« Une introduction informelle à l'interprétation abstraite et à l'analyse statique »

Patrick Cousot

École normale supérieure
45 rue d'Ulm
75230 Paris cedex 05, France

Patrick.Cousot@ens.fr
www.di.ens.fr/~cousot

Analyse statique par interprétation abstraite

Vérification : définir ou prouver (automatiquement) une propriété des comportements possibles d'un système informatique complexe (exemple : sémantique d'un programme)

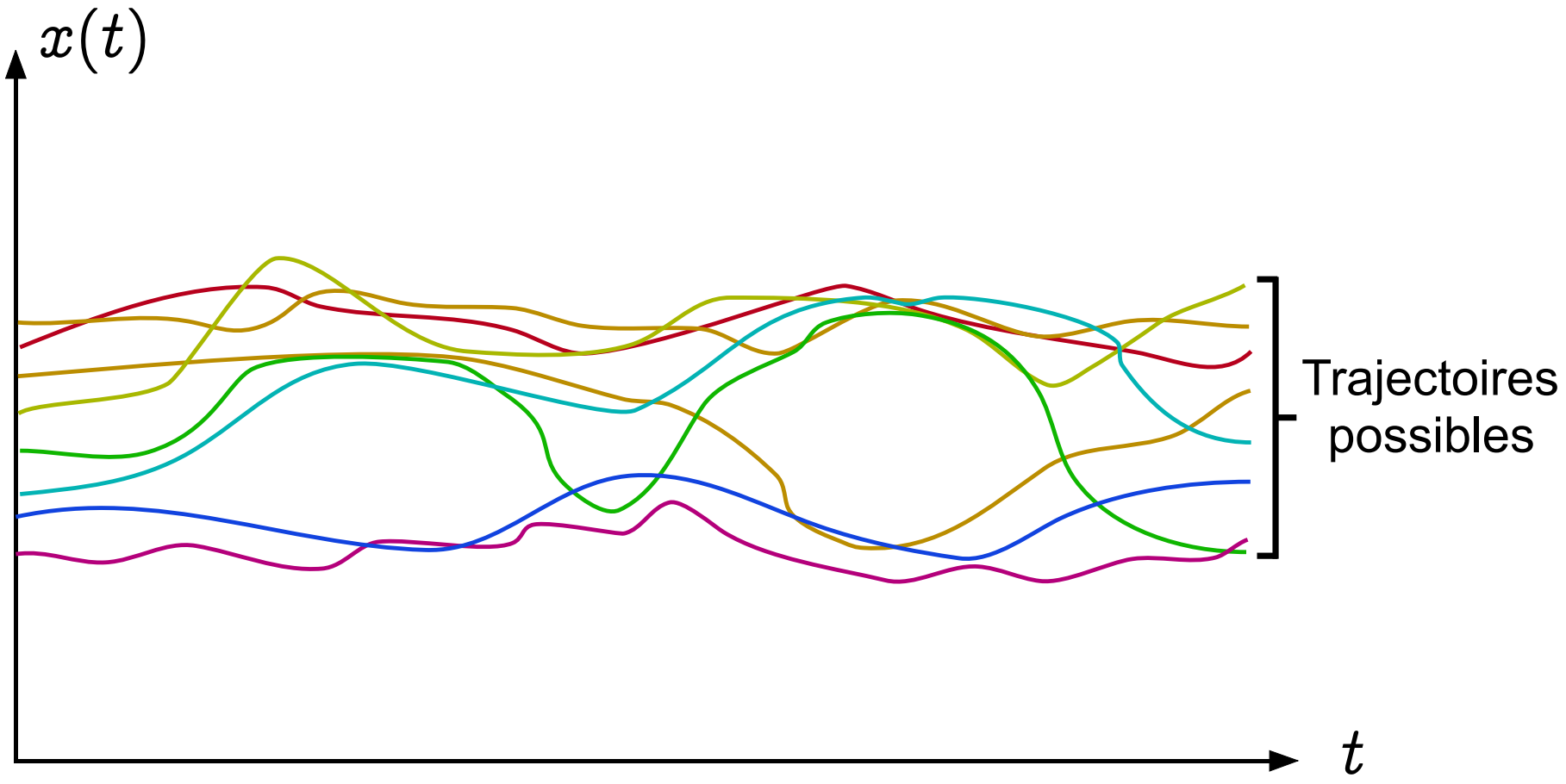
Abstraction : le raisonnement/calcul peut se faire sur une abstraction de ces comportements ne retenant que les éléments relatifs à la propriété considérée

