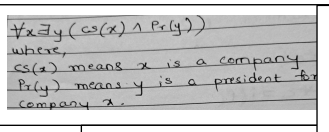
A *Horn clause* is a clause that contains at most one positive literal. A *program clause* is one that contains exactly one positive literal. (In PRO- LOG notation it looks like *A:- B1, B2, . •• ,Bn .)* If a program clause contains some negative literals it is called a *rule (n* > 0 in the notation of (2)). A *fact (or unit clause)* is one that consists of exactly one positive literal (Notation: *A.* or *A:- .*). A *goal clause* is one that contains no positive literals. (Thus, in PROLOG it is entered as a question with the symbol ?- .) A PROLOG *program* is a set of clauses containing only program clauses (rules or facts).

For any $\mathcal{A}$ and S, if $\square \in$ RA(S), then $S \in$ UNSAT.	For any $\mathcal{A}$ and S, if $S \in$ UNSAT, then $\square \in$ RA(S).



Definition 2.2: Terms.

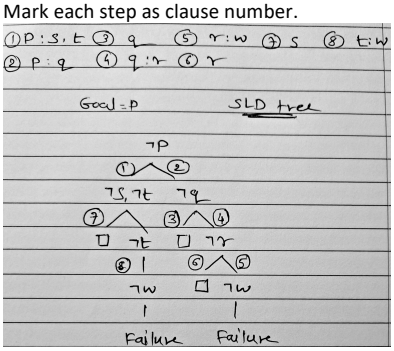
(i) Every variable is a term.

(ii) Every constant symbol is a term.

(iii) If f is an n-ary function symbol ($n = 1, 2, \ldots$) and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is also a term.

SLD Tree –

1. Take Goal value as root value (negative value). Eg. -p.
2. Take expression and negate if we have more than one value (p : s,t => -s, -t).
3. If we have single expression, then do resolution and put empty box at the end (-p => p => box).
4. If we don't have clause, then mark as Failure.
5. Mark each step as clause number.



Assume, Assignment $\mathcal{A}$ for which $\mathcal{A}(P)$ is true
The goal clause G consists of negative literals
$\neg q_i$ and q_i will be true for all $i > 0$, $i \leq n$.
All q_i are facts from the *prolog* program P. Thus it is a consequence of $\mathcal{A}(P)$ is true.
A
$\mathcal{A}$ satisfies P ($\mathcal{A} \vdash P$) and makes G=*False* as q_i=*True* which and its negation $\neg q_i$=*False* , which implies G=*False*.

Definition 2.6:

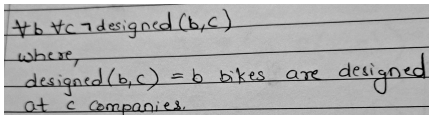
(i) A subformula of a formula φ is a consecutive sequence of symbols from φ which is itself a formula.

(ii) An occurrence of a variable v in a formula φ is bound if there is a subformula ψ of φ containing that occurrence of v such that ψ begins with $(\forall v)$ or $(\exists v)$. (This includes the v in $\forall v$ or $\exists v$ that are bound by this definition.) An occurrence of v in φ is free if it is not bound.

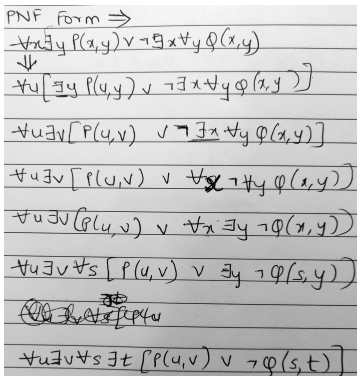
(iii) A variable v is said to occur free in φ if it has at least one free occurrence there.

(iv) A sentence of predicate logic is a formula with no free occurrences of any variable, i.e., one in which all occurrences of all variables are bound.

(v) An open formula is a formula without quantifiers.



If $\square \in$ RT(S), then S is unsatisfiable.	If S is unsatisfiable, then $\square \in$ RT(S).



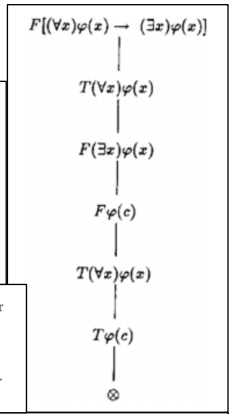
Theorem 7.7: (Skolem-Löwenheim): If a countable set of sentences S is satisfiable (that is, it has some model), then it has a countable model.

