

Apellido y Nombre: **ACHAUME BEZERRA TOMAS**
email (@mi.unc.edu.ar): **TOMAS.ACHAUME@MI.UNC.EDU.AR**
Nota: **10 Cués**

Lenguajes y Compiladores

Primer parcial - 2026

1. Considere el poset $Q = (\mathcal{P}(\mathbb{N}), \supseteq)$ de subconjuntos de los números naturales ordenados por la inclusión al revés. Por ejemplo, $\{0, 1, 2\} \leq \{1\}$ porque $\{0, 1, 2\} \supseteq \{1\}$.

- ¿Hay cadenas interesantes en Q ?
- ¿Es Q un predominio? Justifique indicando el supremo de una cadena arbitraria o dando una cadena que no tenga supremo.
- ¿Es Q un dominio? Justifique.

2. Considere $\mathcal{P}(\mathbb{Z})$ con la inclusión usual como orden.

a) Considere $g : \mathcal{P}(\mathbb{Z}) \rightarrow \mathcal{P}(\mathbb{Z})$ definida como $g(X) = X \cup \{x+1 \mid x \in X\}$.

- Decida (y justifique) si g es monótona.
- Decida (y justifique) si g es continua.

Ayudas: Sean $A, B \subseteq \mathbb{Z}$, entonces $A = B$ si y sólo si $A \subseteq B$ y $B \subseteq A$. Por otro lado, $x \in (A \cup B)$ si y sólo si $x \in A$ ó $x \in B$.

b) Considere la siguiente ecuación recursiva sobre $\mathbb{Z} \rightarrow \mathcal{P}(\mathbb{Z})$.

$$\begin{aligned} f(0) &= \{0\} \\ f(x) &= g(f(x-1)) \end{aligned}$$

- Proponga una solución para f .
- Sea $h : \mathbb{Z} \rightarrow \mathcal{P}(\mathbb{Z})$, tal que $4 \notin h(4)$ ¿Es h una solución para f ?
- ¿Puede utilizar el teorema del menor punto fijo para encontrar la menor solución?

3. Después de encontrar muchos programas de la pinta

`?x; if x > 0 then programa else fail`

quienes diseñaron el lenguaje LIS pensaron en agregar un nuevo comando que "solamente acepte" inputs positivos. Es decir, si se ingresa un número mayor a cero todo es como siempre; si no, se genera un error. Pero es un error distinto al del comando `fail` porque no se puede capturar con `catchin with`.

- Proponga la sintaxis del nuevo comando.
- Proponga el nuevo dominio semántico (si fuera necesario) o justifique por qué no es necesario modificarlo.
- Defina la semántica denotacional. Preste atención a si es necesario revisar otros aspectos de la semántica.

4. Considerá el lenguaje imperativo con input/output y fallas.

$$F : (\Sigma \rightarrow \Omega) \rightarrow (\Sigma \rightarrow \Omega)$$

$$F f \sigma = \begin{cases} \iota_{in} (z \mapsto \iota_{out} (z, f [\sigma \mid x : \sigma x + 1])) & \text{si } \sigma x < 0 \\ \iota_{abort} \sigma & \text{si } \sigma x > 0 \\ \iota_{in} (z \mapsto \iota_{out} (z, f [\sigma \mid x : z])) & \text{si } \sigma x = 0 \end{cases}$$

- Proponé un programa de la forma `while b do c` cuyo funcional sea F .
- Sea σ_k un estado tal que $\sigma_k x = -k$. ¿Vale la igualdad $F^i \perp \sigma_2 = F^i \perp \sigma_3$ para todo i ?
- Sea σ_k como en el ítem anterior. Hacé un diagrama de Hasse con los siguientes elementos de Ω .
 - $\iota_{out} (3, (\iota_{in} (z \mapsto \iota_{out} (z-1, \perp))))$.
 - $\iota_{in} (z \mapsto \iota_{term} [\sigma_0 \mid x : -1])$.
 - $\iota_{in} (z \mapsto \iota_{out} (z, \iota_{term} [\sigma_0 \mid x : -1]))$.
 - $\iota_{in} (z \mapsto \iota_{out} (z, \perp))$.
 - $\iota_{out} (2, (\iota_{in} (z \mapsto \iota_{out} (z+1, \perp))))$.