1st Sem. 2021-22
CS F214 Logic in Computer Science
End Sem (Closed Book)

================================================================================

Time: 3 hours Marks: 100

================================================================================

**Instructions:**

1. **All subparts of the questions should be answered collectively.**
2. **Questions where Boolean answers are sought with an explanation, provide the Boolean answer explicitly following the explanation. The explanation will be considered only if the Boolean answer is correct.**
3. **Please cross the unused sheets. If you answer any question more than once, the answer will not be considered for evaluation.**
4. **Be concise and to the point in answering.**

--------------------------------------------------------------------------------

**Q1.** Consider the following formulas:

$\phi1 = p \rightarrow ((q \vee r) \wedge \neg(q \wedge r))$
$\phi2 = p \rightarrow ((s \vee t) \wedge \neg(t \wedge s))$
$\phi3 = s \rightarrow q$
$\phi4 = \neg r \rightarrow t$
$\phi5 = t \rightarrow s$

Using resolution, prove that $\neg p$ is the logical consequence of $\{\phi1, \phi2, \phi3, \phi4, \phi5\}$. **[10M]**

First, we convert the premises to their CNF form and convert to CNF the negation of the conclusion

$$\begin{aligned}\phi_1 = p \to ((q \vee r) \wedge \neg(q \wedge r)) &\equiv \neg p \vee ((q \vee r) \wedge (\neg q \vee \neg r)) \\ &\equiv (\neg p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r) \\ &\equiv \{\bar{p}qr, \bar{p}\bar{q}\bar{r}\}\end{aligned}$$

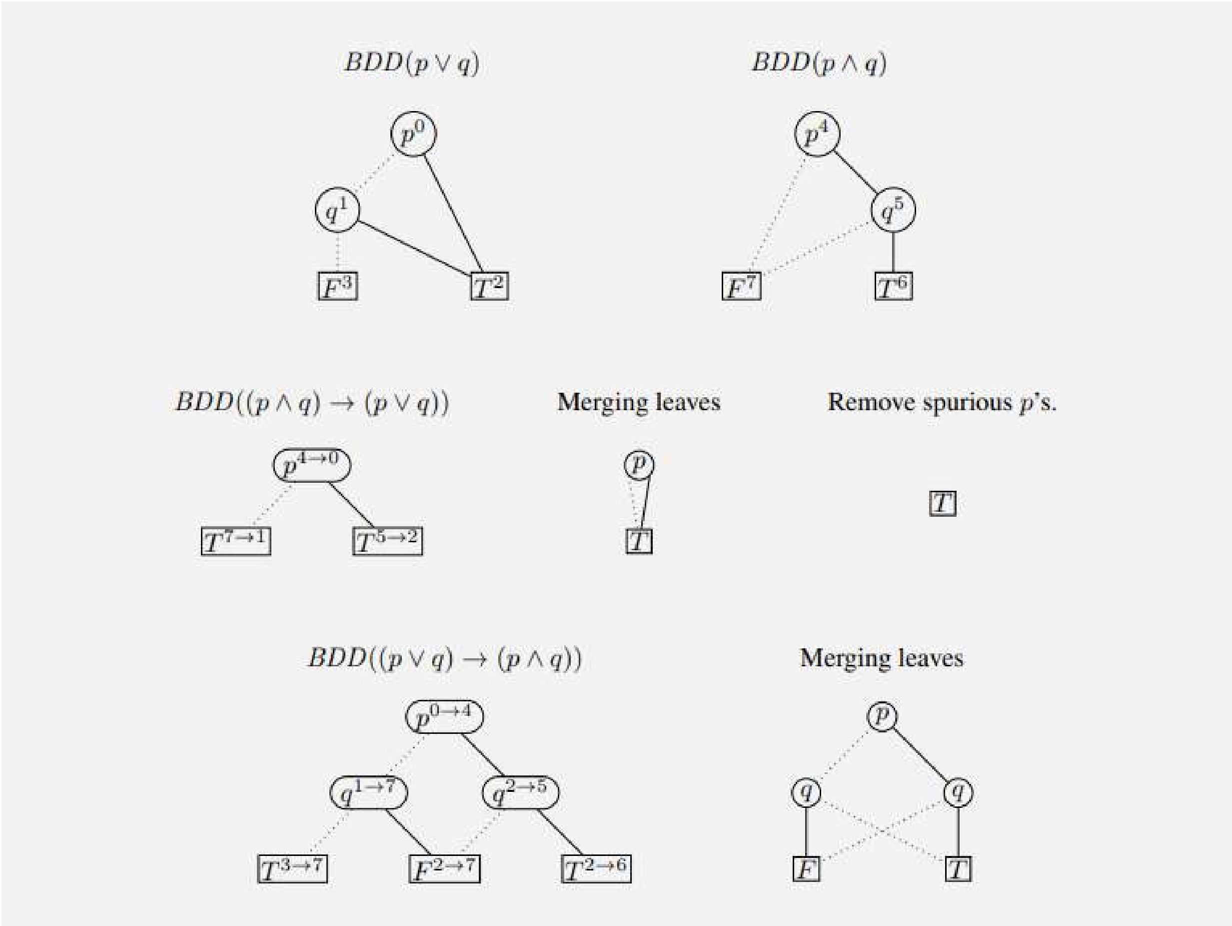
$$\begin{aligned}&\phi_2 = (\neg p \vee s \vee t) \wedge (\neg p \vee \neg s \vee \neg t) \equiv \{\bar{p}st, \bar{p}\bar{t}\bar{s}\} \\ &\phi_3 = \neg s \vee q \equiv \bar{s}q \\ &\phi_4 = r \vee t \equiv rt \\ &\phi_5 = \neg t \vee s \equiv \bar{t}s\end{aligned}$$

