Ayudante: Julián García (jgarcg@uc.cl)

Solución Ayudantía 1

IIC2213 - Lógica para Ciencia de la Computación

Problema 1

Sea EQ un conectivo ternario definido como $\text{EQ}(p, q, r) = 1$ si y sólo si $3 \cdot p - 2 \cdot (q + r) \geq 0$. Defina el conectivo EQ utilizando los conectivos $\wedge$, $\vee$ y $\neg$.

Solución:

Primero analizamos la tabla de verdad para la relacion $\text{EQ}(p, q, r)$

p	q	r	EQ(p,q,r)	3p-2(q+r)
0	0	0	1	0
0	0	1	0	-2
0	1	0	0	-2
0	1	1	0	-4
1	0	0	1	3
1	0	1	1	1
1	1	0	1	1
1	1	1	0	-1

Luego, podemos gracias a las formas normales encontrar una fórmula lógica para esta tabla de verdad. Si utilizamos DNF, debemos recorrer las filas que tienen como valuación 1:

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

En cambio, si utilizamos CNF, debemos recorrer las filas que tienen como valuación 0:

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

Problema 2

El conectivo lógico `NOR` es definido de la siguiente forma:

p	q	p `NOR` q
0	0	1
0	1	0
1	0	0
1	1	0

Demuestre que `NOR` es funcionalmente completo.

Solución:

Para demostrar que {`NOR`} es un conjunto funcionalmente completo, debemos intentar reconstruir los operadores de un conjunto que sepamos que es funcionalmente completo, a partir de sólo el operador `NOR`. Como sabemos que el conjunto $\{\neg, \wedge, \vee\}$ es funcionalmente completo, podemos intentar reconstruir esos operadores sólo mediante el operador `NOR`. Para verificar la igualdad, podemos visualizar las tablas de verdad de los operadores y confirmar que son idénticas.

Partiremos por el operador $\neg\phi$. Este lo podemos construir mediante la fórmula $\phi \,\texttt{NOR}\, \phi$. Lo revisamos con la tabla de verdad:

p	p	$p \,\texttt{NOR}\, p$	$\neg p$
0	0	1	1
1	1	0	0

Vemos que las tablas de verdad son idénticas, por lo tanto se puede construir el operador $\neg$ a partir solamente del operador NOR. Los otros operadores se construyen de manera similar: $\phi \vee \psi$ se construye como $(\phi \,\texttt{NOR}\, \psi) \,\texttt{NOR}\, (\phi \,\texttt{NOR}\, \psi)$ y $\phi \wedge \psi$ se construye como $(\phi \,\texttt{NOR}\, \phi) \,\texttt{NOR}\, (\psi \,\texttt{NOR}\, \psi)$.
Como construimos los operadores $\neg$, $\wedge$, $\vee$ a partir del operador NOR, entonces podemos decir que el operador NOR es funcionalmente completo.

Problema 3

Decimos que una fórmula φ está en 3-CNF si φ está en CNF y cada una de sus cláusulas contiene a lo más tres literales. Por ejemplo, $(p \vee q \vee \neg r) \wedge (\neg p \vee s)$ está en 3-CNF mientras que $(p \vee \neg q \vee \neg r \vee s)$ no está en 3-CNF.
Demuestre que existen fórmulas que no son equivalentes a ninguna fórmula en 3-CNF.
Solución:
Una fórmula que no es equivalente a ninguna otra en 3-CNF es la fórmula $(p \vee q \vee r \vee s)$, tomando $P = \{p, q, r, s\}$. Vayamos por casos:

1. Digamos que la primera cláusula de la fórmula tiene un sólo literal, y sin pérdida de generalidad digamos que ese literal es p. Luego, podemos revisar la tabla de verdad y darnos cuenta de que hay discrepancia en al menos una fila. Llamaremos la fórmula $\phi = (p \vee q \vee r \vee s)$ y $\phi_1 = (p) \wedge \ldots$:

p	q	r	s	ϕ	ϕ_1
0	0	0	1	1	0

2. Analizamos el caso donde la primera cláusula tiene dos literales, SPDG son p y q. Llamamos la fórmula $\phi_2 = (p \vee q) \wedge \ldots$:

p	q	r	s	ϕ	ϕ_2
0	0	0	1	1	0

3. Analizamos el caso donde la primera cláusula tiene tres literales, SPDG son p, q y r. Llamamos la fórmula $\phi_3 = (p \vee q \vee r) \wedge \ldots$:

p	q	r	s	ϕ	ϕ_2
0	0	0	1	1	0

Como analizamos todos los casos posibles de fórmulas en 3-CNF y ninguna es equivalente a la fórmula inicial, entonces podemos decir que las fórmulas en 3-CNF no pueden representar todas las fórmulas en CNF.

Problema 4

Dado $\Sigma \subseteq L(P)$ y $\alpha, \beta \in L(P)$, demuestre que $\Sigma \models \alpha \to \beta$ si y sólo si $\Sigma \cup \{\alpha\} \models \beta$.
Solución:
Demostraremos ambos lados de la doble implicancia:

De izquierda a derecha: Partimos de la premisa de que $\sigma(\Sigma) = 1 \Rightarrow \sigma(\alpha \Rightarrow \beta) = 1$. Si consideramos que $\sigma(\Sigma \cup \{\alpha\}) = 1$, entonces $\sigma(\Sigma) = 1$ y $\sigma(\alpha) = 1$. Luego como $\sigma(\Sigma) = 1$, por hipótesis inicial $\sigma(\alpha \Rightarrow \beta) = 1$. Finalmente, como $\sigma(\alpha) = 1$, entonces $\sigma(\beta) = 1$. Concluimos que $\Sigma \cup \{\alpha\} \models \beta$.

De derecha a izquierda: Partimos de la premisa de que $\sigma(\Sigma \cup \{\alpha\}) = 1 \Rightarrow \sigma(\beta) = 1$. Si consideramos que $\sigma(\Sigma) = 1$, entonces nos podemos situar en 2 casos:

1. $\sigma(\alpha) = 1$. En este caso, $\sigma(\Sigma \cup \{\alpha\}) = 1$. Luego por hipótesis de inducción, $\sigma(\beta) = 1$. Como $\sigma(\alpha) = 1 \Rightarrow \sigma(\beta) = 1$, entonces $\sigma(\alpha \Rightarrow \beta) = 1$.
2. $\sigma(\alpha) = 0$. Luego trivialmente $\sigma(\alpha \Rightarrow \beta) = 1$ (por tabla de verdad).

Concluimos que $\Sigma \models \alpha \Rightarrow \beta$.