Analyse ascendante, syntaxe

Exercice 1 On considère les expressions arithmétiques avec $\times$, $+$ et $C(n)$ (token de constante numérique de représentation n).

1. Justifier que la grammaire suivante est ambiguë :

$$E \longrightarrow E \times E \mid E + E \mid (E) \mid C(n)$$

2. Proposer une grammaire non ambiguë pour ces expressions en considérant que $\times$ associe à gauche et est prioritaire sur $+$ qui associe à droite.

3. Proposer des actions telles que l'enracinement final retourne la valeur (entière) de l'expression.

4. Proposer des actions telles que l'enracinement final retourne la moyenne des constantes présentes dans une expression.

5. Calculer la moyenne des constantes dans l'expression de syntaxe concrète 3 * 5 + 2 + 1 (le token associé à * est $\times$, le token associé à + est $+$).

6. Proposer une syntaxe abstraite pour ces expressions.

7. Calculer l'arbre de syntaxe abstraite de 3 * 5 + 2 + 1.

Exercice 2 Le langage des polynômes à coefficients entiers sur l'indéterminée x est donné par la grammaire concrète suivante :

Poly $\longrightarrow$ *Poly* + *Poly* | *Poly* * *Poly* | *Poly* ^ *Nat* | *Nat* | -*Nat* | x | (*Poly*)
Nat $\longrightarrow$ $(0 \cdots 9)^+$

1. Montrer que cette grammaire est ambiguë.

2. Afin de lever les ambiguïtés on considère que + et * associent à gauche et que ^ est prioritaire sur * qui est prioritaire sur +. Donner une grammaire non ambiguë pour le langage des polynômes.

3. Proposer une grammaire abstraite pour les polynômes avec les étiquettes abstraites *Add*, *Mult*, *Puiss*, *Cte* et *X*. (On pourra étiqueter les *Cte* directement par un entier.)

4. Donner les actions de grammaire permettant de construire l'arbre de syntaxe abstraite au cours de l'analyse ascendante (à chaque enracinement).

5. Calculer l'arbre de syntaxe abstraite de (3 * x ^ 4 + 2) * (x + -1).