We apply the resolution procedure on the set $\{(1)\bar{p}qr, (2)\bar{p}\bar{q}\bar{r}, (3)\bar{p}st, (4)\bar{p}\bar{s}\bar{t}, (5)\bar{s}q, (6)rt, (7)\bar{t}s, (8)p\}$.

$$\begin{aligned}(9)\ \bar{p}\bar{q}t &= Res(2,6) \\ (10)\ \bar{p}\bar{q}s &= Res(7,9) \\ (11)\ \bar{p}\bar{q}\bar{t} &= Res(10,4) \\ (12)\ \bar{p}\bar{q}r &= Res(11,6) \\ (13)\ \bar{p}\bar{q} &= Res(12,2)\end{aligned}$$

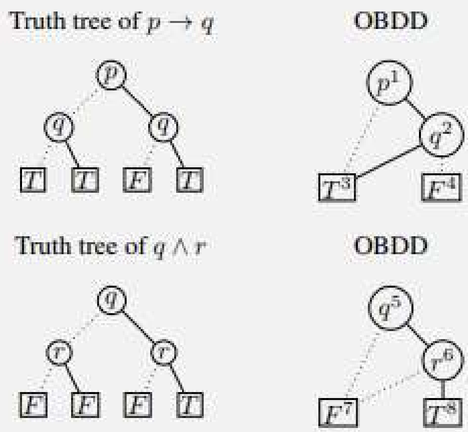
$$\begin{aligned}(14)\ \bar{p}s &= Res(3,7) \\ (15)\ \bar{p}q &= Res(14,5) \\ (16)\ \bar{p} &= Res(15,13) \\ (17)\ \square &= Res(16,8)\end{aligned}$$

