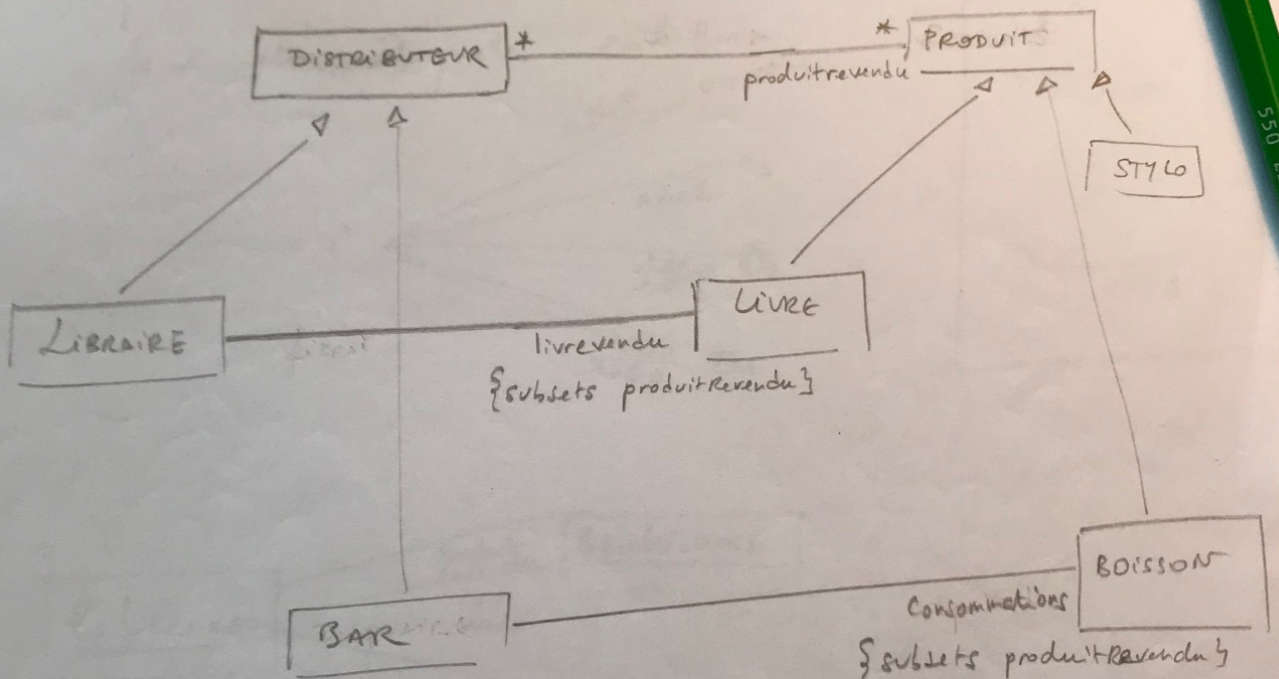


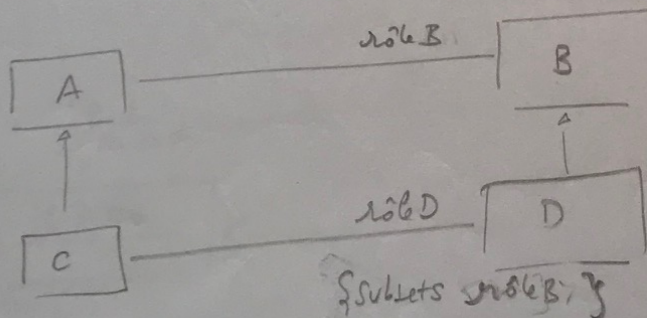
1
PROPRIÉTÉS SUR LES
EXTREMITÉS D'ASSOCIATION

- En UML 2.0 on retrouve `{ordered}` par ex.
Mais également : `subsets <en>`
`redefined <en>`
`union`

Subsets <en>



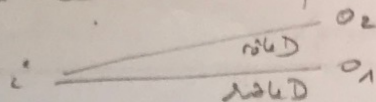
Subsets



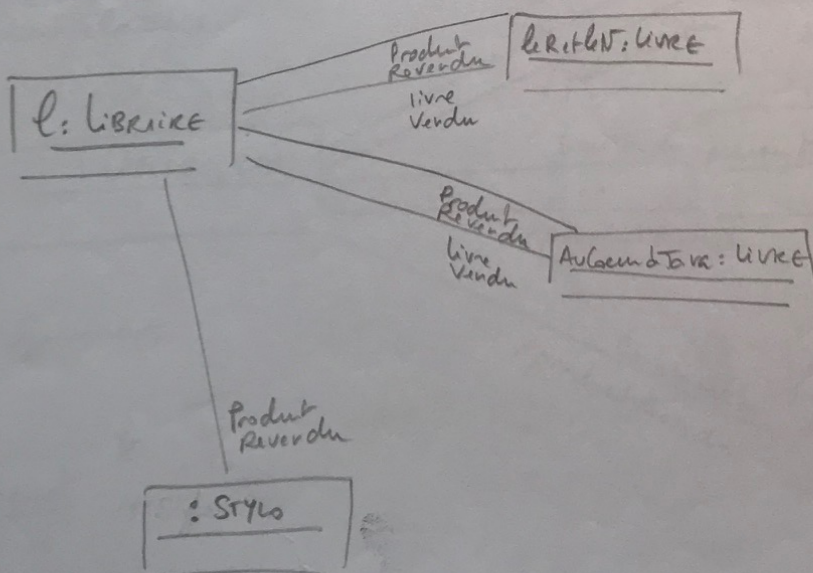
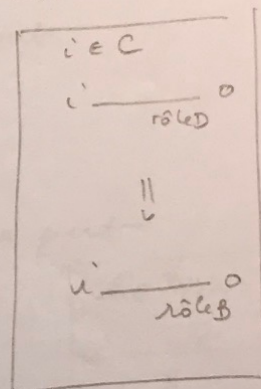
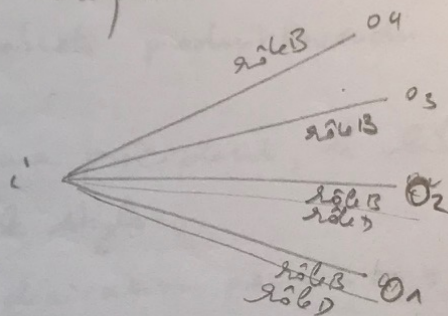
Donner un objet de la classe C, le rôle de D