Théorie : l'interprétation abstraite

Sémantique

La *sémantique concrète* d'un programme formalise l'ensemble de toutes ses exécutions possibles dans tous les environnements d'exécution possibles.

Exemple intuitif : Comportements possibles



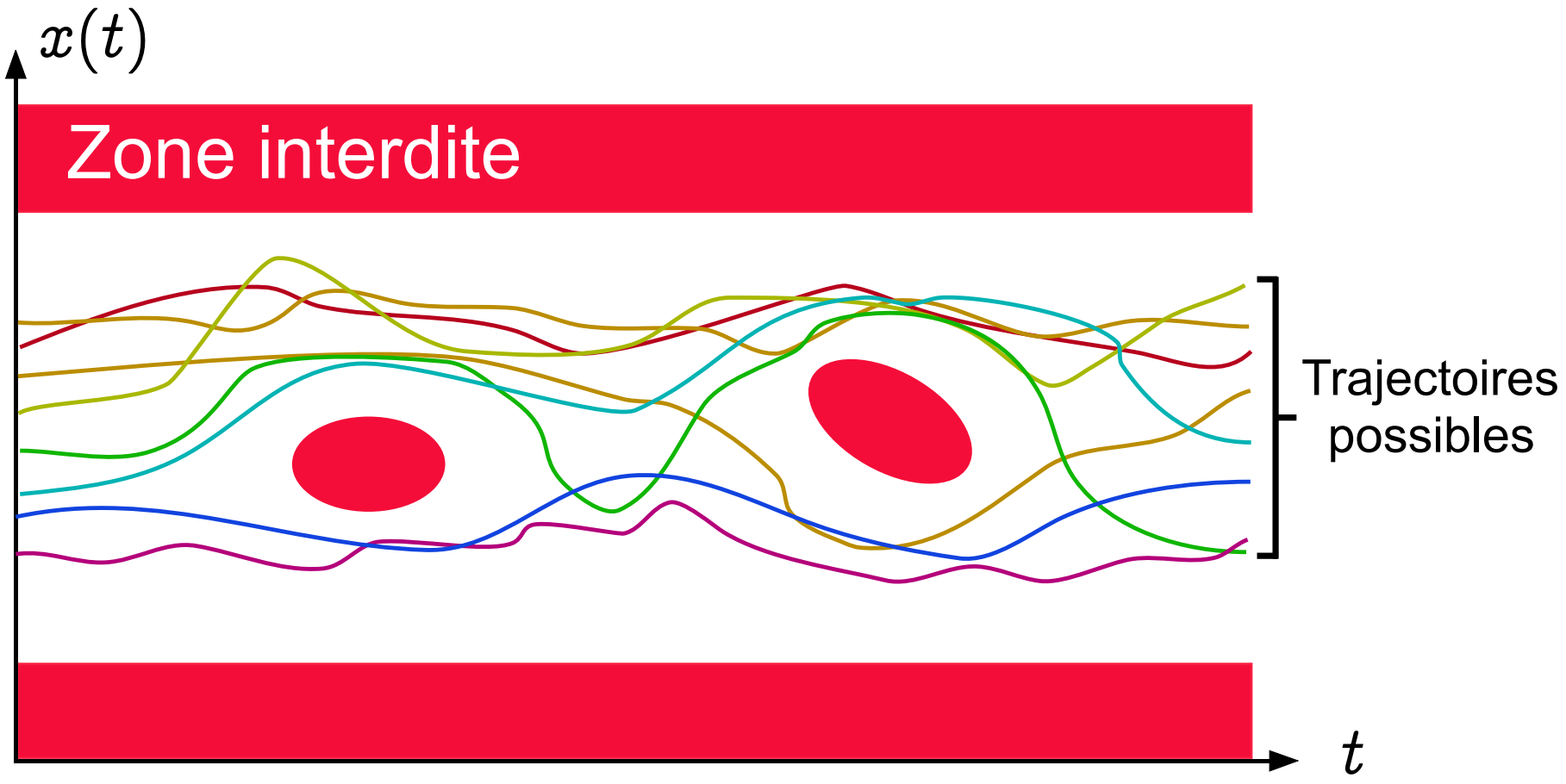
Indécidabilité

– La sémantique concrète d'un programme est un objet mathématique « infini », *non calculable* ;
– Toutes les questions non triviales sur la sémantique concrète d'un programme sont *indécidables*.

spécification d'une propriété de sûreté

Les *propriétés de sûreté* d'un programme expriment qu'aucune exécution possible dans tous les environnements d'exécution possibles ne peut atteindre un état erroné.

