BUT Informatique
1A - Semestre 1
Introduction aux bases de données
(R1.05)

R. Fleurquin

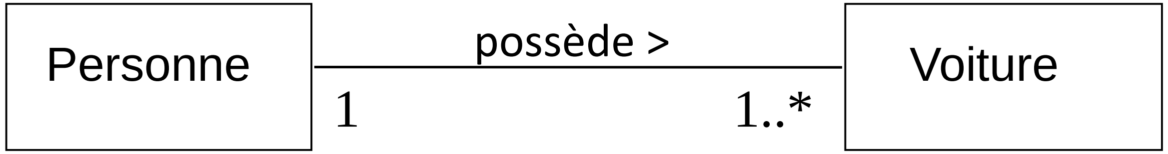
Chapitre 5

Diagramme de classes UML : approche formelle des associations binaires

3 choses à maîtriser absolument sur les associations binaires

1. Les cardinalités n’empêchent pas les diagrammes d’objets d’évoluer dans le temps => Notion d’état instable de durée (théorique) nulle
2. Pour une même association il est interdit d’avoir plus d’un lien reliant une (cas d’une association réflexive) ou deux instances (cas d’une association binaire entre deux classes)
3. **Attention aux associations binaires réflexives**! Par défaut :
   - Elles autorisent un lien d’un objet vers lui-même.
   - Elles autorisent un lien de l’objet A vers B et un autre de B vers A.
   - Les cardinalités minimales >0 imposent nécessairement des cycles dans les diagrammes d’objets ou des diagrammes infinis (pas pratique sur nos ordinateurs ;))

Point 1 : L’invariant temporel qui fait (très) mal à vitesse finie!



Durant l’exécution du logiciel, il faut en théorie respecter ce diagramme à tout instant t : « une personne possède au moins 1 voiture et une voiture est la propriété nécessairement d’une et une seule Personne ».

Mais sur un plan pratique nos processeurs (même avec plusieurs cœurs), disques, bus travaillent à des vitesses qui ne sont pas infinies!

Problème à t=0 (**État initial cohérent**). A l’origine Il est impossible de créer une personne, une voiture puis le lien entre les deux dans la base de données de notre logiciel « simultanément ». Chacune de ses actions prend du temps. De plus la création d’un lien présuppose l’existence des deux objets à lier en amont…

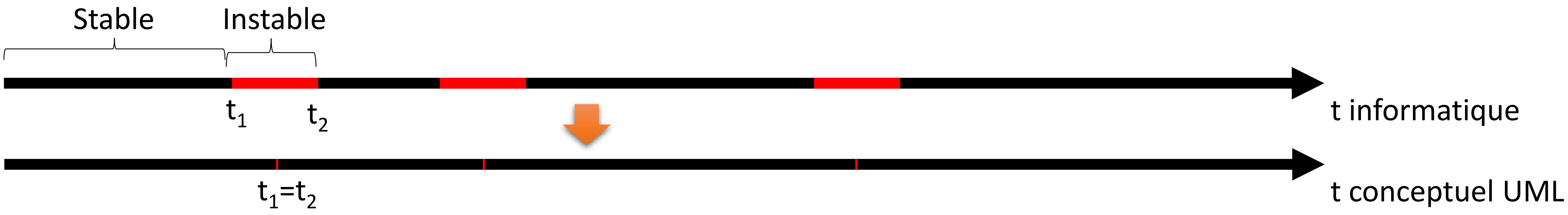
Problème à $\forall t>0$ (**Évolutivité de l’information manipulée**) : le domaine modélisé ne souhaite sans doute pas empêcher le fait qu’une personne puisse revendre une voiture à une autre et éventuellement qu’il en achète de nouvelles. On ne souhaite pas figer dans le marbre un système d’information depuis l’origine jusqu’à la fin des temps! Mais il faudrait pour cela être en capacité, là encore, de simultanément supprimer des liens et d’en créer de nouveaux….

État stable et instable du système modélisé

On accepte que pendant un délai considéré comme conceptuellement nul (mais en pratique non nul!) le système modélisé passe par un état qualifié d'<u>instable</u> pour lui permettre d'« évoluer » avant de revenir nécessairement à un état respectueux de l'invariant qualifié lui d'état <u>stable</u>.