(2)

Pour une instance i de la classe C ,
les objets qui seront extrémités d'un
lien partant de i étiqueté $rolD$ ---



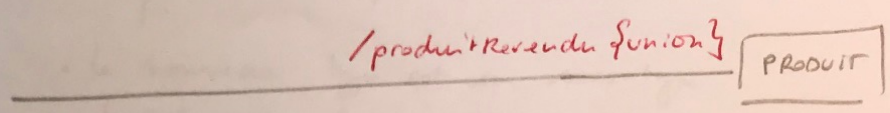
forment un sous-ensemble des objets
qui seront extrémités d'un lien partant
de i et étiqueté $rolB$



• Union

3

Si on ajoute {union} près du rôle produitRevenu
(qui doit être dérivé) ~~rôle~~



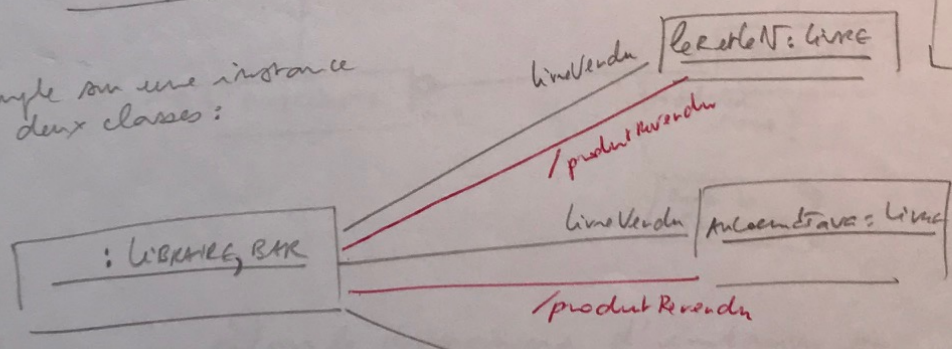
L'ensemble des produits ~~Revenus~~ est dérivé
de l'union stricte des rôles qui sont
déclarés {sous-sets produitRevenu}.

Sur le schéma précédent, le libraire ne peut
plus vendre le stylo.

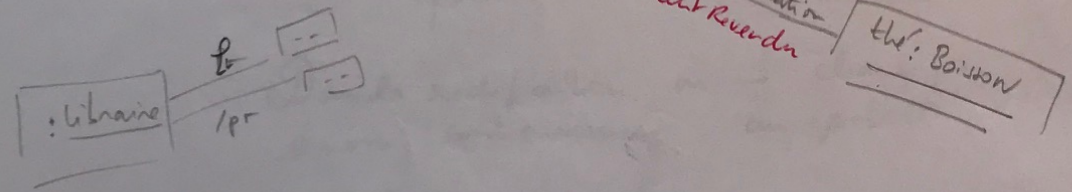
Exemple de dérivation par union :

A ∈ Distib	
i	/pr 0
i	/pr 0
i	ou 0
i	0

exemple sur une instance
des deux classes :



mais également :



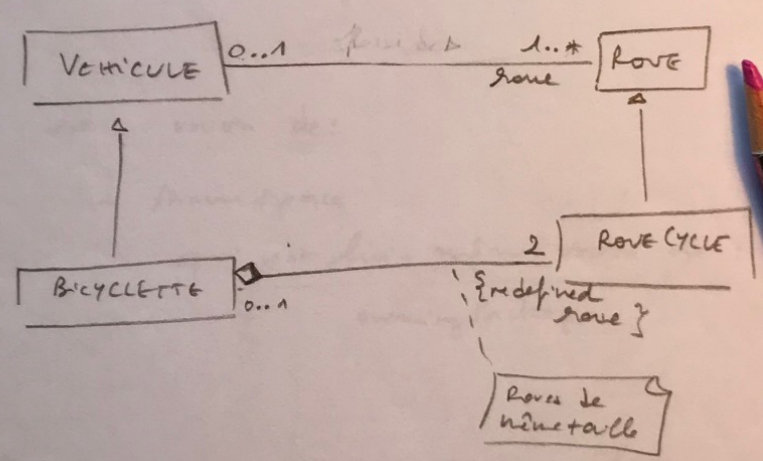
Redefined <en>

Indique que l'extrémité redéfinit en.

Qui est-ce qu'une redéfinition valide?

- le nouveau type est un sous-type de celui de la propriété redéfinie
- la multiplicité est incluse ..
- si la pté redéfinie est dérivée, la redéfinition aussi.

[Ex. ? des Contraintes sur subsets et ~~subset~~ union?]



- Dans le diagramme d'instances, on se rend pas le lien rove 1 quand on a rove 2

- Dans la redéfinition, on ne change pas le nom, contrairement au subset

Et en UML 2.5,

on a aussi de la spécialisation entre associations

