Lógica Computacional, 2020-1

Facultad de Ciencias, UNAM
Práctica 02

Francisco Javier Tonatiuh Fuentes Juárez Kevin Ricardo Villegas Salvador

Roberto Carlos Uribe Cerda

5 de noviembre de 2019

1. Complicaciones.
   En la parte de la semantica hicimos nuevas declaraciones de tipo para poder llevar acabo una evaluación y en la parte de satisfacibilidad y en en validez llego a un StackOverFloat :(. y en la parte de las deducciones, en el ejercicio 1, utilizamos la equivalencias de $\neg p \equiv p \to \bot$.
   Ésto para poder llevar acabo la doble negación de p, ésto ya que no tenemos la introducción de la doble negación y se nos complico a la hora de querer derivarlo.

2. Preguntas.
   - *Describe la diferencia entre una fórmula satisfacible y una válida.*
     Una fórmula valida es aquella que para cualquier modelo dado, la fórmula tiene una valor True, es decir la fórmula es una tautología.
     En el caso de una fórmula satisfacible, es aquella que para al menos un valor obtiene un valor False, es decir la fórmula es una

contradicción o una contingencia.

- *Explica como se utiliza el algoritmo de índices complementarios para comprobar que las fórmulas en forma normal disyuntiva son satisfacibles.*
  En cada termino de la fórmula buscamos que no haya una literal complementaria, ya como como son conjunciones y sabemos que $p \wedge \neg p \equiv \perp$, entonces la fórmula sería Falsa, pero para poder clasificar la fórmula con insatisfacible, entonces tendríamos que llegar a que cada termino es Falso. Pero si encontramos una fórmula que no tiene literales complementarias, entonces el termino sería T, y sabemos que $T \vee f \equiv T$, entonces la fórmula es satisfacible.

- *Indica si las siguientes fórmulas son satisfacibles y explica como se llego a dicha conclusión*

  - $(p \wedge q \wedge s \to p) \wedge (p \wedge r \to p) \wedge (p \wedge s \to s)$
    Utilizando la satisfacibilidad para Horn, marcamos todas las ocurrencias de T, como en la fórmula no hay ocurrencias, terminamos. Ahora buscamos si algún está marcado pero como igual no hay ocurrencias de éste por lo tanto la fórmula es satisfacible.
    Ahora conforme al algoritmo de literales complementarias, tenemos la siguiente fórmula:
    $\equiv (\neg p \vee \neg q \vee \neg s \vee p) \wedge (\neg p \vee \neg r \vee p) \wedge (\neg p \vee \neg s \vee s)$
    $\equiv (T \vee \neg q \vee \neg s) \wedge (T \vee \neg r) \wedge (T \vee \neg p)$
    Y sabemos que $T \vee f \equiv T$, así pues tenemos que
    $\equiv T \wedge T \wedge T \equiv T$ por lo cual la fórmula es satisfacible

  - $(p \wedge q \wedge s \to \perp) \wedge (q \wedge r \to \perp) \wedge (s \to \perp)$
    De la misma manera que el ejercicio anterior, marcamos todas las ocurrencias para T pero como no hay ocurrencias de éstas, terminamos. Ahora nos abocamos a buscar si algún está marcado, pero como no se inicio a marcar nada, entonces cualquier en la fórmula, no está marcado por lo tanto la

fórmula es satisfacible.
Ahora conforme al algoritmo de literales complementarias, tenemos la siguiente fórmula:
$\equiv (\neg p \vee \neg q \vee \neg s) \wedge (\neg q \vee \neg r) \wedge (\neg s)$
Como en cada cláusula no hay literales complementarias, entonces cada literal puede tomar cualquier valor, a lo cual llegamos a:
Sabiendo que $\perp \wedge f \equiv \perp$, así pues tenemos que
$\equiv (\neg p \vee \neg q \vee \neg s) \wedge (\neg q \vee \neg r) \wedge \perp \equiv \perp$, por lo cual la fórmula no es satisfacible.