INTRODUCTION TO MATHEMATICAL LOGIC AND LOGIC PROGRAMMING

1. SYNTAX AND SEMANTIC SYSTEM

Def. <u>Syntax system</u> is

$$[\mathcal{F}, \mathcal{A}, \mathcal{D}]$$

where

$\mathcal{F} \neq \emptyset$ is a set of **formulas**,

$\mathcal{A} \subset \mathcal{F}$ is so-called set of **axioms**,

$\mathcal{D}$ is a set of **deduction rules**

$$d(F_1, \dots, F_n) = G$$

($F_1, \dots, F_n, G$ are formulas)

Note. We will write

$$\frac{F_1, \dots, F_n}{G}$$

instead of

$$d(F_1, \dots, F_n) = G$$

G **follows immediately** from the deduction rule d and the formulas $F_1, \dots, F_n$

Def. $G \in \mathcal{F}$ is a theorem (provable formula, syntactically valid formula) from premises $\mathcal{B} \subset \mathcal{F}$, if there exists a sequence $F_1, \dots, F_n$ (so-called proof) such that

(a) $F_i \in \mathcal{A}$ or $F_i \in \mathcal{B}$ or F_i follows immediately from a deduction rule $d \in \mathcal{D}$ and some formulas F_j, $j < i$

(b) $G \equiv F_n$

We sign: $\mathcal{B} \vdash G$

Note. If $\mathcal{B} = \emptyset$ we write $\vdash G$ (theorem)

Def. $\mathcal{B} \subset \mathcal{F}$ is called inconsistent if $\mathcal{B} \vdash G$ for any formula $G \in \mathcal{F}$.

Def. Semantic system is

$$[\mathcal{F}, \mathcal{T}]$$

where $\mathcal{T}$ is a set of interpretations t (truth-value functions over formulas, i.e. $t(F)$ is either 1 or 0)

Def. $\mathcal{G}$ is semantically valid from $\mathcal{B} \subset \mathcal{F}$ ($\mathcal{B} \models \mathcal{G}$) if

$$(\forall t \in \mathcal{T}) \quad t(\mathcal{B}) \text{ is } 1 \Rightarrow t(\mathcal{G}) = 1$$

(Any model of $\mathcal{B}$ is a model of $\mathcal{G}$ as well).

Note. If $\mathcal{B} = \emptyset$ then

$$(\forall t \in \mathcal{T}) \quad t(\mathcal{G}) = 1$$

$\models \mathcal{G}$ $\qquad$ $\mathcal{G}$ is semantically valid (tautology)

Def. Combined system is

$$[\mathcal{F}, \mathcal{A}, \mathcal{D}, \mathcal{T}]$$

Def. A combined system is complete if

$$\mathcal{B} \vdash \mathcal{G} \quad \text{iff} \quad \mathcal{B} \models \mathcal{G}$$

2. PROPOSITIONAL CALCULUS

Formulas of ProC :

(1) each propositional variable is a formula of ProC;

(2) if F, G are formulas of ProC, then

$F \Rightarrow G$, $\neg F$

are formulas of ProC

($\Rightarrow$ propositional functor of implication,

$\neg$ propositional functor of negation)

Set of axioms for ProC :

(Let F, G, H be formulas of ProC)

A1 $F \Rightarrow (G \Rightarrow F)$

A2 $[F \Rightarrow (G \Rightarrow H)] \Rightarrow [(F \Rightarrow G) \Rightarrow (F \Rightarrow H)]$

A3 $(\neg G \Rightarrow \neg F) \Rightarrow (F \Rightarrow G)$

Deduction rules :

MP (modus ponens) $\dfrac{F,\ F \Rightarrow G}{G}$

<u>Example:</u> Prove $\vdash F \Rightarrow F$

$\vdash F \Rightarrow ((\underbrace{F \Rightarrow F}) \Rightarrow F)$ A_1

$\vdash [F \Rightarrow ((\underbrace{F \Rightarrow F}) \Rightarrow F] \Rightarrow [(F \Rightarrow (\underbrace{F \Rightarrow F})) \Rightarrow (F \Rightarrow F)]$ A_2

$\vdash (F \Rightarrow (F \Rightarrow F)) \Rightarrow (F \Rightarrow F)$ MP

$\vdash F \Rightarrow (\underbrace{F} \Rightarrow F)$ A_1

$\vdash F \Rightarrow F$ MP

Semantic system of ProC:

Boolean algebra

$t(F)$	$t(\neg F)$
0	1
1	0

$t(\neg F) = \overline{t(F)}$

Boolean negation

$t(F)$	$t(G)$	$t(F \Rightarrow G)$
0	0	1
0	1	1
1	0	0
1	1	1

$t(F \Rightarrow G) = t(F) \rightarrow t(G)$

Boolean implication

Th. Combined system of ProC is <u>complete</u>.