Exemple intuitif : Propriété de sûreté



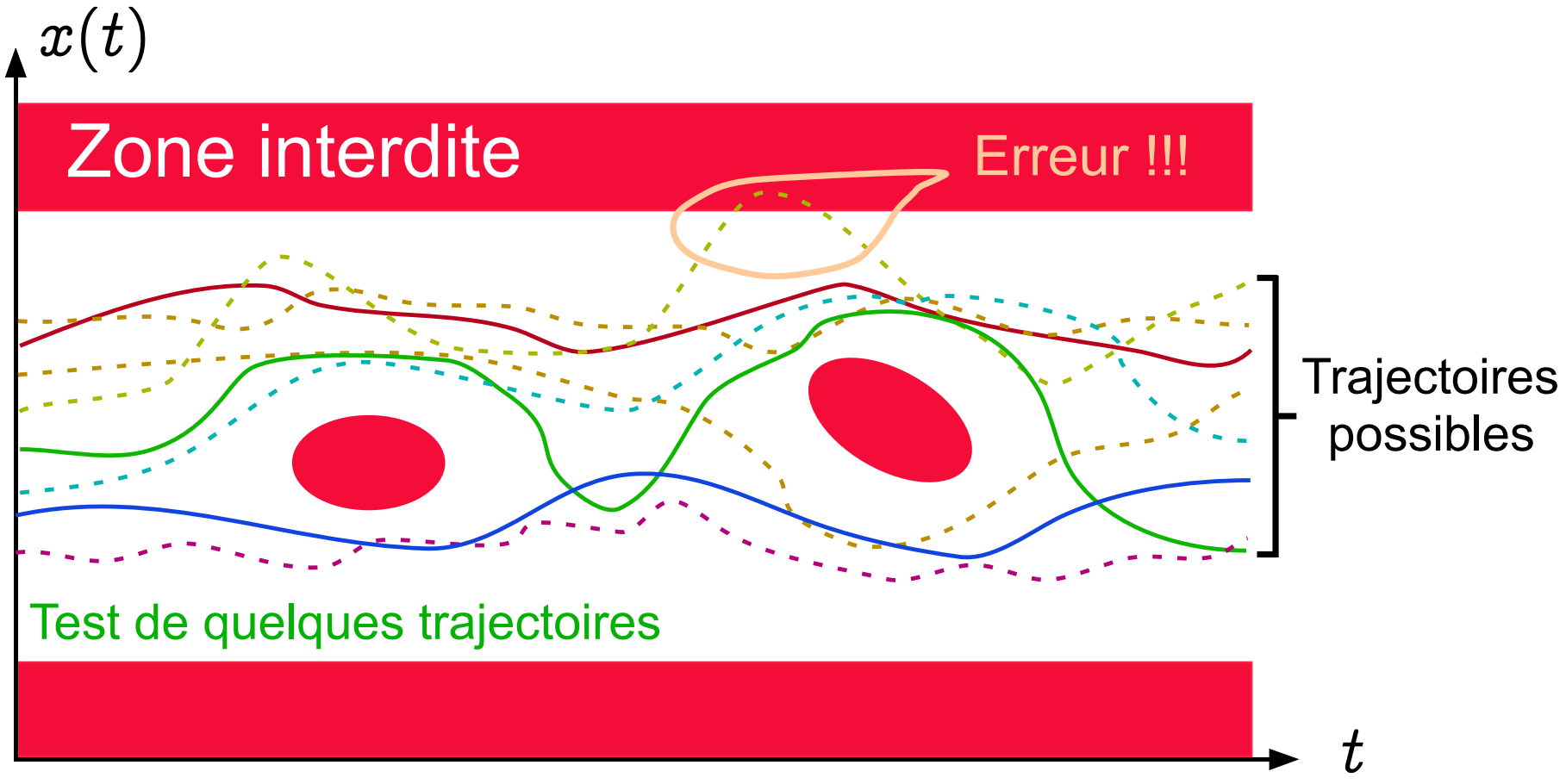
Preuve de propriétés de sûreté

- démontrer que l'intersection de la sémantique concrète du programme et de la zone interdite est vide ;
- problème indécidable (la sémantique concrète est non-calculable) ;
- impossible de répondre de manière complètement automatique avec des ressources informatiques finies et sans aucune incertitude à cette question, sans intervention humaine.

Test

- consiste à considérer un sous-ensemble des exécutions possibles ;
- pas une preuve ;
- problème de l'absence de couverture.

