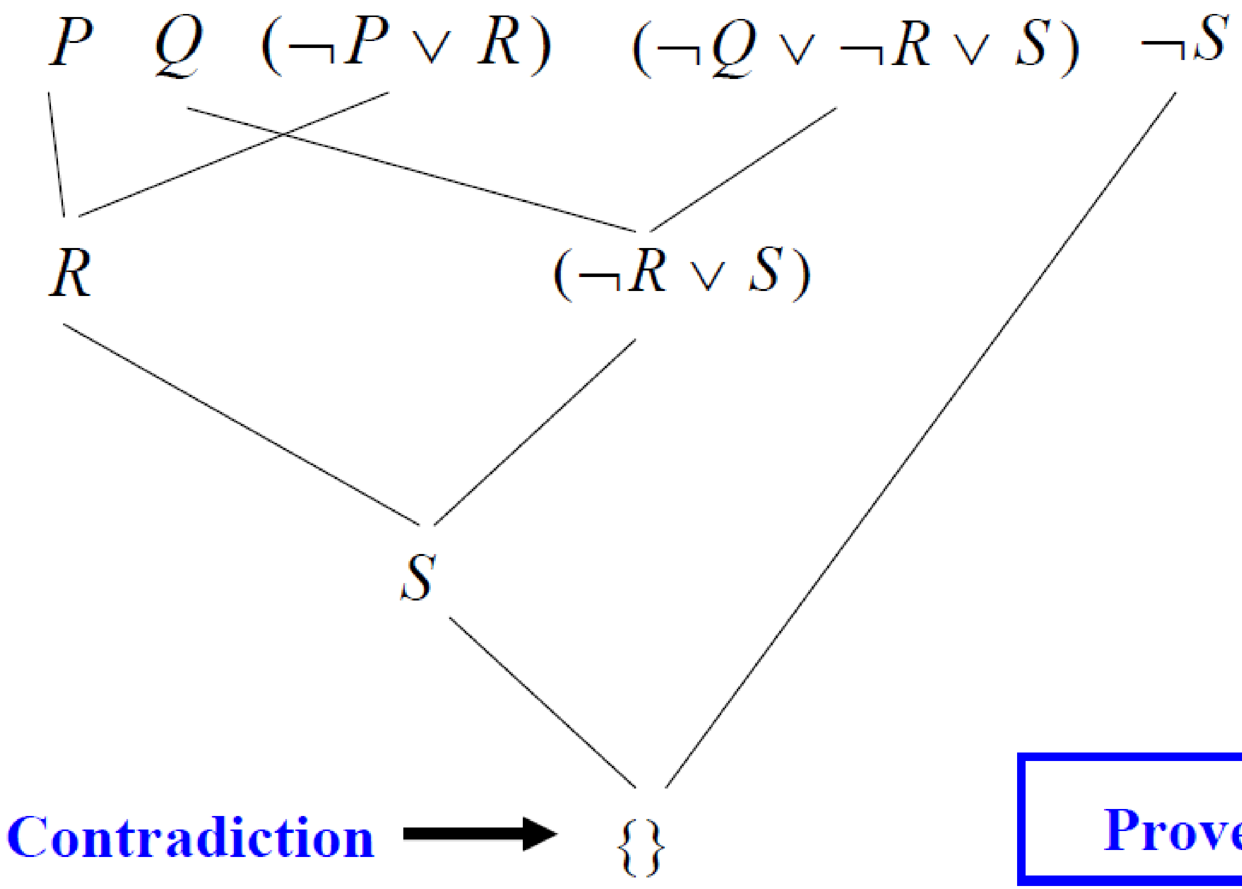


$$\boxed{\frac{A \vee B, \neg A}{B}}$$

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

归结算法例子

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S



$\boxed{\textbf{Proved: } S}$

其他例子

其他例子

1	$\alpha \vee \beta$
2	$\alpha \Rightarrow \gamma$
3	$\beta \Rightarrow \gamma$

已知如上命题成立，
请证明命题γ是成立的

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

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

其他例子

1	$\alpha \vee \beta$
2	$\alpha \Rightarrow \gamma$
3	$\beta \Rightarrow \gamma$

已知如上命题成立，
请证明命题γ是成立的

1	$\alpha \vee \beta$	已知
2	$\neg\alpha \vee \gamma$	2进行蕴涵消除
3	$\neg\beta \vee \gamma$	3进行蕴涵消除
4	$\neg\gamma$	假设命题γ不成立
5	$\beta \vee \gamma$	1和2进行归结
6	$\neg\alpha$	2和4进行归结
7	$\neg\beta$	3和4进行归结
8	γ	5和7进行归结
9	假设不成立，命题γ成立	

其他例子

1	$\alpha \vee \gamma$
2	$\neg\beta \vee \gamma$
3	$\neg\gamma \vee \alpha$
4	$\neg\alpha \vee \beta$
5	$\neg\alpha \vee \neg\gamma$

证明如上命题集是不可满足的

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

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

Horn clauses (Horn子句)

Horn clauses (Horn子句)

- Horn clause: a disjunction of literals in which, at most one literal is not negated
- Horn Form: a CNF whose clauses are all Horn

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

- 注意：命题逻辑中的句子并不都能被转换为Horn form

- Horn form的KB可有三种类型的句子
  - **Rules** $(\neg B_1 \vee \neg B_2 \vee \ldots \neg B_k \vee A)$

$$(\neg(B_1 \wedge B_2 \wedge \ldots B_k) \vee A) \qquad (B_1 \wedge B_2 \wedge \ldots B_k \Rightarrow A)$$

  - **Facts** B
  - **Integrity constraints** $(\neg B_1 \vee \neg B_2 \vee \ldots \neg B_k)$

Horn clauses (Horn子句)

- Horn clause: a disjunction of literals in which, at most one literal is not negated
- Horn Form: a CNF whose clauses are all Horn

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

如果KB表示为Horn form，那么以下两种规则对推理**命题符号**是*可靠*且*完备*的：

- 归结（Resolution）
- 肯定前件（Modus ponens）

Horn clauses (Horn子句)

- 肯定前件（Modus ponens）

$$\frac{B \Rightarrow A, \quad B}{A}$$

更一般的版本：$$\frac{(B_1 \wedge B_2 \wedge \ldots B_k \Rightarrow A)\,, B_1, B_2, \ldots B_k}{A}$$

如果KB表示为Horn form，那么肯定前件对推理**命题符号**是*可靠*且*完备*的

只考虑定理$\boldsymbol{\alpha}$**为命题符号**的逻辑推理问题

Horn clauses (Horn子句)