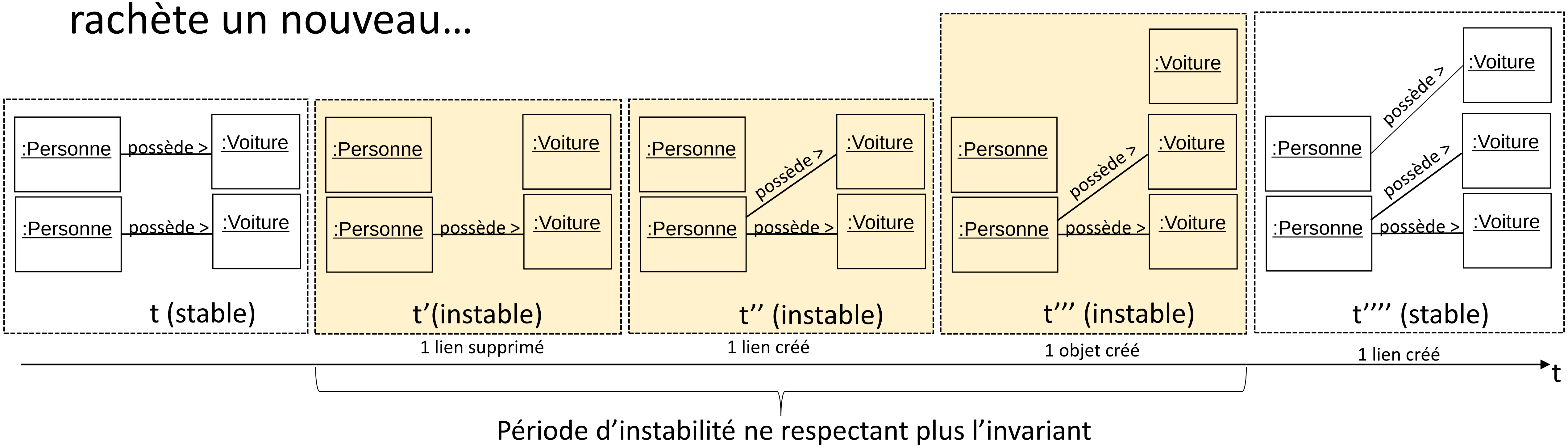
- Pendant cette période (de durée conceptuelle nulle) d'instabilité tout est possible! Perdre des liens entre objets, créer des liens entre objets, faire mourir ou naître des objets, changer des valeurs d'attributs sans pour autant respecter le diagramme de classes.

- La vie du système est une succession de périodes de stabilité et de périodes instables (ces dernières supposées de durée nulle et qui ne « comptent » pas).



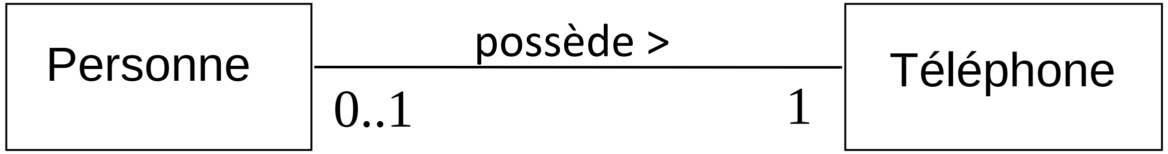
Exemple

Un automobiliste vend à l'instant t son unique véhicule à un autre et en rachète un nouveau…



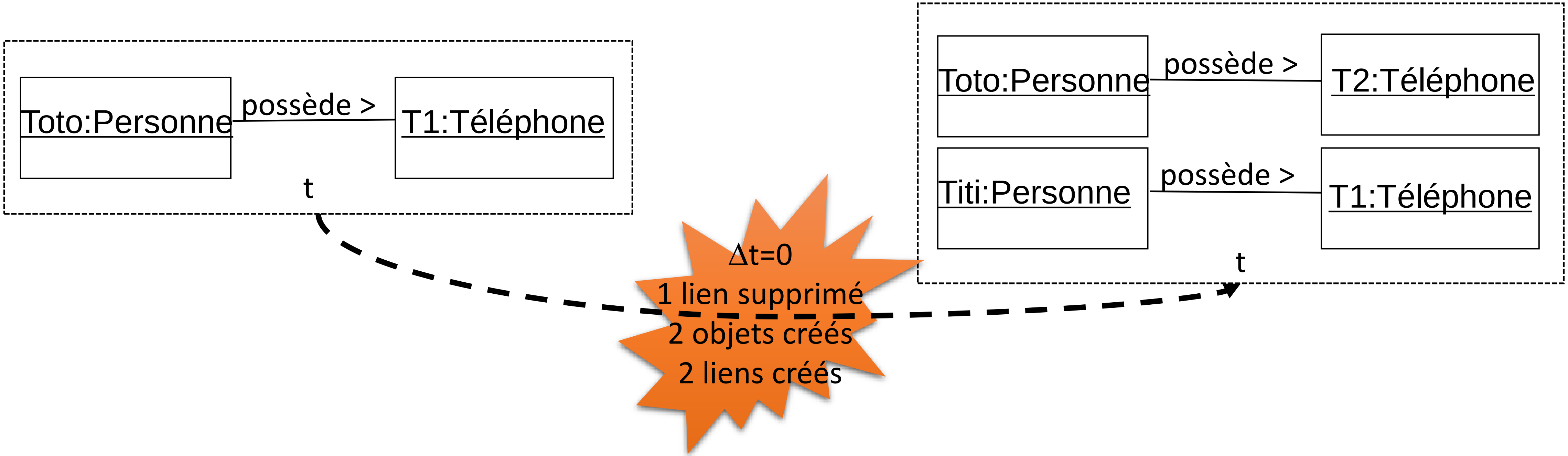
Conceptuellement en UML , on suppose que tout cela se fait en temps nul soit t=t''''
La partie en jaune qui ne respecte pas l'invariant n'est pas comptée.
On fait comme ci on passait de l'état initial au final *instantanément*.