Example. Prove $\models A_1$

F	G	$G \Rightarrow F$	$F \Rightarrow (G \Rightarrow F)$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

← we should rather $t(\ldots)$ but . . .

We have got 1 for all ~~instances~~, interpretations therefore ~~$t(A_1) = 1$~~ $\models A_1$

Example. Prove $\models F \Rightarrow F$

F	$F \Rightarrow F$
0	1
1	1

~~$t(F \Rightarrow F) = 1$~~ $(\forall t)$ $t(F \Rightarrow F) = 1$

Further propositional functors:

disjunction $F \vee G = \neg F \Rightarrow G$

conjunction $F \,\&\, G = \neg(F \Rightarrow \neg G)$

equivalence $F \Leftrightarrow G = (F \Rightarrow G) \,\&\, (G \Rightarrow F)$

3. PREDICATE CALCULUS

Language $\mathcal{L}$ of predicates P $(= \quad <)$
functors f $(+ \quad *)$
constants c $(0 \quad 1)$
variables x

Terms of the language $\mathcal{L}$:

(1) variables and constants are terms,
(2) if $\tau_1, \dots, \tau_n$ are terms and f is n-ary functor, then $f(\tau_1, \dots, \tau_n)$ is a term.

Formulas of the language $\mathcal{L}$:

(1) Each **atomic formula** $P(\tau_1, \dots, \tau_n)$, where $\tau_1, \dots, \tau_n$ are terms and P is n-ary predicate of $\mathcal{L}$, is a formula;
(2) If F, G are formulas, then $\neg F$, $F \Rightarrow G$ are formulas;
(3) If F is a formula, x is a variable, then $(\forall x) F$ is a formula.

($\forall$ general quantifier)

Def. **Literal** is either atomic formula or $\neg$ atomic formula

Example. Formula

$$(\forall x)\ P(x, y, z)$$

x is **bound variable**

y, z are **free variables**

Def. A formula is **closed** if it has no free variables.

Set of axioms for Pre C :

A1 $F \Rightarrow (G \Rightarrow F)$

A2 $[F \Rightarrow (G \Rightarrow H)] \Rightarrow [(F \Rightarrow G) \Rightarrow (F \Rightarrow H)]$

A3 $(\neg G \Rightarrow \neg F) \Rightarrow (F \Rightarrow G)$

A4 $(\forall x)\ (G \Rightarrow F) \Rightarrow [G \Rightarrow (\forall x)\ F]$ if x is not a free variable of G

A5 $(\forall x)\ F \Rightarrow Sb^{x}_{\tau_0}(F)$ if there is no ambiguity

substitute term τ_0 in formula F for all free occurances of the variable x

Deduction rules of Pre C:

MP $\dfrac{F,\quad F \Rightarrow G}{G}$

Gen_x $\dfrac{F}{(\forall x)\ F}$

Note.   & $\vee$ $\Leftrightarrow$

Existentional quantifier $\exists$

$$(\exists x)\, F = \neg (\forall x) \neg F$$

4. RESOLUTION PRINCIPLE

4.1. REFUTATION PRINCIPLE

$\mathcal{B}$ set of closed formulas
G a closed formula
? How to prove $\mathcal{B} \vdash G$

> Th. (Refutation principle). Let $\mathcal{B}$ be a set of closed formulas, G a closed formula.
>
> $\mathcal{B} \vdash G$ iff $\mathcal{B} \cup \{\neg G\}$ is inconsistent

