Université Libre de Bruxelles

Rapport de projet

INFO-F302

Baudru Julien & Kerckhof Anthony

November 13, 2020

Question 1

Question 1.1

Il nous est demandé de trouver une formule ϕ telle que : *"Pour toute position i, il existe exactement une lettre $a \in \Sigma$ qui apparaît dans cette case."*
On pose la variable $x_{i,a}$ qui vaut *"Vrai"* si la lettre a se trouve en position i. À cela s'ajoute la contrainte que pour chaque position i, une seule lettre peut être associée. Donc, pour chaque $a \neq a'$, on n'a pas $x_{i,a} \wedge x_{i,a'}$.
On note donc :

$$\phi = \bigwedge_{i=1}^{n} \left(\bigvee_{\substack{a,a' \in \Sigma \\ a \neq a'}} \neg(x_{i,a} \wedge x_{i,a'}) \right)$$

Ce qui vaut en FNC, par De Morgan :

$$\phi = \bigwedge_{i=1}^{n} \left(\bigvee_{\substack{a,a' \in \Sigma \\ a \neq a'}} \neg x_{i,a} \vee \neg x_{i,a'} \right)$$

Question 1.2

Il nous est demandé de trouver une formule ϕ telle que :

Un mot de taille n contient un a :

$$\phi = \bigvee_{i=1}^{n} x_{i,a}$$

Un mot de taille n ne contient pas de b :

$$\phi = \bigwedge_{i=1}^{n} \neg x_{i,b}$$

Un mot de taille n contient le facteur abc :

On se trouve dans le cas où on a :

$$(x_{1,a} \wedge x_{2,b} \wedge x_{3,c}) \vee (x_{2,a} \wedge x_{3,b} \wedge x_{4,c}) \vee ... \vee (x_{n-2,a} \wedge x_{n-1,b} \wedge x_{n,c})$$

Qui se note succinctement :

$$\phi = \bigvee_{i=1}^{n-2} (x_{i,a} \wedge x_{i+1,b} \wedge x_{i+2,c})$$

Un mot de taille n contient la sous-séquence abc :

On trouve une expression qui pour toute valeur de i, j, k, des positions dans le mots, avec $i < j < k$ (respectivement pour i', j', k'), prend la forme :

$$(x_{i,a} \wedge x_{j,b} \wedge x_{k,c}) \vee (x_{i',a} \wedge x_{j',b} \wedge x_{k',c})$$

De cette idée, on tire la formule :

$$\phi = \bigvee_{i=1}^{n-2} \bigvee_{j=i+1}^{n-1} \bigvee_{k=j+1}^{n} (x_{i,a} \wedge x_{j,b} \wedge x_{k,c})$$

Question 2

Question 2.1

On cherche la formule ψ tel qu'elle satistait la phrase : "Pour toutes positions i dans le mot, si on trouve un Nil à une position, alors le caractère à la position suivante sera un Nil". Ce qui s'écrit :

$$\begin{aligned}\psi &= \bigwedge_{i=1}^{n-1} x_{i,Nil} \rightarrow x_{i+1,Nil} \\ &= \bigwedge_{i=1}^{n-1} \neg x_{i,Nil} \vee x_{i+1,Nil}\end{aligned}$$

Ce qui est bien une FNC.
Prouvons maintenant que ψ est bel et bien valide :

$$\begin{aligned}\neg\psi &= \neg \bigwedge_{i=1}^{n-1} (\neg x_{i,Nil} \vee x_{i+1,Nil}) \\ &= \bigvee_{i=1}^{n-1} (\neg\neg x_{i,Nil} \wedge \neg x_{i+1,Nil}) \\ &= \bigvee_{i=1}^{n-1} (x_{i,Nil} \wedge \neg x_{i+1,Nil})\end{aligned}$$

Ce qui souligne une incohérence par rapport à notre formulation ; pour que cette formule soit satisfaisable il faudrait trouver un Nil suivi d'un caractère de l'alphabet Σ. Cela contredit donc bien l'énoncé, $\neg\psi$ ne sera jamais satisfaite. Puisque $\neg\psi$ n'est pas satifaisable, ψ est valide.

