Syntax (of the language by which we can *represent* information (in our mind or in English), called *propositions*). Semantics (of the language by which we can *reason* with the propositions). ***Meta Variables*** - It can be defined as a symbol or string of symbols used to denote or refer to another object.

Def (arithmetic expression)
1. An *arithmetic constant* is one of 1, 2, 3 …(real number)
2. A *variable* is *x, y, z* 3. An *operator* is +, -, /, *
3. If t1 is an arithmetic expression and t2 is an arithmetic expression, *t1 op t2, op* is an operator, is an *arithmetic expression*.

Def (The language of propositional logic) *The language of propositional logic* consists of the following symbols- *Connectives* : ∧, ∨, ¬, →,←→, *Parentheses* : ),(, *Propositional Variables (letters)* : A, A1,B, B1,

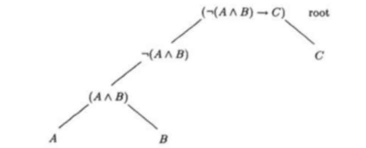
**Formation trees -** A proposition has a structure and we use tree to visualize. The structure shows a "decomposition" of the proposition. **Problem Decomposition** is a problem solving method. For understanding properties of propositions, we need to understand its decomposition (or know its components) and thus apply the problem decomposition methods.

**Definition 2.1** (Propositions):

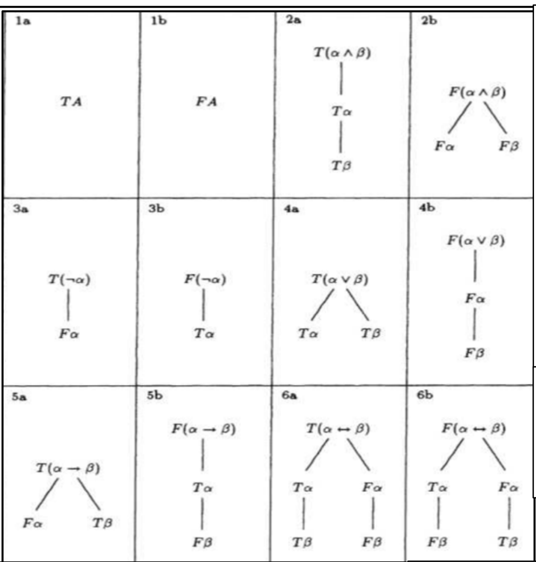
(i) Propositional letters are propositions.

(ii) If $\alpha$ and $\beta$ are propositions, then $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\neg\alpha)$, $(\alpha \to \beta)$ and $(\alpha \leftrightarrow \beta)$ are propositions.

(iii) A string of symbols is a proposition if and only if it can be obtained by starting with propositional letters (i) and repeatedly applying (ii).




| Steps | Proof | Reason |
|---|---|---|
| a. | Assume $\forall_x, x \in \alpha$ | -Given $\alpha$ is any proposition value. |
| b. | For any valuation $V$, assume $V$ is model of $\alpha$. | |
| c. | For any $\sigma \in \alpha$, $V(\sigma)$=T. | -By (b) and Definition of truth valuation. |
| d. | $V(x)$ = T. | -By (a), (b) and (c). |
| e. | For any valuation $V$, If $V$ is a model of $\alpha$, then $V(x)$ = T. | -By Definition, $V(\tau)$ = T for all $\tau \in \Sigma \Rightarrow V(\sigma)$ = T. |
| f. | $\alpha \models x$ | -By definition $x$ is consequences of $\alpha$ and by (e). |
| g. | $x \in C_n\alpha$. | -By definition, $C_n\Sigma$ be a set of consequences of $\Sigma$ and by (f). |
| h. | $\forall_x, x \in \alpha \Rightarrow x \in C_n\alpha$. | -By (g). |
| i. | $\alpha \subseteq C_n\alpha$. | -By (h). |
| j. | $\alpha$ is consequences of $\alpha$. | |




⑦ ((A∧B )→ C )

| | | |
|---|---|---|
| ① | A is a propositional letter | By def of propositional letter |
| ② | A is a proposition | By (1) & def of propo |
| ③ | B is a propositional letter | By def of propo. letter |
| ④ | B is a proposition | By (3) & def of prop |
| ⑤ | ~~The~~ A & B are propositions | |
| ⑥ | If A & B are propositions (A∧B) is a propo. | |
| ⑦ | C is a propositional letter | |
| ⑧ | C is a proposition | |
| ⑨ | (A∧B) and C is a proposition | |
| ⑩ | If (A∧B) and C is a proposition then (A∧B → C) is a proposition | By lang of prop & def of propo |
| ⑪ | A∧B → C is a propo. | |

