(* The start of a nix expression *)
Expression →

(*ExpressionNoAssignment* | *Assignment*)

ExpressionNoAssignment →

(*Attrset* | *Function* | *Arithmetic*)

(* A let .. in or with *)
AssignmentPreamble →
{ (*let-in* | *with*) }⁺,

(* Let in *)
(* (* `let a = 5; in ...` *) *)
let-in →
`let` { (*AssignmentNoPreamble* | *Inherit*) }⁺ `in`

with →
`with`

(*Attrset* | ? Identifier of Attrset ?) `;`

Inherit →
`inherit` [? Identifier of Attrset ? | *Attrset*] { ? Identifier ? }⁺ `;`

(* Expression of an attrset *)
Attrset →
[*AssignmentPreamble*] `{` { (*Assignment* | *Inherit*) }⁺ `}`

(* An assignment *)
(* (* `let x = 5; in a = x;` *) *)
Assignment →
[*AssignmentPreamble*] *AssignmentNoPreamble*

(* An assignment without the preamble *)
(* (* `a = 5;` *) *)
AssignmentNoPreamble →
? someIdentifier ? `=` *ExpressionNoAssignment* `;`

(* A string, integer or float *)
Primary →
(? string ? | ? integer ? | ? float ? | `true` | `false` | *List* | (`(` *ExpressionNoAssignment* `)`))

List →
`[` { ExpressionNoAssignment } `]`

(* A function *)
(* (* `input: output` *), *)
Function →
[AssignmentPreamble] ? InputIdentifier ? `:` [AssignmentPreamble] *ExpressionNoAssignment* `;`

Arithmetic →
ArithmeticMul { (`-` | `+`) *ArithmeticMul* }

ArithmeticMul →
PrimaryOrIdentifier { (`*` | `/`) *PrimaryOrIdentifier* }

PrimaryOrIdentifier →
(*Primary* | *Identifier*)