Question 2.2

Un tableau de taille n contient un mot de longueur $\leq l$:

Il est équivalent de dire que le mot doit avoir une longueur $< l$+1. Si on vérifie donc que la $(l+1)^{eme}$ place est occupée par un Nil, c'est une condition suffisante pour dire qu'il n'y a pas de mot plus long que l. Notons qu'on considère un mot de taille 0 comme valide.
On a :

$$\phi = x_{l+1,Nil}$$

Un tableau de taille n contient un mot de longueur $\geq l$:

Il est équivalent de dire que le mot doit avoir une longueur $> l$-1. Si on vérifie donc que la l^{eme} position du tableau est occupée par une lettre, c'est une condition suffisante pour dire qu'il n'y a pas de mot de longueur inférieure à l.
On a :

$$\phi = x_{l,a}$$

Un tableau de taille n contient un mot de longueur l :

Il faut vérifier que la position l est bien un caractère de l'alphabet et que la position $l+1$ est un Nil.
On a :

$$\phi = x_{l,a} \wedge x_{l+1,Nil}$$

Question 3

Question 3.1

On checher la formule ϕ telle que : "Un mot contient le facteur *abc*".
Ce qui s'écrit :

$$\phi = \bigvee_{i=1}^{n} (x_{i,a} \wedge x_{i+1,b} \wedge x_{i+2,c})$$

Ce qui n'est pas en FNC, mais puisqu'on a un disjonction de conjonction, on peut faire l'application de Tseitin à une forrmule du type $C_1 \vee C_2 \vee ... \vee C_n$. On introduit la variable y_i **pour chaque disjonction**, on obtient alors :

$$\phi = (y_1 \vee y_2 \vee ... \vee y_n) \wedge (y_1 \leftrightarrow C_1) \wedge (y_2 \leftrightarrow C_2) \wedge ... \wedge (y_n \leftrightarrow C_n)$$

Puisque chaque C_i est la conjonction de littéraux : $x_{i,a} \wedge x_{i+1,b} \wedge x_{i+2,c}$,

$$\begin{aligned} C_i \leftrightarrow y_i &\equiv ((x_{i,a} \wedge x_{i+1,b} \wedge x_{i+2,c}) \rightarrow y_i) \wedge (y_i \rightarrow (x_{i,a} \wedge x_{i+1,b} \wedge x_{i+2,c})) \\ &\equiv (\neg(x_{i,a} \wedge x_{i+1,b} \wedge x_{i+2,c}) \vee y_i) \wedge (\neg y_i \vee (x_{i,a} \wedge x_{i+1,b} \wedge x_{i+2,c})) \\ &\equiv (\neg x_{i,a} \vee \neg x_{i+1,b} \vee \neg x_{i+2,c} \vee y_i) \wedge (\neg y_i \vee x_{i,a}) \wedge (\neg y_i \vee x_{i+1,b}) \wedge (\neg y_i \vee x_{i+2,c}) \end{aligned}$$

Si on réintroduit ça dans la formule ϕ, on a :

$$\phi = \left(\bigvee_{i=1}^{n-2} y_i\right) \wedge (\neg x_{i,a} \vee \neg x_{i+1,b} \vee \neg x_{i+2,c} \vee y_i) \wedge (\neg y_i \vee x_{i,a}) \wedge (\neg y_i \vee x_{i+1,b}) \wedge (\neg y_i \vee x_{i+2,c})$$

Question 3.2

On cherche la formule ϕ telle que : "Un mot contient la sous-séquence *abc*.
On sait que : $y_{i,a}$ est vrai si la lettre a se trouve à la position i, $y_{i,b}$ est vrai si la lettre b se trouve à la position i et $y_{i,c}$ est vrai si la lettre c se trouve à la position i.
On a donc :

$$\begin{aligned}\phi = \bigwedge_{i=1}^{n} \bigwedge_{j=i}^{n} \bigwedge_{k=j}^{n} & y_{i,a} \wedge y_{j,b} \wedge y_{k,c} \\ & \wedge (y_{i,a} \leftrightarrow x_{i,a}) \\ & \wedge (y_{j,b} \leftrightarrow x_{j,b}) \\ & \wedge (y_{k,c} \leftrightarrow x_{k,c})\end{aligned}$$

Qu'on transforme en FNC :

$$\begin{aligned}\phi = \bigwedge_{i=1}^{n} \bigwedge_{j=i}^{n} \bigwedge_{k=j}^{n} & y_{i,a} \wedge y_{j,b} \wedge y_{k,c} \\ & \wedge (y_{i,a} \rightarrow x_{i,a}) \wedge (x_{i,a} \rightarrow y_{i,a}) \\ & \wedge (y_{j,b} \rightarrow x_{j,b}) \wedge (x_{j,b} \rightarrow y_{j,b}) \\ & \wedge (y_{k,c} \rightarrow x_{k,c}) \wedge (x_{k,c} \rightarrow y_{k,c})\end{aligned}$$