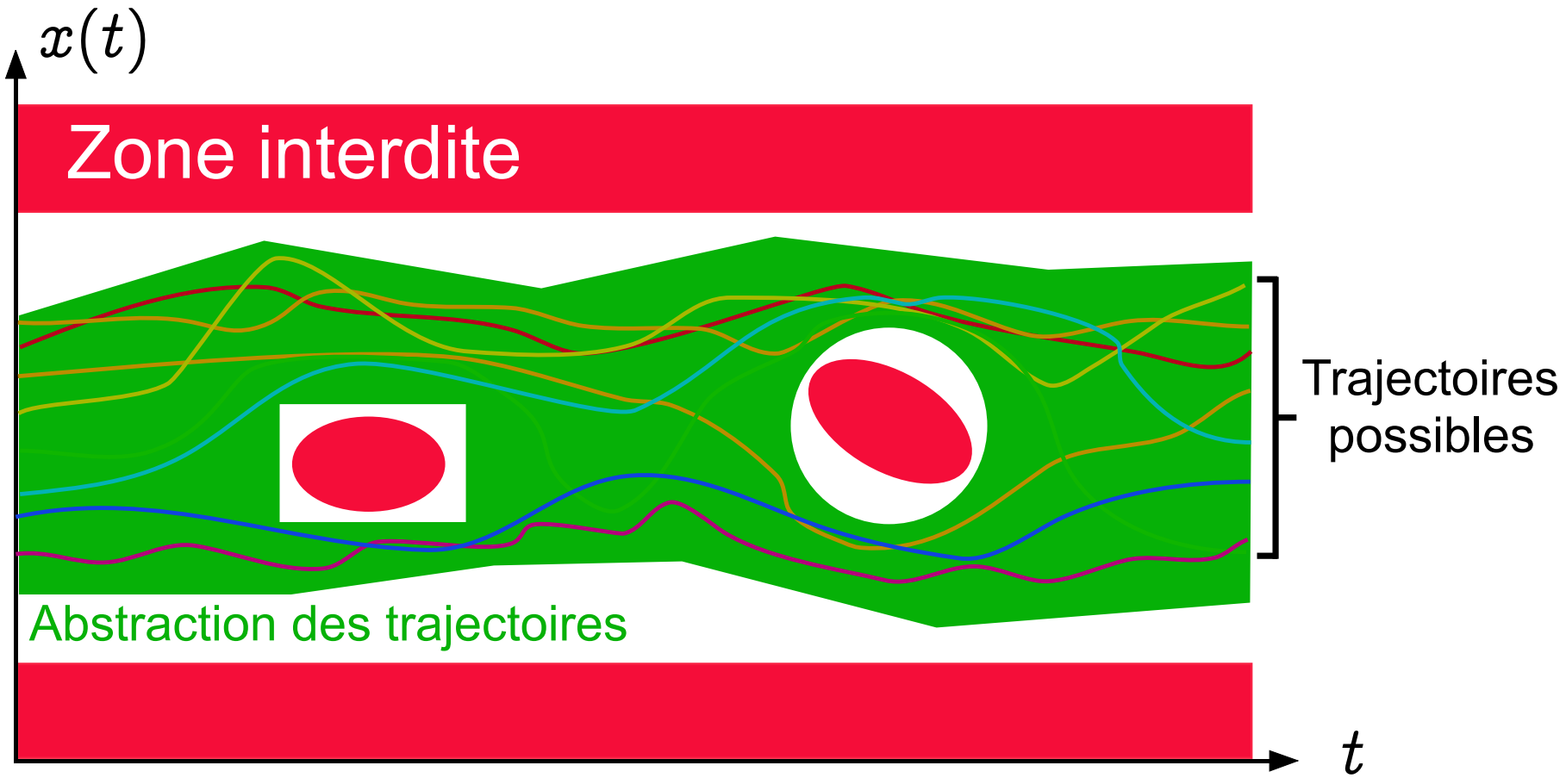
Exemple intuitif : Test/simulation de la propriété



Interprétation abstraite

- consiste à considérer une *sémantique abstraite*, c'est-à-dire sur-ensemble de la sémantique concrète du programme ;
- la sémantique abstraite couvre tous les cas possibles ;
- correct : si la sémantique abstraite est sûre (n'intersecte pas la zone interdite), la sémantique concrète l'est également.

Exemple intuitif : Interprétation Abstraite



Méthodes formelles

Interprétations abstraites qui différent dans la façon d'obtenir la sémantique abstraite :

