Tarea 1

Profesores: Andrés Abeliuk - Jocelyn Simmonds
Auxiliares: Carlos Antil - Blaz Korecic - Diego Salas - Lucas Torrealba

Instrucciones.-

- La tarea deber ser resuelta en grupos de a los más 3 integrantes. No olviden especificar claramente sus nombres y secciones en el informe.
- Deberá entregar un único archivo en formato `.zip`, que deberá contener 4 archivos: P1.pdf, P2.pdf, P3.pdf, P3.{py,java,cpp}
- Procure que sus demostraciones sean claras, sin ambigüedades y ordenadas. Se recomienda, aunque no es obligatorio, que elabore sus informes usando LaTeX.

P1: Lógica proposicional.-

1. *(3 pt)* Suponga que le entregan un algoritmo de "caja negra" de resolución SAT - es decir, un dispositivo que toma una fórmula de lógica proposicional ϕ y devuelve si ϕ es o no satisfactible. Usted no sabe nada sobre el funcionamiento de este algoritmo. Vamos a denotar este algoritmo A, por lo que $A(\phi)$ es verdadero si ϕ es satisfacible.

   Esta pregunta plantea qué más se puede hacer con un algoritmo de resolución SAT.

   a) *(1 pt)* Cree un algoritmo que utilice A como subrutina para determinar si ϕ es una tautología. Demuestre que su algoritmo es correcto. No se limite a enumerar todas las posibles asignaciones y comprobar cada una individualmente

   b) *(1 pt)* Suponga que tiene dos fórmulas proposicionales ϕ y ψ. Te interesa determinar si $\varphi \equiv \psi$; es decir, si ϕ y ψ tienen siempre los mismos valores de verdad. Crea un algoritmo que utilice A como subrutina para responder a esta pregunta, y demuestra que tu algoritmo es correcto.

   c) *(1 pt)* Suponga que tiene una fórmula proposicional ϕ con n variables que sabe que es satisfacible. Cree un algoritmo que utilice A como subrutina para obtener una asignación satisfactoria para ϕ utilizando como máximo n llamadas a A. Demuestre que su respuesta es correcta.

2. *(3 pt)* En esta pregunta usted construirá fórmulas de la lógica proposicional que definen como sumar números binarios.

   Considere el conjunto de variables proposicionales $P = \{a_0, a_1, ..., a_{n-1}, b_0, b_1, ..., b_{n-1}\}$. Podemos suponer que cada valuación $\sigma : P \to \{0, 1\}$ define un par de números binarios X_1^σ y X_2^σ dados por la evaluación de σ sobre las secuencias de variables $a_{n-1}...a_1a_0$ y $b_{n-1}...b_1b_0$. Por ejemplo, si $n = 3$ y σ es tal que, $\sigma(a_2) = \sigma(a_1) = \sigma(b_1) = 0$ y $\sigma(a_0) = \sigma(b_2) = \sigma(b_0) = 1$ entonces $X_1^\sigma = 001$ y $X_2^\sigma = 101$.

Construya fórmulas (solamente con los conectivos $\{\vee, \neg\}$) $\phi_0, \phi_1, ..., \phi_{n-1}, \phi_n$ en $\mathcal{L}(P)$ tales que para toda valuación σ se cumpla que $\sigma(\phi_i) = 1$, si y solo si, el bit en la posición i de $X_1^\sigma + X_2^\sigma$ es 1.

*Sus fórmulas las deben dejar en función de $\{\neg, \vee\}$, en caso de que usen algún otro conectivo lógico ($\{\wedge, \implies, \oplus, etc\}$), **DEBEN** definirlo en base a $\{\neg, \vee\}$, por más trivial que sea (esto es por enunciado solamente).*

P2: Lógica de predicados.-

Considere la estructura de la lógica relacional definida como $\mathcal{A} = (\mathbb{N}; +; \times)$. Es decir, el dominio de $\mathcal{A}$ son los números naturales y tenemos accesos a las relaciones ternarias de adición $(+)$ y multiplicación $(\times)$ sobre tal dominio, respectivamente. Formalice los siguientes enunciados, definiendo los predicados que sean apropiados.

Importante: En esta pregunta nuestro dominio son los naturales, es decir, **NO** pueden realizar x^{-1} para representar $\frac{1}{x}$ ni tampoco pueden hacer $-x$, al menos que se definan una relación de esa manera con los elementos en los naturales. Además denotaremos $+(a, b, c)$ que equivale a $a + b = c$, mientras que $\times(a, b, c)$ equivale a $a \cdot b = c$. Tampoco pueden usar constantes como $1, 2...$ sin antes definirlas como relaciones.

1. *(1.5 pt)* El conjunto de todos los triples (m, n, p) tal que $n \neq 0$ y $p = \lfloor \frac{m}{n} \rfloor$.
2. *(1.5 pt)* El conjunto de los triples (m, n, p) tal que $m \geq n$ y $m \equiv n \mod p$, es decir, $m - n$ es divisible por p.
3. *(1 pt)* El conjunto de todos los n que son de la forma 2^p, para algún $p \geq 0$.
4. *(1 pt)* El conjunto de todos los pares (p, q) de números primos gemelos. Donde los números primos (p, q) son números primos gemelos si, siendo $q > p$, se cumple $q - p = 2$.
5. *(1 pt)* El conjunto de los números irracionales. Para este problema, suponga que la definición de número racional significa que p y q no deben tener factores en común distintos de 1 y -1.

P3: Diseño de Algoritmos por Inducción.-

Considere el problema de la galleta: Se le entrega una lista de n galletas donde la galleta $i-$ésima tiene $-A \leq a_i \leq A$ como valor de satisfacción. Usted quiere escoger un conjunto de galletas tal que para todo par de galletas escogidas, estas no estaban adyacentes en la lista, y quiere que la suma de satisfacción de las galletas escogidas sea la máxima posible.

Se tiene la función $F\colon [n] \to \mathbb{N}$ tal que $F(i)$ es la solución considerando solo las primeras i galletas. Además consideraremos que todo número $n \cdot A$ puede guardarse en memoria usando $O(1)$ de memoria.

1. *(1 pt)* Entregue una fórmula recursiva para F y demuestre su correctitud.
2. *(1 pt)* Diseñe un algoritmo que calcule F de forma recursiva con $O(1)$ de memoria extra (es decir memoria constante). Indique la complejidad del algoritmo.

3. *(1 pt)* Mejore el algoritmo anterior a uno iterativo que calcule $F(n)$ con $O(1)$ de memoria extra e indique la complejidad.

4. *(3 pts)* Programe un algoritmo $P3.\{py, cpp, java\}$ que reciba n en la entrada y una lista de n números enteros y entregue el índice de las galletas a escoger para maximizar la satisfacción junto con la satisfacción máxima. El algoritmo debe usar la relación encontrada en la parte (2). Ejemplo:

Entrada	**Salida**
5	1 3 5
1 2 3 4 7	11