L’invariant n’empêche donc jamais l’évolution d’un état compatible vers un autre état compatible peu importe la quantité de changements



Ce diagramme précise qu’à chaque instant une personne a un téléphone.

Mais il n’empêche pas cette personne de changer de téléphone au cours du temps car on suppose que les opérations sur les données (création/suppression de liens ou d’objets) à faire pour réaliser ces changements peuvent se faire en temps nul.

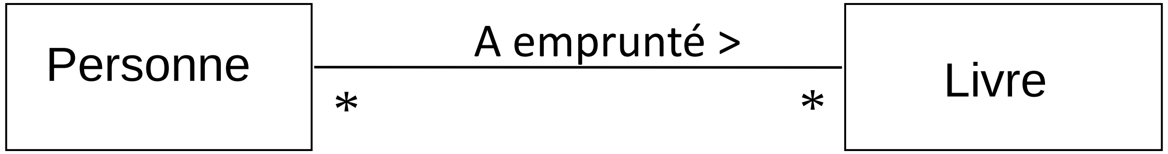


Point 2 et 3 : L’association binaire UML c’est des maths!

Une association binaire UML définit entre les ensembles d’instances des deux classes participantes une famille de *relations binaires* mathématiques indexée par le temps (cf. cours de math R1.06 de mathématiques discrètes ou fin de ce poly).

Ce fondement mathématique a des implications fortes qu’il faut bien maîtriser!

Premièrement : on interdit les doublons de lien!



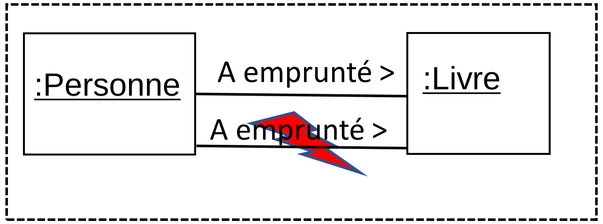


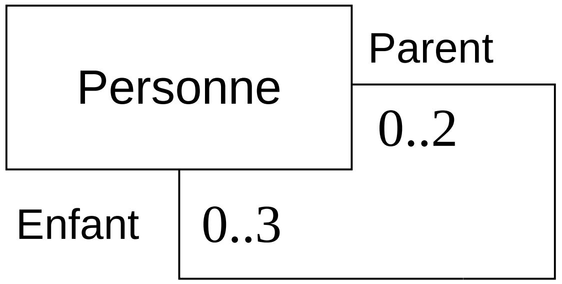
Diagramme d'objets non compatibles

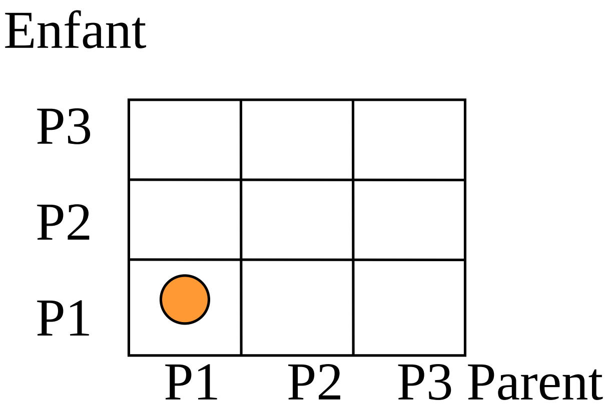
Comme une relation est un ensemble. Un même couple (personne, livre) ne peut apparaître qu'une seule fois et donc il ne peut y avoir qu'un seul lien instance de l'association entre deux objets.

Il faut donc lire : « Une personne a emprunté un nombre quelconque de livres <u>différents</u>. Un livre a été emprunté par un nombre quelconque de personnes <u>différentes</u>. »

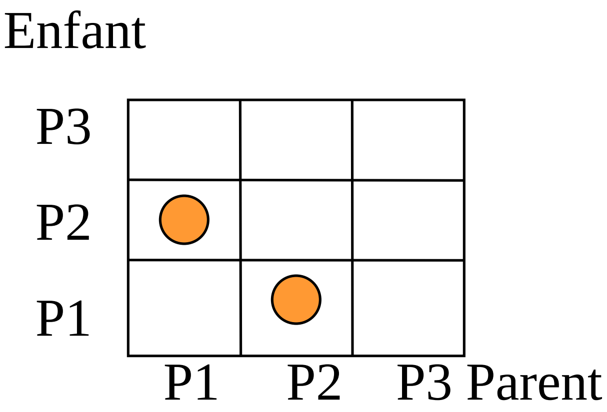
Avec ce diagramme de classes il est donc impossible de mémoriser le fait qu'une personne a emprunté plusieurs fois le même livre…

Les associations réflexives : très utiles mais… permissives!