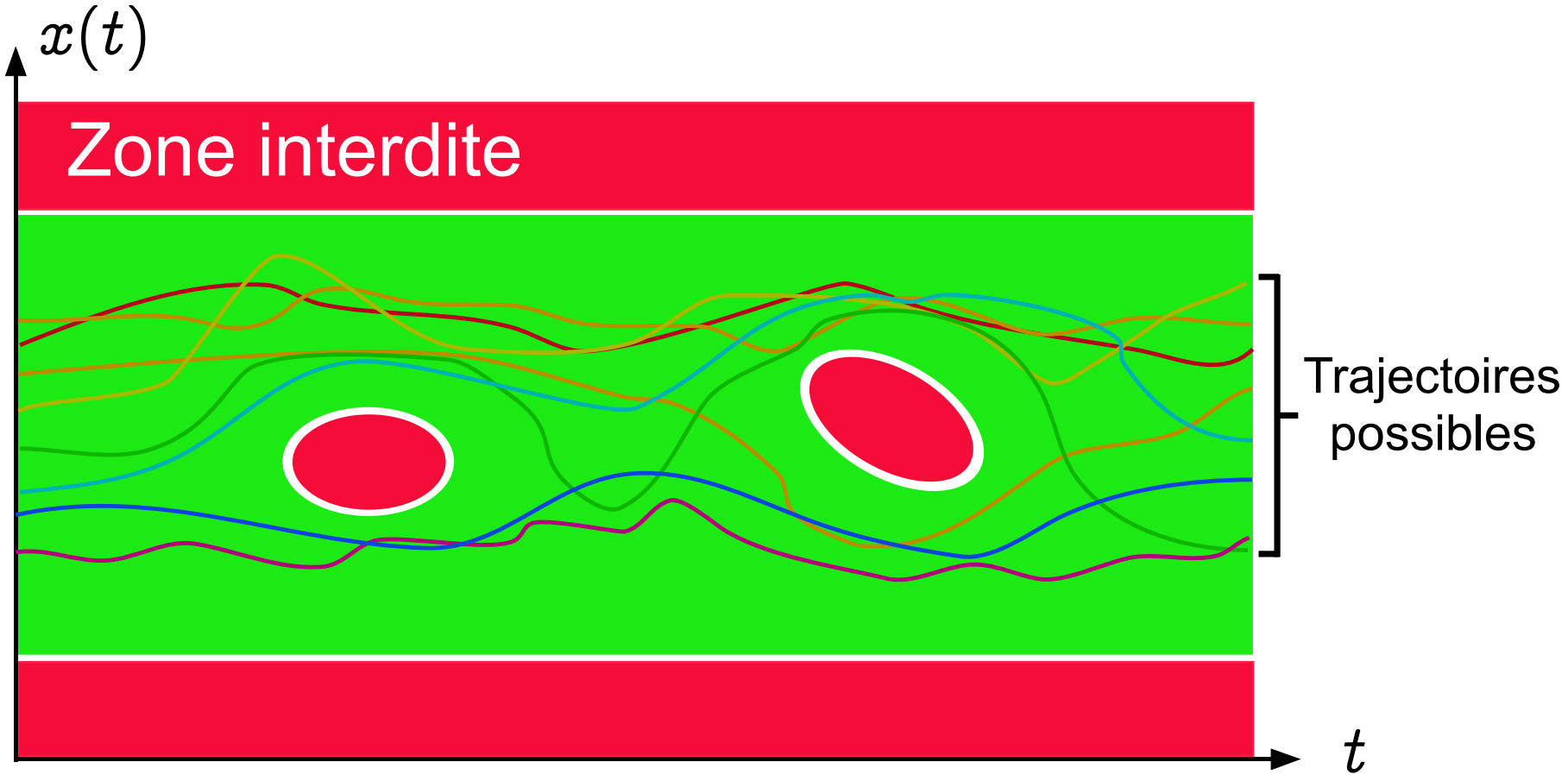
– « *vérification de modèle* » (*model checking*) :
  - la sémantique abstraite est donnée manuellement par l'utilisateur ;
  - sous forme d'un modèle finitaire de l'exécution du programme ;
  - peut être calculé automatiquement, par des techniques relevant de l'analyse statique.

– *« méthodes déductives »* :
  - la sémantique abstraite est spécifiée par des conditions de vérification ;
  - l'utilisateur doit la fournir sous forme de propriétés inductives ;
  - peuvent être calculées par des techniques relevant de l'analyse statique).
– *« analyse statique »* : la sémantique abstraite est calculée automatiquement selon des approximations prédéfinies (éventuellement paramétrables manuellement par l'utilisateur).

