Programación Avanzada

Pablo F. Castro

Universidad Nacional de Río Cuarto, Departamento de Computación

Información de la Materia

- **Teóricos** (Pablo F. Castro, pcastro@dc.exa.unrc.edu.ar):
  - Miércoles de 14 a 16hs.
  - Viernes de 10 a 12hs.

- **Prácticos y Laboratorios** (M.Marta Novaira, Pablo Ponzio, Ernesto Cerdá):
  - Martes de 10 a 12hs. (Lab.)
  - Martes de 16 a 18hs. (Lab.)
  - Lunes de 14 a 16hs.
  - Jueves de 8 a 10hs.

- **Página de la materia:** `http://dc.exa.unrc.edu.ar/moodle/` (Hay que registrarse.)

Modalidad de la Materia:

El regimen de regularización de la materia exige:

- La aprobación de dos parciales.
- La aprobación de un trabajos prácticos. (Se van a tomar defensas.)
- Los parciales tienen un recuperatorio cada uno.

Promoción

- Aprobación de los dos parciales, o sus respectivos recuperatorios, con nota mayor a 7.
- Aprobación de los trabajos prácticos con nota mayor a 7.
- Promedio mayor a 8.

Los alumnos regulares que no hayan promocionado deben rendir un examen final para aprobar la materia. Los trabajos prácticos son de a grupos de tres personas.

Temas de la materia:

En la materia se verán nociones avanzadas de programación en particular nos enfocaremos en la construcción de programas correctos.

Programas Correctos

Se dice que un programa es correcto si hace exáctamente lo que es requerido por su especificación

Para esto veremos los siguientes temas:

- Repaso de lógica proposicional y de primer orden.
- Programación funcional y cálculo de programas funcionales.
- Lógica de Hoare y cálculo de programas imperativos.
- Nociones básicas de Autómatas y Lenguajes.
- Nociones básicas de computabilidad.

Bibliografía

Durante la materia utilizaremos los siguientes libros:

- **Cálculo de Programas**. Autores: Javier Blanco, Silvina Smith y Damián Bartsotti. Disponible como .pdf. (Lógica, Introducción a la Programación Funcional, Corrección de Programas). **Online**
- **Program Construction:** Calculating Implementations from Specifications. Roland Backhouse. (Lógica y Corrección de programas Imperativos). **C.Estudiantes**
- **Introduction to Functional Programming using Haskell.** Richard Bird. (Programación Funcional) **Biblioteca**.

Desarrollo Riguroso de Programas

Es común que en la informática se desarrollen programas mediante **Ensayo y Error**. Sin embargo, muchas veces utilizar esta clase de técnicas puede tener consecuencias no deseadas:

- **Hubble:** Las computadoras del satélite se apagaron debido a que el software quiso girar una antena más rápido de lo permitido.
- **Misiles Patriot:** En la primera guerra de Irak muchos misiles patriot no funcionaron debido a que el software tenia dos representaciones diferentes del número $0{,}1$.
- **Airbus (1988):** Un error en el software del piloto automático causo 5 muertes.
- **Therac-25:** Un acelerador de partículas utilizado para curar cáncer produjo una sobredosis de energía. La causa fue un error en el software.
- **Airbags:** General Motors retiró del mercado más de un millón de autos (1996-1997 Chevy Cavaliers y Pontiac Sunfires) debido a que un mal funcionamiento del software que controlaba los airbags.

Ejemplo de Razonamiento Riguroso

Trataremos de resolver el siguiente problema:

Tenemos una barra de chocolate y queremos saber cuantos cortes tenemos que hacer para quedarnos con todos los bloques.

Como solucionamos esto?

- Una forma es agarrar una barra de chocolate e intentar...
- Podemos resolver este problema (sin una barra de chocolate) si lo planteamos con ayuda de la matemática.
  - Cada vez que cortamos un pedazo agregamos un bloque mas.
  - Si definimos P:Cantidad de partes obtenidas hasta este momento, C:Cantidad de cortes.