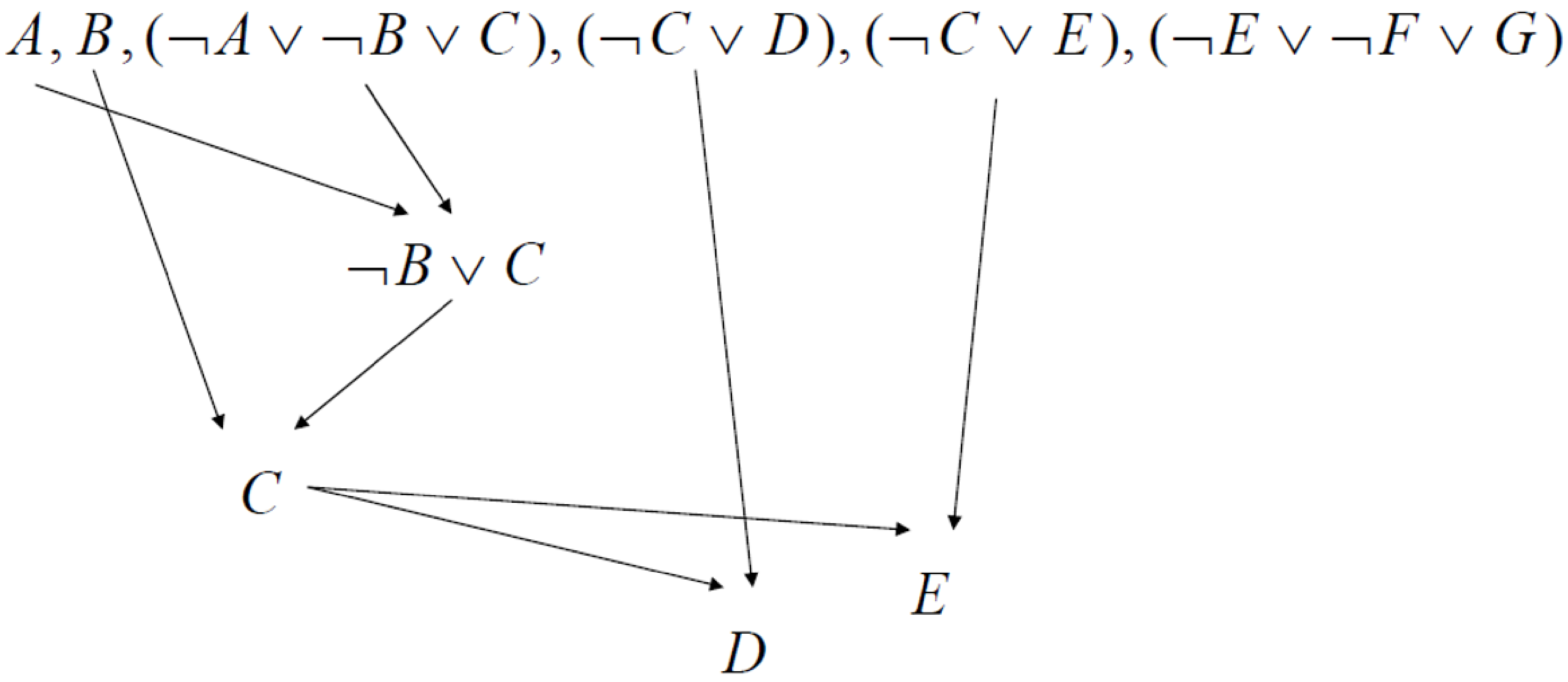
- 归结（Resolution）

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

如果KB表示为Horn form，那么归结规则对推理**命题符号**是*可靠*且*完备*的

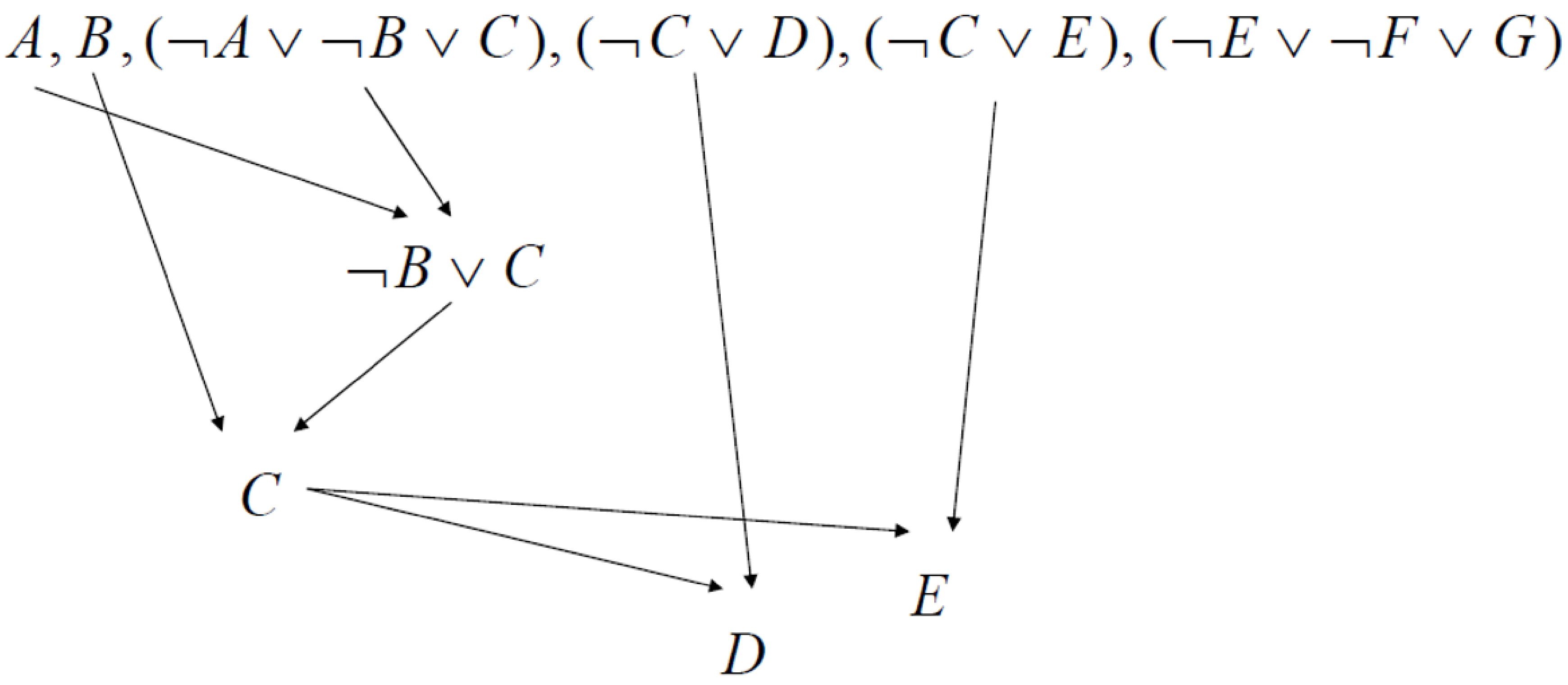
Horn clauses (Horn子句)

