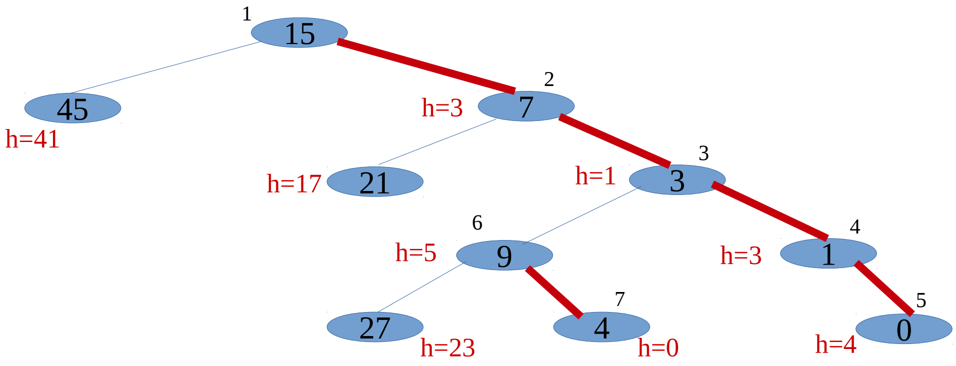
Corrigé Examen Semestriel TPGO (2CS) – ESI 2017/2018

Exercice 1 :
Dans le problème vu en cours, du passage du nombre 15 au nombre 4 en utilisant uniquement les fonctions : f(x) = 3x   et   g(x) = x div 2
Donnez l'espace de recherche généré par *Best First Search*, ainsi que l'ordre des visites des différents états, si on utilise la fonction d'estimation suivante : h(e) = |e – 4|

f(x) : le fils-droit   g(x):le fils-gauche
L'ordre des visites : 15, 7, 3, 1, 0, 9, 4



Exercice 2 :
Soit A le problème de décision suivant : (Stable ou Ensemble Indépendant de taille k)

Entrée : un graphe G non orienté et un entier k

Existe-t-il un ensemble de k sommets indépendants (c'est-à-dire tous non reliés 2 à2) ?

Soit B le problème de décision suivant : (Clique de taille k)

Entrée : un graphe G non orienté et un entier k

G a-t-il une clique de taille k (une clique de taille k est un ensemble de k sommets tous reliés les uns aux autres) ?

Montrer que : A est polynomialement réductible à B. Est-ce que l'inverse est vrai ?

La fonction f qui transforme toute instance x de A en instance f(x) de B est comme suit :
Soit x un graphe non orienté ayant comme ensemble de sommets S et comme ensemble d'arêtes A
Le graphe f(x) est formé par le même ensemble de sommets S et par le complémentaire de A comme ensemble d'arêtes. Soit A' cet ensemble complémentaire d'arêtes.
Pour chaque couple de sommets (a,b) dans S, s'il n'y a pas d'arête entre a et b dans A alors A' contiendra une telle arête a-b. S'il existe une arête a-b dans A, alors A' ne contient pas d'arête entre a et b.
De cette manière à chaque fois qu'un sommet a est relié à un sommet b dans le graphe f(x), cela voudrait dire que a et b n'était pas reliés dans le graphe x. Et inversement à chaque fois qu'un sommet a est relié à un sommet b dans le graphe x, ces mêmes sommets ne seront pas reliés dans le graphe f(x).
Il est donc facile de voir que l'existence d'un stable de taille k sommets dans le graphe x implique que ces k sommets ne sont pas reliés entre eux dans x et donc que ces mêmes k sommets seront reliés entre eux complètement dans le graphe f(x) formant ainsi une clique de taille k dans le graphe f(x).
L'inverse est aussi vrai.
La construction du graphe f(x) peut se faire en temps polynomiale, car il y a n(n-1)/2 arêtes au maximum dans un graphe de n sommets. La transformation f est en O(n^2).
On en déduit que les 2 problèmes A et B sont polynomialement réductibles entre eux.

Exercice 3 :

Soit S un ensemble de 4 clauses (c0, c1, c2, c3) :

S = { c0 :p(x,f(y),b) ∨ q(a) ∨ r(y);   c1 : ¬p(a,x,b) ∨ q(x) ∨ r(c);   c2 : ¬q(z) ∨ r(f(a));   c3 : ¬r(x) }

avec x,y, et z des symboles de variables, a,b et c des symboles de constantes, f un symbole de fonction, p,q et r des symboles de prédicats.