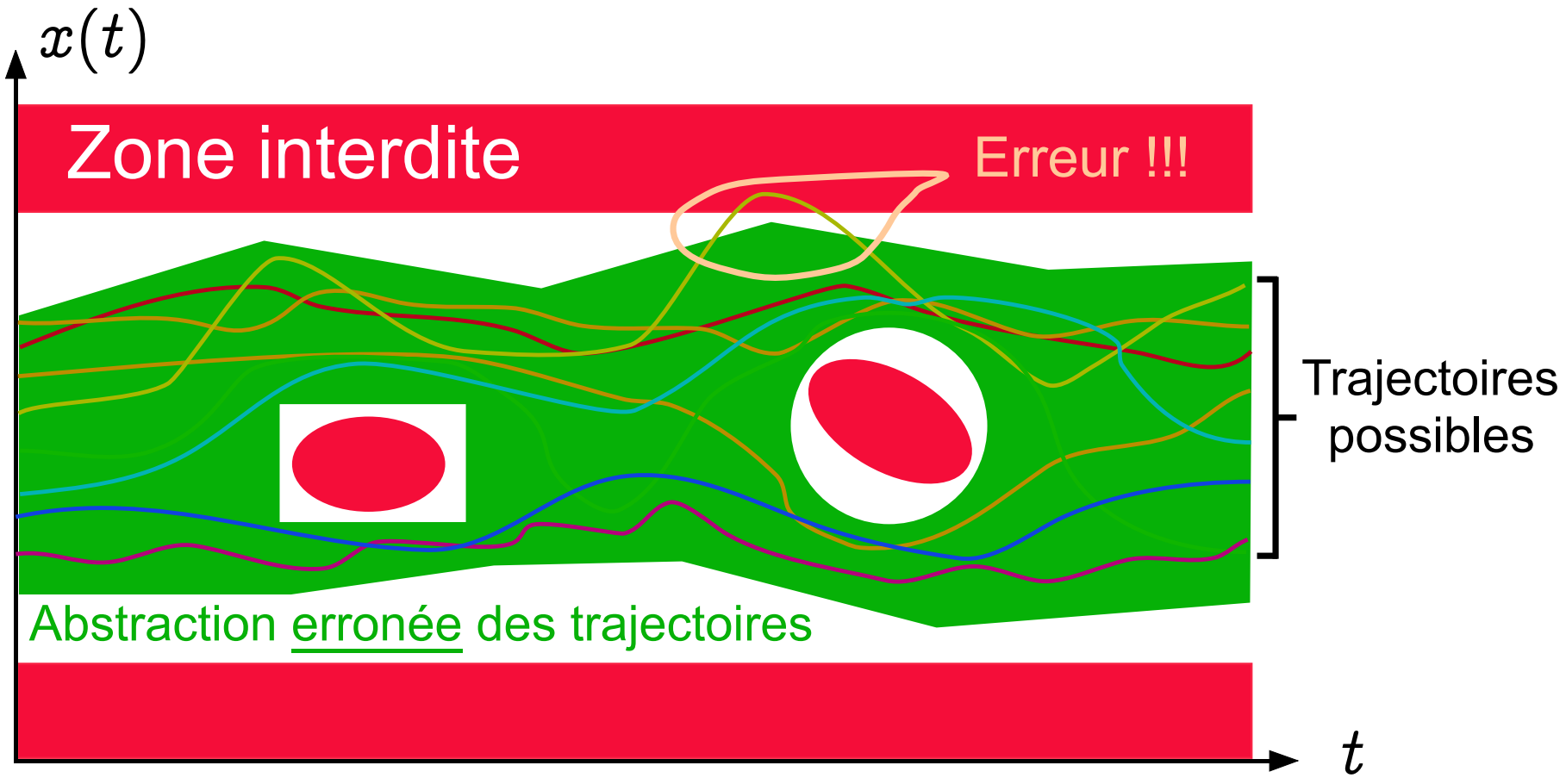
Propriétés requises de la sémantique abstraite

- correcte (*sound*) pour ne pas oublier de cas d'erreur ;
- suffisamment précise (pour éviter les fausses alarmes) ;
- aussi simple/abstraite que possible (pour éviter les phénomènes d'explosion combinatoire).