QED

**finition 3.7** Let $\Sigma$ be a (possibly infinite) set of propositions. We say t $\sigma$ is a consequence of $\Sigma$ (and write $\Sigma \models \sigma$) if, for any valuation $\mathcal{V}$, $\tau) = T$ for all $\tau \in \Sigma) \Rightarrow \mathcal{V}(\sigma) = T$.

te that, if $\Sigma$ is empty, $\Sigma \models \sigma$ (or just $\models \sigma$) iff $\sigma$ is valid. We also write s as $\models \sigma$. This definition gives a semantic notion of consequence.

| Steps | Proof | Reason |
|---|---|---|
| a. | Assume $\forall_x, x \in \alpha$ | -Given $\alpha$ is any proposition value. |
| b. | For any valuation $V$, assume $V$ is model of $\alpha$. | |
| c. | For any $\sigma \in \alpha$, $V(\sigma)$=T. | -By (b) and Definition of truth valuation. |
| d. | $V(x)$ = T. | -By (a), (b) and (c). |
| e. | Assume $M(\Sigma_1)$ the set of all model of $\Sigma_1$ | |
| f. | As $\Sigma_1 \subseteq \Sigma_2$,V(x)$\in \Sigma_2$ | -By definition every model of $\Sigma_2$ is also model of $\Sigma_1$. |
| g. | $\Sigma_1 \models x$. | -By definition, $x$ is consequences of $\Sigma$ and by (d). |
| h. | $x \in C_n(\Sigma_1)$. | -By (g). |
| i. | $x \in \Sigma_1 \Rightarrow x \in C_n(\Sigma_1)$. | -By (h). |
| j. | $\forall_x, x \in \Sigma_1 \Rightarrow x \in C_n(\Sigma_1)$. | By (a) and (i). |
| k. | $\Sigma_1 \subseteq C_n(\Sigma_1)$. | |
| l. | $x \in \Sigma_2 \Rightarrow x \in C_n(\Sigma_2)$. | -By (f) and (k). |
| m. | $C_n(\Sigma_1) \subseteq C_n(\Sigma_2)$. | -By (l). |
| n. | $\Sigma_1 \subseteq \Sigma_2 \Rightarrow C_n(\Sigma_1) \subseteq C_n(\Sigma_2)$. | -By (f), (k) and(m). |

③ Prove B is consequences of {C, D, (C∧D → B)}

① By the definition of consequences we can say that

σ is consequence of Σ if and only if (V(x) = T ∀(x)∈Σ) ⇒ V(σ) = T

② Let us assume σ = B and Σ = {C, D, (C∧D → B)}

③ ~~If~~ From (1), (2) we can say that V(C) = T , V(D) = T

④ From the definition of truth table, (3) V(C∧D) = T

⑤ From (2), (4) and definition of truth table V(C∧D → B) = T

⑥ Let us consider α = C∧D , β = B From (5) our argument will be α → β = T

⑦ From definition of Truth Table α → β will be true if and only if β is true.

⑧ From (6) and (7) we can say that V(B) = T

⑨ Hence, we ~~pro~~ can say that B is consequences of {C, D, C∧D → B}   From def of consequence, and (8)

QED

Prove soundness result Theorem 5.1.

Prove $\vdash \sigma$ implies $\models \sigma$. (\vdash for ⊢)

Proof
(b2.t) Assume $\vdash \sigma$
We next prove (b2.b) by contradiction
(F3) Assume the opposite, i.e., $\models \sigma$ is false.
(F4) there exists a valuation $V$, $V(\sigma) = F$   (by (F3))
(F5) by (b2.t), there exists a contradictory tableau $\tau$
(F6) (F4) implies that $V$ agrees with $F\sigma$
(F7) by Lemma 5.2, there exists a path $P_1$ from $\tau$ such that V agrees with every entry of this path.
(F8) Since $\tau$ is contradictory (by (F5)), $P_1$ is contradictory.
(F9) Since $P_1$ is contradictory, there exists $T\beta$ and $F\beta$ on $P_1$.
(F10) Since $V$ agrees with $P_1$, $V(\beta) = T$ and $V(\beta) = F$.
Contradicting that a valuation does not assign two values to one proposition.
Hence conclusion (b2.b) holds.
(b2.b) $\models \sigma$
(b1) $\vdash \sigma$ implies $\models \sigma$.
QED

⑥ ① A real number is an arithmetic ~~constant~~ expression Eg 1, 2, 3 . . .
② A variable is an arithmetic expression Eg x, y, z . . .
③ From (1) and (2) If 1 and x are arithmetic expressions then (x+1), (x−1), (x*1), (x/1) are arithmetic expression.
④ If x, (x−1) are arithmetic expression then ((x * (x−1))/x) is an arithmetic expression – By defⁿ of propositions.