- 如何推理?
- 使用归结（resolution）：



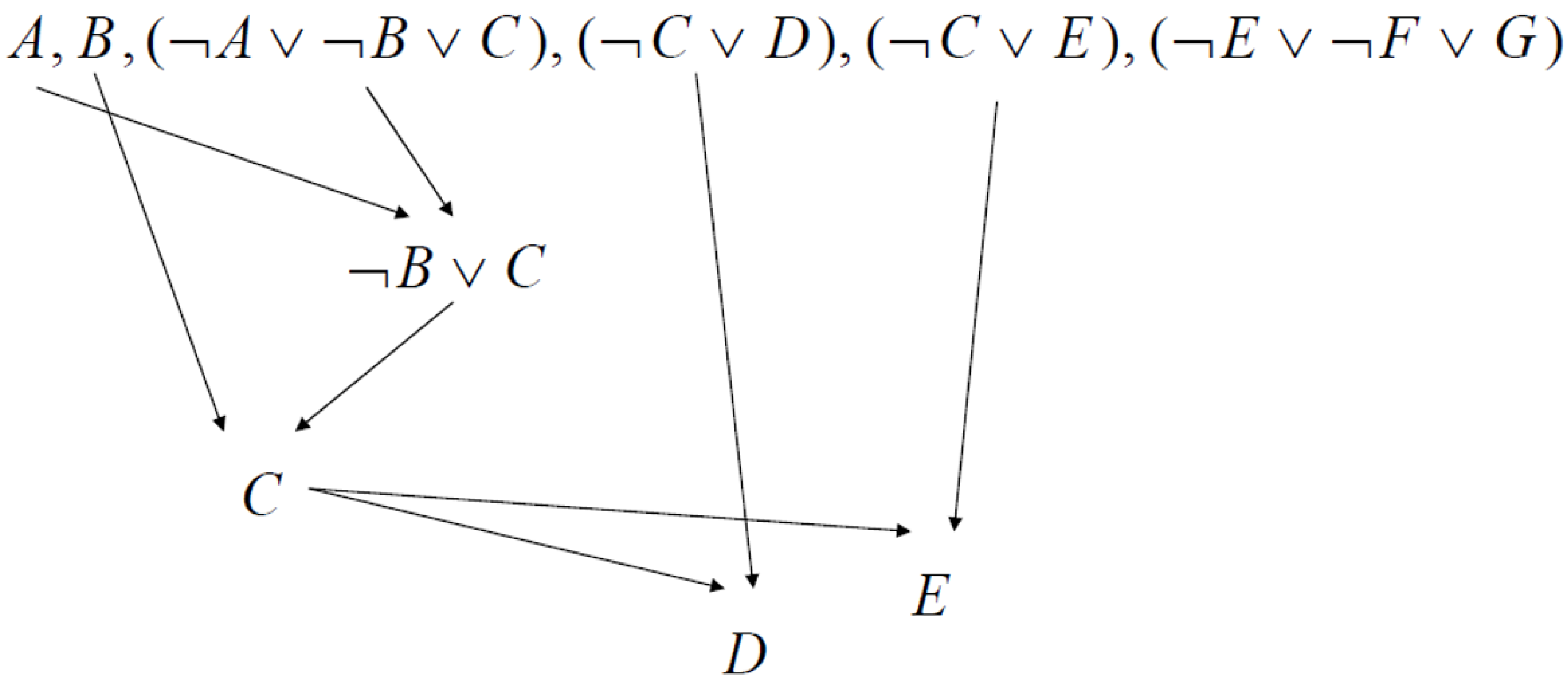
Horn clauses (Horn子句)

- 使用归结（resolution）的特点
  - Every resolution is a positive unit resolution, i.e., a resolution in which **one clause is a positive propositional symbol**



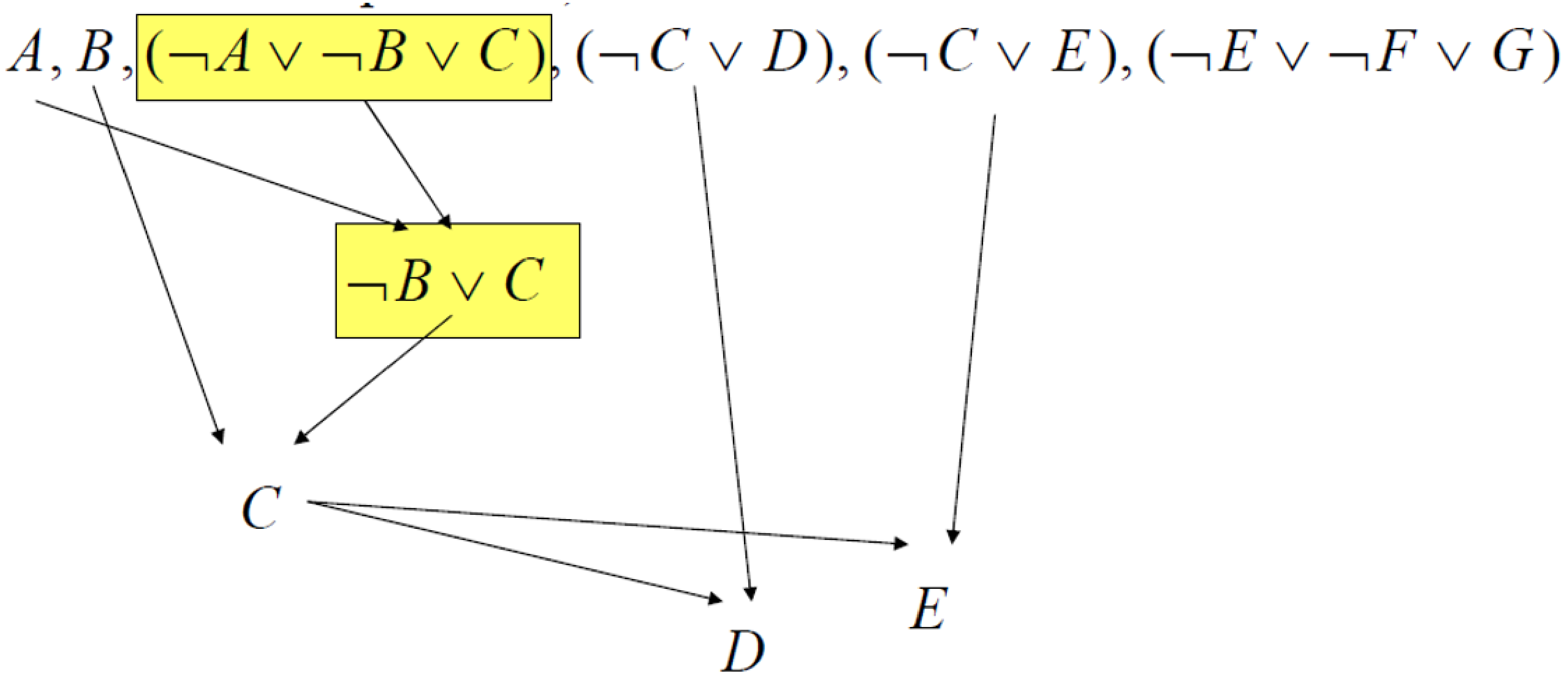
Horn clauses (Horn子句)