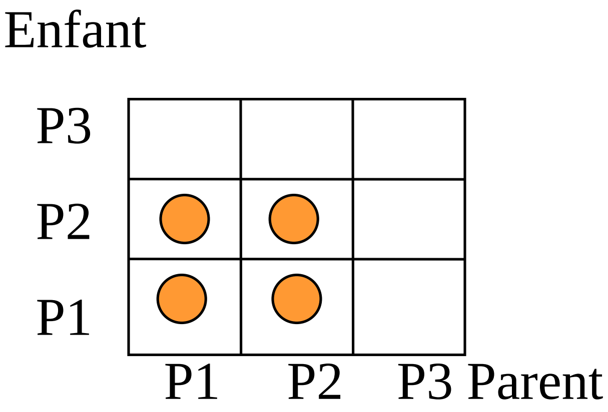




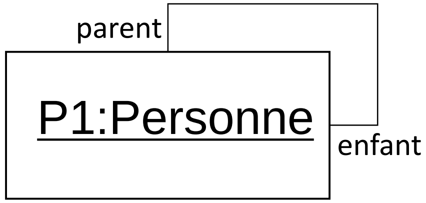
Une personne peut être son propre enfant et parent. Mais une seule fois! $(a,a) \in \mathcal{A}\times\mathcal{A}$

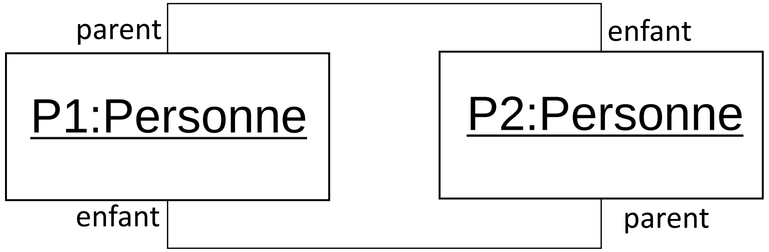


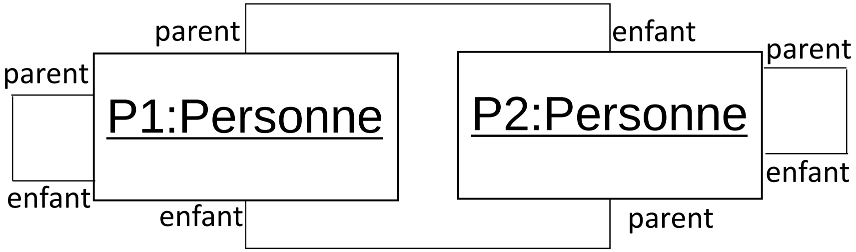
Une personne peut avoir un enfant qui est son parent $(a,b)\neq(b,a) \in \mathcal{A}\times\mathcal{A}$



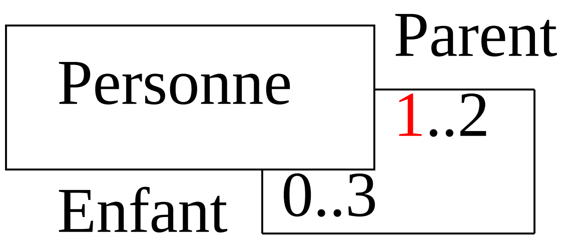
Avec deux Personnes on peut avoir jusqu'à 4 liens! $(a,b)\neq(b,a)\neq(a,a)\neq(b,b) \in \mathcal{A}\times\mathcal{A}$







Méfiez vous des cardinalités minimales >0 dans les réflexives



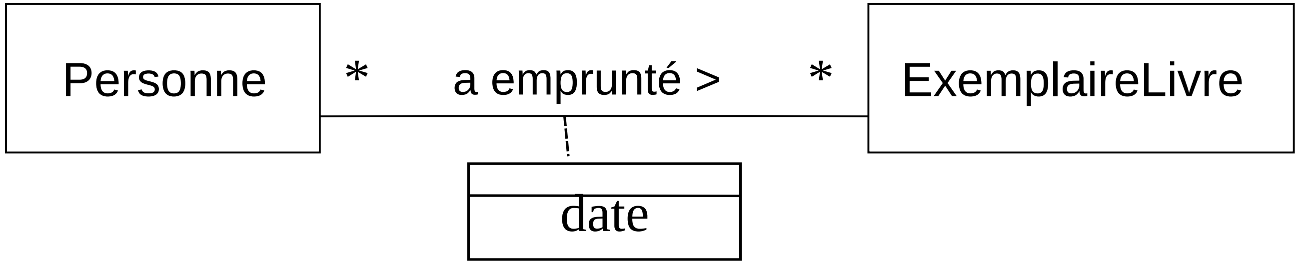
- Une personne a au moins un Parent, qui a à son tour au moins un parent, etc… C'est une progression potentiellement infinie sauf à autoriser l'impensable : qu'une personne ait en parent l'un de ses descendants…!

- Très souvent, on règle le problème (si cela a un sens) en posant à 0 toutes les cardinalités minimales dans une réflexive.

