CENG 424
Fall 2024
Homework 2

Yancı, Baran
e2449015@ceng.metu.edu.tr

November 5, 2023

Q1)

p	q	r	p ∧ q	¬ q	p	q ∨¬ q	p ∧ q → r	r
False	False	False	True	True	False	True	False	False
False	False	True	True	True	False	True	True	True
False	True	False	True	False	False	True	False	False
False	True	True	True	False	False	True	True	True
True	False	False	False	True	True	True	True	False
True	False	True	False	True	True	True	True	True
True	True	False	False	False	True	True	True	False
True	True	True	False	False	True	True	True	True

Premises are marked dark red and the resulting sentence is marked dark blue. As can be seen, the row marked with yellow is the only row where the premises are true and the resulting sentence is false. Therefore, the premises do not entail the resulting sentence.

Q2)

$$
\begin{aligned}
&p \to q && Premise && (1)\\
&q \to r && Premise && (2)\\
&(q \to r) \to (p \to (q \to r)) && II && (3)\\
&p \to (q \to r) && MP(2,3) && (4)\\
&(p \to (q \to r)) \to ((p \to q) \to (p \to r)) && ID && (5)\\
&(p \to q) \to (p \to r) && MP(4,5) && (6)\\
&(p \to r) && MP(1,6) && (7)\\
&(p \to r) \to ((p \to \neg r) \to \neg p) && CR && (8)\\
&(p \to \neg r) \to \neg p && MP(7,8) && (9)
\end{aligned}
$$

Q3)

$$
\begin{array}{lrr}
\neg\neg p & \mathit{Premise} & (1) \\
(\neg p \to \neg p) \to ((\neg p \to \neg\neg p) \to p) & CR & (2) \\
\neg\neg p \to (\neg p \to \neg\neg p) & II & (3) \\
\neg p \to \neg\neg p & MP(1,3) & (4) \\
(\neg p \to ((\neg p \to \neg p) \to \neg p)) \to ((\neg p \to (\neg p \to \neg p)) \to (\neg p \to \neg p)) & ID & (5) \\
(\neg p \to ((\neg p \to \neg p) \to \neg p)) & II & (6) \\
(\neg p \to (\neg p \to \neg p)) \to (\neg p \to \neg p) & MP(5,6) & (7) \\
(\neg p \to (\neg p \to \neg p)) & II & (8) \\
(\neg p \to \neg p) & MP(7,8) & (9) \\
(\neg p \to \neg\neg p) \to p & MP(2,9) & (10) \\
p & MP(4,10) & (11)
\end{array}
$$

Q4)

$$
\begin{array}{lrr}
(p \vee q \to r) \to (p \to (q \to r)) & \mathit{Sentence} & (1) \\
\neg(p \vee q \to r) \vee (p \to (q \to r)) & \mathit{Implications\ out} & (2) \\
\neg(\neg(p \vee q) \vee r) \vee (p \to (q \to r)) & \mathit{Implications\ out} & (3) \\
\neg(\neg(p \vee q) \vee r) \vee (\neg p \vee (q \to r)) & \mathit{Implications\ out} & (4) \\
\neg(\neg(p \vee q) \vee r) \vee (\neg p \vee (\neg q \vee r)) & \mathit{Implications\ out} & (5) \\
\neg((\neg p \wedge \neg q) \vee r) \vee (\neg p \vee (\neg q \vee r)) & \mathit{Negations\ In} & (6) \\
(\neg(\neg p \wedge \neg q) \wedge \neg r) \vee (\neg p \vee (\neg q \vee r)) & \mathit{Negations\ In} & (7) \\
((p \vee q) \wedge \neg r) \vee (\neg p \vee (\neg q \vee r)) & \mathit{Negations\ In} & (8) \\
((p \vee q) \wedge \neg r) \vee \neg p \vee \neg q \vee r & \mathit{Distribution} & (9) \\
(((p \vee q) \vee \neg p) \wedge (\neg p \vee \neg r)) \vee \neg q \vee r & \mathit{Distribution} & (10) \\
(((p \vee q) \vee \neg p \vee \neg q) \wedge (\neg p \vee \neg r \vee \neg q)) \vee r & \mathit{Distribution} & (11) \\
(((p \vee q) \vee \neg p \vee \neg q \vee r) \wedge (\neg p \vee \neg q \vee \neg r \vee r)) & \mathit{Distribution} & (12) \\
(p \vee q \vee \neg p \vee \neg q \vee r) \wedge (\neg p \vee \neg q \vee \neg r \vee r) & \mathit{Distribution} & (13) \\
\hline
\{p, q, \neg p, \neg q, r\} & \mathit{Operators\ Out} & (14) \\
\{\neg p, \neg q, \neg r, r\} & \mathit{Operators\ Out} & (15) \\
\hline
\{r, \neg r\} & & (16) \\
\{p, \neg p\} & & (17) \\
\{q, \neg q\} & & (18)
\end{array}
$$