4.2. CLAUSES

Closed formula F

$\downarrow$

Prenex conjunction normal form $(Q_1x_1)(Q_2x_2)\dots(Q_nx_n)\ M$

(1) Eliminate $\Rightarrow$ $\Leftrightarrow$
(2) Bring $\neg$ immediately before atomic formulas

$$\begin{aligned}
\neg\neg F &\Leftrightarrow F \\
\neg(F \& G) &\Leftrightarrow \neg F \vee \neg G \\
\neg(F \vee G) &\Leftrightarrow \neg F \& \neg G \\
\neg(\forall x)\ F(x) &\Leftrightarrow (\exists x)\ \neg F(x) \\
\neg(\exists x)\ F(x) &\Leftrightarrow (\forall x)\ \neg F(x)
\end{aligned}$$

(3) Move the quantifiers to the left

$$(Qx)\ F(x) \overset{\&}{\vee} G \iff (Qx)\,(F(x) \overset{\&}{\vee} G)$$

$$(\forall x)\,F(x)\ \&\ (\forall x)\,H(x) \iff (\forall x)\,(F(x)\ \&\ H(x))$$

$$(\exists x)\,F(x) \vee (\exists x)\,H(x) \iff (\exists x)\,(F(x) \vee H(x))$$

$$(Q_1 x)\,F(x) \overset{\&}{\vee} (Q_2 x)\,H(x) \iff (Q_1 x)(Q_2 z)\,(F(x) \overset{\&}{\vee} H(z))$$

(4) Use the distributive law to get conjunction normal form:

$$F \vee (G \,\&\, H) \iff (F \vee G)\ \&\ (F \vee H)$$

↓

Skolem normal form $(\forall x_1)\,(\forall x_2) \cdots (\forall x_k)\ N$

(1) If $(\exists x)$ is the left-most quantifier, choose a new constant c and replace all x appearing in the formula by c, and delete $(\exists x)$

(2) If $(\forall y_1), \ldots, (\forall y_m)$ are all the universal quantifiers appearing before $(\exists x)$, choose a new m-ary functor f, replace all x in the formula by $f(y_1, \cdots, y_m)$, and delete $(\exists x)$

↓

Set of clauses

(1) Remove all universal quantifiers and rewrite the formula

$$F = C_1\ \&\ C_2\ \&\ C_3$$

into the set of clauses

$$\mathcal{B} = \{ C_1, C_2, C_3 \}$$

(2) Rename variables in $\mathcal{B}$ so that each clause has different variables.

Clause is a finite disjunction of zero or more literals.

Empty clause $\Box$ is a clause which contains no literal.

● $F = C_1 \,\&\, C_2 \,\&\, C_3$ is inconsistent iff $\mathcal{B} = \{ C_1, C_2, C_3 \}$ is inconsistent.

Example.

$$F \equiv (\exists x)\,[P(x) \,\&\, (\forall y)\,(D(y) \Rightarrow L(x,y))]$$

$$(\exists x)\,[P(x) \,\&\, (\forall y)\,(\neg D(y) \vee L(x,y))]$$

$$(\exists x)\,(\forall y)\,[P(x) \,\&\, (\neg D(y) \vee L(x,y))]$$

$$(\forall y)\,[P(c) \,\&\, (\neg D(y) \vee L(c,y))]$$

$$\underbrace{P(c)}_{C_1} \,\&\, \underbrace{(\neg D(y) \vee L(c,y))}_{C_2}$$

$$\mathcal{B} = \{ P(c) \,,\, \neg D(\underset{z}{y}) \vee L(c,\underset{z}{y}) \}$$

Example.

$$F \equiv (\forall x)(\forall y)\,[(\exists z)(P(x,z) \,\&\, P(y,z)) \Rightarrow (\exists z)\,Q(x,y,z)]$$

$$(\forall x)(\forall y)\,[\neg(\exists z)(P(x,z) \,\&\, P(y,z)) \vee (\exists z)\,Q(x,y,z)]$$

$$(\forall x)(\forall y)\,[(\forall z)(\neg P(x,z) \vee \neg P(y,z)) \vee (\exists z)\,Q(x,y,z)]$$

$$(\forall x)(\forall y)(\forall z)(\exists u)\,[\neg P(x,z) \vee \neg P(y,z) \vee Q(x,y,u)]$$

$$(\forall x)(\forall y)(\forall z)\,[\neg P(x,z) \vee \neg P(y,z) \vee Q(x,y,f(x,y,z))]$$

$$\underbrace{\neg P(x,z) \vee \neg P(y,z) \vee Q(x,y,f(x,y,z))}_{C_1}$$

$$\mathcal{B} = \{C_1\}$$

4.3. RESOLUTION PRINCIPLE

Modus ponens

$$\frac{F, \quad F \Rightarrow G}{G}$$

If we rewrite it into the conjunctive normal form:

$$\begin{array}{ll} \boxed{F} & \equiv C_1 \\ \boxed{\neg F} \vee G & \equiv C_2 \\ \hline G & \equiv C_3 \end{array}$$

