27.11.2023

Semantic tableau method

1.1) Decide the type of our formula.

Decomposition rules

α rules: (conj. formulas)

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

β rules (disj. formulas)

$$\begin{array}{ccc} & A \vee B & \\ A & & B \end{array} \qquad \begin{array}{ccc} & \overset{\neg A \vee B}{A \to B} & \\ \neg A & & B \end{array}$$

$A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$

$A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$

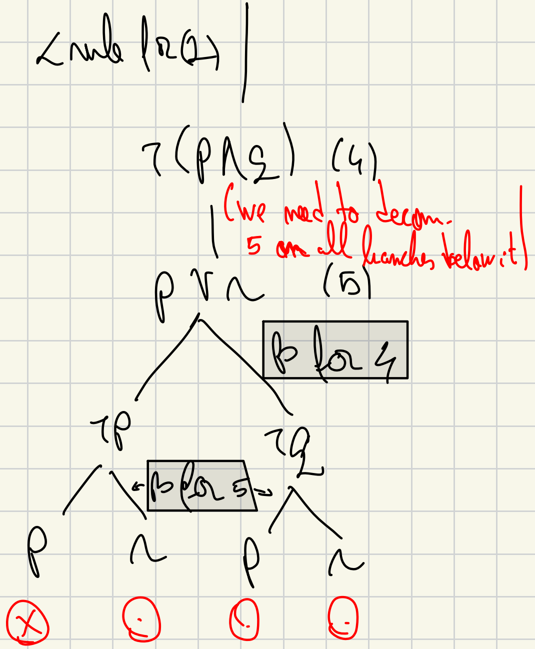
Theoretical results

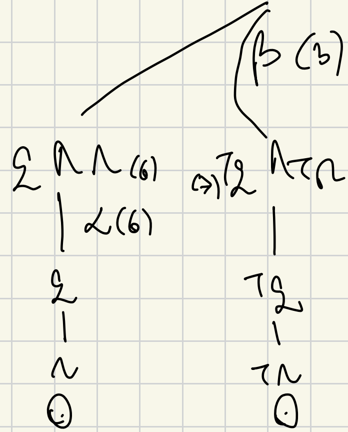
1) A branch is $\boxed{\text{closed}}$ ⊗ if it contains a pair of opposite literals $(p, \neg p)$
2) U is consistent if its semantic tabl. is open, at least one open branch.
3) The open branches of the sem. tabl. of U provide the models of U.
4) U is inconsistent when if its semantic tabl. closed (all branches are closed)
5) U is valid (taut.) if $\neg U$ has a closed sem. tableau

$$U_1 = (p \wedge q) \vee (\neg p \wedge \neg r) \to (q \leftrightarrow r) \quad (1)$$

β rules for (1)

$$\neg((p \wedge q) \vee (\neg p \wedge \neg r)) \ (2) \qquad (3)\ (q \leftrightarrow r) \equiv (q \wedge r) \vee (\neg q \wedge \neg r)$$





⊗ symbolises inconsistency

☺ symbolises consistency

- independency law:

$p \vee p \equiv p$

- absorbtion law:

$r \vee (\neg p \wedge r) \equiv r$

DNF of U_1 can be written as a disjunction of all branches. A branch will be a cube!

$$\mathrm{DNF}(U_1) = (\neg p \wedge p) \vee (\neg p \wedge r) \vee (\neg q \wedge p) \vee (\neg q \wedge r) \vee (q \wedge r) \vee (\neg q \wedge \neg r) \equiv$$

$$\equiv (r \wedge (q \vee \neg q)) \vee (\neg q \wedge (r \vee \neg r)) \vee (\neg p \wedge r) \vee (\neg q \wedge p)$$

$$\equiv r \lor \neg q \lor (\neg p \land r) \lor (\neg q \land p)$$

$$\equiv r \lor \neg q \quad \text{(simplified DNF)}$$

r: Cube 1 $\qquad$ $\neg q$: Cube 2

Cube 1: $r \equiv T$ provides 4 models

$$i_{1,2,3,4} : \{p, q, r\} \to \{T, F\}$$

$i_1(p) = T$	$i_1(q) = T$	$i_{1,2,3,4}(r) = T$
$i_2(p) = T$	$i_1(q) = F$	
$i_3(p) = F$	$i_1(q) = T$	
$i_4(p) = F$	$i_1(q) = F$	

