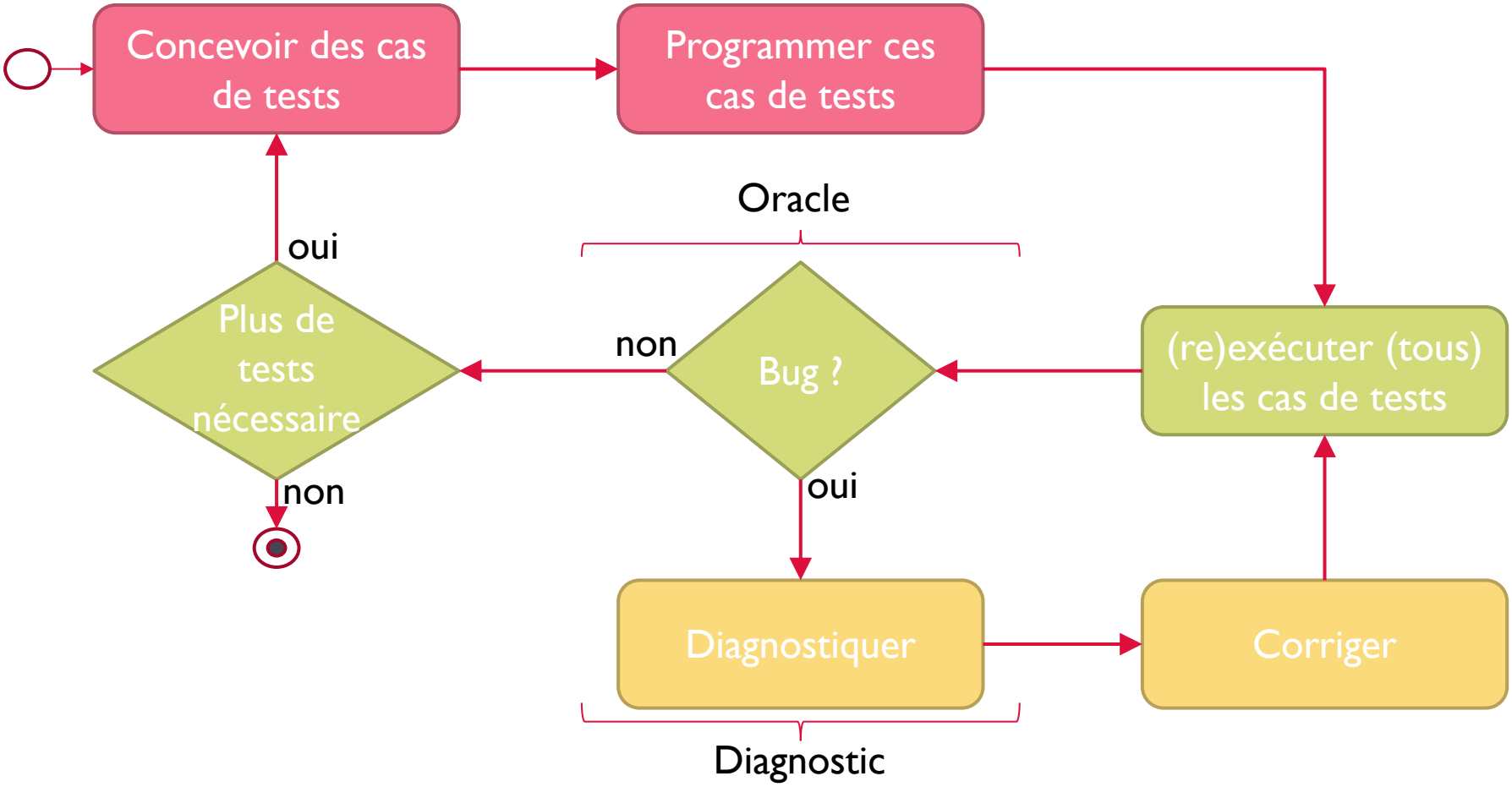
IUT Nantes
Pôle Sciences et technologie

Nantes Université

R2.03 - Qualité de développement 1
Automatisation des tests

CM10 - Oracle

Le cycle de test dynamique



Oracle du test

- L'oracle est supposé détecter un mauvais comportement
- Comportement
  - Fonctionnel
    - Mauvaise sortie
      - résultat renvoyé par la méthode testée
      - nouvel état d'un objet
    - Mauvais comportement
      - Évènements
      - IHM, etc.
  - Extra-Fonctionnel
    - Temps de réponse
    - Quantité de mémoire utilisée, etc.