As can be seen, the only literals in the given sentence p, q and r can be given any value and the sentence will still be valid. Therefore, the sentence is valid.

Q5)

```
function DPLL( ϕ)
    // unit propagation:
    while there is a unit clause {l} in ϕ do
        ϕ  ← unit-propagate(l,ϕ);
    // pure literal elimination:
    while there is a literal l that occurs pure in ϕ do
        ϕ  ← pure-literal-assign(l,ϕ);
    // stopping conditions:
    if  ϕ is empty then
        return true;
    if  ϕ contains an empty clause then
        return false;
    // DPLL procedure:
    l ← choose-literal(ϕ);
    return DPLL(ϕ ∧ {l}) or DPLL(ϕ ∧ {¬ l});
```

(taken from Wikipedia.)

$$\{\neg p, q, s\} \qquad \textit{Clause } \#1 \tag{1}$$
$$\{p, s, t\} \qquad \textit{Clause } \#2 \tag{2}$$
$$\{p, s, \neg t\} \qquad \textit{Clause } \#3 \tag{3}$$
$$\{p, s, t\} \qquad \textit{Clause } \#4 \tag{4}$$
$$\{p, \neg s, \neg t\} \qquad \textit{Clause } \#5 \tag{5}$$
$$\{p, \neg s, t\} \qquad \textit{Clause } \#6 \tag{6}$$
$$\{p, q, \neg s\} \qquad \textit{Clause } \#7 \tag{7}$$
$$\{p, \neg q, s\} \qquad \textit{Clause } \#8 \tag{8}$$

$$p \qquad \textit{Set } p \equiv \textit{True} \tag{9}$$
$$T \qquad \textit{Clauses } \#2, \#3, \#4, \#5, \#6, \#7, \#8 \tag{10}$$
$$\{s, q\} \qquad \textit{Clause } \#1 \tag{11}$$

$$q \qquad \textit{Set } q \equiv \textit{True} \tag{12}$$
$$T \qquad \textit{Clauses } \#1, \#2, \#3, \#4, \#5, \#6, \#7, \#8 \tag{13}$$
$$\{\} \tag{14}$$

As can be seen above, the set of clauses is satisfiable.
Here are the steps of the algorithm:

1. Randomly pick p as the literal and set $p \equiv \textit{True}$ (`choose-literal`).
2. Remove all clauses containing p. The removed clauses are #2, #3, #4, #5, #6, #7 and #8 (`unit-propagate`). The only clause that remains is #1, which is now $\{s, q\}$.

3. Our set of clauses contains pure literals q and s. Let us set $q \equiv True$ and remove all clauses containing q. There is only one clause that contains q, which is #1, the only remaining clause in our set of clauses. (`pure-literal-assign`)

4. Since there are no clauses left, the set of clauses is satisfiable (`if  (*$\phi$*) is empty then return true`).

Any interpretation that satisfies the set of clauses is $\{p \equiv True, q \equiv True\}$. The literals s and t can be given any value and the set of clauses will still be satisfiable, as long as p and q are set to $True$.