Cube 2: $\neg q \equiv T \iff q \equiv F$ provides 4 models

$i_5(p) = T$	$i_{5,6,7,8}(q) = F$	$i_5(r) = T$
$i_6(p) = T$		$i_6(r) = F$
$i_7(p) = F$		$i_7(r) = T$
$i_8(p) = F$		$i_8(r) = F$

- $i_2 = i_5$
- $i_4 = i_7$

Conclusion: U_1 has a complete and open sem. tabl. with one closed branch $(p, \neg p)$ and 5 open branches, so it is consistent.

U_1 has 6 models: $i_1(U_1) = i_2(U_1) = i_3(U_1) =$
$= i_4(U_1) = i_6(U_1) = i_7(U_1) = T$.

2.1) Prove distribution of '$\to$' over '$\wedge$'

$$U = (p \to q \wedge r) \to (p \to q) \wedge (p \to r)$$

$$\neg U = \neg((p \to q \wedge r) \to (p \to q) \wedge (p \to r)) \quad (1) \checkmark$$

$\alpha(1)$

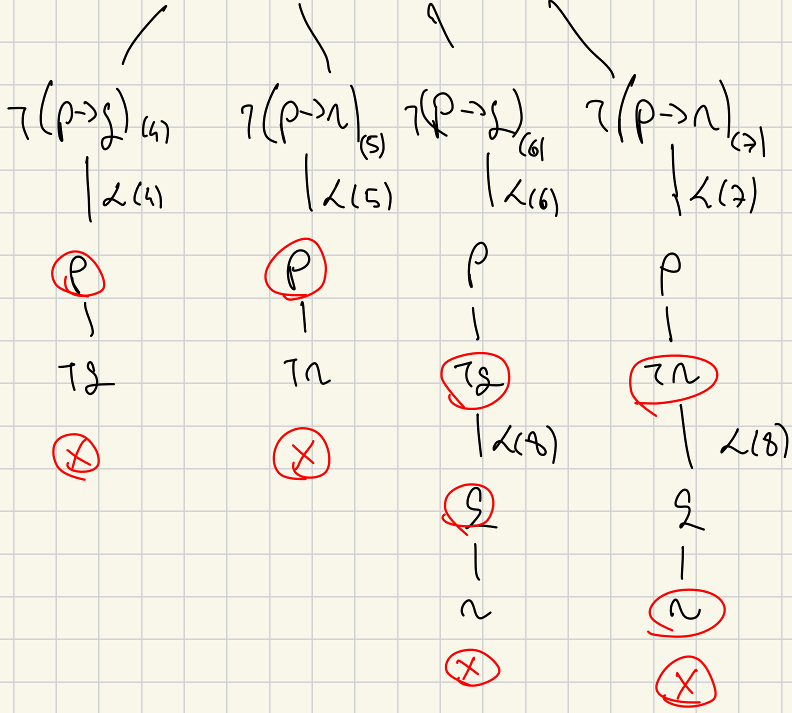
$$p \to q \wedge r \quad (2)$$

$$\neg((p \to q) \wedge (p \to r)) \quad (3)$$

$\beta(2)$

$\neg p$ $\qquad$ $q \wedge r$ (8)

$\beta(3)$ $\qquad$ $\beta(3)$



C: $\neg U$ has a closed sem. tab. with 4 closed branches $(p, \neg p)$ $(p, \neg p)$ $(q, \neg q)$ $(r, \neg r)$, so $\neg U$ is inconsistent. $\Longrightarrow$ U is $\boxed{\text{VALID}}$.

Theoretical results

6. $U_1, \dots, U_n \models V$ if $U_1 \wedge U_2 \wedge \dots \wedge U_n \wedge \neg V$ has a closed tableau.

