Translator from C To VB



Par

**Mohamad CHALHOUB Abdallah JAFFAL**

Sous-direction de

Dr **Ihab SBEITY**

Contents

[1. Introduction : 3](#_Toc31964157)

[2. Objectif: 3](#_Toc31964158)

[3. Structure: 3](#_Toc31964159)

[4. Fonctionnement (l’algorithme) : 4](#_Toc31964160)

[5. Grammaire: 4](#_Toc31964161)

[6. Conclusion: 7](#_Toc31964162)

[7. Perspectives: 7](#_Toc31964163)

[8. Annexe: 7](#_Toc31964164)

[9. Références : 7](#_Toc31964165)

# Introduction :

Tous d’abord on va parler des logiciels utilisés dans ce Traducteur.

* 1. Flex : Pour effectuer une analyse lexicale ; pour convertir un flux de caractères en un flux de jetons (Tokens) comme préalable à l'analyse.
  2. Bison : Bison est un programme qui convertit la description formelle d'une grammaire du langage informatique en un programme du langage C qui peut analyser la syntaxe et les symboles de cette grammaire en instructions que l'ordinateur peut exécuter. La grammaire à convertir doit être une grammaire hors contexte Lookahead de gauche à droite (LALR).
  3. GCC : est le compilateur de langage c fournit par le Flex et Bison .

# Objectif:

Le but de ce projet est de construire un traducteur à partir d'un code valide écrit en langage C vers un autre code qui effectue la même opération mais en langage VB (Visual Basic).

# Structure:

Le Traducteur contient trois fichiers principaux :

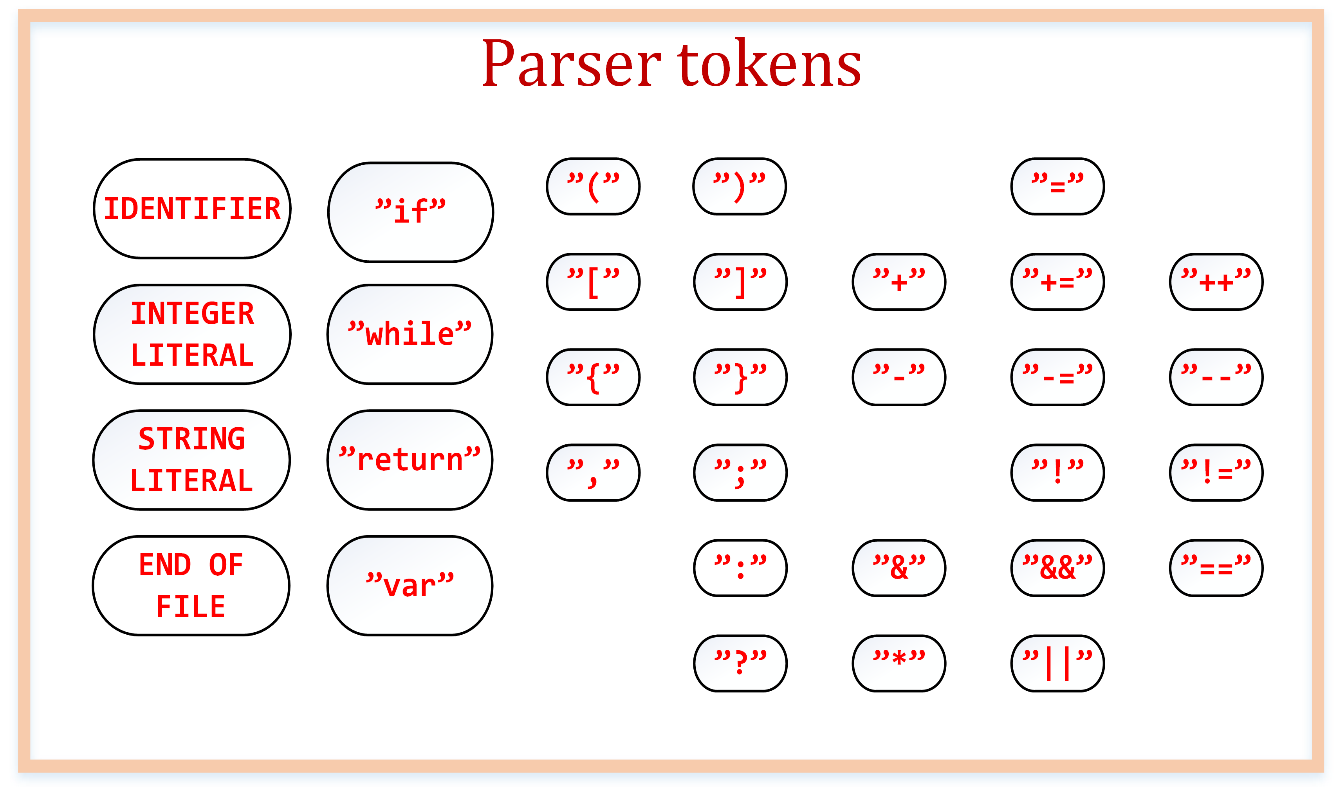
* 1. L’analyseur Lexical ctvb.l :
     1. Qui sera transformer en lex.yy.c .
  2. L’analyseur Syntaxique ctvb.y :
     1. Qui sera transformer en y.tab.c .
     2. Qui sera transformer en y.tab.h .
  3. Le fichier à convertir C\_code.txt .