En utilisant l'algorithme de la résolution générale, montrer si S est consistant ou non.

(mettre en évidence les substitutions générées par le processus de preuve)

Soit S un ensemble de clauses

S = { c0 :p(x,f(y),b) ∨ q(a) ∨ r(y);   c1 : ¬p(a,x,b) ∨ q(x) ∨ r(c);   c2 : ¬q(z) ∨ r(f(a));   c3 : ¬r(x) }

En utilisant l'algorithme de la résolution générale, montrer si S est consistant ou non

(mettre en évidence les substitutions générées par le processus de preuve)

étape 1 :

S1 = S ∪ {c4=res(c0,c1) , c5=res(c0,c2) , c6=res(c0,c3) , c7=res(c1,c2) , c8=res(c1,c3)   , c9=res(c2,c3) }

c4=res(c0 :p(x,f(y),b) ∨ q(a) ∨ r(y) , c1 :¬p(a,x,b) ∨ q(x) ∨ r(c))

  renommage de x en x1 dans c1,

  l'unificateur entre p(x,f(y),b) et ¬p(a,x1,b) est : [x ← a, x1 ← f(y)]

  c4= q(a) ∨ r(y) ∨ q(f(y)) ∨ r(c)

c5=res(c0 :p(x,f(y),b) ∨ q(a) ∨ r(y) , c2 :¬q(z) ∨ r(f(a)))

  l'unificateur entre q(a) et ¬q(z) est : [z ← a]

  c5= p(x,f(y),b) ∨ r(y) ∨ r(f(a))

c6=res(c0 :p(x,f(y),b) ∨ q(a) ∨ r(y) , c3 :¬r(x))

  renommage de x en x2 dans c3,

  l'unificateur entre r(y) et ¬r(x2) est : [y ← x2]

  c6= p(x,f(x2),b) ∨ q(a)

c7=res(c1 :¬p(a,x,b) ∨ q(x) ∨ r(c) ,   c2 : ¬q(z) ∨ r(f(a)))

  l'unificateur entre q(x) et ¬q(z) est : [x ← z]

  c7= ¬p(a,z,b) ∨ r(c) ∨ r(f(a))

c8=res(c1 :¬p(a,x,b) ∨ q(x) ∨ r(c) , c3 : ¬r(x))

  renommage de x en x3 dans c3,

  l'unificateur entre r(c) et ¬r(x3) est : [x3 ← c]

  c8= ¬p(a,x,b) ∨ q(x)

c9=res(c2 : ¬q(z) ∨ r(f(a)) ,   c3 : ¬r(x))

  l'unificateur entre r(f(a)) et ¬r(x) est : [x ← f(a)]

  c9= ¬q(z)

S1 = {c0 :p(x,f(y),b) ∨ q(a) ∨ r(y);   c1 : ¬p(a,x,b) ∨ q(x) ∨ r(c);   c2 : ¬q(z) ∨ r(f(a));   c3 : ¬r(x) ;
  c4 : q(a) ∨ r(y) ∨ q(f(y)) ∨ r(c) ; c5 : p(x,f(y),b) ∨ r(y) ∨ r(f(a)) ; c6= p(x,f(x2),b) ∨ q(a) ;
  c7= ¬p(a,z,b) ∨ r(c) ∨ r(f(a)) ;   c8= ¬p(a,x,b) ∨ q(x) ;   c9= ¬q(z)   }

étape 2 :

S2 = S1 ∪ {… c10=res(c6,c9) , c11=res(c8,c9) , ... }

c10=res(c6 :p(x,f(x2),b) ∨ q(a) , c9 :¬q(z))

l'unificateur entre q(a) et ¬q(z) est : [z← a]

c10= p(x,f(x2),b)

c11=res(c8 :¬p(a,x,b) ∨ q(x) , c9 :¬q(z))

l'unificateur entre q(x) et ¬q(z) est : [z← x]

c11= ¬p(a,x,b)

S2 = {… ; c10= p(x,f(x2),b) ; c11= ¬p(a,x,b) ; … }

étape 3 :

S3 = S2 ∪ {… c12=res(c10,c11) , ... }

c12=res(c10 :p(x,f(x2),b) , c11 : ¬p(a,x,b))

renommage de x en x3 dans c11,

l'unificateur entre p(x,f(x2),b) et ¬p(a,x3,b) est : [x← x3]

c12= ◊ (la clause vide)

Arrêt de l'algorithme avec succès, car l'ensemble S3 contient la clause vide

Donc l'ensemble initial **S est inconsistant**.