} parent clauses

resolvent of C_1 and C_2

resolved literals

Resolution principle (Robinson, 1965) is an extension of MP

$$\begin{array}{l} C_1 \equiv L_0 \vee L_1 \vee L_2 \\ C_2 \equiv \neg L_0 \vee L_3 \vee L_4 \vee L_5 \\ \hline C_3 \equiv L_1 \vee L_2 \vee L_3 \vee L_4 \vee L_5 \end{array}$$

resolvent of C_1, C_2

Th. $\{C_1, C_2\} \vdash C_3$

Def. Let $\mathcal{B}$ be a set of clauses. **Resolution deduction** of C from $\mathcal{B}$ is a finite sequence $C_1, C_2, \dots, C_n$ such that

(a) each C_i is either a clause in $\mathcal{B}$ or a resolvent of clauses preceding C_i,

(b) $C_n \equiv C$

Th. Resolution principle is **complete**, i.e. a set of clauses $\mathcal{B}$ is inconsistent iff there is a resolution deduction of $\Box$ from $\mathcal{B}$.

Consequently,

$\mathcal{B} \vdash G$ iff $\mathcal{B} \cup \{\neg G\}$ is inconsistent

iff there is a res. deduction of $\Box$ from $\mathcal{B} \cup \{\neg G\}$

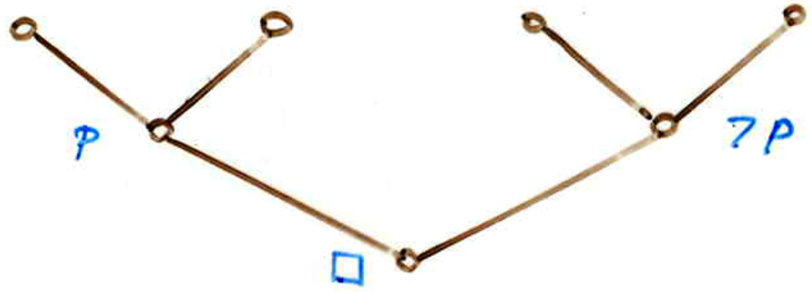
Example.

$\mathcal{B} = \{ Q \Rightarrow P, \ P \Rightarrow Q, \ P \Rightarrow \neg Q \}$

$G \equiv \neg P \ \& \ \neg Q$

Prove $\mathcal{B} \vdash G$

$\mathcal{B} \cup \{\neg G\} = \{ P \vee Q, \ P \vee \neg Q, \ \neg P \vee Q, \ \neg P \vee \neg Q \}$



Done.

4.4. UNIFICATION ALGORITHM

Now consider Predicate Calculus:

$$C_1' \equiv P(f(y)) \vee Q(f(y))$$
$$C_2 = \neg P(f(y)) \vee R(y).$$

We can resolve $P(x)$ and $\neg P(f(y))$ only if

$$x \equiv f(y)$$

Resolvent of C_1', C_2

$$C_3 \equiv Q(f(y)) \vee R(y)$$

The process of matching two literals is called unification algorithm.

Example. Unify $P(a, x, f(g(y)))$
and $P(z, f(z), f(u))$

1. step $z \equiv a$: $P(a, x, f(g(y)))$
$P(a, f(a), f(u))$

2. step $x \equiv f(a)$: $P(a, f(a), f(g(y)))$
$P(a, f(a), f(u))$

3. step $u \equiv g(y)$: $P(a, f(a), f(g(y)))$

The above literals match if the above substitutions are applied.

Resolution principle for PreC:

Before resolving literals, try to unify literals of the two clauses.