## Definition of Truth valuation

Definition 3.2: A **truth valuation** $\mathcal{V}$ is a function that assigns to each proposition $\alpha$ a unique truth value $\mathcal{V}(\alpha)$ so that its value on a compound proposition (that is, one with a connective) is determined in accordance with the appropriate truth tables. Thus, for example, $\mathcal{V}((\neg\alpha)) = T$ iff $\mathcal{V}(\alpha) = F$ and $\mathcal{V}((\alpha \vee \beta)) = T$ iff $\mathcal{V}(\alpha) = T$ or $\mathcal{V}(\beta) = T$. We say that $\mathcal{V}$ **makes** $\alpha$ **true** if $\mathcal{V}(\alpha) = T$. A precise def on the condition of the "function" $\mathcal{V}$ in the def 3.2 For any compound proposition ($\alpha \wedge \beta$) where $\alpha$ and $\beta$ are proposition, $\mathcal{V}(\alpha \wedge \beta)$ = T iff $\mathcal{V}(\alpha)$ = T and $\mathcal{V}(\beta)$ = T

A finite tableau is a binary tree, labeled with signed propositions called entries, that satisfies the following inductive definition: (i) All atomic tableaux are finite tableaux. (ii) If T is a finite tableau, P a path on T, E an entry of T occurring on P and 7' is obtained from T by adjoining the unique atomic tableau with root entry E to T at the end of the path P, then 7' is also a finite tableau.

A truth valuation V function gives each proposition a distinct truth value V(a) in order to determine the value of a proposition's value on a compound proposition (i.e., one with a connective). The pertinent truth tables are then used to calculate this value.

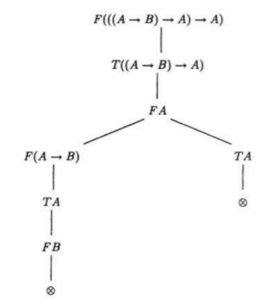
Consider the propositional letters A and B. as defined by propositional logic. Letters with a proposal are always propositions. If the assertions A and B are true, then the propositions (¬A), (A ∨B), (A ∧ B), (A → B), and (A ↔ B) are also true. The Truth values of a compound statement of A and B are always True or False according to propositional logic. Additionally, based on their compound statements and connectives, their truth values will be determined.

Consider the statement, "If and only if A, then B," in the proposition (A ↔ B). This statement will be understood as ((A → B) wedge (B → A)) or ((neg A ∨B) ∧ (A ∨ ¬B)) if it is rewritten.

According to the truth table below, B is true if and only if A is true. Alternatively, B is false if and only if A is false.

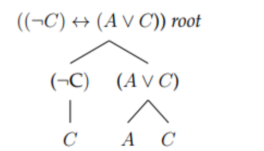
| A | B | (A ↔ B) |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

Prove $(((A \to B) \to A) \to A)$ is a tautology




Soundness. For any proposition $\alpha$, if there is a tableau proof of $\alpha$, then $\alpha$ is valid or a tautology.
Completeness. For any proposition $\alpha$, if $\alpha$ is valid, there is a tableau proof of $\alpha$.

CST




i) reduced - if all the entries on one path through the atomic tableau with root E occur on P.
(ii) P is contradictory if, for some proposition a, Ta and Fa are both entries on P. P is finished if it is contradictory or every entry on P is reduced on P.(iii) T is finished if every path through T is finished.
(iv) T is contradictory if every path through T is contradictory.

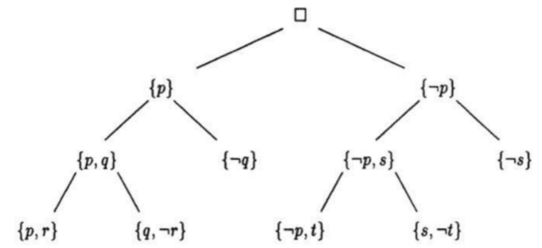
**Definition 8.5** A resolution tree proof of $C$ from $S$ is a labeled binary tree $T$ with the following properties:

(i) The root of $T$ is labeled $C$.