On obtient finalement :

$$\begin{aligned}\phi = \bigwedge_{i=1}^{n} \bigwedge_{j=i}^{n} \bigwedge_{k=j}^{n} & y_{i,a} \wedge y_{j,b} \wedge y_{k,c} \\ & \wedge (\neg y_{i,a} \vee x_{i,a}) \wedge (\neg x_{i,a} \vee y_{i,a}) \\ & \wedge (\neg y_{j,b} \vee x_{j,b}) \wedge (\neg x_{j,b} \vee y_{j,b}) \\ & \wedge (\neg y_{k,c} \vee x_{k,c}) \wedge (\neg x_{k,c} \vee y_{k,c})\end{aligned}$$

Question 4

Notons que pour cette question nous allons légèrement modifier la syntaxe de l'énoncé pour ne pas confondre les valeurs m qui représente la longueur des mots dans la matrice des mots, avec les $m_{i,j,a}$. Prenons donc plutôt la variable $x_{i,j,a}$ qui représente que la lettre a se trouve en position j du mot i.

Question 4.1

Le mot 1 a une lettre en commun avec le mot 2

On trouve :

$$\phi = \bigvee_{j=1}^{m} \bigvee_{a \in \Sigma} x_{1,j,a} \wedge x_{2,j,a}$$

Notons qu'avec $\bigvee_{j=1}^{m}$, on itère bien sur chaque position de la matrice des mots. Que puisqu'on compare uniquement les 2 premiers mots de cette matrice, forcer leur indice i est suffisant et il ne faut donc pas itérer sur tous les i. Et que puisque Nil n'appartient pas à l'alphabet Σ, pour 2 mots de taille inférieure à m, la condition ne sera pas vérifiée sur les Nil.

Le mot 1 est plus long que le mot 2

On peut formuler : "S'il existe une position j telle que $x_{1,j,a}$ est vrai et $x_{2,j,Nil}$ est vrai, alors le mot 1 est plus long que le mot 2." On trouve alors :

$$\phi = \bigvee_{j=1}^{m} \bigvee_{a \in \Sigma} x_{1,j,a} \wedge x_{2,j,Nil}$$

Les mêmes remarques que pour "Le mot 1 a une lettre en commun avec le mot 2" s'appliquent.

Le mot 2 est un préfixe du mot 1

La variable m étant la longueur du mot 2, on trouve :

$$\phi = \bigwedge_{j=1}^{m} \bigvee_{a \in \Sigma} ((x_{1,j,a} \wedge x_{2,j,a}) \vee (x_{2,j,Nil}))$$

Qui satisfait bien la propriété puisque cette formule vérifie bien que **pour toutes** les positions j, on a soit une égalité entre les caractères des 2 mots, soit un Nil dans le second mot (donc le second mot a une taille inférieure ou égale à celle du premier et comme les positions avant le Nil ont été vérifiées, la propriété a déjà été vérifiée. Notons que 2 mots de taille identiques sont bien préfixes l'un de l'autre.

Note : On suppose qu'un mot vide est bien le préfixe d'un autre mot (lui même pouvant être vide).

Question 4.2

Les mots 1 et 2 sont de la même taille et diffèrent en une positon exactement

Pour être sur que les 2 mots ne diffèrent que d'exactement une seule position, on va utiliser 2 formules en conjonction l'une de l'autre. La première va vérifier

que si une position diffère, la suite ne diffère pas (ce qui n'est pas suffisant : ça revient à dire que 1 seule position diffère **au maximum**). La deuxième vérifie que les 2 mots ne sont pas exactement identiques.
Pour être sur que les 2 mots ont la même taille; on vérifie toujours qu'à chaque position il y a soit une lettre dans chacun des 2 mots, soit un Nil dans chacun des 2 mots.
On a donc :

$$\phi_1 = \bigwedge_{j=1}^{m} \bigvee_{a\in\Sigma} \neg(x_{1,j,a} \wedge x_{2,j,a}) \rightarrow \bigwedge_{j'=j+1}^{m} (x_{1,j',a} \wedge x_{2,j',a}) \vee (x_{1,j',Nil} \wedge x_{2,j',Nil})$$

Et :

$$\phi_2 = \neg \left(\bigwedge_{k=1}^{m} \bigvee_{a\in\Sigma} (x_{1,k,a} \wedge x_{2,k,a}) \vee (x_{1,k,Nil} \wedge x_{2,k,Nil}) \right)$$

