**Partie II: Question 1 :**

**On enlève la récursivité à gauche et l’ambiguïté :**

* P → **program** id ;

        Dcl

        Inst Composée .

* Dcl → ε dcl’

dcl’ → var List\_id : Type ; dcl’ | ε

* list\_id → id list-id’

list\_id’ → , id list\_id’ | ε

* Type  → **integer** | **char**
* Inst\_composée  → **begin**

        Inst

**end**

* Inst  → Liste\_inst | ε
* Liste\_inst → I Liste\_inst'

Liste\_inst' → ; I Liste\_inst' | ε

* I **→** Id := Exp\_simple | **if** Exp **then** I **else** I |

**while** Exp **do** I |

**read**(id) | **readln**(id)| **write**(id) | **writeln**(id)

* Exp → Exp\_simple Exp’

Exp’ → ε | oprel Exp

* Exp\_simple → Facteur Exp\_simple’

Exp\_simple’→ opadd Terme Exp\_simple’ | ε

* Terme → Facteur Terme’

Terme’ → opmul Facteur terme’ | ε

* Facteur → Id Facteur' | nb Facteur' | (Exp\_simple) Facteur'

Facteur' → ε