- 使用归结（resolution）的特点
  - 每次归结，不是正命题符号（positive propositional symbol）的那项可删除。



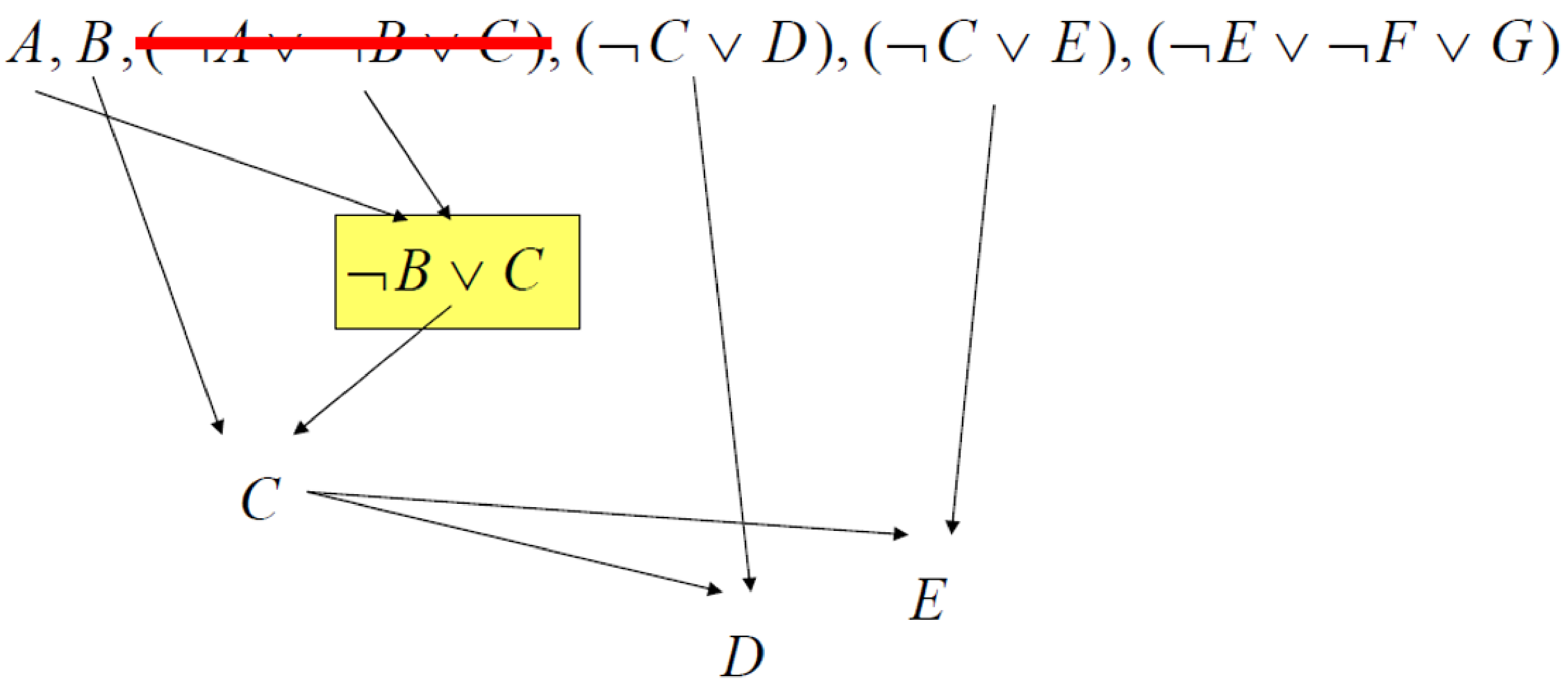
Horn clauses (Horn子句)

- 使用归结（resolution）的特点
  - 每次归结，不是正命题符号（positive propositional symbol）的那项可删除。



Horn clauses (Horn子句)

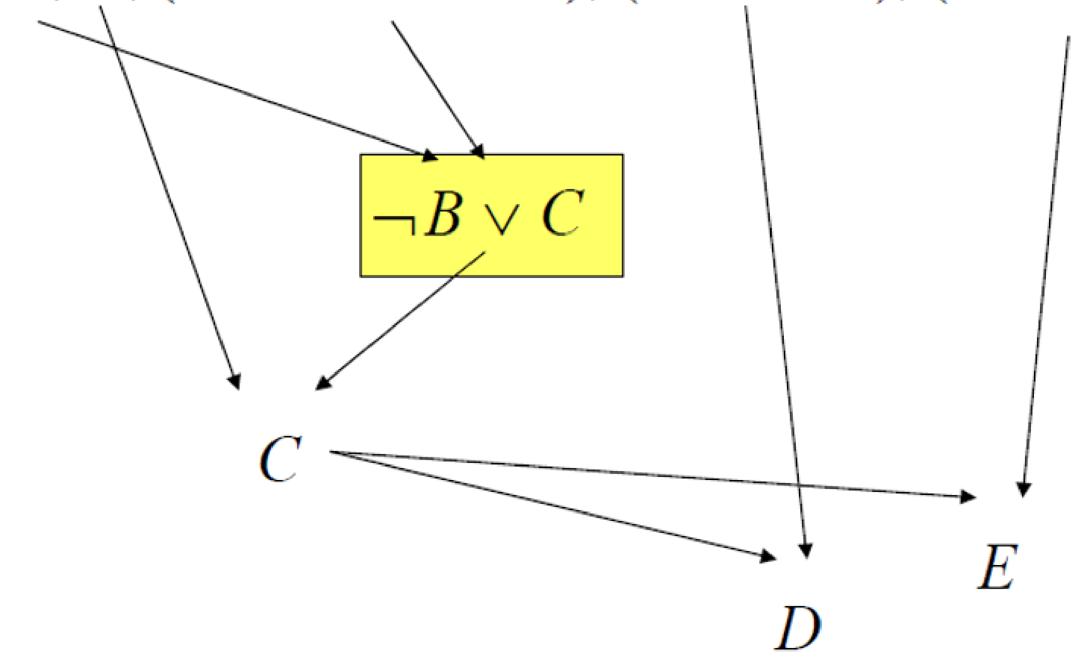
- 使用归结（resolution）的特点
  - 每次归结，不是正命题符号（positive propositional symbol）的那项可删除。