Oracle du test

- Définition
  - L'oracle vérifie que l'exécution d'un test
    respecte la spécification
    en analysant le comportement du test.

- L'oracle produit le verdict du test : passe ou échoue
  - Peut donner davantage d'indications, cf partie du CM suivante

- L'oracle n'est pas systématiquement la donnée de sortie attendue

Difficulté de l'oracle

- L'oracle doit analyser des données parfois complexes pour produire le verdict
  - Type simple dans le cas de méthode renvoyant un résultat
  - Type complexe
    - Objets (avec toutes leurs propriétés)
    - Structures complexes
      - Base de données
      - Programme compilé
      - IHM, etc.
    - Propriétés extra-fonctionnelles
      - Temps, environnement (consommation mémoire, processeur), etc.

Difficulté de l'oracle

- L'oracle vérifie le respect de spécifications variées

- Les spécifications sont exprimées
  - Formellement (idéal)
  - Semi-formellement
  - Textuellement

- Les spécifications peuvent être
  - Difficilement interprétables
  - Impossible à traiter automatiquement (majorité des cas)

Pas d'oracle = pas de vérification

- Simple exécution de données de test à partir d'un état donné du logiciel sous test
  - Ne vérifie pas le comportement du logiciel
  - Une seule utilité : le test de robustesse, l'absence de crash du système

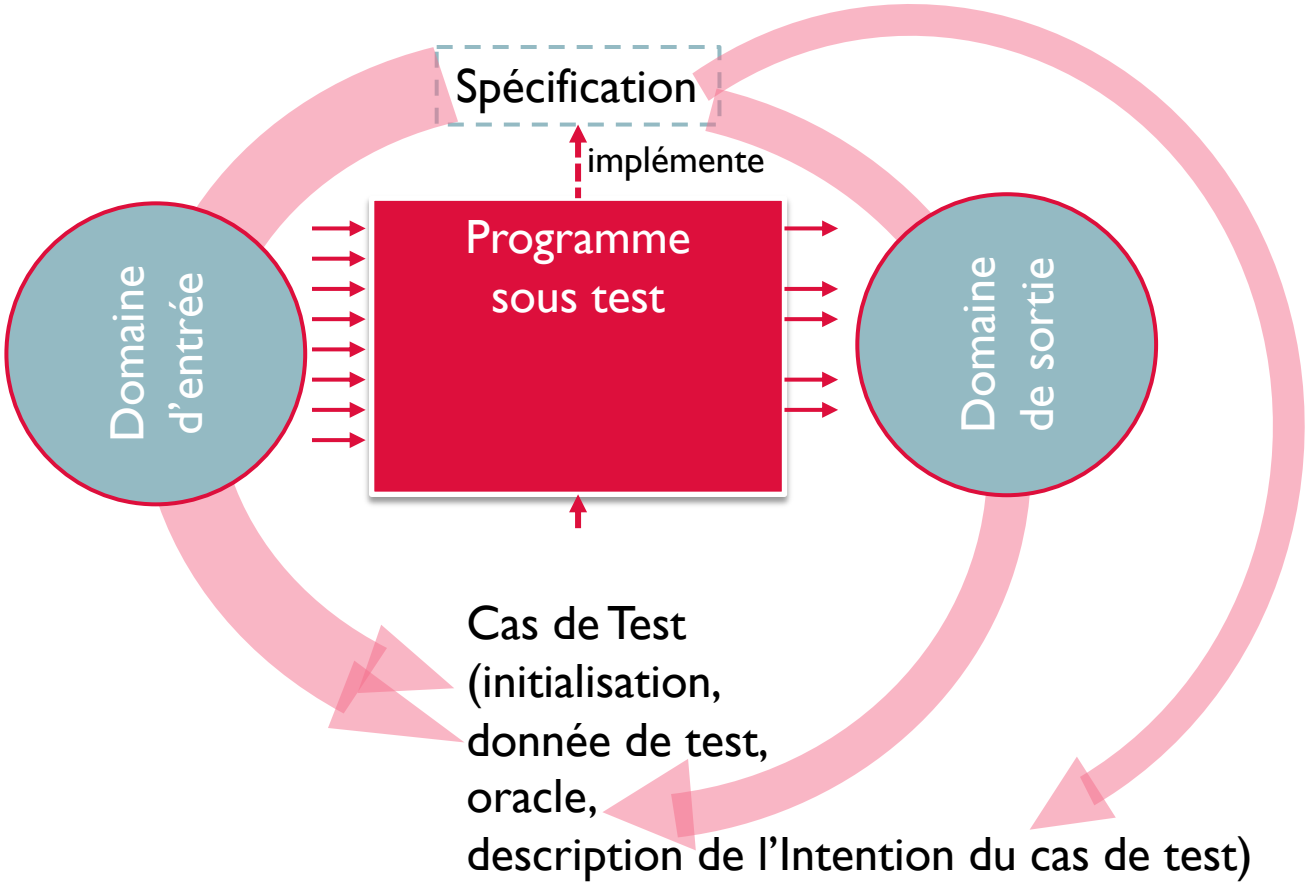
- Il ne suffit pas de générer et d'exécuter des données de test pour faire des suppositions sur l'exactitude d'un logiciel