Example. Let $\mathcal{B} = \{B_1, B_2\}$ where

$$B_1 \equiv (\exists x)\,[P(x)\ \&\ (\forall y)\,(D(y) \Rightarrow L(x,y))]$$

$$B_2 \equiv (\forall x)\,[P(x) \Rightarrow (\forall y)\,(Q(y) \Rightarrow \neg L(x,y))]$$

and let $G \equiv (\forall x)\,(D(x) \Rightarrow \neg Q(x))$

Prove $\mathcal{B} \vdash G$

We will prove that $\mathcal{B} \cup \{\neg G\}$ is inconsistent.

Set of clauses corresponding to $\mathcal{B} \cup \{\neg G\}$:

B_1: $P(c_1)$ (1)

$\neg D(y_2) \vee L(c_1, y_2)$ (2)

B_2: $(\forall x)(\forall y)\,[\neg P(x) \vee \neg Q(y) \vee \neg L(x,y)]$

$\neg P(x_3) \vee \neg Q(y_3) \vee \neg L(x_3, y_3)$ (3)

$\neg G$: $\neg(\forall x)\,(\neg D(x) \vee \neg Q(x))$

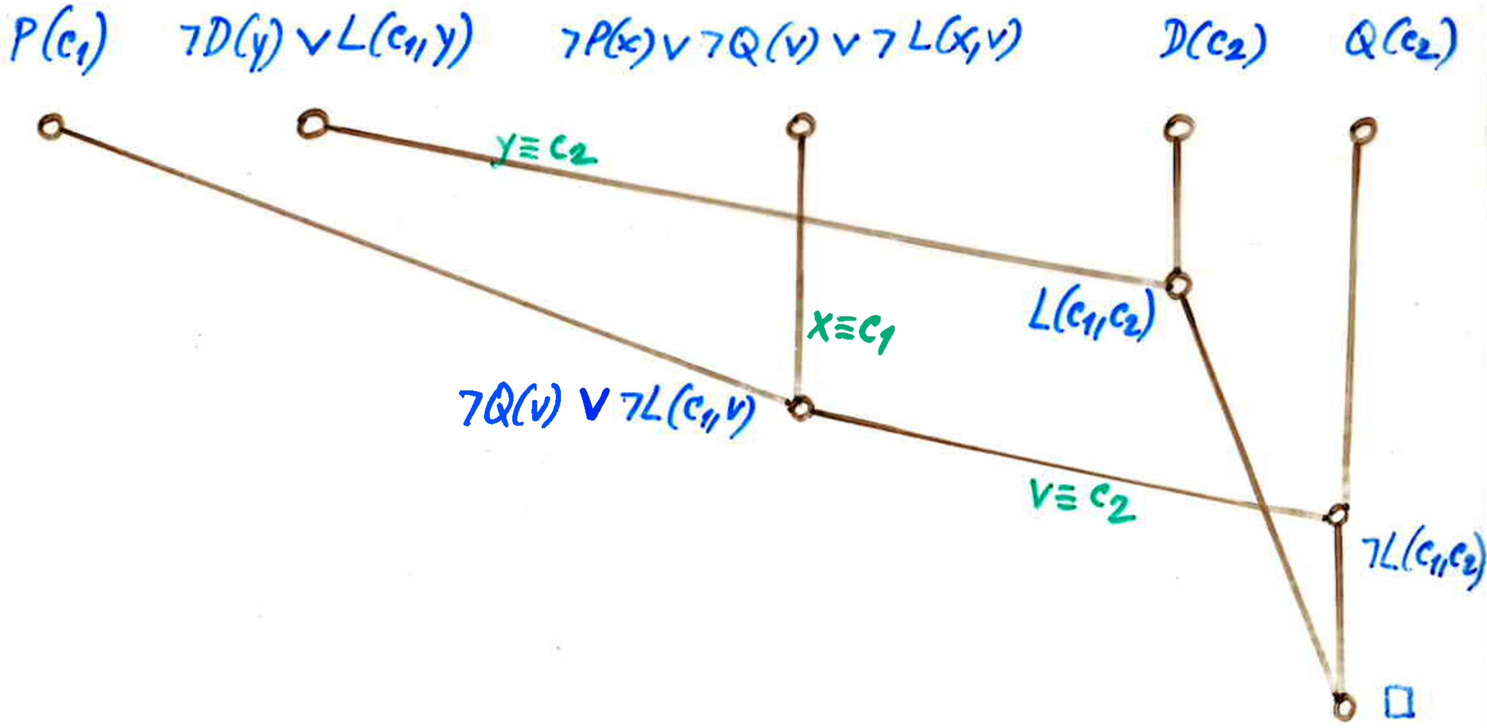
$(\exists x)\,(D(x)\ \&\ Q(x))$

$D(c_2)\ \&\ Q(c_2)$

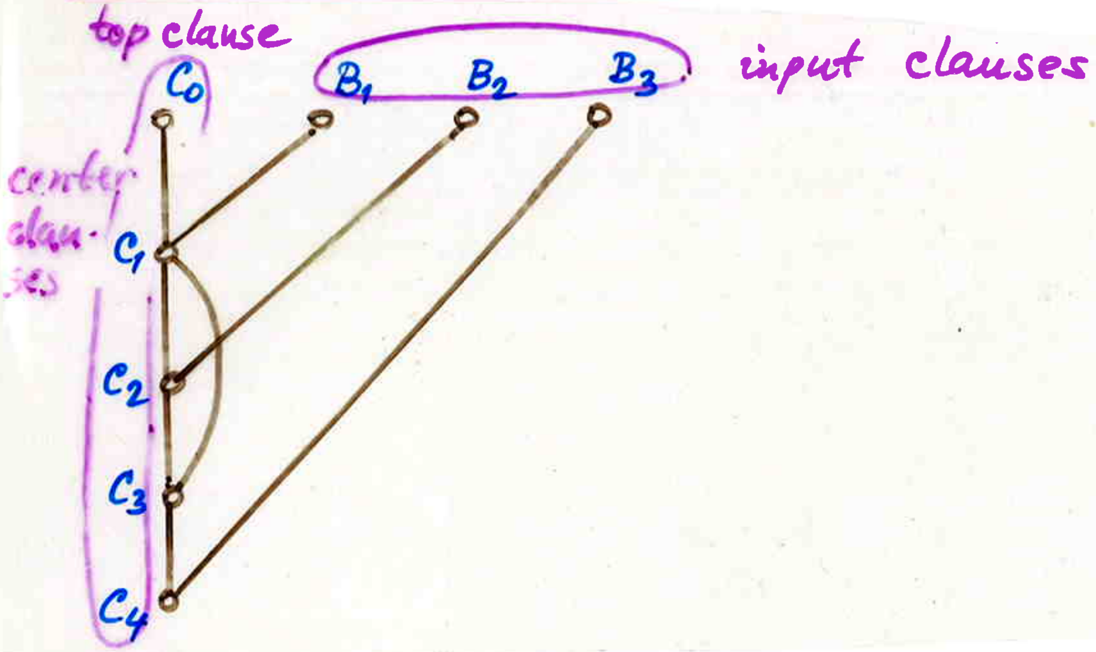
$D(c_2)$ (4)

$Q(c_2)$ (5)

Resolution deduction of □ from the above set:



4.5. LINEAR STRATEGY



Linear resolution:

C_{i+1} is a resolvent of the center clause C_i and

$\begin{cases} \text{an input clause} \\ \text{a center clause } C_j,\ j<i \end{cases}$

Note. The top clause C_0 is usually the negation of the formula which we are to prove to be a theorem from some premises.

Example.