(ii) The leaves of $T$ are labeled with elements of $S$.

(iii) If any nonleaf node $\sigma$ is labeled with $C_2$ and its immediate successors $\sigma_0, \sigma_1$ are labeled with $C_0, C_1$, respectively, then $C_2$ is a resolvent of $C_0$ and $C_1$.

A resolution tree refutation of the formula
$S = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}\}$, i.e., a resolution tree proof of $\square$ from $S$:




9) completeness of deductions from premises
if $\alpha$ is a consequence of a set $\Sigma$ of premises.
then there is a tableau deduction of $\alpha$ from $\Sigma$.
i.e $\Sigma \models \alpha \Rightarrow \Sigma \vdash \alpha$.
if $\Sigma \models \alpha$, every valuation $V$ that makes every proposition in $\Sigma$ true also makes $\alpha$ true. consider the CST from $\Sigma$ with root $F\alpha$. It is finished by theorem 6.4. Now apply Lemma 6.7.
again we consider the problem of finiteness of proofs.
if $\Sigma$ is finite, then the CST from $\Sigma$ itself is also finite.

(2) $F(\neg(A \wedge B) \rightarrow (\neg A \vee \neg B))$

$T(\neg(A \wedge B))$

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

$F(A \wedge B)$

$F(\neg A)$

$F(\neg B)$

$T(A)$

$T(B)$

$F(A)$ $\qquad$ $F(B)$

× $\qquad$ ×

**Definition 8.2** (i) $p, q, r, \neg p, \bar{q}(= \neg q), \bar{r}$ and $\neg\bar{q}(= q)$ are literals.

(ii) $\{p, r\}$, $\{\neg q\}$ and $\{q, \neg r\}$ are clauses.

(iii) $S = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{s, \neg t\}\}$ is a formula that, in our original notation system, would be written as $((p \vee r) \wedge (q \vee \neg r) \wedge (\neg q) \wedge (\neg p \vee t) \wedge (s \vee \neg t))$.

(iv) If $\mathcal{A}$ is given by $\{p, q, r, s, t\}$, i.e., the (partial) assignment such that $\mathcal{A}(p) = T = \mathcal{A}(q) = \mathcal{A}(r) = \mathcal{A}(s) = \mathcal{A}(t)$, then $\mathcal{A}$ is an assignment no satisfying the formula $S$ in (iii). $S$ is, however, satisfiable.

**Definition 8.3** [Resolution] From clauses $C_1$ and $C_2$ of the form $\{l\} \sqcup C$ and $\{\bar{l}\} \sqcup C_2'$, infer $C = C_1' \cup C_2'$ which is called a resolvent of $C_1$ and $C_2$ (Here $l$ is any literal and $\sqcup$ means that we are taking a union of disjoint sets.) We may also call $C_1$ and $C_2$ the parent and $C$ their child and say that we resolved on (the literal) $l$.

**Definition 8.4** A (resolution) deduction or proof of $C$ from a given formula $S$ is a finite sequence $C_1, C_2, ..., C_n = C$ of clauses such that eac $C_i$ is either a member of $S$ or a resolvent of clauses $C_j, C_k$ for $j, k < i$. If there is such a deduction, we say that $C$ is (resolution) provable from $S$ and write $S \vdash_{\mathcal{R}} C$. A deduction of $\square$ from $S$ is called a (resolution) refutation of S. If there is such a deduction we say that $S$ is (resolution) refutable and write $S \vdash_{\mathcal{R}} \square$.

- Prove the case of entries with $n+1$ connectives, i.e., $\mathcal{V}$ agrees with all entries, *with $n+1$ connectives*, of $P$.