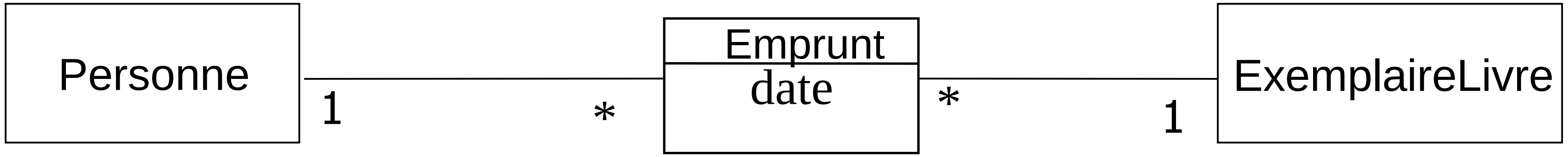
Intérêt de maitriser cette formalisation

Vous devez toujours prendre garde à ne pas confondre les sémantiques que vous affichez au travers des nommages et rôles et les contraintes effectivement (mathématiquement) assurées par la notation UML

Exemple Classique : le modèle d’une bibliothèque dans laquelle, on veut conserver l’historique des emprunts

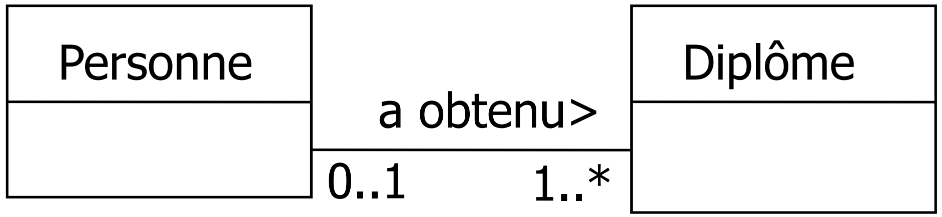


Une personne ne peut avoir emprunté qu’une seule fois un ouvrage
car un couple (a,b) ne peut apparaître qu’une seule fois dans $\mathcal{A}\times\mathcal{B}$.
Le modèle proposé est donc faux.



Le modèle est presque correct!
Mais il n ’empêche pas qu’un même exemplaire puisse être emprunté à la même date :
par deux personnes différentes (ubiquité!)
ou par la même personne (absurdirté!)

Exemple classique 2

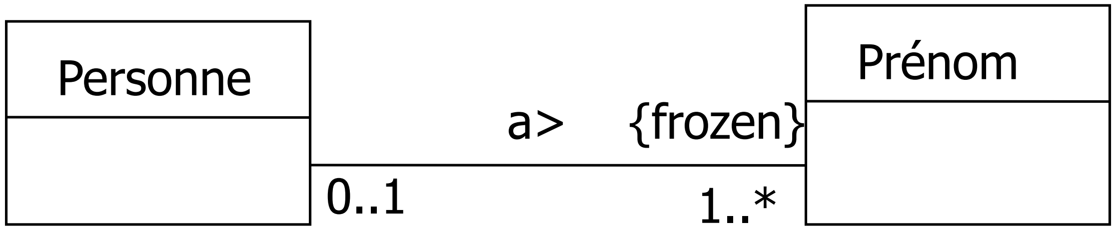


- Ce que dit bien ce diagramme :
- A chaque instant une personne a au moins un diplôme
- A chaque instant un diplôme n’est possédé que par au plus une personne
- et ce qu’il ne dit pas!
- Une personne conserve (à vie) les diplômes obtenus (pas de suppression de liens ou d’instances)
- Un diplôme est nominatif (pas de réaffectation d’un diplôme d’une personne à une autre)
- Le nombre des diplômes est figé (pas de création d’un nouveau diplôme pour une personne)

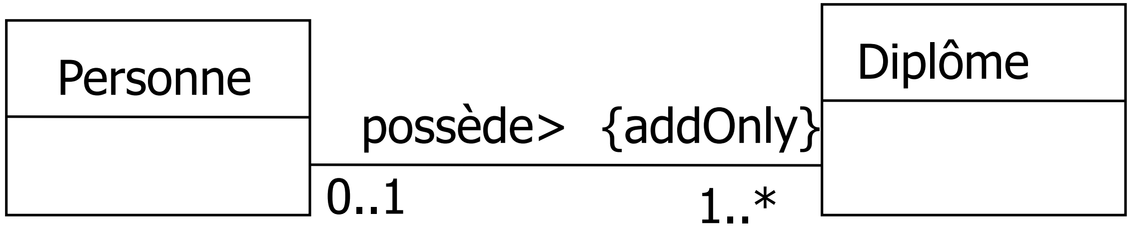
frozen : on ne peut ni supprimer ni rajouter

addOnly : on ne peut pas supprimer

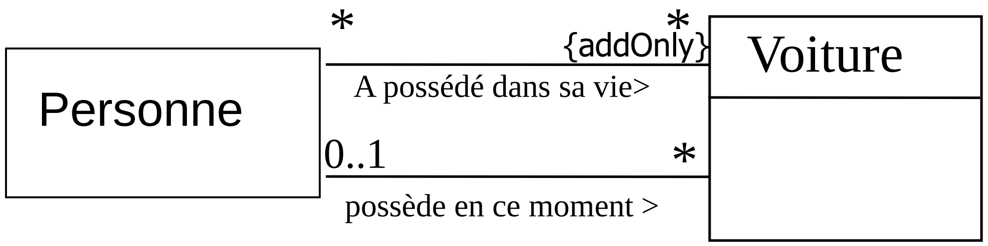
- Ainsi selon le domaine :
- Une personne à des prénoms définis une fois pour toute à sa naissance



- Un diplôme un fois acquis, l’est pour la vie OUF!