Verdict des oracles

- Verdict complet ou partiel
  - L'exécution d'un cas de test peut entraîner de nombreuses modifications
  - Un verdict partiel assure que ce test a vérifié une partie de la spécification (partie de la spécification, dont le domaine de sortie)
  - Un verdict complet assure que ce cas de test respecte la globalité de la spécification
    - Attention les tests ne sont pas complets pour autant

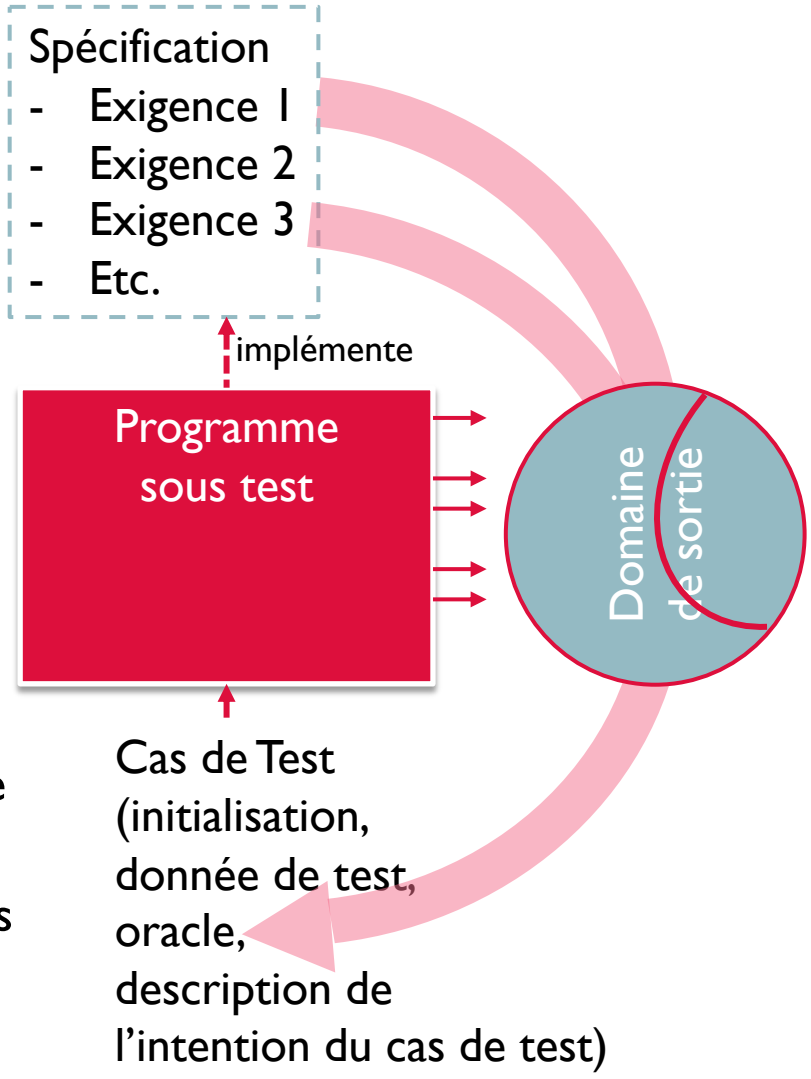
Verdict des oracles

- Un cas de test et son oracle considèrent :



