



Auxiliar 3

Estructuras recursivas

P1. Árboles N-Arios: Basándonos en la implementación de árboles binarios vistos en clase, ahora trabajaremos con árboles n-arios.

- Defina el tipo **NTree**
- Defina **fold-ntree** que abstraiga el patrón recursivo de uso de funciones en árboles n-arios.
- Defina **sum-n-tree** que calcule la suma total de los valores internos y hojas del árbol.
- Defina **contains-n-tree?** que retorna si un árbol *nt* contiene el valor *v* en alguno de sus nodos.
- Defina **select-n-tree** que retorna la lista de valores de un árbol *nt* que cumplen con un predicado *p*.

P2. Polinomios: Podemos definir polinomios mediante las siguientes reglas inductivas:

$$\frac{\text{nullpol} \in \text{Polynomial}}{\quad} \quad \frac{c \in \mathbb{I} \quad e \in \mathbb{N} \quad p \in \text{Polynomial}}{(\text{plus } c \ e \ p) \in \text{Polynomial}}$$

Por ejemplo, el polinomio $p(x) = 4x^5 + 3x^2 + 5$ se representa mediante:

(plus 4 5 (plus 3 2 (plus 5 0 (nullp))))

- Defina el tipo **Polynomial**
- Defina la función **parser** que transforme la sintaxis concreta en abstracta. Ojo que los símbolos en la sintaxis concreta están juntos, es decir, $2x^5$ es **UN SOLO SÍMBOLO** y no hay forma de separarlo fácil con un `match`. Programe funciones auxiliares que a partir del símbolo entero entreguen el coeficiente y el grado.
- Defina **fold-p** que abstraiga el patrón recursivo de uso de funciones.
- Defina **eval** que reciba *n* un número entero y *p* un polinomio y evalúe $p(n)$. Use **fold-p**.
- Decimos que la representación de un polinomio está en forma normal si:
 - Los exponentes son listados de mayor a menor, estrictamente, de izquierda a derecha (\rightarrow).
Ej: (plus 3 2 (plus 4 5 (plus 5 0 (nullp)))) y (plus 4 5 (plus 7 5 (nullp))) no están en forma normal.
 - Los monomios de coeficiente 0 se omiten.
Ej: (plus 0 10 (plus 4 5 (plus 0 3 (plus 3 2 (plus 5 0 (nullp))))) no está en forma normal.

Defina la función **nf?** que recibe un polinomio *p* y retorna *#t* si está en forma normal y *#f* si no. **NO** use **fold-p**.