Le problème de l'archivage



Ce modèle empêche qu'une voiture qui a été possédée à t_n par une personne, ne le soit plus à $t_m > t_n$.

Cependant, il n'impose pas en rien qu'une voiture possédée puis cédée soit archiveée comme ayant été possédée!

En fait ici, on suppose que le modèle objet à l'instant t est la vérité ultime et qu'il n'est pas modifié dans les traitements dans un sens contraire à la sémantique qu'il cherche à exprimer (ici en n'archivant pas une possession qui se termine).

- Dans les faits, on ne marque pas ces contraintes implicites à la sémantique du nommage pour ne pas alourdir les diagrammes. Il faudra cependant veiller au respect de ces contraintes lorsque l’on écrira la partie traitement
- Ne décorez donc pas les associations de contraintes *addOnly* ou *frozen*

Annexe : Définition mathématique de l’association binaire

Ensemble et élément

Ensemble : "collection d'entités <u>toutes distinctes".</u>

Les entités sont appelé "<u>éléments</u>". <u>L'ensemble vide</u> se note $\emptyset$.

Deux modes de définition :

- <u>en extension</u> :
- <u>en compréhension</u> :

$$A = \{1,2,7\}$$

$$B = \{x / (x \in \Re) et (x^2 > 1)\}$$

Il n'y a pas d'ordre dans un ensemble : {a,b}={b,a}

L'assertion définissant l'<u>appartenance</u> d'un élément x à un ensemble E se note : $x \in E$ (et sa négation $x \notin E$)

Exemple $1 \in A$ et $-3 \in B$ et $0 \notin B$

Partie (sous ensemble) d'un ensemble

Un sous ensemble ou partie d'un ensemble E est un ensemble composé d'éléments piochés dans E.

Le fait d'être sous ensemble se note par le symbole inclusion (non stricte) :

- si A={1,2} et B={1,2,3} on a $A \subseteq B$
- $\{1\} \subseteq \{1\}$
- $\emptyset \subseteq \{1,a\}$
- Par contre il est faut de dire que $\{1,3\} \subseteq \{1,2\}$

Le plus grand sous ensemble d'un ensemble est lui-même.

Ensemble des parties

L’ensemble des parties d’un ensemble E noté P(E): “Ensemble de tout les sous ensembles que l’on peut construire sur E”.

Exemple : A={a,b}, P(A)={$\emptyset$,{a},{b},{a,b}}

Notez que lorsque l’on passe au P(E) d’un ensemble E on obtient un nouvel ensemble dont les éléments sont des ensembles (on change la nature des contenus) !

{a}$\in$P(A)!

Evitez la confusion $\in$ et $\subseteq$

Un élément est différent de l'ensemble qui le contient : $a \neq \{a\}$

L'inclusion $\subseteq$ compare deux entités de type ensemble.

L'appartenance $\in$ distingue deux entités de niveaux différents (l'un est élément pour l'autre qui est un ensemble).

Exemple : $A=\{a,b\}$, $P(A)=\{\emptyset,\{a\},\{b\},\{a,b\}\}$

Il est vraie de dire :

$a \in A$ (a est bien élément de l'ensemble A)

$\{a\} \in P(A)$ (l'ensemble {a} est bien un élément de P(A))

$\{a\} \subseteq A$ ({a} et A sont des ensembles}

$\{\{a\}\} \subseteq P(A)$ (notez les doubles accolades sinon c'est faux!)

Il est faux de dire :

$\{a\} \subseteq P(A)$ car l'ensemble {a} n'est pas construit avec des éléments piochés dans P(A)

$a \subseteq A$ ou $a \subseteq a$ (car a n'est pas un ensemble, cela n'a donc aucun sens de l'écrire).

Cardinal d’un ensemble

Le cardinal d’un ensemble est le nombre de ses éléments.

$card(\{1,2,a\})=3$

$card(\mathcal{P}(A))=2^{card(A)}= 2^3 = 8$

Certains ensembles ont un cardinal fini d’autres infini (comme IN ou IR).

