Lógica Computacional
Tarea Formal 1

Prof. Miguel Carrillo
Ayte: Estefanía Prieto Ayte: Mauricio Esquivel

Fecha de entrega: 19 de Septiembre 2018
La evaluación de la tarea formal es sobre 100 puntos

1. Define recursivamente la función `icd` que se especifica como sigue: icd toma como entrada una fórmula ϕ y devuelve la fórmula resultante al intercambiar en ϕ todas las conjunciones por disyunciones, y las disyunciones por conjunciones respectivamente. Por ejemplo se debe cumplir que:

$$icd(p \wedge (q \vee \neg r) \to \neg(r \vee s) \wedge t) \ = p \vee (q \wedge \neg r) \to \neg(r \wedge s) \vee t$$

2. Verifica tu definición mostrando paso a paso el cálculo del ejemplo de arriba.

3. Define la función *atom(ϕ)* que, para $\phi \in PL$, devuelve el número de fórmulas atómicas ($\top, \bot$ *o variables*) en ϕ.

4. Demuestra que para cualquier fórmula $\phi \in PL$ se cumple que:

$$atom(\phi) \leq con(\phi) + 1$$

donde *con(ϕ)* es la función que devuelve el número de conectivos de ϕ.

5. Realiza las siguientes sustituciones uniformes eliminando los paréntesis innecesarios en el resultado:

   (a) $((q \vee r)\,[q, p := \neg p, s] \to (r \wedge \neg(r \leftrightarrow p)))\,[p, r, q := r \vee q, q \wedge p, s]$

   (b) $(u \vee t) \to (\neg r \leftrightarrow (u \leftrightarrow s))\,[r, u, t := u, t, r]$

6. Demuestra empleando la semántica de la lógica proposicional, que cada uno de los siguientes pares de fórmulas son equivalentes:

   (a) $(p \wedge q)$ *y* $\neg(p \to \neg q)$

   (b) $\neg(p \leftrightarrow q)$ *y* $(q \leftrightarrow \neg p)$

   (c) $(p \wedge (p \vee q))$ *y* p

7. Decide si las siguientes fórmulas son válidas, utilizando la transformación a la forma normal de conjunción.

   (a) $(p \lor q) \leftrightarrow (\neg p \to q)$

   (b) $((p \lor q) \land \neg(\neg p \lor q)) \leftrightarrow p$

8. Obtener la forma normal de conjunción de las siguientes fórmulas (mencionando la operación realizada en cada paso):

   (a) $((q \to r) \to q) \land (r \to q)$

   (b) $\neg p \land q \to p \land (r \to q)$

9. Obtener la forma normal de disjunción de las siguientes fórmulas (mencionando la operación realizada en cada paso):

   (a) $\neg(w \neg p) \lor \neg((\neg s \leftrightarrow w) \lor (p \land s))$

   (b) $\neg(p \to \neg r \to s \leftrightarrow \neg(\neg q \land (p \lor r)))$

10. Aplica el algoritmo de decibilidad para la satisfacibilidad de una fórmula de Horn vista en clase en las siguientes fórmulas.

    (a) $(p \land q \land w \to \bot) \land (t \to \bot) \land (r \to p) \land (\top \to r) \land (\top \to q) \land (u \to s) \land (\top \to u)$

    (b) $(p \land q \land w \to \bot) \land (t \to \bot) \land (r \to p) \land (\top \to r) \land (\top \to q) \land (r \land u \to w) \land (u \to s) \land (\top \to u)$

    (c) $(p \land q \land s \to p) \land (q \land r \to p) \land (p \land s \to s)$

    (d) $(p \land q \land s \to \bot) \land (q \land r \to p) \land (\top \to s)$

11. Define la función recursiva `impl_free` que recibe una fórmula $\phi \in PL$ y regresa una fórmula lógicamente equivalente a ϕ que resulta de quitar las implicaciones en ϕ.

    (a) Define la función `NNF` tal que recibe una fórmula ϕ y devuelve la fórmula *phi* en forma normal de negación.

    (b) Define la función `CNF` tal que recibe una fórmula ϕ y devuelve la fórmula ϕ en forma normal de conjunción.

    (c) Una vez definidas, ejecuta:

    `CNF (NNF (impl_free` $\neg(p \to (\neg(q \land (\neg p \to q)))$`)))`