Tenemos:

$$P - C = 1$$

Esta propiedad se llama **Invariante** debido a que se cumple siempre, antes y después de efectuar un corte.

Barra de Chocolate

Queremos averiguar cuantos cortes hacen falta para obtener el número B de bloques. Es decir, esto va a suceder cuando:

$$P = B$$

Reemplazando en la ecuación anterior obtenemos:

Calculando los cortes

$B - C = 1$ lo cual es equivalente a: $B - 1 = C$.

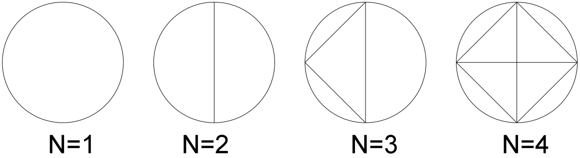
Si la barra de chocolate tiene n bloques, necesitaremos $n - 1$ cortes para obtener todos los bloques.

Un ejemplo más complicado

El problema de la torta:

- Tenemos una torta.
- Tenemos N puntos en su contorno.
- Se trazan todas las cuerdas posibles sobre los puntos.
- Se supone que nunca se cortan mas de dos cuerdas en un punto.
- Cuantas porciones de torta obtenemos?

Veamos...



Ejemplo de la torta (2)

Analicemos unos cuantos casos:

- 1 punto $\rightarrow$ **1 porción**.
- 2 puntos $\rightarrow$ **2 porciones**.
- 3 puntos $\rightarrow$ **4 porciones**.
- 4 puntos $\rightarrow$ **8 porciones**.
- 5 puntos $\rightarrow$ **16 porciones**.

Podríamos decir que: $f = 2^{n-1}$, sin embargo...

- 6 puntos $\rightarrow$ **31 porciones**!
- 7 puntos $\rightarrow$ **57 porciones**!

Ejemplo de la torta (3)

En este caso el método de ir de lo particular a lo general no sirve. Intentemos deducir un formula que nos permita calcular el número para cualquier N:

- f : número de porciones.
- c : número de cuerdas.
- p : cantidad de intersecciones internas.

Veamos:

$$\begin{aligned}
&\text{número de porciones agregadas por una cuerda} \\
&= \{ \text{ las cuerdas dividen porciones en dos } \} \\
&\text{número de porciones cortadas por la cuerda}
\end{aligned}$$

Podemos seguir razonando.

Ejemplo de la torta (4)

número de porciones agregadas por una cuerda

$= \{$las cuerdas dividen las porciones en dos $\}$

número de porciones cortadas por la cuerda

$= \{$Una porción se divide por un segmento de la cuerda$\}$

número de segmentos de la cuerda

$= \{$Los segmentos estan determinados por la cantidad de puntos de intersección$\}$

$1 +$ cantidad de puntos de intersección

Ejemplo de la torta (5)

Es decir, podemos decir:

$$
\begin{array}{l}
\text{Cantidad de porciones agregadas por c cuerdas} \\
= \{ \text{ deducción anterior } \} \\
\text{c} + \text{Cantidad de puntos de intersección en las c cuerdas}
\end{array}
$$

Al principio si $N = 0$, tenemos una porción. Es decir, obtenemos la siguiente formula:

$$f = 1 + c + p$$

Ejemplo de la torta (final)

$$
\begin{aligned}
& f \\
= & \{ \text{ formula anterior } \} \\
& 1 + c + p \\
= & \{\text{una cuerda cada dos puntos}\} \\
& 1 + \binom{N}{2} + p \\
= & \{\text{un punto de intersección cada cuatro puntos}\} \\
& 1 + \binom{N}{2} + \binom{N}{4} \\
= & \{Algebra\} \\
& 1 + \frac{N^4 - 6N^3 + 23N^2 - 18N}{24}
\end{aligned}
$$

No era tan sencillo!