Theorem 7.8: (Completeness and Soundness):

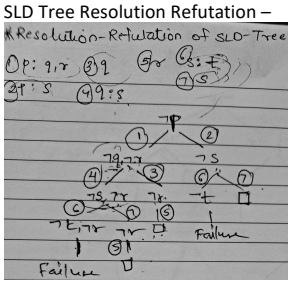
(i) α is tableau provable from $S \Leftrightarrow \alpha$ is a logical consequence of S.

(ii) If we take α to be any contradiction such as $\beta \wedge \neg\beta$ in (i), we see that S is inconsistent if and only if S is unsatisfiable.

Theorem 7.9: Let $S = \{\alpha_1, \alpha_2, \ldots\}$ be a set of sentences of predicate logic. S is satisfiable if and only if every finite subset of S is satisfiable.



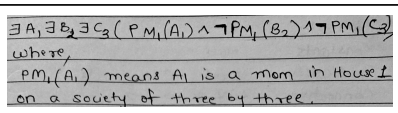
SLD Tree Resolution Refutation –

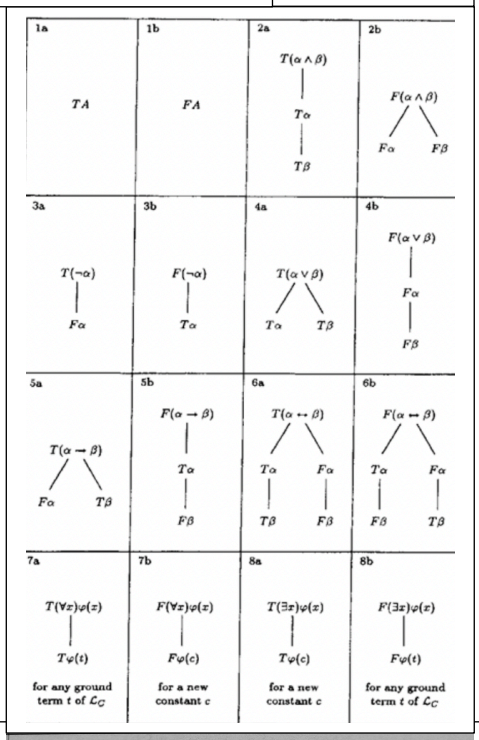


Definition 2.3: Terms with no variables are called variable-free terms or ground terms.

Definition 2.4: An atomic formula is an expression of the form $R(t_1, \ldots, t_n)$ where R is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms.

If there is a *resolution refutation* of S, then S is unsatisfiable.	If S is unsatisfiable, then there is a resolution refutation of S.





Definition 2.1: A *language* $\mathcal{L}$ consists of the following disjoint sets of distinct primitive symbols:

(i) Variables: $x, y, z, v, x_0, x_1, \ldots, y_0, y_1, \ldots, \ldots$ (an infinite set)

(ii) Constants: $c, d, c_0, d_0, \ldots$ (any set of them)

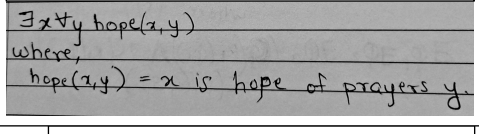
(iii) Connectives: $\wedge, \neg, \vee, \rightarrow, \leftrightarrow$

(iv) Quantifiers: $\forall, \exists$

(v) Predicate symbols: $P, Q, R, P_1, P_2, \ldots$ (some set of them for each arity $n = 1, 2, \ldots$. There must be at least one predicate symbol in the language but otherwise there are no restrictions on the number of them for each arity).

(vi) Function symbols: $f, g, h, f_0, f_1, \ldots, g_0, \ldots$ (any set of them for each arity $n = 1, 2, \ldots$. The 0-ary function symbols are simply the constants listed by convention separately in (ii). The set of constant symbols may also be empty, finite or infinite).

(vii) Punctuation: the comma , and (right and left) parentheses) , (.



ion 2.5: *Formulas.*

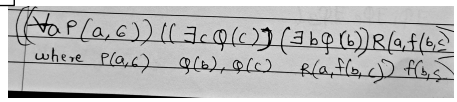
(i) Every atomic formula is a formula.

(ii) If α, β are formulas, then so are $(\alpha \wedge \beta)$, $(\alpha \rightarrow \beta)$, $(\alpha \leftrightarrow \beta)$, $(\neg\alpha)$ and $(\alpha \vee \beta)$.

