

## 1-Introduction

Le projet consiste à produire un service qui permette une fois une expression de bool insérée, de retourner sa forme simplifiée.

Ce projet pourrait servir à aider professeurs comme étudiants à connaître directement la forme simplifiée d'une expression de bool, ils peuvent ainsi l'utiliser comme correction ou autre.

Nous avons été inspiré directement par notre cursus, puisque plusieurs fois lors de celui ci nous avons rencontré cet exercice typique de l'électronique de simplifier au maximum une expression de bool, on aurait bien aimé disposé de ce service là pour aller beaucoup plus vite.

## 2-Description générale du projet

Pour réaliser ce projet, il faudrait disposer d'une structure qui permette de reconnaître les portes AND, OR et NOT qui compose l'expression puis via une matrice qui va servir de table de Karnaugh, réaliser la simplification.

Les fonctionnalités principales vont être un système d'interface utilisateur pour pouvoir récupérer l'expression de bool sur laquelle on va travailler, une traduction de cette expression en 1 ou 0 dans la table de Karnaugh, puis pouvoir traduire le résultat en une expression qui sera l'expression simplifiée et la proposer à l'utilisateur.

Nous ciblons par ce projet, professeurs comme étudiants qui sont au quotidien confronté à des expressions de bool qui ne sont pas les plus optimales, qui grâce à ce service vont aller beaucoup plus vite et devenir plus optimales.

## 3- Exigences fonctionnelles

Le système que l'on propose permettra ainsi :

- Une récupération de l'expression de bool, ainsi il sera nécessaire de réaliser une interaction utilisateur/machine soit via une interface tkinter soit via directement un site web. Peu importe le choix de l'interface, celle ci permettra aussi l'affichage de l'expression de bool simplifiée.

- Le traitement de l'expression obtenue, en effet une fois l'expression de bool reçu il y aura un travail à effectuer sur son traitement puisque selon la pré-condition qu'on aura émise, cette expression ne sera initialement qu'un ensemble de caractère sans sens dans notre contexte. Il faudra ainsi contextualiser cette ensemble de caractère.

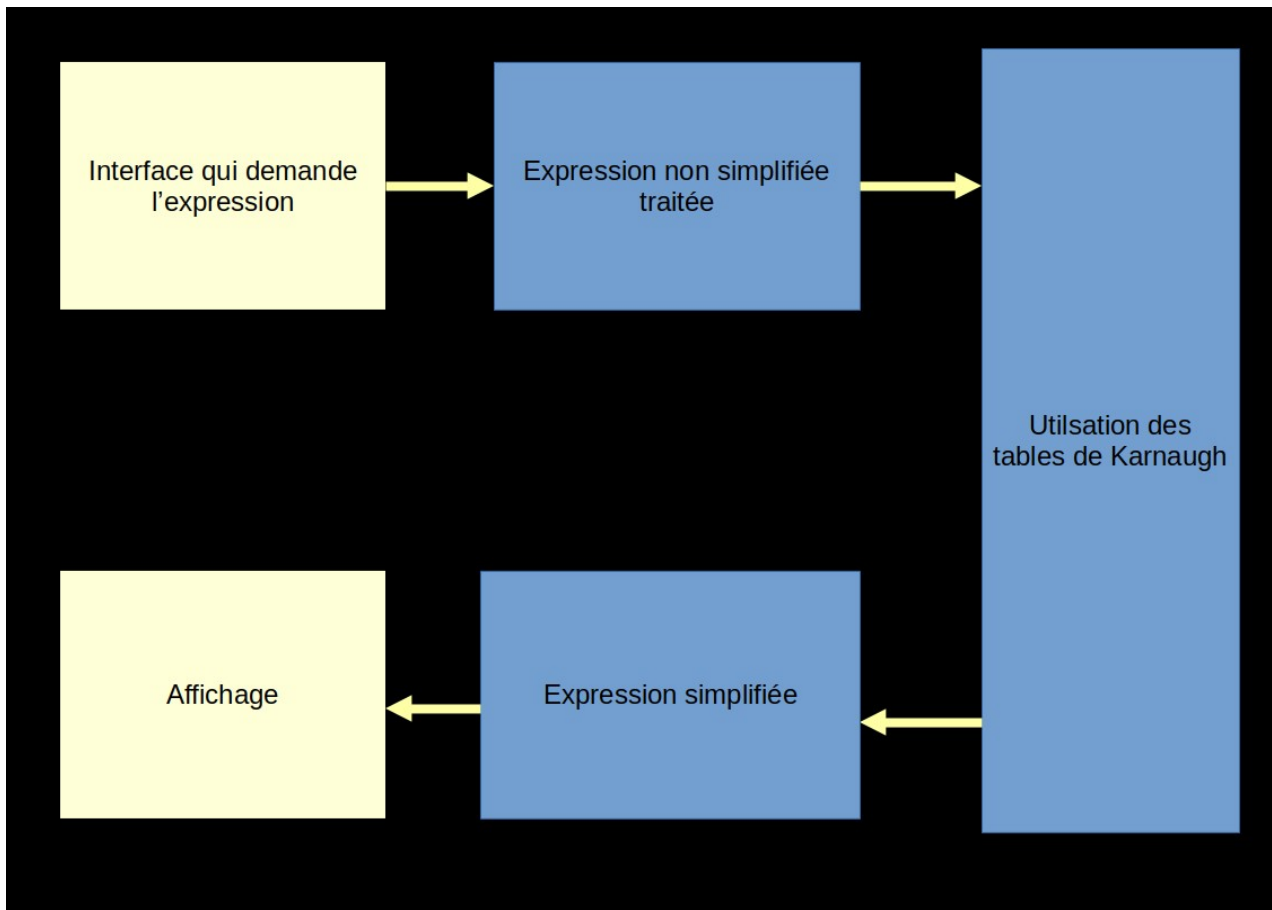
- Sa traduction en table de Karnaugh, une fois l'expression reconnu dans notre contexte il faut maintenant la traduire dans la matrice, chaque position (i,j) va représenter une issue logique avec la bijection entre chaque ligne et colonne avec une des quatre ou trois entrées. Il faudra ainsi reconnaître chaque composant de l'expression de bool et la représenter par un 1 ou un 0 à la position (i,j) qui correspond.