Verdict des oracles

- Un cas de test et son oracle considèrent :
  - une partie (multiple) de la spécification
    - les exigences qui sont concernées par le cas de test
      - Un verdict partiel en considère certaines
      - Un verdict complet les considère toutes
  - l'effet du programme sur une partie du domaine de sortie
    - des propriétés sont concernées
      - Un verdict partiel dira si certaines d'entre elles ont été bien modifiées
      - Un verdict complet les considérera toutes
        - Voire même les effets de bords potentiels



Mise en œuvre : Oracle discret

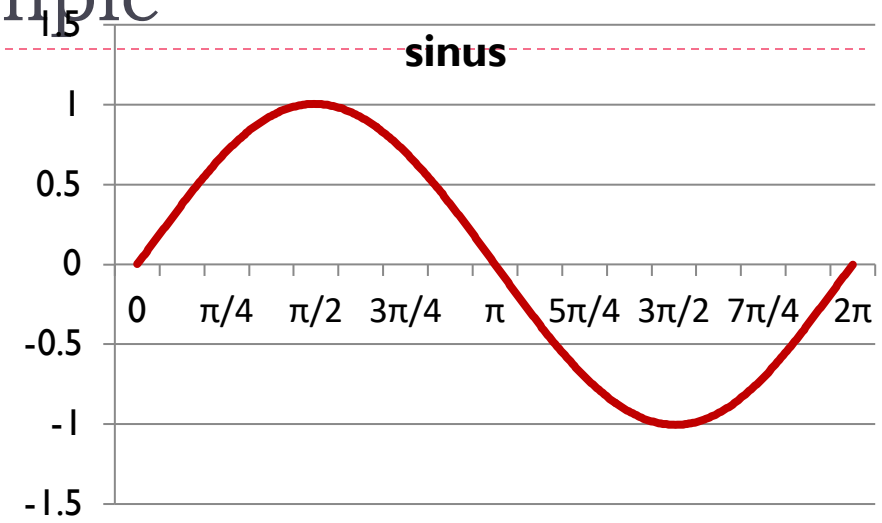
- Directement relié à la détermination d'un ensemble fini de données de test

- Un oracle par résultat de sortie qui vérifie
  - Les données renvoyées en sortie
  - L'état du système le cas échéant

- Solution la plus importante
  - Parce que l'écriture (majoritairement) manuelle des oracles nécessite une limitation de la quantité de test à un nombre fini
  - Néanmoins les tests discrets peuvent laisser passer des erreurs

Oracle discret
Exemple – Contre-exemple

- Exemple du test de la fonction sinus : y = sin(x)
- Entre 0 et 2π
  - Données de test identifiables fonctionnellement
    - Les valeurs provoquant un changement de sens : π/2, 3π/2
    - Les valeurs dont le sinus est nul : 0, π, 2π
    - Des valeurs dans les intervalles : π/4, 3π/4, 5π/4, 7π/4

