

Spécification

- Description :
 - 1) **Abstraite** : pas trop de détail
 - 2) **Clarté et précision**: sans Ambiguïté

—————→ Langages (**formalismes**) de spécification
ou de modélisation

Si on utilise des représentations graphique, on parle aussi de modélisation et de modèle

Langages de Sépcification

- Informels: langage naturel, table, ...;
- Semi-formels: UML, DFD, DEA, ...;
- Formels;

Langages Formels

Qualités d'un
langage formel



1. *L'expressivité*
2. *La vérifiabilité*
3. *L'abstraction*
4. *La lisibilité*

Aspects d'un langage formel:

- 1) Une syntaxe bien définie (alphabet, mots, règles d'écriture) et une sémantique formelle (interprétation)
- 2) Un fondement mathématique : permettre la vérification et la preuve

Langages Formels: classes

- Les approches basées sur **la logique** : théorie des ensembles (le langage Z, Vienna Development Method (VDM), méthode B), logiques temporelles (TRIO, TLA, TLCO), logiques d'ordre supérieur (Coq, PVS), logique linéaire.
- Les approches **algébriques** : algèbres de processus (CCS, LCS, LOTOS, RTL).
- Les approches par **modèles à états**: tels que les systèmes de transitions (avec StateCharts, SDL) et les Réseaux de Petri.

Vérification

- Vérification: **prouver** qu'un système vérifie bien certaines **spécifications** (des propriétés).
- preuve **mathématique** du **bon fonctionnement** d'un **modèle** du système.

Que peut on vérifier?

- On ne peut pas vérifier tout dans un système
- Plutôt il y'a un ensemble **limité** de propriétés qui sont **possibles à vérifier** et qui sont aussi **nécessaires à vérifier**.

Propriétés à vérifier?

- **Atteignabilité**: un certain état est atteignable durant l'exécution du système.

Exemple: *Il existe un moyen pour que la machine serve du café*

- **Vivacité**: possibilité d'exécuter une action dans le système

Exemple 1: *s'il reste du café et que l'utilisateur a mis le montant adéquat, la machine lui sert un café,*

Exemple 2: *le message envoyé d'un côté sera finalement délivré de l'autre côté d'un réseau*

Propriétés à vérifier?

Sûreté: le mauvais n'arrivera jamais.

- **Exemple 1:** *si le montant adéquat n'a pas été mis, la machine ne servira jamais de café,*
 - **Exemple 2:** *un état de violation de l'exclusion mutuelle ne doit pas se présenter,*
 - **Exemple 3:** la barrière doit obligatoirement se fermer un certain temps avant le passage du train.
- **Équité, absence de blocage, ...**

Comment Vérifier: techniques?

- Preuve de programme : preuve de théorème, axiomes, ...
- **Model-checking** sur un modèle de système;
- Preuve d'équivalence : Bisimulation entre modèles (exemple: modèle de spécification vs modèle de l'implémentation)

Limites :

Preuve = très couteuse, non automatisable

Model-checking = limités dans ses applications