4CCS1ELA - Elementary Logic with Applications
Programming with Logic I:
Propositional Definite Clause Programming

Tutorial List 6 Solutions

Question 1

Solution:

See Slides

Question 2

Solution:

See Slides

Question 3

Solution:

1. $\neg P \rightarrow (\neg Q \wedge R)$

- First, we transform it into CNF using equivalence laws (see lecture slide 15).

$\neg\neg P \vee (\neg Q \wedge R)$ (we applied law 1)
$P \vee (\neg Q \wedge R)$ (we applied law 5)
$(P \vee \neg Q) \wedge (P \vee R)$ (we applied law 6)

- Second, we identify definite clauses. Not all clauses are definite clauses (a definite clause has exactly one positive literal)!

$(P \vee \neg Q)$ is a definite clause, but $(P \vee R)$ is not (two positive literals!)

- Third, we represent the definite clause as definite rules.

(P ∨¬Q) can be represented as rule **Q** → **P**

2. ¬(S ∨ R) ∧ T

- First, we transform it into CNF using equivalence laws.

(¬S ∧¬R) ∧ T (we applied law 3)
¬S ∧¬R ∧ T (associativity – getting rid of brackets)

- Second, we identify definite clauses. Not all clauses are definite clauses!

T is a definite clause, but ¬S and ¬R are not (they have no positive literal!)

- Third, we represent the definite clause as definite rules.

T can be represented as the rule →***T***

Question 4

Solution:

¬(R →¬T)

- First, we transform it into CNF using equivalence laws.

¬(¬R ∨¬T)
¬¬R ∧ ¬¬T
R ∧ T

- Second, we identify definite clauses.

R and *T* are both definite clauses.

- Third, we represent the definite clause as definite rules.

R can be represented as the rule →***R***
T can be represented as the rule →***T***