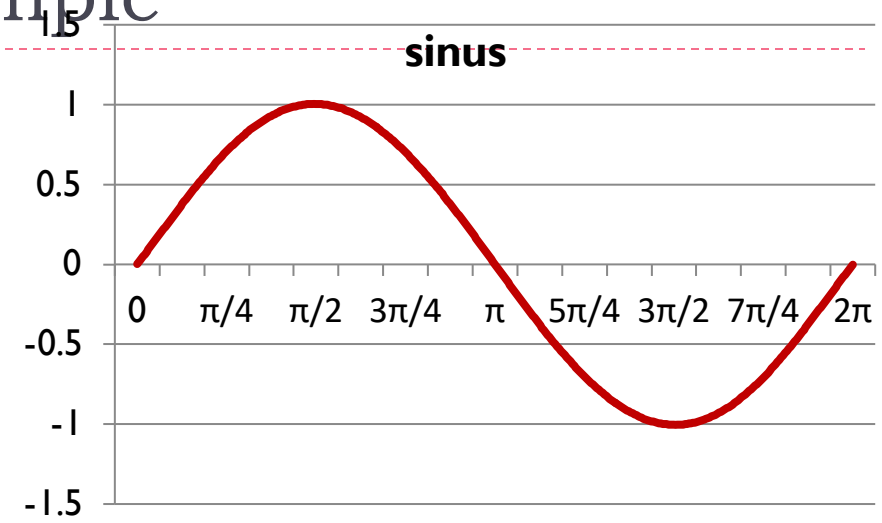


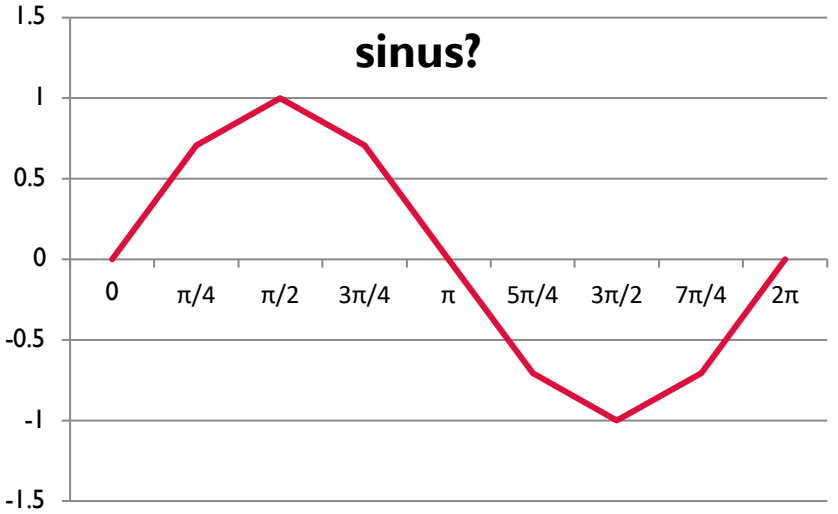
- CT:

DT	Oracle
0	0
π/4	√2/2
π/2	1
3π/4	√2/2
π	0
5π/4	-√2/2
3π/2	-1
7π/4	-√2/2
2π	0

Oracle discret
Exemple – Contre-exemple

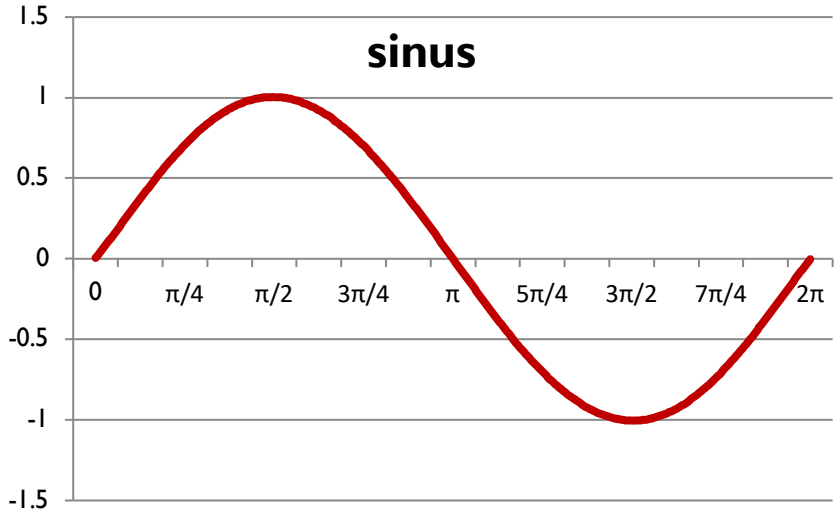
- Exemple du test de la fonction sinus : y = sin(x)
- Entre 0 et 2π
  - Données de test identifiables fonctionnellement
    - Les valeurs provoquant un changement de sens : π/2, 3π/2
    - Les valeurs dont le sinus est nul : 0, π, 2π
    - Des valeurs dans les intervalles : π/4, 3π/4, 5π/4, 7π/4