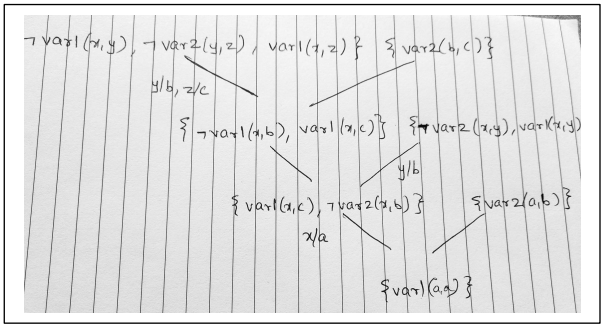
(iii) If v is a variable and α is a formula, then $((\exists v)\alpha)$ and $((\forall v)\alpha)$ are also formulas.

A is a subset of B if for any x, if x belongs to A then x belongs to B.
subset(A, B) ← ((∀x)(belongsT o(x, A) → belongsT o(x, B)))
Predicates - subset(X, Y): X is subset of Y
belongsTo(X, Y): X belongs to Y

If there is a SLD-resolution refutation of $P \cup \{G\}$	If $P \cup \{G\} \in$ UNSAT and R is any selection rule, then there is a SLD-resolution refutation of $P \cup \{G\}$ via R.



Consider the formula proven unsatisfiable, S = {{*p* , r}, {*q*, *-r*}, {*-q*} , {*-p*, *t*}, {*-s*} , {*s*, *-t*}}. When we assume *p* to be true, we eliminate the clauses containing *p* (as they are true) and omit *-p* from the others (since being false *-p* cannot help to satisfy them) to get *SP* = {{*q*, *-r*}, {*-q*}, {*t*}, {'8}, {*s*, *-t*}} . On the other hand, when we assume that *p* is false we eliminate clauses containing *-'p* and remove *p* from the others to get *sP* = {{r} , {*q*, *-r*}, {*-q*} , {*-s*} , {*s*, *-t*}} .



General unifier – {P(x, g(x), y), P(z,u,g(a)), P(a,g(a),v)} where 'a' is a constant with x,y,z,u,v being a variable as below – (x, g(x), y) = (z,u,g(a)) = (a,g(a),v);
x = z = a, g(x) = u = g(a), y = g(a) = v;
z = z= a, u = y = g(a) = v;
hence the given set becomes {P(a,g(a),g(a)), P(a,g(a),g(a)), P(a,g(a),g(a)} = {P(a,g(a),g(a)}

Lemma 9.1: *For any string of quantifiers* $\overrightarrow{Qx} = Q_1x_1Q_2x_2\ldots Q_nx_n$ *(each* Q_i *is* $\forall$ *or* $\exists$*) and any formulas* φ, ψ *we have the following provable equivalences:*

(1a) $\vdash \overrightarrow{Qx}\neg\forall y\varphi \leftrightarrow \overrightarrow{Qx}\exists y\neg\varphi$

(1b) $\vdash \overrightarrow{Qx}\neg\exists y\varphi \leftrightarrow \overrightarrow{Qx}\forall y\neg\varphi$

(2a) $\vdash \overrightarrow{Qx}(\forall y\varphi \vee \psi) \leftrightarrow \overrightarrow{Qx}\forall z(\varphi(y/z) \vee \psi)$

(2a′) $\vdash \overrightarrow{Qx}(\varphi \vee \forall y\psi) \leftrightarrow \overrightarrow{Qx}\forall z(\varphi \vee \psi(y/z))$

(2b) $\vdash \overrightarrow{Qx}(\exists y\varphi \vee \psi) \leftrightarrow \overrightarrow{Qx}\exists z(\varphi(y/z) \vee \psi)$

(2b′) $\vdash \overrightarrow{Qx}(\varphi \vee \exists y\psi) \leftrightarrow \overrightarrow{Qx}\exists z(\varphi \vee \psi(y/z))$.

where z *is a variable not occurring in* φ *or* ψ *or among the* x_i.

Removing existence quantifiers (Skolemization)

$\varphi = \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_m R(x_1, \ldots, x_n, y_1, \ldots, y_m)$

is equi-satisfiable to

$\psi = \forall x_1 \ldots \forall x_n R(x_1, \ldots, x_n, \quad f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$. (*n* should be *m*)

For example, $\forall x \exists y\ mother(y, x)$

Clearly y depends on x, we can introduce function f such that y is $f(x)$.

$\forall x\ mother(f(x), x)$

Formation tree of a formula