Couple

- On appelle couple de deux éléments x et y noté (x,y) l'ensemble :
  - $(x,y)=\{\{x\},\{x,y\}\}$.
- Cette définition met en valeur le caractère ordonné d'un couple :
  - $(a,b)=\{a,\{a,b\}\} \neq (b,a)=\{b,\{a,b\}\}$
- Cet ensemble très utile vérifie la propriété d 'ordre suivante :
  - $(a,b) =(a',b') \Leftrightarrow a=a'$ et $b=b'$

Produit cartésien

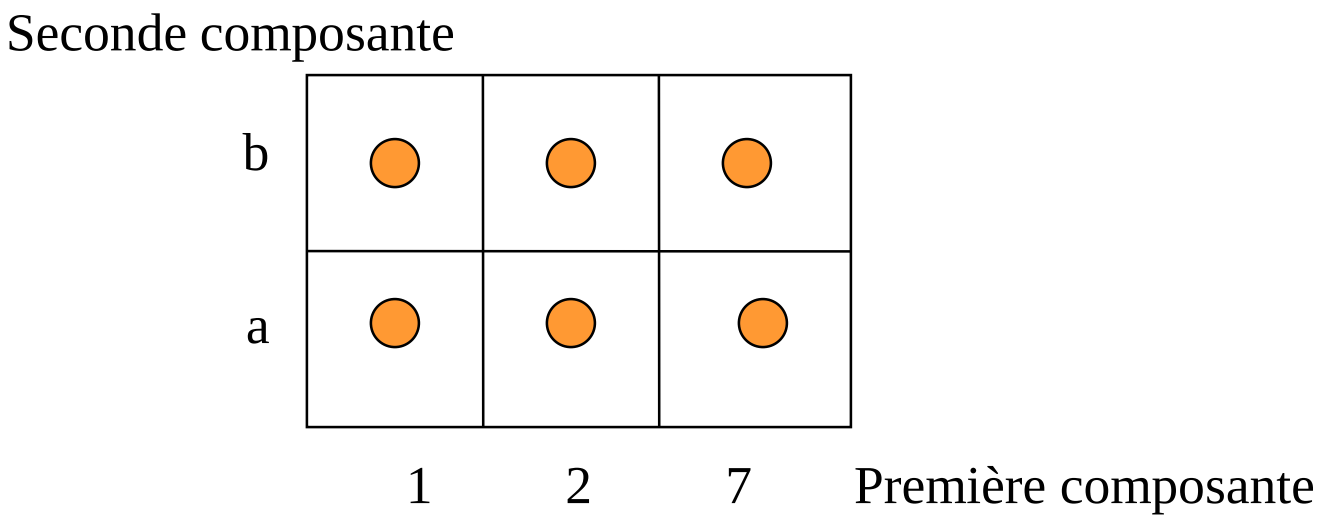
On appelle produit cartésien de A et B noté $\mathcal{A}\times\mathcal{B}$, l'ensemble des couples (x,y) tels que $x\in\mathcal{A}$ et $y\in\mathcal{B}$.

Comme c'est un ensemble il ne peut pas y avoir deux fois le même couple!

$$\{1,2,7\}\times\{a,b\}=\{(1,a),(1,b),(2,a),(2,b),(7,a),(7,b)\}$$

Représentation tabulaire d’un produit cartésien : le diagramme cartésien binaire

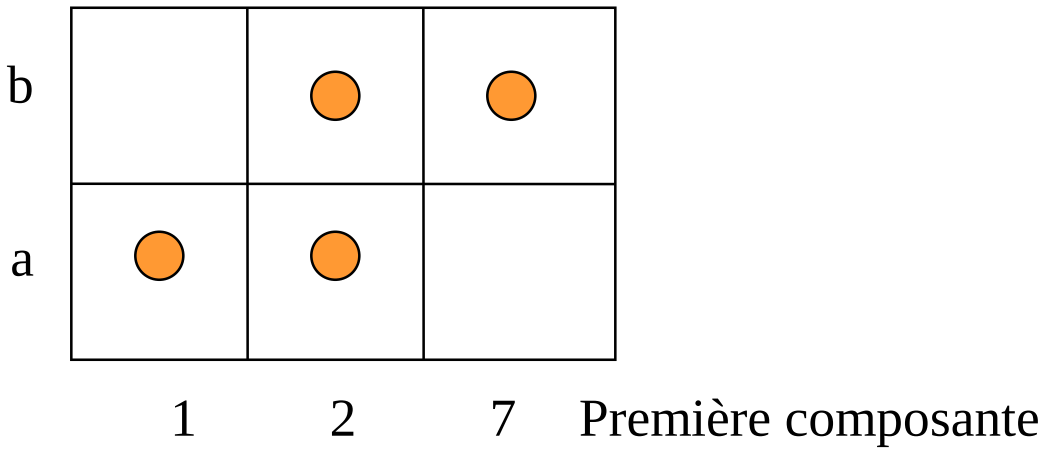
$$\{1,2,7\}\times\{a,b\}=\{(1,a),(1,b),(2,a),(2,b),(7,a),(7,b)\}$$



