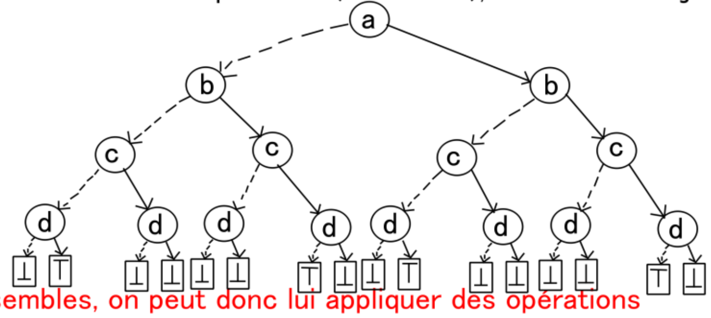
Q16 6.3

Operations on SFDD

Définition formelle des SFDD :
Soit T un ensemble de termes. L'ensemble des SFDD S est défini inductivement :
$\bot \in S$: 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)

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



SFDD est une famille d'ensembles, on peut donc lui appliquer des opérations

Union :
$A \cup B = B \cup A$ // $A \cup A = A$ // $\bot \cup A = A$ ($\bot$ = famille vide)
$\top \cup \langle t, \tau, \sigma \rangle = \langle t, \tau, \top \cup \sigma \rangle$ ($\top$= ensemble vide)

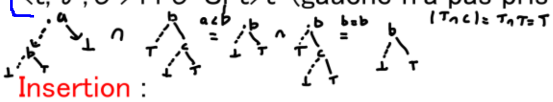
si t != t', on reste tjrs avec le plus petit des deuxet o

$\langle t, \tau, \sigma \rangle \cup \langle t', \tau', \sigma' \rangle =$
- $\langle t, \tau, \sigma \cup \langle t', \tau', \sigma' \rangle \rangle$ Si t<t' (l'arbre de droite a rejeté t (t<t') mais on en a besoin donc on fais l'union sur skip de t)
- $\langle t, \tau \cup \tau', \sigma \cup \sigma' \rangle$ Si t=t'
- $\langle t', \tau', \sigma' \cup \langle t, \tau, \sigma \rangle \rangle$ Si t>t'

si t=t' -> on fait inter de acce

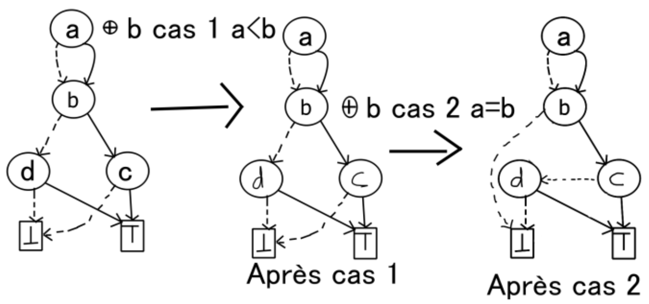
sinon, on prend le rejetant

Intersection :
$A \cap B = B \cap A$ // $A \cap A = A$ // $\bot \cap A = \bot$
top $\top \cap \langle t, \tau, \sigma \rangle = \top \cap \sigma$ (garder que l'ensemble vide, les noeuds pris ne sont pas dans le résultat)
$\langle t, \tau, \sigma \rangle \cap \langle t', \tau', \sigma' \rangle =$
- $\sigma \cap \langle t', \tau', \sigma' \rangle$ Si t<t' (droite a rejeté t (t<t'), donc on enlève les taken nodes de t (cas où t est choisi))
- $\langle t, \tau \cap \tau', \sigma \cap \sigma' \rangle$ Si t=t'
- $\langle t, \tau, \sigma \rangle \cap \sigma'$ Si t>t' (gauche n'a pas pris t', donc on enlève les taken nodes de t')

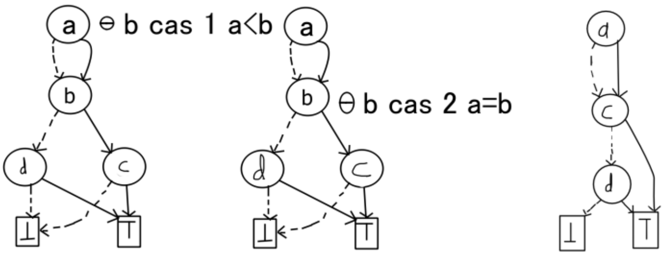


Insertion :
$\bot \oplus a = \bot$ (la famille vide n'est pas modifiée)
TOP $\top \oplus a = \langle a, \top, \bot \rangle$ car $\{\emptyset\} \oplus a = \{\{a\}\}$
$\langle t, \tau, \sigma \rangle \oplus a =$
$\langle t, \tau \oplus a, \sigma \oplus a \rangle$ Si t<a (ajout de a que t soit pris ou pas)
$\langle t, \tau \cup \sigma, \bot \rangle$ Si t=a (on ne skip plus rien car on prend t=a)
$\langle a, \langle t, \tau, \sigma \rangle, \bot \rangle$ Si t>a (ajout de a en haut)
$\oplus$ est un homomorphisme : $(A \cup B) \oplus a = (A \oplus a) \cup (B \oplus a)$

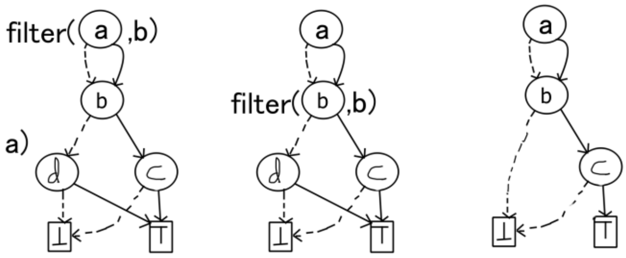
Exemple : enc({a,b,c}, {a,d}, {b,c}, {d})⊕b



Suppression :
$\bot \ominus a = \bot$ // $\top \ominus a = \top$
$\langle t, \tau, \sigma \rangle \ominus a =$
$\langle t, \tau \ominus a, \sigma \ominus a \rangle$ si t<a
$\tau \cup \sigma$ si t=a (on garde le reste)
$\langle t, \tau, \sigma \rangle$ si t>a (plus de a à enlever)
$\ominus$ est un homomorphisme



Filtrage : Supprimer une structure ne contenant pas a
filter($\bot$,a)=$\bot$ // filter($\top$,a)=$\bot$ (on enlève l'ensemble vide)
filter($\langle t, \tau, \sigma \rangle$,a)=
$\langle t$,filter(τ,a),filter(σ,a)$\rangle$ si t<a
$\langle t, \tau, \bot \rangle$ si t=a (on enlève les skip nodes qui ne prennent pas a)
$\bot$ si t>a (on a pas trouvé a)



filter est un homomorphisme