Horn clauses (Horn子句)

- 使用归结（resolution）的特点
  - 因而最多需执行归结的次数不大于N（KB的大小）。
  - KB大小：KB中每个句子的literal数之和

$$A, B, \cancel{(\neg A \vee \neg B \vee C)}, (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$$



复杂度可线性于KB大小

Horn clauses (Horn子句)

- 使用肯定前件（Modus ponens）的两种方法

- Forward chaining
  - 思路：当一条规则的前提满足时，就推理出它的结论。迭代直到无法推出新的。

- Backward chaining
  - 思路：直接考虑一条规则的结论，通过递归地证明其前提来证明它。

两种方法都是完备的

Forward chaining

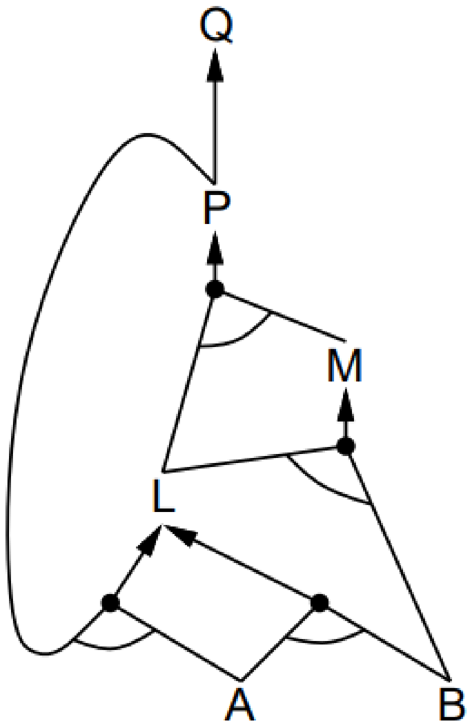
- 算法:

```
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
```