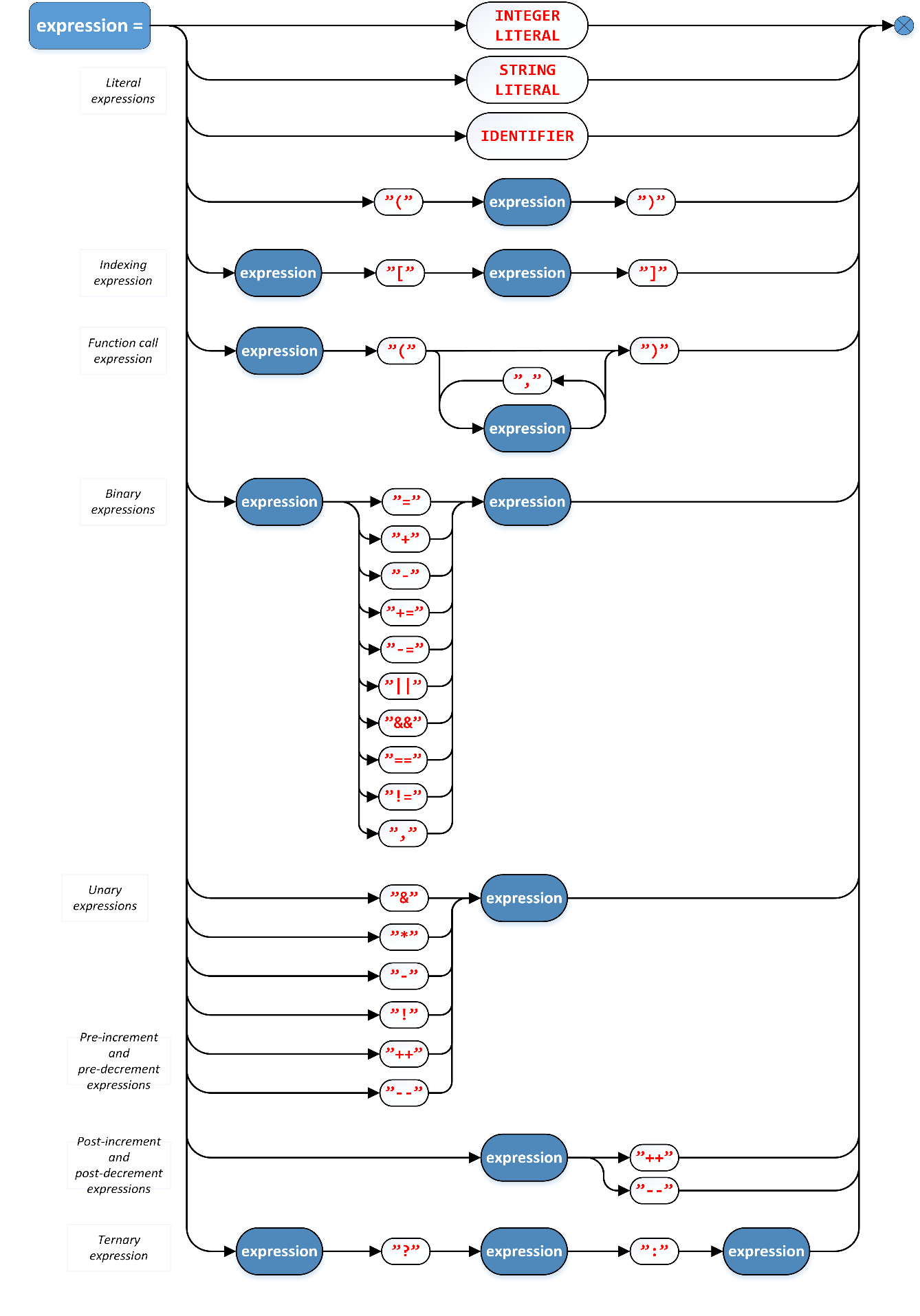
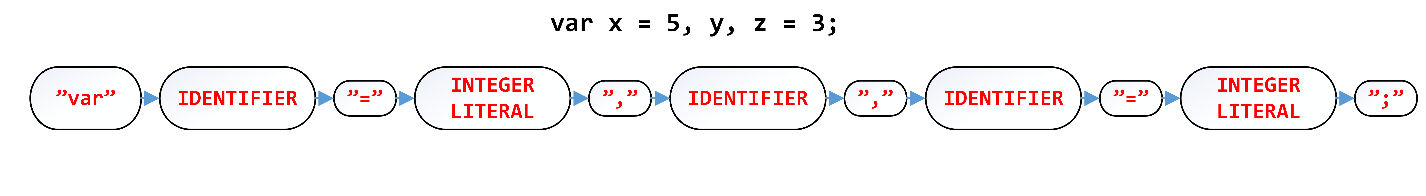