7. The open branches of the semantic tableau of $\neg U$ provide the models of $\neg U$, which are the anti-models of U.

3.1) $\overbrace{p \to (\neg q \vee r \wedge s)}^{U_1}, \overbrace{p}^{U_2}, \overbrace{\neg s}^{U_3} \models \overbrace{\neg q}^{V}$

$U_1 \wedge U_2 \wedge U_3 \wedge \neg V$ (1)

| α(1)

$p \to (\neg q \vee r \wedge s)$ (2)

|

p

|

$\neg s$

|

q

β(2)

- $\neg p$
  - X
- $\neg q \vee r \wedge s$ (3)
  - β(3)
    - $\neg q$
      - X
    - $r \wedge s$ (4)
      - | α(4)
      - r
      - |
      - s
      - X

$U_1 \wedge U_2 \wedge U_3 \wedge \neg V$ has a closed semantic tableau with 3 closed branches $(p, \neg p), (q, \neg q), (s, \neg s)$ $\Longrightarrow$

Th. results 6 ⇒ $U_1, U_2, U_3 \models U$

Predicate-logic: 6.1) $H_1, H_2, H_3, H_4 \models C$?

H_1: Any CS student likes logic and likes any programming language.

H_2: Someone who likes logic is a CS student or a philosophy student.

H_3: Java is a programming language.

H_4: John does not like Java but does like logic.

C: John is a philosophy student but is not a CS student.

prog. lang.

H_1: $(\forall x)(\forall y)\big((CS(x) \wedge pl(y)) \to likes(x, logic) \wedge likes(x, y)\big)$

H_2: $(\forall x)(likes(x, logic) \to CS(x) \vee PS(x))$

H_3: $pl(Java)$

H_4: $\neg likes(John, Java) \wedge likes(John, logic)$

C: $PS(John) \wedge \neg CS(John)$

$H_1 \wedge H_2 \wedge H_3 \wedge H_4 \wedge \neg C$ (1)

| α (1)

$H_1: (\forall x)(\forall y)(CS(x) \wedge pl(y) \rightarrow likes(x, logic) \wedge likes(x, y))$ (2)

|

$H_2: (\forall x)(likes(x, logic) \rightarrow CS(x) \vee PS(x))$ (3)

|

H_3: $pl(Java)$

|

H_4: $\neg likes(John, Java) \wedge likes(John, logic)$ (4) same rule

|

$\neg C$: $\neg PS(John) \vee CS(John)$ (5)

| α (4)

$\neg likes(John, Java)$

|

$likes(John, logic)$

| γ(3) (John used for instantiation)

$likes(John, logic) \rightarrow CS(John) \vee PS(John)$ (6)

|

(3)

$(\forall x)\ U_{(x)}$

| $\boxed{\gamma \text{ rule}}$

$U_{(c_1)}$

|

$U_{(c_2)}$

|

$\vdots$

|

$U_{(c_n)}$

|

$(\forall x)\ U_{(x)}$

$\beta(6)$

$\neg likes(John, logic)$ ✗ $\qquad CS(John) \vee PS(John)$ (7)

$\mid \gamma(2)\ (x \leftarrow John)$

$(\forall y)\ CS(John) \wedge pl(y) \rightarrow likes(John, logic) \wedge likes(John, y)$ (8)

$\mid \gamma(8)\ (y \leftarrow Java)$

$CS(John) \wedge pl(Java) \rightarrow likes(John, logic) \wedge likes(John, Java)$ (9)

$\mid$

(7)

$\beta(9)$

$\neg(CS(John) \wedge pl(Java))$ (11) $\qquad likes(John, logic) \wedge likes(John, Java)$ (10)

$\beta(11) \qquad\qquad \mid \alpha(10)$

$\neg CS(John) \qquad \neg pl(Java)$ ✗ $\qquad likes(John, logic)$

$\qquad\qquad\qquad\qquad\qquad\qquad likes(John, Java)$ ✗

$\beta(5)$

$\neg PS(John) \qquad CS(John)$

$\beta(7)$

$CS(John)$ ✗ $\qquad PS(John)$ ✗

C: $H_1 \wedge H_2 \wedge H_3 \wedge H_4 \wedge \neg C$ has all 5 branches closed $\Rightarrow H_1, H_2, H_3, H_4 \models C$ by reductio ad absurdum.