Exercice 4 :

On peut représenter les entiers en lambda-calcul pur comme ceci :
0 : λf.λx. x
1 : λf.λx. f x
2 : λf.λx. f (f x)
3 : λf.λx. f (f (f x))
… etc …

Quelle fonction unaire est alors représentée par l'expression suivante :

λn.λf.λx. n (λg.λh. h (g f)) (λu. x) (λu. u)

Appliquons la fonction au nombre 3 :
[λnfx. n (λgh. h (g f)) (λu. x) (λu. u)] [λf.λx. f (f (f x))] →
[λfx. [λf.λx. f (f (f x))] (λgh. h (g f)) (λu. x) (λu. u)] →
[λfx. [λx. (λgh. h (g f)) ((λgh. h (g f)) ((λgh. h (g f)) x))] (λu. x) (λu. u)] →
[λfx. [(λgh. h (g f)) ((λgh. h (g f)) ((λgh. h (g f)) (λu. x)))] (λu. u)] →
[λfx. [(λgh. h (g f)) ((λgh. h (g f)) ((λh. h ((λu. x) f))))] (λu. u)] →
[λfx. [(λgh. h (g f)) ((λgh. h (g f)) (λh. h (x)))] (λu. u)] →
[λfx. [(λgh. h (g f)) ((λh. h ((λh. h (x)) f)))] (λu. u)] →
[λfx. [(λgh. h (g f)) (λh. h (f (x)))] (λu. u)] →
[λfx. [(λh. h ((λh. h (f (x))) f))] (λu. u)] →
[λfx. (λh. h (f (f (x)))) (λu. u)] →
[λfx. ((λu. u) (f (f (x))))] →
[λfx. (f (f (x)))] = λfx. f (f (x)) = 2

Posons F_n l'expression [λf.λx. f (f … f (f x))] contenant n occurrences liées de f et représentant l'entier n

Hypothèse de récurrence : l'expression donnée représente la fonction **prédécesseur(n)**
[λnfx. n (λgh. h (g f)) (λu. x) (λu. u)] $F_n \rightarrow F_{n-1}$

Vérifions :
[λnfx. n (λgh. h (g f)) (λu. x) (λu. u)] $F_{n+1} \rightarrow F_n$

[λnfx. n (λgh. h (g f)) (λu. x) (λu. u)] F_{n+1} →
[λfx. F_{n+1} (λgh. h (g f)) (λu. x) (λu. u)] →
[λfx. [λx. (λgh. h (g f)) ((λgh. h (g f)) … ((λgh. h (g f)) x))] (λu. x) (λu. u)] →
[λfx. [(λgh. h (g f)) ((λgh. h (g f)) … ((λgh. h (g f)) (λu. x)))] (λu. u)] →
[λfx. [(λgh. h (g f)) ((λgh. h (g f)) … ((λh. h ((λu. x) f))))] (λu. u)] →
[λfx. [(λgh. h (g f)) ((λgh. h (g f)) … (λh. h x))] (λu. u)]
chaque sous terme de la forme (λgh. h (g f)) (λh. h x) → (λh. h ((λh. h x) f)) → (λh. h (f x))
comme il reste n sous termes de cette forme imbriquée vers la droite, ces réduction généreront un terme contenant n occurrence de f avec des application imbriquées vers la droite :
… →
[λfx. (λh. h (f (f … (f (x))...))) (λu. u)] →
[λfx. (λu. u) (f (f … (f (x))...)))] →
[λfx. (f (f … (f (x))...)))] = F_n

Donc l'hypothèse de récurrence est correcte, c'est donc bien la fonction prédécesseur(n) qui est représentée par l'expression : [λnfx. n (λgh. h (g f)) (λu. x) (λu. u)]

Exercice 5 :
Soit P le programme Lisp suivant :
```
(DE P(x y)
        (if (= y 0)  0   (+  (P  (* 2 x)  (Div  y  2))  (*  x  (Mod  y  2)) ) )
)
```
// Div et Mod les fonctions retournant respectivement le quotient et le reste de la division entière de deux nombres.
En utilisant le théorème du point fixe trouver la fonction exactement calculée par P.