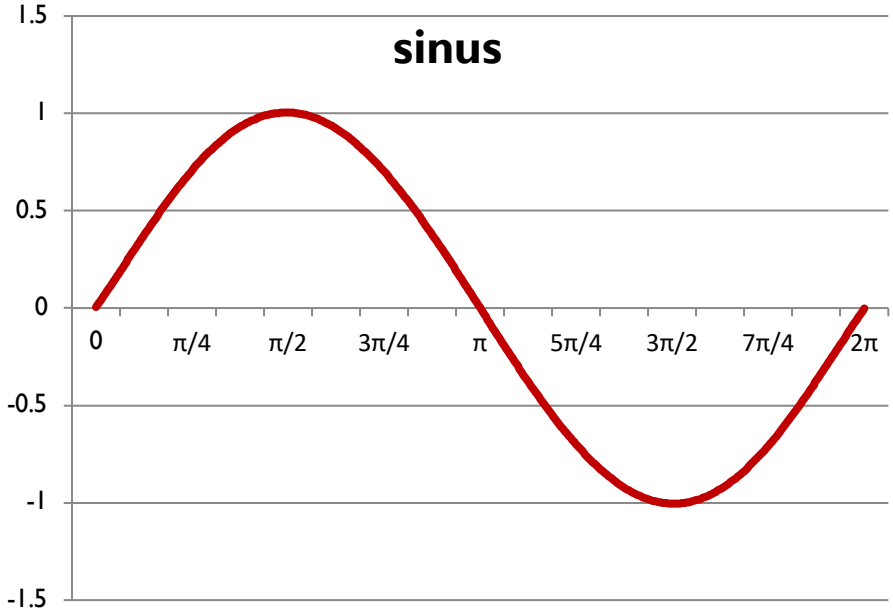
Mise en œuvre : Oracle heuristique

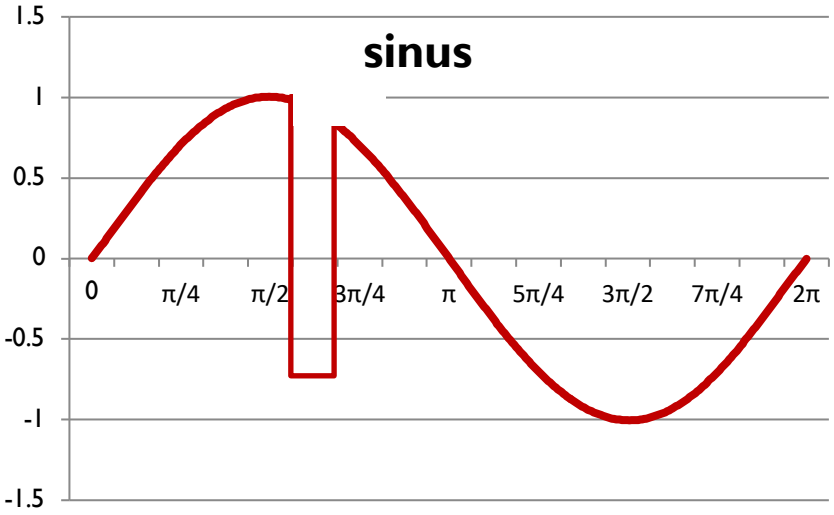
- Un oracle heuristique permet de vérifier certaines propriétés d'un groupe de résultats
- Se base particulièrement sur des liens entre données d'entrée et données de sortie
- Exploite le partitionnement
- cf. : Property Based Testing



Exemple d'heuristiques

- y = sin(x) :
  - -1=< y <= 1 quel que soit x
  - 0 > y si x mod 2π < π
  - 0 < y si x mod 2π > π



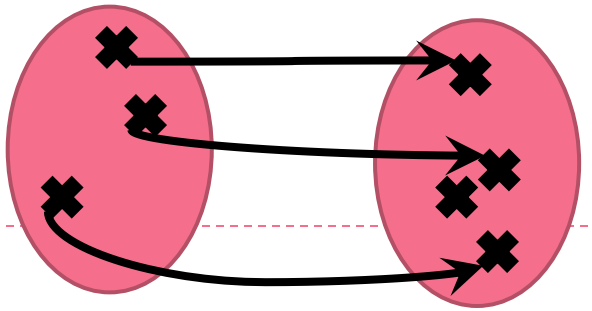


Oracle vrai (true oracle)