- La traduction dans l'autre sens, une fois la matrice remplie il faudra regrouper les 1 en la plus grande case de dimension de puissance de 2 puis lire d'une manière ou d'une autre l'expression de bool qui correspond.

- L'affichage de l'expression simplifiée.

Ce système sera particulièrement utile pour les étudiants qui pourront obtenir en support d'exercices une correction directe. On peut aussi imaginer un professeur qui donne des expressions à simplifier à ses élèves et quand ils ont fini, leur dit de vérifier leur travail avec notre système.

Schéma qui résume le fonctionnement :



#### 4- Exigences non fonctionnelles

Les contraintes techniques sont les suivantes : soit on adopte python comme langage de programmation et on utilise la bibliothèque tkinter pour l'interface. Ce qui nous arrange étant tous les trois à l'aise avec python, alors il nous restera comme défis technique que d'abord, de réaliser une classe objet qui nous permettra de décoder la logique de bool, on peut imaginer une classe dont les instances sont deux ou une expression de bool et les attributs représente des portes d'entrées comme la porte AND, OR, NOT ou rien.

Une fois cette étape passée, il faudra renseigner la matrice, tout en sachant qu'on demande en précondition pour l'expression une forme canonique disjonctive, pour chaque composante de l'expression placer un 1 à la position (i,j) qui correspond, (le tableau sera initialement rempli de 0). Pour ce faire, il faudra une relation entre chaque ligne et colonne avec une entrée logique puis sauvegarder quelle ligne/ colonne correspond à quelle entrée. Et il ne suffira plus que de lire les entrées que compose l'expression et faire la connection avec sa position. On va faire cela pour chaque composante de notre expression.

Une fois, le tableau rempli il va falloir faire tous les plus grand regroupement possible de 1 de puissance 2. Pour ce faire, pour chaque 1, on va analyser ses alentours dans la matrice pour voir les plus grand regroupement possible et on va le faire pour certains n de manière à ce qu'aucun 1 ne soit laissé de côté et qu'en meme temps, qu'il n'y ait aucun regroupement inutile. Une fois les regroupement fait et les positions enregistrées, il suffira de réutiliser la relation pré-établi entre les positions et les entrées pour pouvoir traduire ce tableau en l'expression simplifiée.

Enfin, il faudra afficher cette expression simplifiée dans l'interface.

(on pourrait parler aussi de quelle nb d'entrée max on prend si jamais on fait un tableau de Karnaugh dynamique)

(on pourrait aussi parler de quelles portes on prend et lesquelles on ignore)

## 5- Exigences relatives aux données

La seule donnée qui sera à traiter et qui vient de l'extérieur est l'expression de bool que l'utilisateur va renseigner. Cette expression devra être codifiée de manière à ce qu'on puisse reconnaître les opérandes logiques et les entrées. De plus, on demande en précondition que l'expression soit une forme canonique disjonctive, FCD, c'est à dire une somme de produit puisque une FCC, forme canonique conjonctive demande un traitement particulier que nous n'avons pour l'instant pas la volonté de réaliser.

On pourrait imaginer que les entrées demandées soient normalisées en A B C D, que l'opérande « et » soit représenté par un « . » et l'opérande ou par un « + » et l'opérande non par un « - ».

Un exemple d'expression serait : « A.B.C.D + -A.B.-C.D ».

## 6-Interfaces utilisateur

(parler de l'interface dont laquelle l'utilisateur devra renseigner l'expression de bool, et de celle qui affichera l'expression simplifiée)

(réaliser une maquette de ce qu'on a en tête pour l'interface)

## 7-Contraintes de développement

(parler du site web)

(parler technique)

(faire un planning de ce qu'on compte faire)

## 8-Documentation

(faire un manuel test pour renseigner l'utilisateur du service/des préconditions)

(faire une docu technique pour si jamais des dev doivent rejoindre le projet, pour savoir quelles techniques on a utilisé pour que eux puisse ajouter des trucs)

(notre politique sur la maintenance du projet (correction des erreurs et mise à jour du projet au fur et à mesure du temps))

## 9-Validation et tests

(parler de comment on pourrait tester notre système, tests basiques, tests chauds, et jusqu'à où on considère que le projet est fini est prêt à sortir)

## 10-Budget et ressources

(les ressources nécessaires dont on a eu besoin pour réaliser le projet (bibliothèque tkinter si jamais on finit par l'utiliser))

## 11-Annexes

(je vois pas ce qu'on pourrait mettre ici mais on verra)