Finalement, on a que :

$$\phi = \phi_1 \wedge \phi_2$$

Le mot 2 est un suffixe du mot 1

L'idée est de vérifier pour chaque sous-séquence commençant dans le mot 1 par le premier caractère du mot 2 si la suite est bien identique jusqu'à la taille maximale du mot 1 ou jusqu'à ce qu'on rencontre un Nil dans les mots 1 **et** 2.
On trouve :

$$\phi = \bigvee_{j=1}^{m} \bigvee_{a\in\Sigma} \left[(x_{1,j,a} \wedge x_{2,1,a}) \rightarrow \bigwedge_{j'=j+1}^{m} \bigwedge_{k=2}^{m} (x_{1,j',a} \wedge x_{2,k,a}) \vee (x_{1,j',Nil} \wedge x_{2,k,Nil}) \vee x_{2,k,Nil} \right]$$

On remarque la présence du terme $x_{2,k,Nil}$ dont l'utilité est de vérifier que si le mot 1 se termine parce qu'il est de taille m, le mot 2 se termine bien également à la fin de ce qu'on a trouvé être le préfixe (et donc que pour toutes les valeurs de k supérieures à la taille du candidat préfixe trouvé, le seul moyen de retourner une valeur vraie est qu'il ne reste plus que des Nil dans le mot 2). Si ce n'était pas le cas, le mot 2 ne serait pas un préfixe du mot 1.

Le mot 2 est un facteur du mot 1

C'est en fait une simplification du cas précédent. L'idée est la même, vérifier qu'à un moment si on trouve une sous-séquence du mot 1 qui commence comme le mot 2, alors on vérifie que la suite concorde. La seule différence est qu'il ne faut plus s'inquiéter de si le mot 1 se termine ou non mais juste si le mot 2 concorde à la sous-séquence du mot un ou est terminé.
On trouve :

$$\phi = \bigvee_{j=1}^{m} \bigvee_{a\in\Sigma} \left[(x_{1,j,a} \wedge x_{2,1,a}) \rightarrow \bigwedge_{j'=j+1}^{m} \bigwedge_{k=2}^{m} (x_{1,j',a} \wedge x_{2,k,a}) \vee x_{2,k,Nil} \right]$$

Question 5

Question 5.1

Le mot 2 est à une substitution près du mot 1

On se rend compte que ça revient au même de dire que "Les mots 1 et 2 sont de la même taille et diffèrent en une postion exactement" qui est une sous-question de la question 4.2
La solution est donc identique :

$$\phi = \phi_1 \wedge \phi_2$$

Avec :

$$\phi_1 = \bigwedge_{j=1}^{m} \bigvee_{a\in\Sigma} \neg(x_{1,j,a} \wedge x_{2,j,a}) \rightarrow \bigwedge_{j'=j+1}^{m} (x_{1,j',a} \wedge x_{2,j',a}) \vee (x_{1,j',Nil} \wedge x_{2,j',Nil})$$

Et :

$$\phi_2 = \neg \left(\bigwedge_{k=1}^{m} \bigvee_{a\in\Sigma} (x_{1,k,a} \wedge x_{2,k,a}) \vee (x_{1,k,Nil} \wedge x_{2,k,Nil}) \right)$$

Le mot 2 est à une délétion près du mot 1

Si la délétion est à la position k, alors il faut que tout ce qu'il il y avant j soit identique dans les 2 mots et que les positions j+1 à m du mot 1 correspondent aux positions j à m-1 du mot 2 (et donc que la position m du mot 2 soit un Nil).
On a :

$$\psi = \bigvee_{j=1}^{m} \bigvee_{a\in\Sigma} \neg(x_{1,j,a} \wedge x_{2,j,a}) \rightarrow \left(\bigwedge_{j'=1}^{j} (x_{1,j',a} \wedge x_{2,j',a}) \wedge \bigwedge_{j''=j}^{m-1} (x_{1,j''+1,a} \wedge x_{2,j'',a}) \wedge x_{2,m,Nil} \right)$$

Le mot 2 est à une addition près du mot 1

La logique est la même que pour une délétion sauf qu'on doit inverser les indices après la position j où apparaît l'insertion par rapport à la délétion, et on doit aussi s'assurer que maintenant c'est le mot 1 pour lequel la position m contient un Nil.
On trouve :

$$\omega = \bigvee_{j=1}^{m} \bigvee_{a\in\Sigma} \neg(x_{1,j,a} \wedge x_{2,j,a}) \rightarrow \left(\bigwedge_{j'=1}^{j} (x_{1,j',a} \wedge x_{2,j',a}) \wedge \bigwedge_{j''=j}^{m-1} (x_{1,j'',a} \wedge x_{2,j''+1,a}) \wedge x_{1,m,Nil} \right)$$