- Un oracle vrai vérifie n'importe quelle exécution d'une donnée de test en renvoyant un verdict complet
  - Très souple car il peut être utilisé dans n'importe quel test
  - Très difficile à obtenir
    - Aussi complexe que le logiciel qu'on vérifie
      - Le découpage en tests perd son intérêt
      - Risque d'erreur
- Logiciel de référence comme oracle
  - Permet les vérifications fonctionnelles
    - Généralement pas adapté aux vérifications extra-fonctionnelles
  - Valable si un logiciel de référence est disponible
    - Cas des changements de plate-forme
    - Cas des améliorations extra-fonctionnelles
  - On n'écrit pas un logiciel de référence spécifiquement pour en faire un oracle
    - Il faudrait aussi le tester

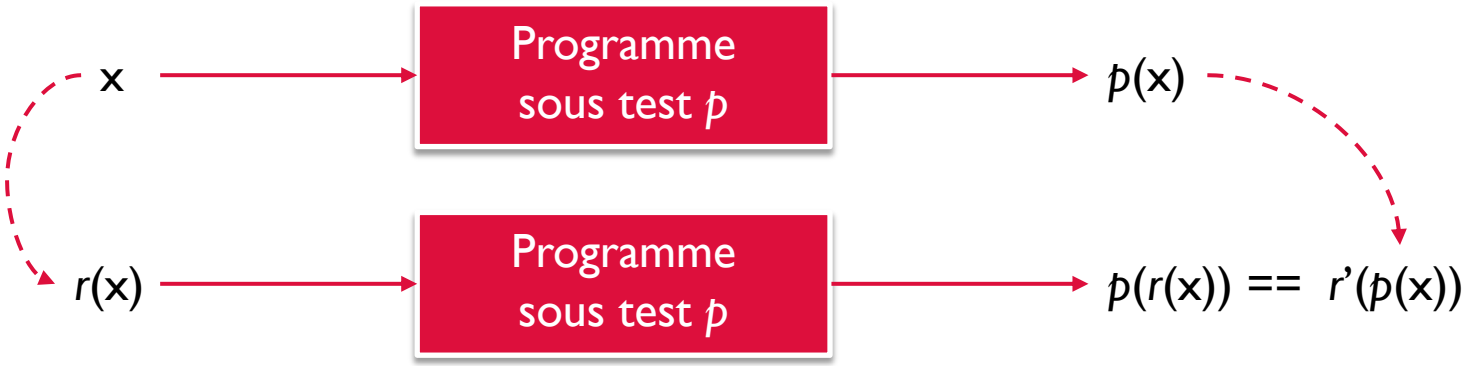
Logiciel inverse comme oracle

- Utiliser un logiciel réalisant l'inverse du logiciel testé
  - Permet les vérifications fonctionnelles
  - Ne permet pas les vérifications extra-fonctionnelles
- On n'écrit pas un logiciel inverse spécifiquement pour en faire un oracle
  - Il faudrait aussi le tester
- Valable si un logiciel inverse est disponible
  - Utopique à l'échelle d'un logiciel
  - Possible à l'échelle d'une fonction
    - En particulier les fonctions mathématiques sous la condition qu'elles soient des applications injectives



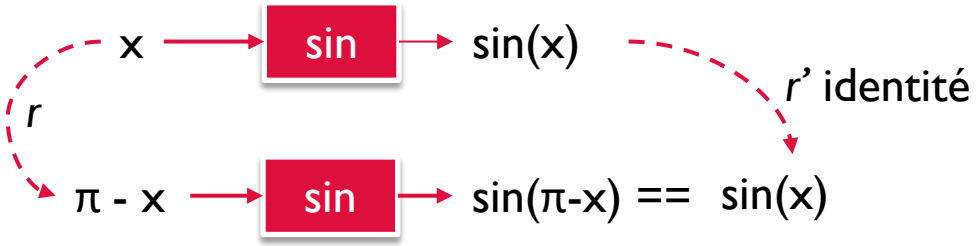
Metamorphic testing

- Utiliser des propriétés liant entrées et sorties
  - Si 2 entrées ont une relation,
    alors leurs sorties peuvent avoir une relation connue



  - Ainsi si l'égalité n'est pas respectée, le test échoue
  - Pour appliquer cette méthode, il faut identifier les relation r et r'

  - Par exemple avec *p*
    la fonction sinus,
    alors pour *r*(x) = π-x
    on a *r'*(y)=y : l'identité



Synthèse Oracle

- Différentes mises en œuvre
- Quel que soit l'oracle on ne sera pas exhaustif
  - Même difficulté que pour les données de test

- Compromis
  - Utilisation d'oracle discret et heuristique