La fonction exactement calculée par le programme P est la plus petite solution (le plus petit point fixe) de l'équation : P = T(P), c-a-d :
P = λx.λy. SI (= y 0) 0 (+ (P (* 2 x) (Div y 2)) (* x (Mod y 2))))

Soit O la fonction nul part définie (le plus petit élément de l'ensemble des fonctions ordonné par la relation « moins définie que »)

Le plus petit point fixe $f_0 = \mathrm{Sup}(T^n(O))_{n\geq 0}$

T(O) = λx.λy. SI (= y 0) 0 (+ (O (* 2 x) (Div y 2)) (* x (Mod y 2))))
comme le terme (+ (O (* 2 x) (Div y 2)) (* x (Mod y 2)))) n'est pas défini, la fonction T(O) peut s'écrire plus simplement comme :
T(O) = λx.λy. SI (= y 0) 0

T2(O) = λx.λy. SI (= y 0) 0 (+ (T(O) (* 2 x) (Div y 2)) (* x (Mod y 2))))
Le terme (T(O) (* 2 x) (Div y 2)) n'est défini que si y div 2 = 0, c-a-d y = 1 et dans ce cas, la valeur de ce terme est 0.

Dans ces mêmes conditions, le terme (+ (T(O) (* 2 x) (Div y 2)) (* x (Mod y 2)))) vaut alors : (+ 0 (* x (Mod 1 2))) = (+ 0 (* x 1)) = x
Donc la fonction T^2(O) est définie uniquement lorsque y = 0 ou y = 1. Sa valeur est :
T^2(O) = λx.λy. SI (= y 0) 0 (SI (= y 1) x)

T^3(O) = λx.λy. SI (= y 0) 0 (+ (T^2(O) (* 2 x) (Div y 2)) (* x (Mod y 2))))
Le terme (T^2(O) (* 2 x) (Div y 2)) n'est défini que si y div 2 = 0 ou y div 2 = 1, c-a-d y = 1 ou y = 2 dans ce cas, la valeur de ce terme est 0 si y = 1 et 2x si y = 2.
Dans ces mêmes conditions, le terme :
(+ (T^2(O) (* 2 x) (Div y 2)) (* x (Mod y 2)))) = (+ 0 (* x (Mod 1 2))) = (+ 0 (* x 1)) = x si y = 1
(+ (T^2(O) (* 2 x) (Div y 2)) (* x (Mod y 2)))) = (+ 2x (* x (Mod 2 2))) = (+ 2x 0) = 2x si y = 2
Donc la fonction T^3(O) est définie uniquement lorsque y = 0 ou y = 1 ou y = 2. Sa valeur est :
T^3(O) = λx.λy. SI (= y 0) 0 (SI (= y 1) x (SI (= y 2) 2x)

A ce stade on peut émettre une hypothèse de récurrence disant :
T^k(O) = λx.λy. x*y si $y < 2^{k-1}$

Vérifions alors que :
T^{k+1}(O) = λx.λy. x*y si $y < 2^k$

T^{k+1}(O) = λx.λy. SI (= y 0) 0 (+ (T^k(O) (* 2 x) (Div y 2)) (* x (Mod y 2))))
D'après l'hypothèse de récurrence, le terme :
(T^k(O) (* 2 x) (Div y 2)) vaut 2x(y div 2) si y div 2 < 2^k
dans le cas où y est pair, on aura y div 2 = y/2 et dans le cas où y est impair, y div 2 = (y-1)/2, donc :
(T^k(O) (* 2 x) (Div y 2)) vaut xy si y est pair et y/2 < 2^k
et
(T^k(O) (* 2 x) (Div y 2)) vaut x(y-1) si y est impair et (y-1)/2 < 2^k
De ce fait le terme :
(+ (T^k(O) (* 2 x) (Div y 2)) (* x (Mod y 2)))) vaudra alors :

(+ xy (* x 0)) = xy si y est pair et y < 2^{k+1} et

(+ x(y-1) (* x 1)) = xy si y est impair et y-1 < 2^{k+1} (c-a-d y < 2^{k+1} aussi)

on a donc bien : T^{k+1}(O) = λx.λy. x*y si $y < 2^k$

On en déduit donc que le plus petit point fixe ($T^{+\infty}$(O)) est :
f_0 = λx.λy. (* x y)