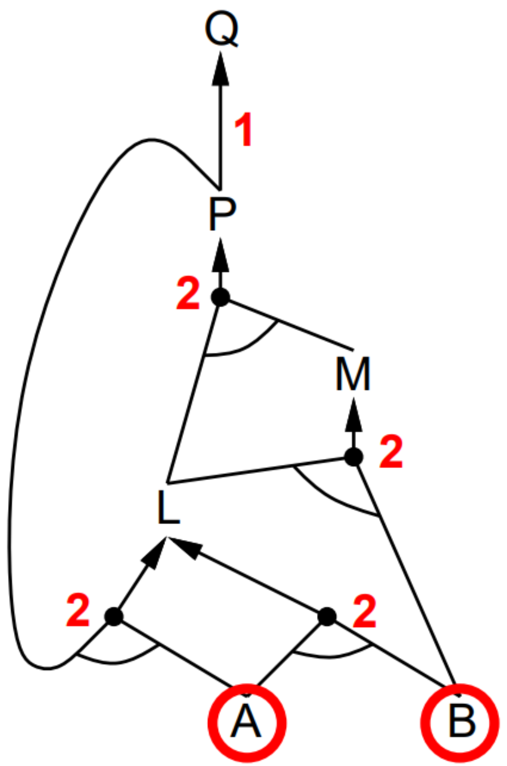
Forward chaining

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



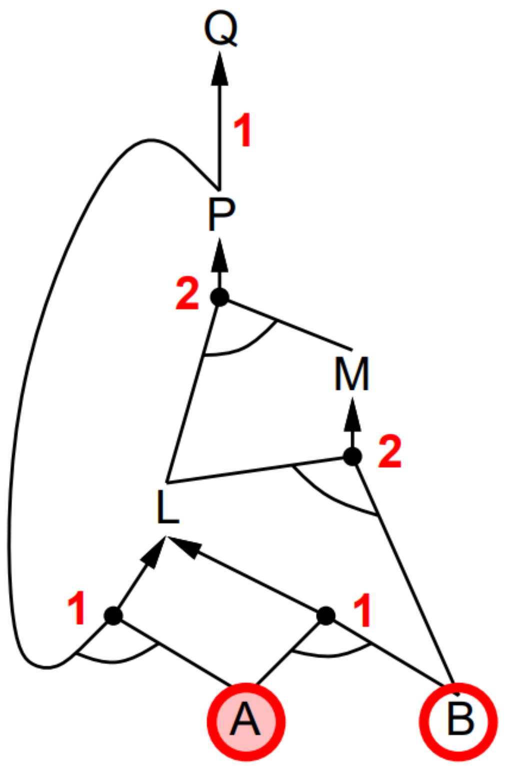
Forward chaining

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



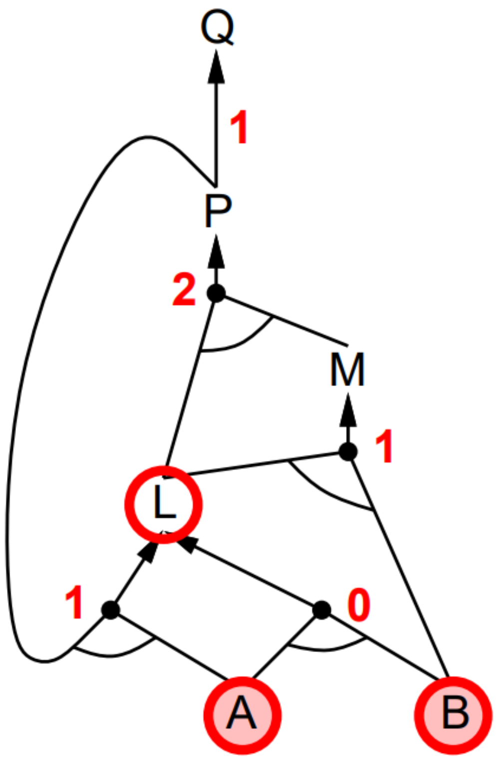
Forward chaining

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



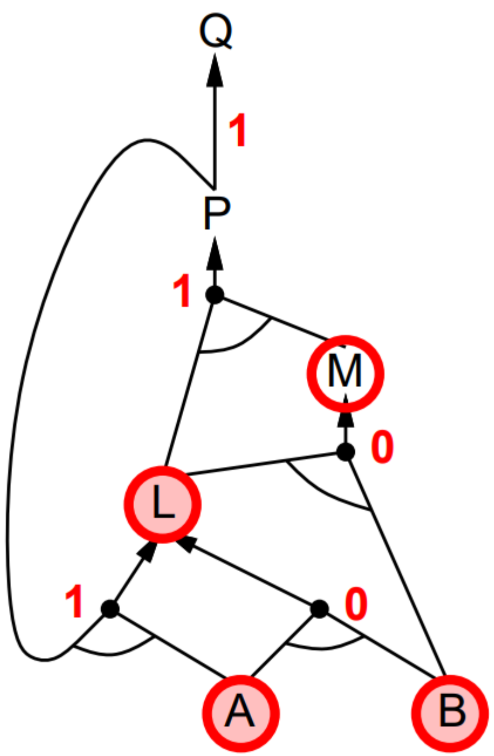
Forward chaining

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



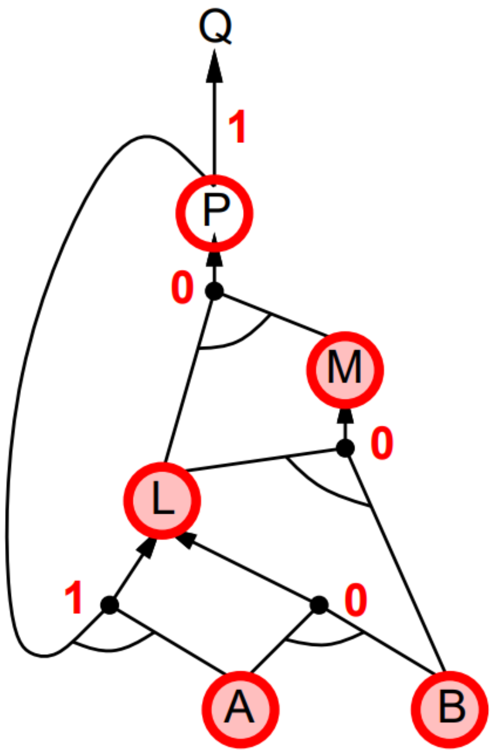
Forward chaining

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



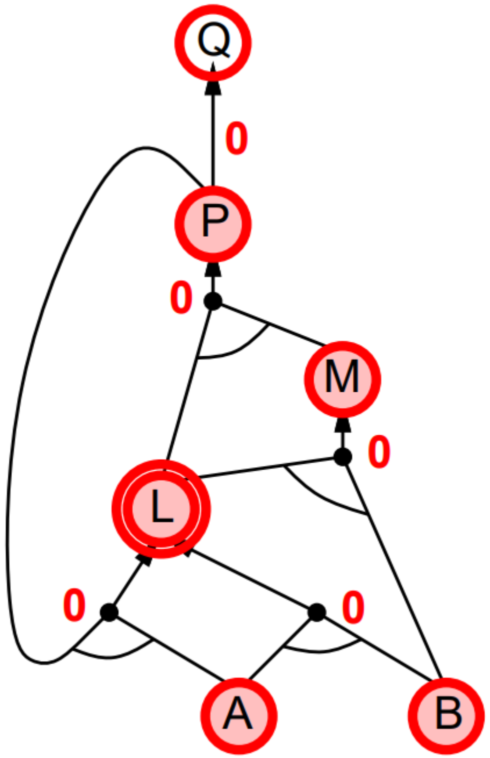
Forward chaining

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



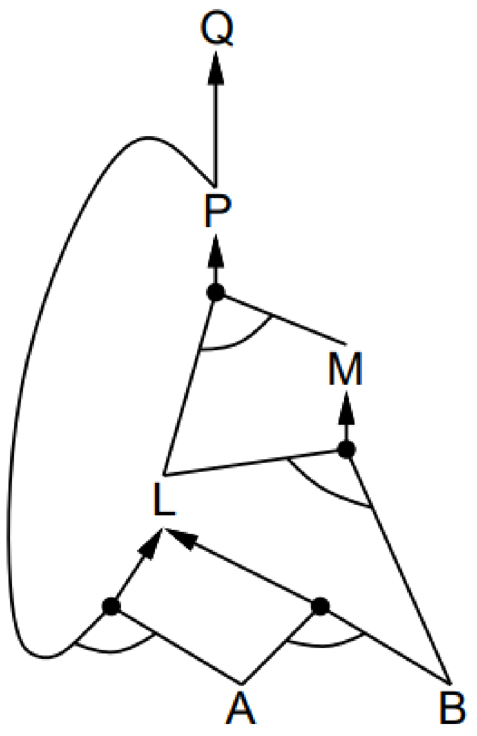
Forward chaining

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



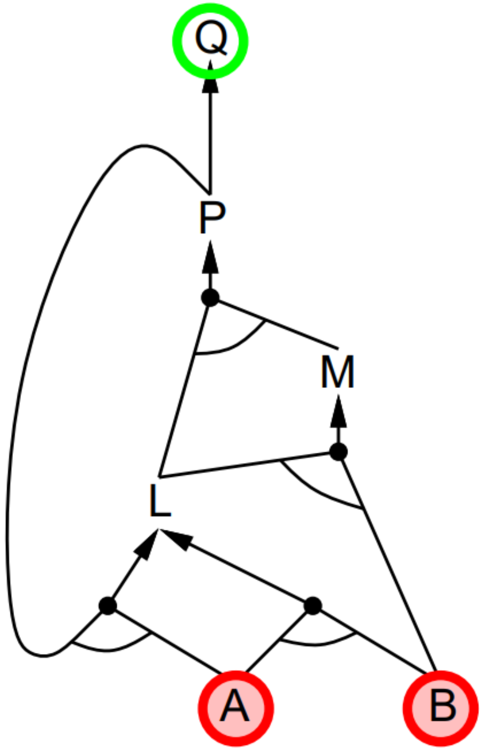