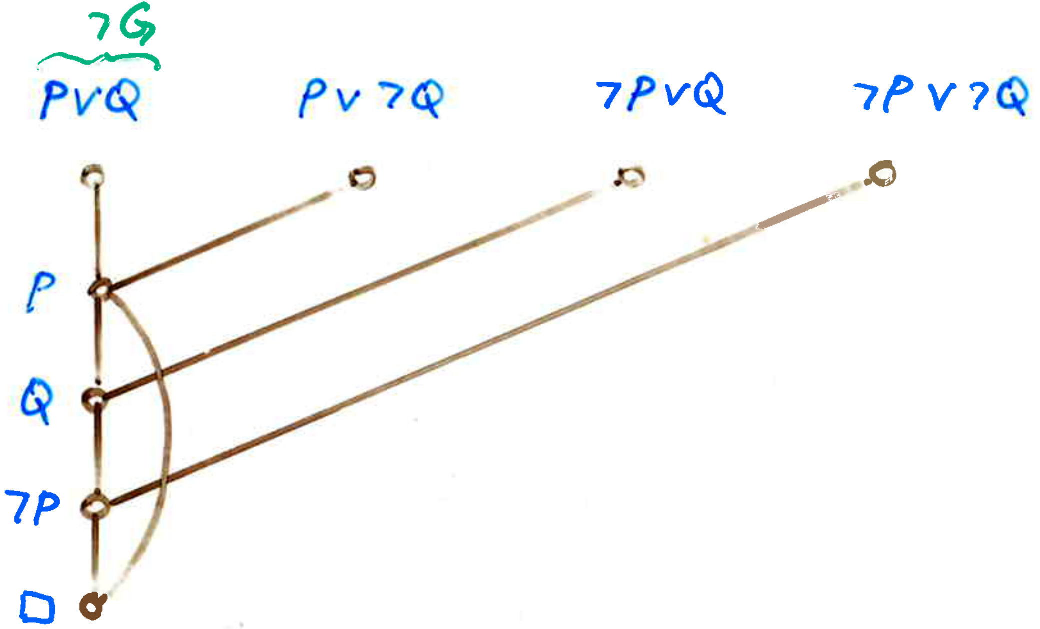
$\mathcal{B} = \{ Q \Rightarrow P,\ P \Rightarrow Q,\ P \Rightarrow \neg Q \}$

$G = \neg P \& \neg Q$

Prove $\mathcal{B} \vdash G$

a) Linear resolution

We will prove that $\mathcal{B} \cup \{\neg G\}$ is inconsistent.



b) Syntactically

$\vdash (Q \Rightarrow P) \,\&\, (P \Rightarrow \neg Q) \Rightarrow (Q \Rightarrow \neg Q)$

known theorem

$\vdash (F \Rightarrow G) \,\&\, (G \Rightarrow H) \Rightarrow (F \Rightarrow H)$

$\mathcal{B} \vdash Q \Rightarrow \neg Q$ 2× MP

$\vdash (Q \Rightarrow \neg Q) \Rightarrow \neg Q$

known theorem

$\vdash (\neg F \Rightarrow F) \Rightarrow F$

→ $\mathcal{B} \vdash \neg Q$ MP

$\vdash (P \Rightarrow Q) \Rightarrow (\neg Q \Rightarrow \neg P)$

→ $\mathcal{B} \vdash \neg Q \Rightarrow \neg P$ MP

$\mathcal{B} \vdash \neg P$ MP

$\vdash \neg P \Rightarrow (\neg Q \Rightarrow \neg P \,\&\, \neg Q)$

known theorem

$\vdash F \Rightarrow (G \Rightarrow F \,\&\, G)$

$\mathcal{B} \vdash \underbrace{\neg P \,\&\, \neg Q}_{G}$ 2× MP

c) Semantically

		$\mathcal{B}$			G
P	Q	$Q \Rightarrow P$	$P \Rightarrow Q$	$P \Rightarrow \neg Q$	$\neg P \& \neg Q$
0	0	1	1	1	1
0	1	0	1	1	0
1	0	1	0	1	0
1	1	1	1	0	0

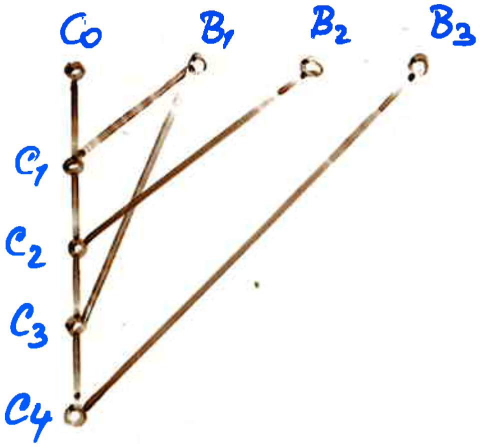
the model of $\mathcal{B}$ is a model of G as well.

- Linear resolution is complete.

4.6. INPUT RESOLUTION

Linear resolution is complete but we must store the entire sequence of center clauses (inefficient!)

Input resolution is linear resolution but a center clause is resolved with an input clause only.



Input resolution is incomplete, but we need not store the center clauses (completeness × efficiency).

Def. Horn clause is a clause which has at most one literal without ¬

Th. Input resolution with Horn clauses and the top clause containing only literals with ¬ is complete.

- PROLOG is equivalent to the Input resolution for Horn clauses.