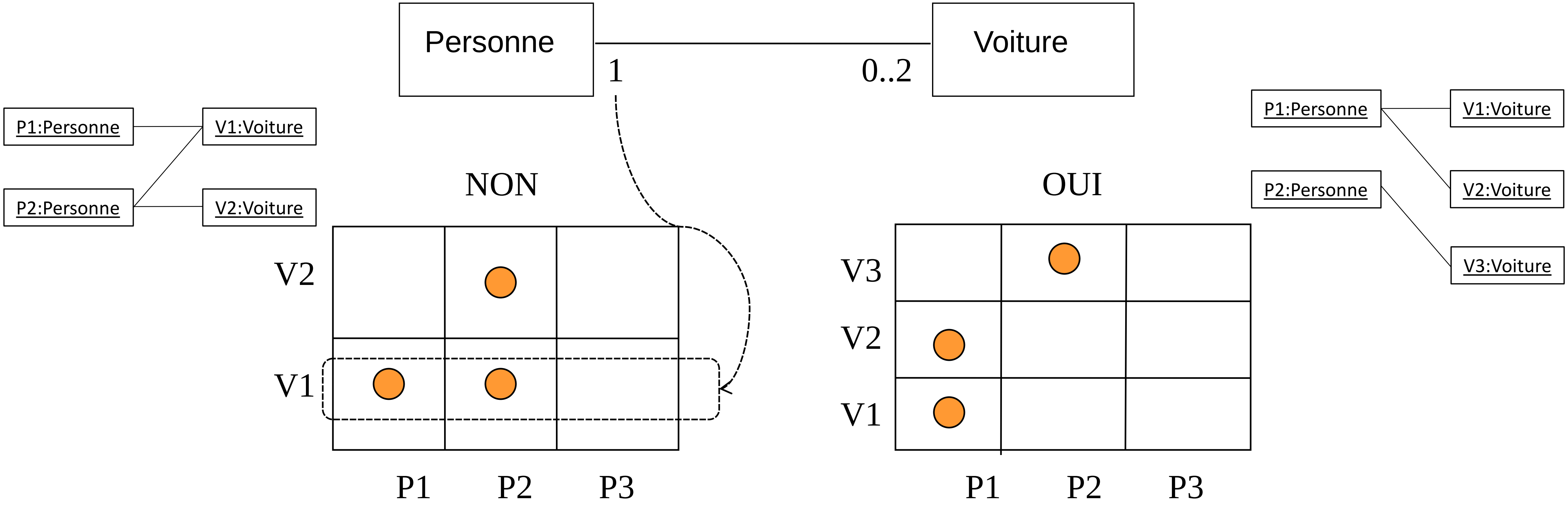
Relation binaire

Une relation binaire est une partie du produit cartésien de 2 ensembles $\mathcal{R} \in \mathcal{P}(\mathcal{A}\times\mathcal{B})$
ou de façon équivalente $\mathcal{R} \subseteq (\mathcal{A}\times\mathcal{B})$

$$R = \{(1,a),(2,a),(2,b),(7,b)\} \subseteq \{1,2,7\}\times\{a,b\}$$



L'association binaire UML a été construite sur la relation binaire mathématique entre les ensembles d'instances des deux classes

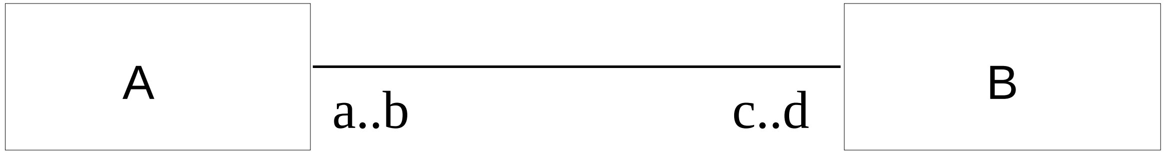


- Une association $\mathcal{A}$ définit donc une famille de relation $\mathcal{R}_{\mathcal{A}}(t)$ indexée par le temps.

- A tout instant t l'ensemble des liens instances de l'association A doit constituer une relation mathématique $\mathcal{R}_{\mathcal{A}}(t)$ vérifiant les contraintes de cardinalité de l'association A.

- Les cardinalités expriment donc des invariants temporels (une propriété qui perdure à tout instant stable du système modélisé).

Let’s go…



- Soit $I(\mathcal{A},t)$ *et* $I(\mathcal{B},t)$ l’ensemble des instances de $\mathcal{A}$ et $\mathcal{B}$ à l’instant t
- Soit $\mathcal{R}(t)$ la relation issue des liens instances de l’association à l’instant t

$\forall t,$

$[\,(\forall\ c_1 \in I(\mathcal{A},t),\ c \leq card(\{\, c_2 \in I(\mathcal{B},t) \mid (c_1,c_2) \in \mathcal{R}(t)\}) \leq d\ )$

Un A est lié au minimum à `c` B et au maximum `d`

$\mathcal{ET}$

$(\forall\ c_2 \in I(\mathcal{B},t),\ a \leq card(\{\, c_1 \in I(\mathcal{A},t) \mid (c_1,c_2) \in \mathcal{R}(t)\}) \leq b\ )]$

Un B est lié au minimum à `a` A et au maximum `b`