Backward chaining

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



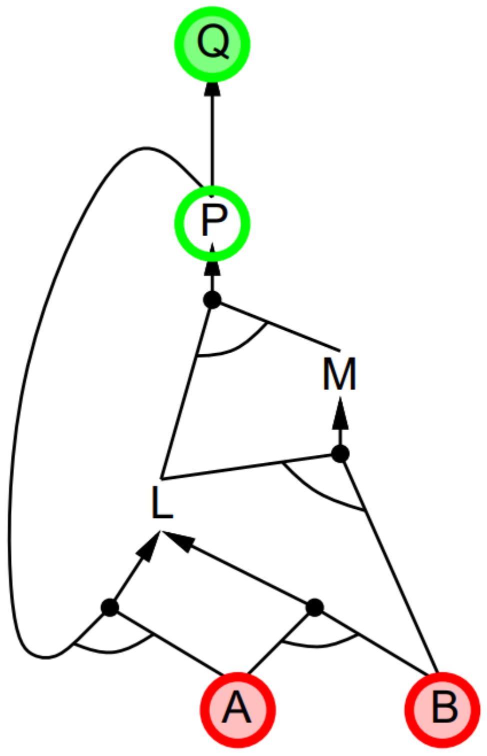
Backward chaining

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



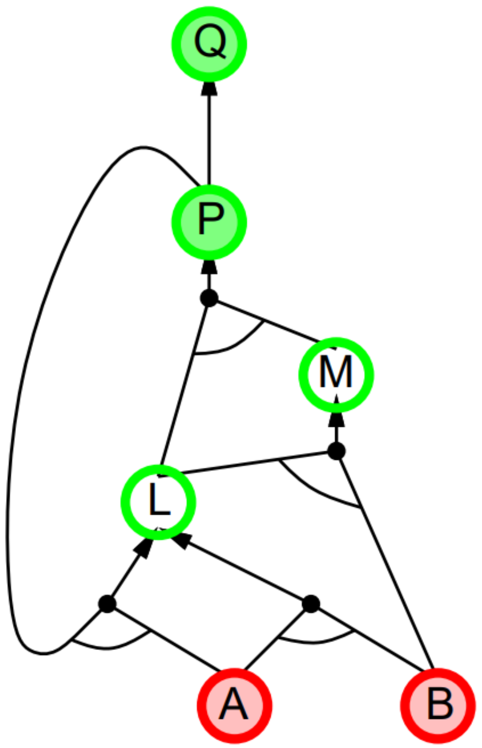
Backward chaining

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



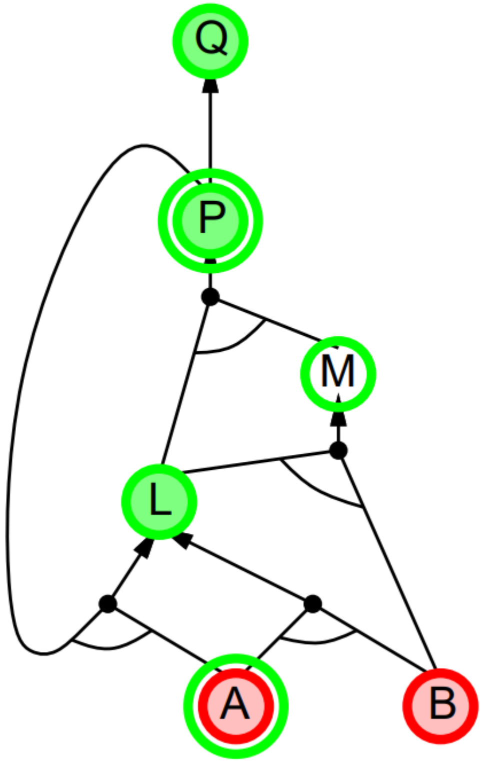
Backward chaining

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



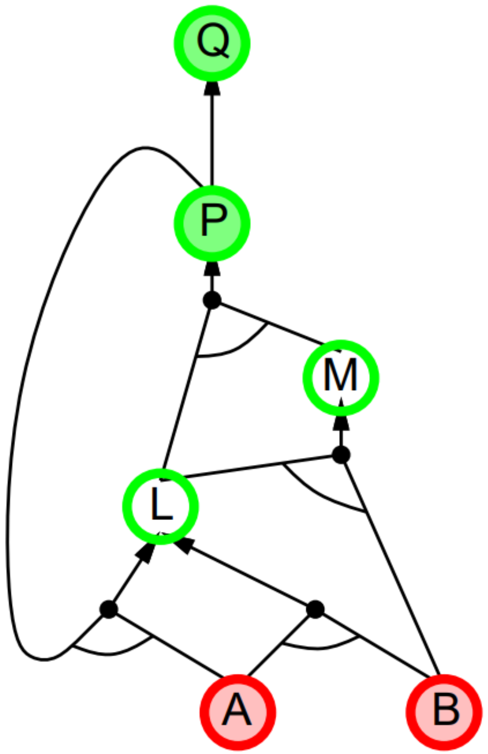
Backward chaining

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



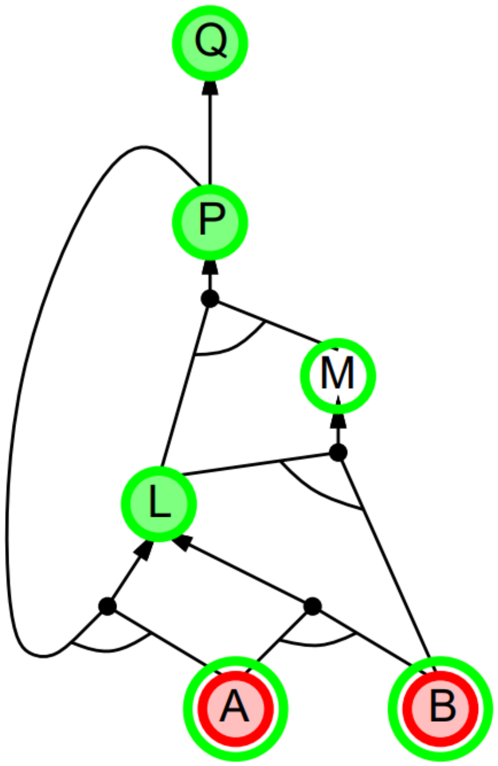
Backward chaining

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



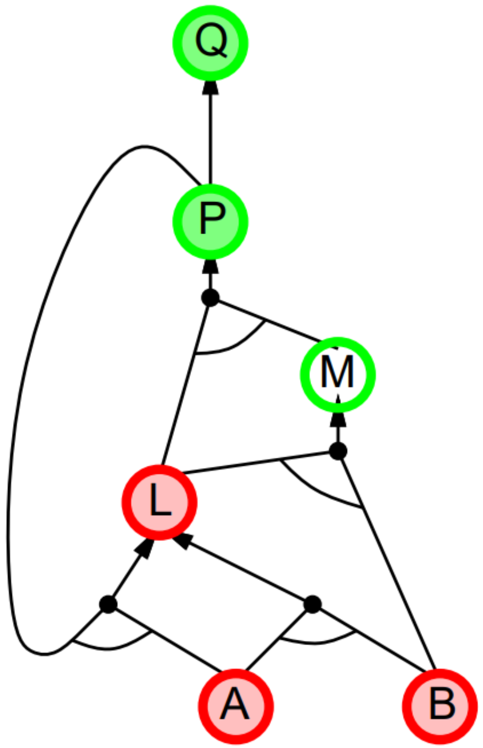
Backward chaining

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



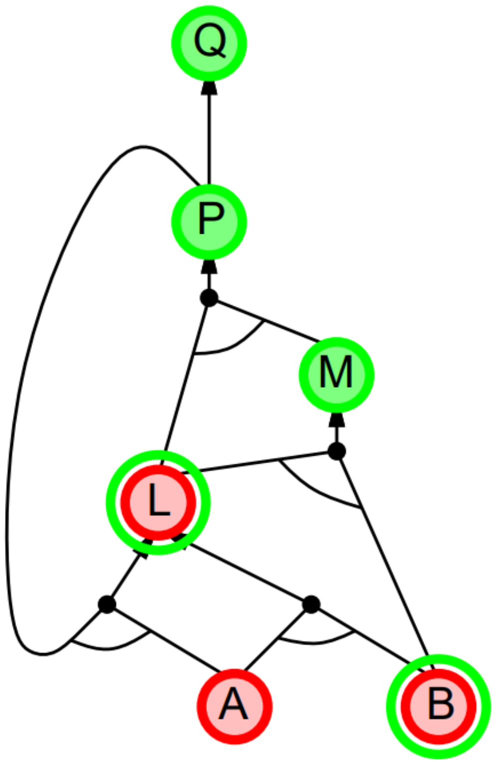