Question 5.2

Étant donnée une matrice de taille n*m, on cherche la formule qui exprime "*Chaque mot i + 1 se trouve à une étape de mot i*". Par étape, on entend soit une substitution notée s_i, une délétion notée d_i ou une addition notée a_i. Dans l'exercice précédent, nous avons noté la formule de la substitution par ϕ, celle de la délétion par ψ et celle de l'addition par ω. Nous reprendrons ces formules dans la solution ci-dessous, à la différence que les variables $x_{1,j,a}$ seront remplacées par $x_{i,j,a}$ et les variables $x_{2,j,a}$ seront remplacées par $x_{i+1,j,a}$ pour correspondre avec la formule demandée. Ainsi la variable s_i équivaut à la formule ϕ, la variable d_i équivaut à la formule ψ et la variable a_i équivaut à la formule ω.
On a donc :

$$\begin{aligned}\bigwedge_{i=1}^{n-1} &(s_i \vee d_i \vee a_i) \\ &\wedge (s_i \leftrightarrow \phi) \\ &\wedge (d_i \leftrightarrow \psi) \\ &\wedge (a_i \leftrightarrow \omega)\end{aligned}$$

Qu'on transforme en FNC :

$$\begin{aligned}\bigwedge_{i=1}^{n-1} &(s_i \vee d_i \vee a_i) \\ &\wedge (s_i \rightarrow \phi) \wedge (\phi \rightarrow s_i) \\ &\wedge (d_i \rightarrow \psi) \wedge (\psi \rightarrow d_i) \\ &\wedge (a_i \rightarrow \omega) \wedge (\omega \rightarrow a_i)\end{aligned}$$

On obtient finalement :

$$\begin{aligned}\bigwedge_{i=1}^{n-1} &(s_i \vee d_i \vee a_i) \\ &\wedge (\neg s_i \vee \phi) \wedge (\neg \phi \vee s_i) \\ &\wedge (\neg d_i \vee \psi) \wedge (\neg \psi \vee d_i) \\ &\wedge (\neg a_i \vee \omega) \wedge (\neg \omega \vee a_i)\end{aligned}$$

Question 6

Question 6.1

Étant donné 2 mots u et v de taille $\leq m$, il nous est demandé de prouver que s'il existe une séquence de n étapes d'éditions pour passer de u en v, alors il existe une séquence de $\leq n$ étapes d'éditions pour aller de u en v sans utiliser de mots de taille $> m$.

Pour montrer que cette proposition est vraie, prenons les mots u et v de taille $\leq m$ tels que :

Mots	1	2	3	4	5
u	a	b	c	b	b
v	c	a	c	c	a

Il existe une séquence d'éditions pour passer de u en v telle que :

u = a b c b b
délétion en position 5
u = a b c b
délétion en position 4
u = a b c
addition de c en position 4
u = a b c c
addition de a en position 5
u = a b c c a
substitution de a par c en position 1
u = c b c c a
substitution de b par a en position 2
u = c a c c a

La séquence ci-dessus utilise 6 étapes d'éditions, d'après la proposition, il existe alors une séquence d'éditions pour passer de u en v en ≤ 6 étapes. Parmi les différentes séquences possibles l'une d'entre elles est la suivante :

u = a b c b b
substitution de a par c, position 1
u = c b c b b
substitution de b par a, position 2
u = c a c b b
substitution de b par c, position 4
u = c a c c b
substitution de b par a, position 5
u = c a c c a

Cette séquence d'édition utilise 4 étapes, ce qui montre que la proposition est bien vérifiée. Le seul cas où la séquence d'éditions sera strictement égale à n se produit lorsque la séquence d'éditions initiale utilise le nombre minimum d'étapes pour passer de u à v. On ne pourra donc jamais avoir un nombre d'étapes $<$ à n_{min} mais toujours au moins égal.

Question 6.2

On cherche la formule ϕ telle que si elle est vraie alors il est possible de passer de u à v en n étapes.

Question 6.3

Par la formule ϕ en 6.2, on peut trouver une séquence de maximum n additions / délétions / substitutions pour passer de u à v. Un algorithme naïf consisterait à rajouter une clause sur n pour fixer sa valeur à chaque tentative de résolution de longueur minimale. Ainsi il suffirait de tester toutes les valeurs de n entre 0 et m (la longueur maximale d'un mot) et de s'arrêter dès que l'appel à la fonction *solve()* du solver renvoie *True*.