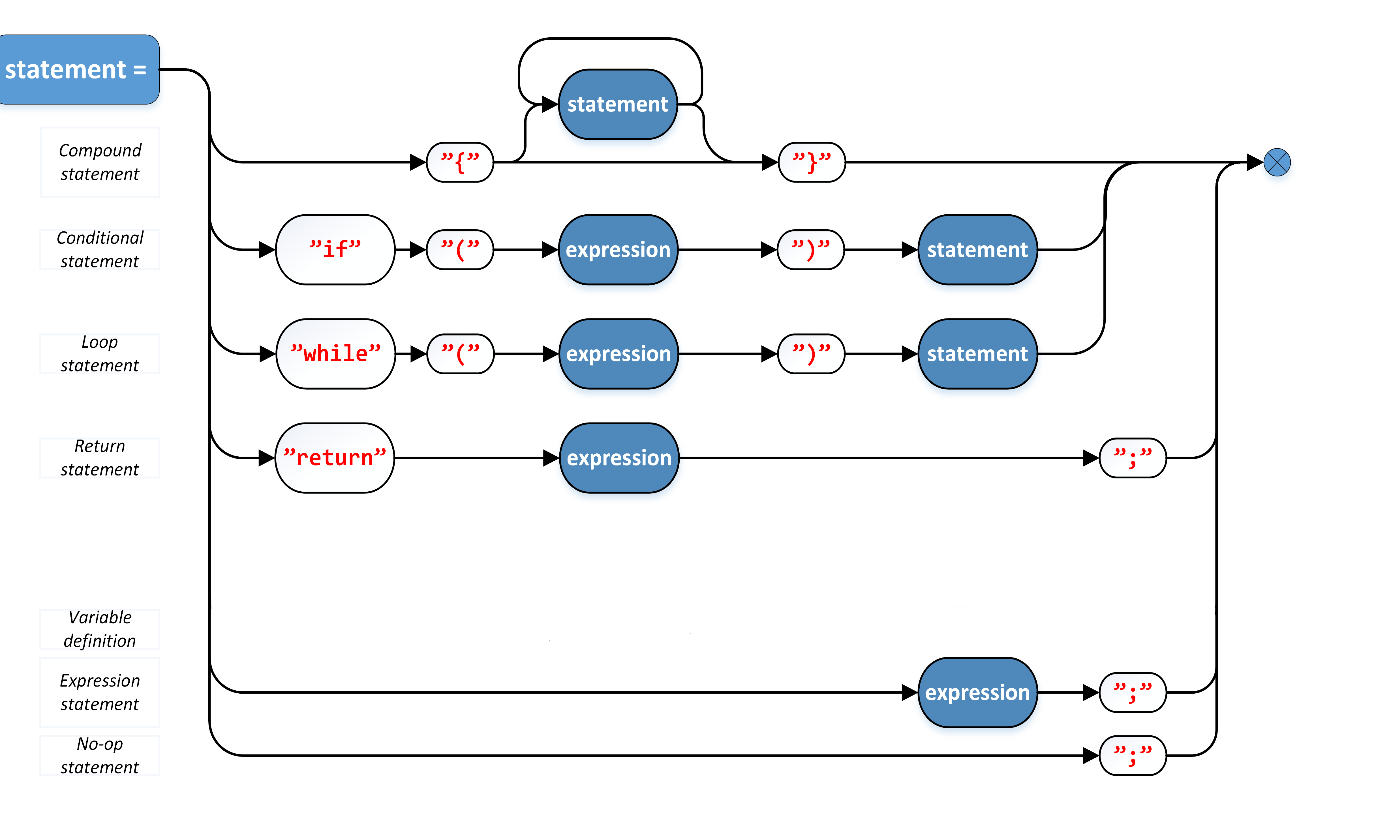
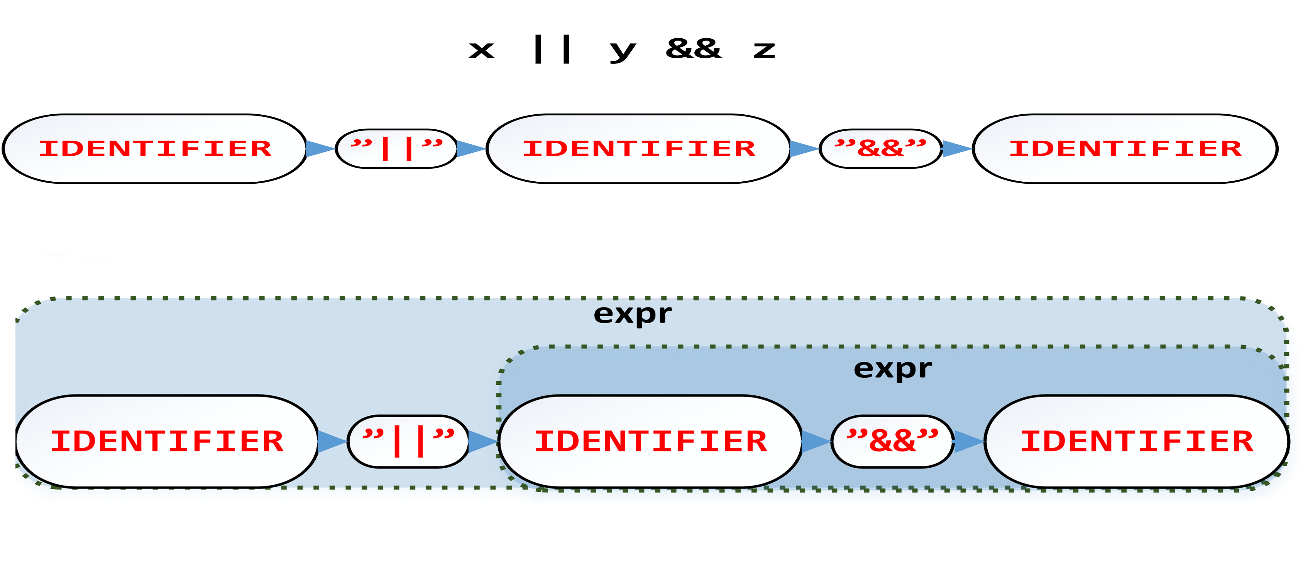
# Fonctionnement (l’algorithme) :