  For every such entry $E$, with $n+1$ connectives, of $P$,
  since it has $n+1$ connectives, it must be of one of the forms:
  $T(\alpha_1 \vee \alpha_2), T(\alpha_1 \wedge \alpha_2), T(\alpha_1 \rightarrow \alpha_2), T(\alpha_1 \leftrightarrow \alpha_2), T(\neg\alpha_1)$,
  $F(\alpha_1 \vee \alpha_2), F(\alpha_1 \wedge \alpha_2), F(\alpha_1 \rightarrow \alpha_2), F(\alpha_1 \leftrightarrow \alpha_2)$, or $F(\neg\alpha_1)$
  where $\alpha_1$ (and $\alpha_2$ respectively) has *at most* $n$ connectives.
  We prove by cases.
  Case 1. $E = T(\alpha_1 \vee \alpha_2)$.
  Since $\tau$ is finished, $P$ is finished and thus $E$ is reduced.
  By definition of *reduced*, $T(\alpha_1)$ or $T(\alpha_2)$ must occur on $P$.
  Case 1.1 $T(\alpha_1)$ occurs on $P$. By IH, $\mathcal{V}$ agrees with $T(\alpha_1)$, and thus $\mathcal{V}(\alpha_1) = T$. Therefore, $\mathcal{V}(T(\alpha_1 \vee \alpha_2)) = T$, hence, $\mathcal{V}$ agrees with $E$.
  Case 1.2 $T(\alpha_2)$ occurs on $P$.
  We can prove, similarly to case 1.1, that $\mathcal{V}$ agrees with $E$.
  $\mathcal{V}$ agrees with $E$, by cases 1.1 to 1.2.
  Case 2 to 10. We can prove similarly,
  $\mathcal{V}$ agrees with $E$.
  Therefore, $\mathcal{V}$ agrees with $E$, by cases 1 to 10.
  $\mathcal{V}$ agrees with $E$.

1. Consider the formula $S = \{C_1, C_2\}$ is satisfiable — By given statement
2. Consider $C = C_1' \cup C_2'$ — By given statement $C$ is the resolvent of $C_1$ and $C_2$
3. $C_1 = \{l\} \sqcup C_1'$, $C_2 = \{\bar{l}\} \sqcup C_2'$ — By (2)
4. There exists an assignment $\mathcal{A}$ such that $\mathcal{A} \cap C \neq \emptyset$ and satisfies S and C, i.e $\mathcal{A} \models S, \mathcal{A} \models C$ — By (1) and (2).
5. $\mathcal{A}$ satisfies $\{C_1, C_2\}$, so $\mathcal{A} \models \{C_1, C_2\}$ satisfies each of $C_1$ and $C_2$ — By (4)
6. $\mathcal{A}$ satisfies each of $C_1$ and $C_2$, i.e $\mathcal{A} \models C_1, \mathcal{A} \models C_2$ — By (5)
7. Let $l \notin \mathcal{A}$
8. As, $C_1 = \{l\} \sqcup C_1'$, we have $\mathcal{A} \models C_1'$ - $\mathcal{A} \cap C_1' \neq \emptyset$ — By (7)
9. $\mathcal{A} \models C$ — By (2) and (8)
10. Let $\bar{l} \notin \mathcal{A}$
11. As, $C_2 = \{\bar{l}\} \sqcup C_2'$, so $\mathcal{A} \models C_2'$ - $\mathcal{A} \cap C_2' \neq \emptyset$ — By (10)
12. Hence, we have $\mathcal{A} \models C$ — By (2) and (10)
13. Hence, we have $\mathcal{A} \models C$ — By (9) and (12)
14. $C$ is satisfiable — By (13)
15. As formula $S = \{C_1, C_2\}$ is satisfiable and $C$ is the resolvent of $C_1$ and $C_2$, we prove that $C$ is satisfiable — By (14)

QED

**Definition 8.1** (i) A literal $l$ is a propositional letter $p$ or its negation $\neg p$. If $l$ is $p$ or $\neg p$, we write $\bar{l}$ for $\neg p$ or $p$, respectively. The propositional letters are also called positive literals and their negati negative literals.

(ii) A clause $C$ is a finite set of literals (which you should think of as th disjunction of its elements). As we think of $C$ as being true iff one c its elements is true, the empty clause $\square$ is always false - it has no tr element.

(iii) A formula $S$ is a (not necessarily finite) set of clauses (which you should think of as the conjunction of its elements). As we think of a formula $S$ as being true if everyone of its elements is true, the empt formula $\emptyset$ is always true - it has no false element.

(iv) An assignment $\mathcal{A}$ is a consistent set of literals, i.e., one not containi both $p$ and $\neg p$ for any propositional letter $p$. (This, of course, is just the (partial) truth assignment in which those $p \in \mathcal{A}$ are assigned $T$ a those $q$ with $\bar{q} \in \mathcal{A}$ are assigned $F$.) A complete assignment is one containing $p$ or $\neg p$ for every propositional letter $p$. It corresponds to what we called a truth assignment in Definition 3.1.

(v) $\mathcal{A}$ satisfies $S$, $\mathcal{A} \models S$, iff $\forall C \in S(C \cap \mathcal{A} \neq \emptyset)$, i.e., the valuation induced by $\mathcal{A}$ makes every clause in $S$ true.

(vi) A formula S is (un)satisfiable if there is an (no) assignment $\mathcal{A}$ that satisfies it.