**Q2**. Construct BDD((p ∨ q) → (p ∧ q)) from BDD(p ∧ q) and BDD(p ∨ q), using the ordering p < q **[10M]**

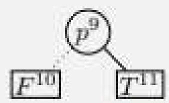



**Q3.** Show (p → q) ∨ ((q ∧ r) ∨ p) ≡ p ∨ ¬p by the construction of the reduced OBDDs of both the formulae. **[15M]**

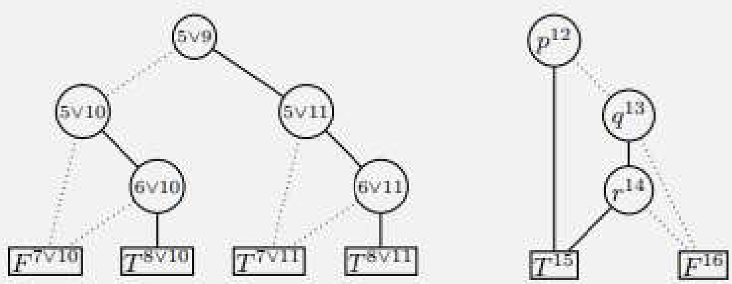
We use $p < q < r$ as ordering. First we construct the OBDDs of the subformulas $p \to q$ and $q \wedge r$ from the truth trees:



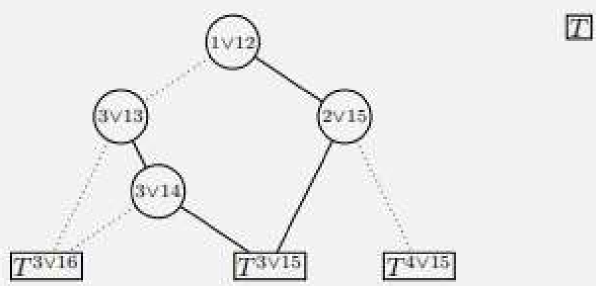

Furthermore, the OBDD of $p$ is



The composition for $(q \wedge r) \vee p$ gives rise to the tree on the left hand side, resulting in the OBDD on the right hand side:



Finally, the composition for $(p \to q) \vee ((q \wedge r) \vee p)$ gives rise to the tree on the left hand side, resulting in the OBDD on the right hand side:



Since this last OBDD equals that of $p \vee \neg p$, we are done.

**Q4**. A fairness constraint is imposed on (the scheduler of) the system that it fairly selects the process to be executed next. One can have three typical fairness constraints.

a. Absolute Fairness, Impartiality: every process should be executed infinitely often
   **GF exi**

b. Strong Fairness: Every process that is infinitely often enabled should be executed infinitely often in a state where it is enabled
(GF eni) ⇒ (GF(eni ∧ exi)).

c. Weak Fairness: Every process that is almost always enabled should be executed infinitely often
(FG eni) ⇒ (GF exi)

Formulate each of these fairness constraint in LTL. Use the following atomic propositions: $ex_i$ denotes process execution. $en_i$ denotes process being enabled.

Does absolute Fairness imply Strong Fairness?

Does Strong Fairness imply absolute fairness? **[10M]**

**Absolute Fairness does not imply Strong Fairness, nor vice versa**

**Q5**. Consider the TS below and the CTL formula ∃◊∀□c. Decide whether the formula is satisfied over the TS. Also, outline (in a numbered manner) the CTL model checking steps used to obtain the answer. **[15M]**

First consider the formula $\Phi_1 = \exists\Diamond\,\forall\Box\, c$:
It can be expressed equivalently in ENF:

$$\begin{aligned}\Phi_1 &= \exists\Diamond\,\forall\Box\, c\\ &\equiv \exists(\mathit{true}\,\mathsf{U}\,\forall\Box\, c)\\ &\equiv \exists(\mathit{true}\,\mathsf{U}\,\neg\exists\Diamond\,\neg c)\\ &\equiv \exists(\mathit{true}\,\mathsf{U}\,\neg\exists(\mathit{true}\,\mathsf{U}\,\neg c)).\end{aligned}$$

The bottom-up computation of the satisfaction sets yields:

- $Sat(\mathit{true}) = S$
- $Sat(c) = \{s_2, s_3, s_4\}$
- $Sat(\neg c) = \{s_0, s_1\}$
- $Sat(\exists\,(\mathit{true}\,\mathsf{U}\,\neg c))$ yields a backward search as follows:
  * $E = T = Sat(\neg c) = \{s_0, s_1\}$
  * Choose $s_0$: As $Pre(s_0) = \varnothing$, we get $E = \{s_1\}$
  * Choose $s_1$: $Pre(s_1) = \{s_0\}$. But $s_0 \notin Sat(\mathit{true}) \setminus T$ (i.e., it has already been visited), we get $E = \varnothing$

$\Longrightarrow T = \{s_0, s_1\}$.