Comme nous l’avons mentionné le traducteur sert à convertir un code C en VB en suivant le chemin suivant :

1. Tout d’abord il commence par lire le code C pour convertir un flux de caractères en un flux de jetons (Tokens) selon des règle (Pattern) prédéfinie dans le lexer (ctvb.l).
2. En suit, il reconnaître cette langue en utilisant un ensemble de règles (BNF) prédéfinie dans le parser (ctvb.y).
3. En cas d’erreur le programme imprime le message d’erreur en spécifiant le type (Lexical ou Syntaxique) et le line et la colonne de l’occurrence de cette erreur.
4. Si le code C est valide il sera converti en code VB qui le corresponde. En plus un Log file (Log.txt) sera généré qui décrit les règles utilisées pour reconnaître le code C.

# Grammaire:

Dans cette partie on va décrire une partie de la grammaire (la grammaire complète est référencée dans la partie [Références](#_Références_:) ci-dessous) et quelque Token : 



# Conclusion:

Nous avons pu appliquer nos connaissances théoriques en analyse lexical et syntaxique sur un exemple pratique implémentant les mécanismes de découvrir et reconnaître une grammaire hors-contexte (le code c) et le transformer en une autre langue (le code VB), en utilisant la langage C, Flex et Bison.

# Perspectives:

Enfin, on peut ajouter des différentes suggestions sur ce projet au futur par exemple :

* 1. Faire l’analyse sémantique correspondant et l’optimisation.
  2. Amélioré l’interface graphique.

# Annexe:

Le programme utilisé Flex, Bison et le GCC (ces trois sont une partie du GNU projet).

# Références :

* 1. How to Create a Compiler ideas : https://www.youtube.com/watch?v=eF9qWbuQLuw&vl=en
  2. ANSI C grammar, Lex specification : https://www.lysator.liu.se/c/ANSI-C-grammar-l.html#VOID
  3. ANSI C Yacc grammar : https://www.lysator.liu.se/c/ANSI-C-grammar-y.html#declarator
  4. Flex Manual : https://www.cs.virginia.edu/~cr4bd/flex-manual/index.html#Top
  5. Bison Manual : http://dinosaur.compilertools.net/bison/bison\_1.html
  6. C Language syntax summary : http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf