

- **Fecha de entrega: 28 de noviembre de 2024 < 23:59.** Ver al final **Condiciones de Entrega.**

**Cada inciso: 10 puntos. Máximo: 100 pts.**

- Sea  $B := \{a^n b^n c^n \mid n \in \mathbb{N}, n \geq 0\}$ .
  - Defina un módulo Haskell**, `ParserAnBnCn`, que contenga un parser, `parserAnBnCn`, para el lenguaje B. El parser `parserAnBnCn` debe regresar resultados exitosos mediante la tupla `(String,String,String)`. Por ejemplo, `parse parserAnBnCn "aabbccXXX"` debe regresar `[(("aa","bb","cc"), "XXX")]`  
Sugerencia: Use algo similar a lo que usamos en la definición del parser `parserLenguajeId`, del módulo `DFAparserLenguajeId` (visto en clase).
  - En un archivo, `parserAnBnCn.txt`, **muestre** tests de `parserAnBnCn` sobre: `"aabbcc"`, `"aabbccc"`, y `""`.
- Tomando como guía el parser `item :: (Parser Char)` del módulo `HuttonParsing`:
  - Defina un módulo Haskell**, `ParsersXyHola`, que importe solamente `Parser(..)` y `parse` del módulo `HuttonParsing`. Es decir el módulo `ParsersXyHola` **debe tener solamente un import**:  
`import HuttonParsing(Parser(..), parse).`  
**Usando el constructor P** del tipo de datos `Parser(..)`, dentro del módulo `ParsersXyHola`:
    - **Defina un parser**, `parserX`, para el caracter `'X'`.
    - **Defina un parser**, `parserHola`, para el string `"Hola"`.
  - En un archivo, `parsersXyHola.txt`, **muestre** tests de:
    - `parserX` sobre los inputs: `"Xzzz"`, `"xzzz"`.
    - `parserHola` sobre los inputs: `"Holazzz"`, `"holazzz"`
- Tomando como guía módulo Haskell `DFAparserLenguajeId` (visto en clase):
  - Defina un módulo Haskell**, `ParserLenguajeId`, que incluya un parser, `parserIdentidad :: Parser String`.  
El parser `parserIdentidad` debe hacer **parsing** del lenguaje  $Identidad := \{1^n = 1^n \mid n \in \mathbb{N}\}$ , donde el alfabeto es  $\Sigma := \{1, =\}$ .  
El parser `parserIdentidad` debe **reflejar la gramática** de `Identidad`:  
 $\langle Identidad \rangle ::= 1 \langle Identidad \rangle 1 \mid =$  (notación BNF)  
Es decir, el parser `parserIdentidad` es un parser **recursivo**, y debe estar **definido con**:
    - La **alternativa** de dos parsers: `parser_1_Identidad_1`, y `parserEq` (para `"1⟨Identidad⟩1 | ="`).
      - donde `parser_1_Identidad_1` es un parser definido con una **combinación secuencial** de tres parsers: `parser1`; `parserIdentidad`; y `parser1`. (para `"1⟨Identidad⟩1"`).
      - donde `parser1 :: Parser Char` es un parser para el caracter `'1'`.
      - donde `parserEq :: Parser Char` es un parser para el caracter `'='`.
  - Muestre** tests de `parserIdentidad` sobre: `"1=1"`, `"="`, `"=1"`, `"1="`, `"111=111"`. **Archivo:** `parserIdentidad.txt`.
- Escriba a mano:**
  - La definición de Hutton para el tipo de datos `Parser a`. **Archivo** `DeffParser.pdf` o `jpg`.
  - Los parsers: `parserX` y `parserHola` (con las restricciones del ejercicio 2a). **Archivo:** `ParsersXyHola.pdf` o `jpg`.
  - El parser `parser_1_Identidad_1`, ejercicio 3a (suponiendo definidos: `parser1` y `parserIdentidad`). **Archivo** `parserIdentidad.pdf` o `jpg`.
  - Las respuestas de:  
¿Es posible definir una Gramática libre de contexto (CFG) para el lenguaje B? ¿Porqué?  
Donde  $B := \{a^n b^n c^n \mid n \in \mathbb{N}, n \geq 0\}$ . **Archivo** `CFGaNbNcN.pdf` o `jpg`.

### Condiciones de Entrega:

1. **Fecha de entrega: 28 de noviembre de 2024 < 23:59.**
2. **Cada inciso: 10 puntos. Máximo: 100 pts.**
3. Archivos de las **preguntas 1–3**:  
Los **módulos Haskell** deben ser **compilables**.  
En un **archivo zip**: T1p1-3EquipoOapellidos.zip  
Enviar **T1p1-3EquipoOapellidos.zip** por e-mail:
  - a) **Dirigido a:** ayudantes del curso
  - b) **Con copia a:** miguel.mcb en gmail, e Integrantes del Equipo.
  - c) **Asunto:** [Comp T1p1-3EquipoOapellidos]
4. Archivos de las **preguntas 4a-4d**, “Escritas a mano”:  
Las **respuestas** debe ser **legibles**. Las **preguntas** son **individuales**.  
En un **archivo zip**: T1aManoApellidos.zip.  
Enviar **T1aManoApellidos.zip** por e-mail:
  - a) **Dirigido a:** ayudantes del curso
  - b) **Con copia a:** miguel.mcb en gmail.
  - c) **Asunto:** [Comp T1aManoApellidos]