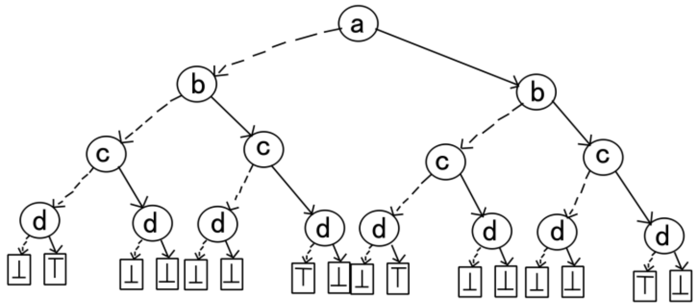
Q 15

~~6.2~~ 6.2 Canonicity Efficient Implementations

Problème de Model Checking : Explosion le l'espace pris par les états.
200 Philosophes -> $2.5*10^{125}$ états
Solutions :
- réduire l'espace de recherche
- Meilleure représentation de l'espace de recherche

SFDD (Set Family Decision Diagrams) :
- graphe acyclique dirigé
- chaque noeud représente un terme (ordonnés) (élément d'un ensemble)
- chaque noeud a 2 enfants : le terme est inclu ou non dans l'ensmble
- chaque chemin fini sur un terminal disant si l'ensemble appartient au système ou non

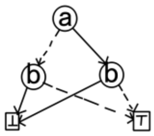
Exemple :
Termes : a<b<c<d
Ensembles :
{a,b,c}
{a,d}
{b,c}
{d}



Définition formelle des SFDD :
Soit T un ensemble de termes. L'ensemble des SFDD S est défini inductivement :
$\perp \in S$: le terminal rejetant
$\top \in S$: le terminal acceptant
$\langle t, \tau, \sigma \rangle \in S \Leftrightarrow t \in T,\ \tau \in S,\ \sigma \in S$
- noeuds avec :
- le terme t, sous-noeud acceptant τ (take node), sous-noeud rejetant σ (skip node)

La manière de représenter la famille d'ensemble n'est pas unique.
Ex : S={Ø,{a}} :

Forme brute :

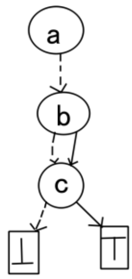


Forme normale :

Réductions : (forme brute->forme normale)
- Supression des noeuds négatifs (avec la branche d'acception qui va sur $\perp$)
- Partage des sous-arbres communs (implémentation, mémoire)

Clean : S->S supprime les noeuds négatifs :
clean($\perp$) = $\perp$
clean($\top$) = $\top$
clean($\langle t, \tau, \sigma \rangle$)= clean($\sigma$) si $\tau = \perp$
= $\langle$t,clean(τ),clean(σ)$\rangle$ sinon

Exemple :
S=$\langle a,\perp,\langle b,\langle c,\top,\perp\rangle,\langle c,\top,\perp\rangle\rangle\rangle$
clean(S)=clean($\langle b,\langle c,\top,\perp\rangle,\langle c,\top,\perp\rangle\rangle$)
=$\langle$b,clean($\langle c,\top,\perp\rangle$),clean($\langle c,\top,\perp\rangle$)$\rangle$
=$\langle$b,$\langle$c,clean(T),clean($\perp$)$\rangle$,$\langle$c,clean(T),clean($\perp$)$\rangle\rangle$
=$\langle b,\langle c,\top,\perp\rangle,\langle c,\top,\perp\rangle\rangle$



Forme canonique :
Soit S le SFDD $\langle t, \tau, \sigma \rangle$.
S est canonique ~~si~~ (informellement):
- le take node et le skip node sont des termes plus grands ou des terminaux
- le take node n'est pas $\perp$

Définition :
Les termes sont ordonnés par < qui est un ordre total. S est canonique si :
- S est $\perp$
- S est $\top$
- S=$\langle t, \tau, \sigma \rangle$ où :
  - $\tau = \langle t_\tau, \tau_\tau, \sigma_\tau \rangle \Rightarrow t < t_\tau$ et $\tau \neq \perp$
  - $\sigma = \langle t_\sigma, \tau_\sigma, \sigma_\sigma \rangle \Rightarrow t < t_\sigma$ ou $\sigma = \perp$ ou $\sigma = \top$
  - τ et σ sont canoniques

(avantages de la forme cananique : mémorisation de toutes les opérations qu'il y a eu dans l'exécution)

L'implémentation permet de réduire encore.
Nous pouvons définir une équivalence : deux arbres sont similaires si :
$\perp \equiv \perp$
$\top \equiv \top$
$\langle t, \tau, \sigma \rangle \equiv \langle t, \tau', \sigma' \rangle$ si $\boxed{\tau \equiv \tau'}$ et $\boxed{\sigma \equiv \sigma'}$
La structure réelle S=clean(Sbrute)/≡ (est alors la structure clean quotientée par la relation d'équivalence.)
- Les sous-arbres équivalents sont partagés : gain de place mémoire
Les différents SFDD sont gardés à travers des références, les skip et take node contiennent des références
-> 2 SFDD équivalents ne sont gardés que par une seule référence
-> partage d'une partie de l'information
- Nous pouvons faire de la mémorisation (une fonction calculée n'a pas besoin d'être recalculée) :
gain de temps de calcul

Exemple : SFDD S = <a,<b,T,T>,<b,T,T>>. Nous aurions alors à l'implémentation : <a,#0x14,#0x14>
L'adresse #0x14 contient <b,T,T>.
Avec la représentation sous forme d'ensemble il n'y aurait pas de partage d'information : {{},{a},{b},{a,b}}