Backward chaining

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



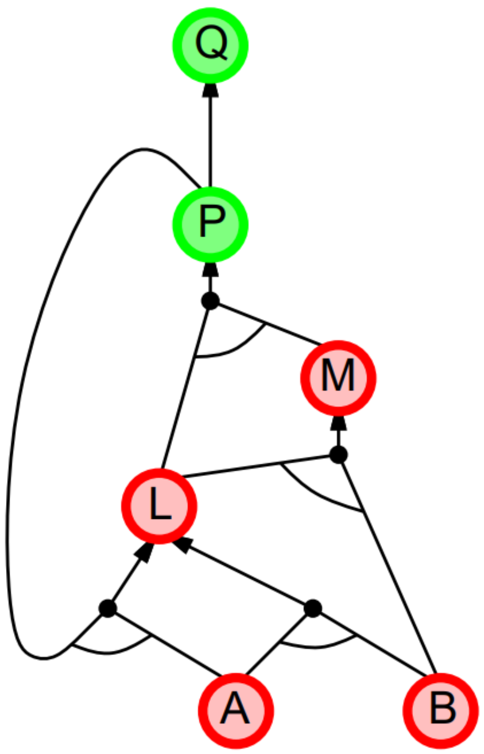
Backward chaining

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



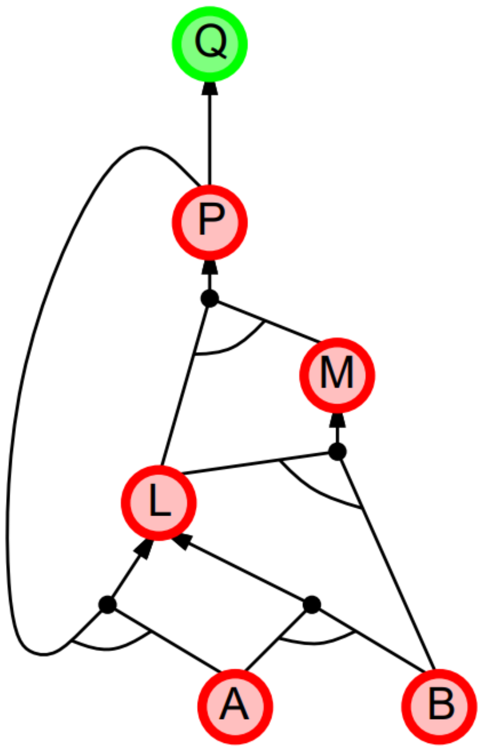
Backward chaining

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



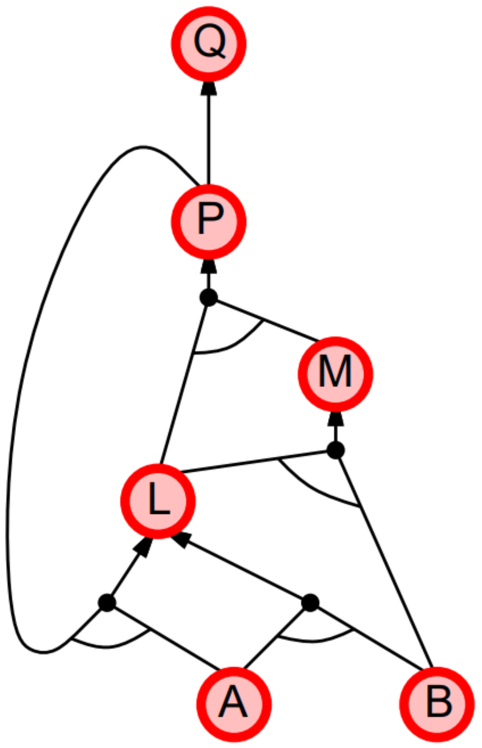
Backward chaining

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



Backward chaining

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



回顾

- 命题逻辑符号：命题符号、真值符号、联结词（¬、∧、∨、→、↔）
- 矛盾式和重言式
- 解释（interpretation）、模型、有效性和可满足性
- 知识库（KB）、语义蕴含（Entailment）、逻辑推理问题
- 可靠和完备的推理（Inference）
- 真值表法、推理规则法
- 逻辑等价、逻辑推理规则（肯定前件、与消除、…）
- 合取范式（CNF）、析取范式
- 可满足性（SAT）问题、$KB \models \alpha$当且仅当$KB \wedge \neg\alpha$不可满足
- Horn Clause

问题

- All men are mortal.
- Socrates is a man.
- Socrates is mortal.

END