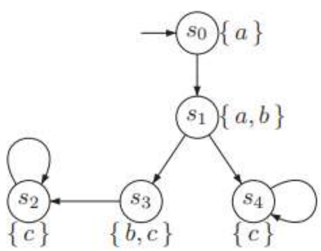
$Sat(\neg\exists\,(true\,\mathsf{U}\,\neg c)) = \{s_2, s_3, s_4\}$

$Sat(\exists\,(true\,\mathsf{U}\,\neg\exists\,(true\,\mathsf{U}\,\neg c)))$ again yields a backward search:

* $E = T = Sat(\neg\exists\,(true\,\mathsf{U}\,\neg c)) = \{s_2, s_3, s_4\}$
* Choose $s_2$: $Pre(s_2) = \{s_2, s_3\}$, but all predecessors are already in $T$. $\{s_3, s_4\}$
* Choose $s_3$: $Pre(s_3) = \{s_1\}$. Here we have $s_1 \notin T$ and $s_1 \in Sat(true)$. $T = T \cup \{s_1\} = \{s_1, s_2, s_3, s_4\}$ and $E = E \cup \{s_1\} = \{s_1, s_4\}$ ($s_3$ gets rem $E$)
* Choose $s_1$: $Pre(s_1) = \{s_0\}$. Again $s_0 \in Sat(true)\setminus T$ and therefore $T =$ : $S$ and $E = E \cup \{s_0\} = \{s_0, s_4\}$
* Choose $s_0$: $Pre(s_0) = \varnothing$ and we directly continue with $s_4$:
* Choose $s_4$: $Pre(s_4) = \{s_1, s_4\}$ but $s_1, s_4$ are already in $T$. Therefore we $E = \varnothing$.

$\Longrightarrow T = \{s_0, s_1, s_2, s_3, s_4\}$

:efore, we have $Sat(\Phi_1) = \{s_0, s_1, s_2, s_3, s_4\}$ and $I \subseteq Sat(\Phi_1) \Longrightarrow TS \models \Phi_1$.




**Q6.** Write the following statements in predicate logic:
(a) A natural number is a prime if it has no factors other than 1 and itself.
(b) Given any natural number, there is a larger prime number.
(c) Any natural number can be written as a product of primes.
You can use predicates of your choice. [**2x3 = 6M**]

**Q7**. Prove the sequent: $\exists Y\ \forall X\ \big(a(X) \to b(Y)\big) \vdash \forall X\ \exists Y\ \big(a(X) \to b(Y)\big)$ [**4M]**

**Q8**. Use predicates: $\boldsymbol{add}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$ denoting $\boldsymbol{X}+\boldsymbol{Y}=\boldsymbol{Z}$**;** $\boldsymbol{mult}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$ denoting $\boldsymbol{X}*\boldsymbol{Y}=\boldsymbol{Z}$**;** to define $\boldsymbol{pow}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$ to denote $\boldsymbol{X}^{\boldsymbol{Y}} = \boldsymbol{Z}$**;** and to define $\boldsymbol{divof}(\boldsymbol{X},\boldsymbol{Y})$ to denote X is a divisor of Y. **[5M]**

**Q9**. Define proof rules in natural deduction style for "introduction" and "elimination" of the following operators. [**5M**]

(a) **NOR** operator
(b) **NAND** operator

**Q10**.(a) Prove the sequent **pVq, p→q, q →p,** $\neg(\mathbf{p}\wedge\mathbf{q})$ **|-- r** **[5M]**
(b) Prove the sequent **|-- (p V (p →q))** **[5M]**

**Q11.** Prove the sequent $\forall$**x** $\exists$**y R(x, y) |--** $\forall$**x** $\neg\forall$**y** $\forall$**z** $\neg$ **(R(x, y)** $\wedge$ **R(y, z))** **[10M]**