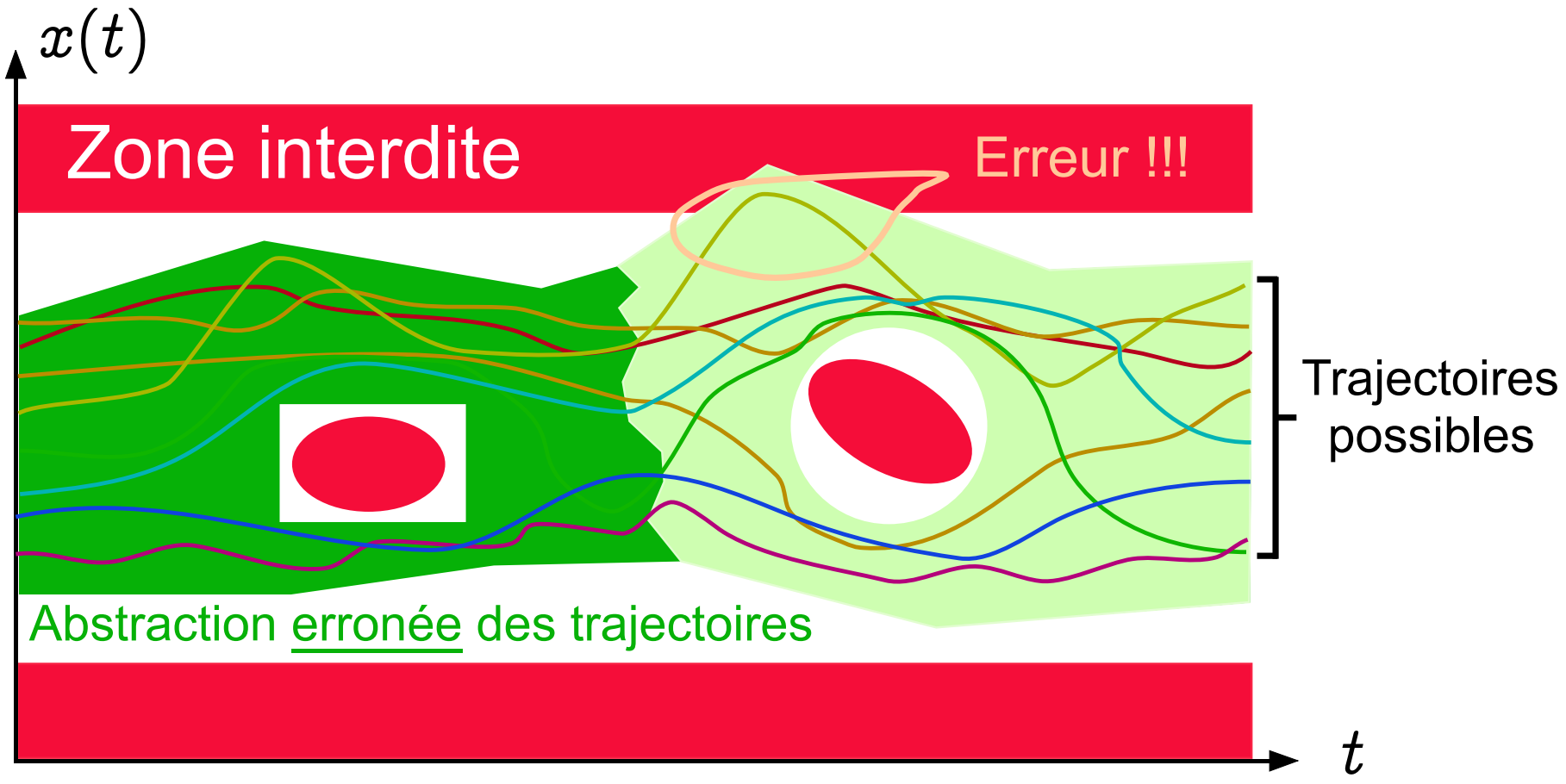
Exemple intuitif : Sémantique la plus abstraite, correcte et précise



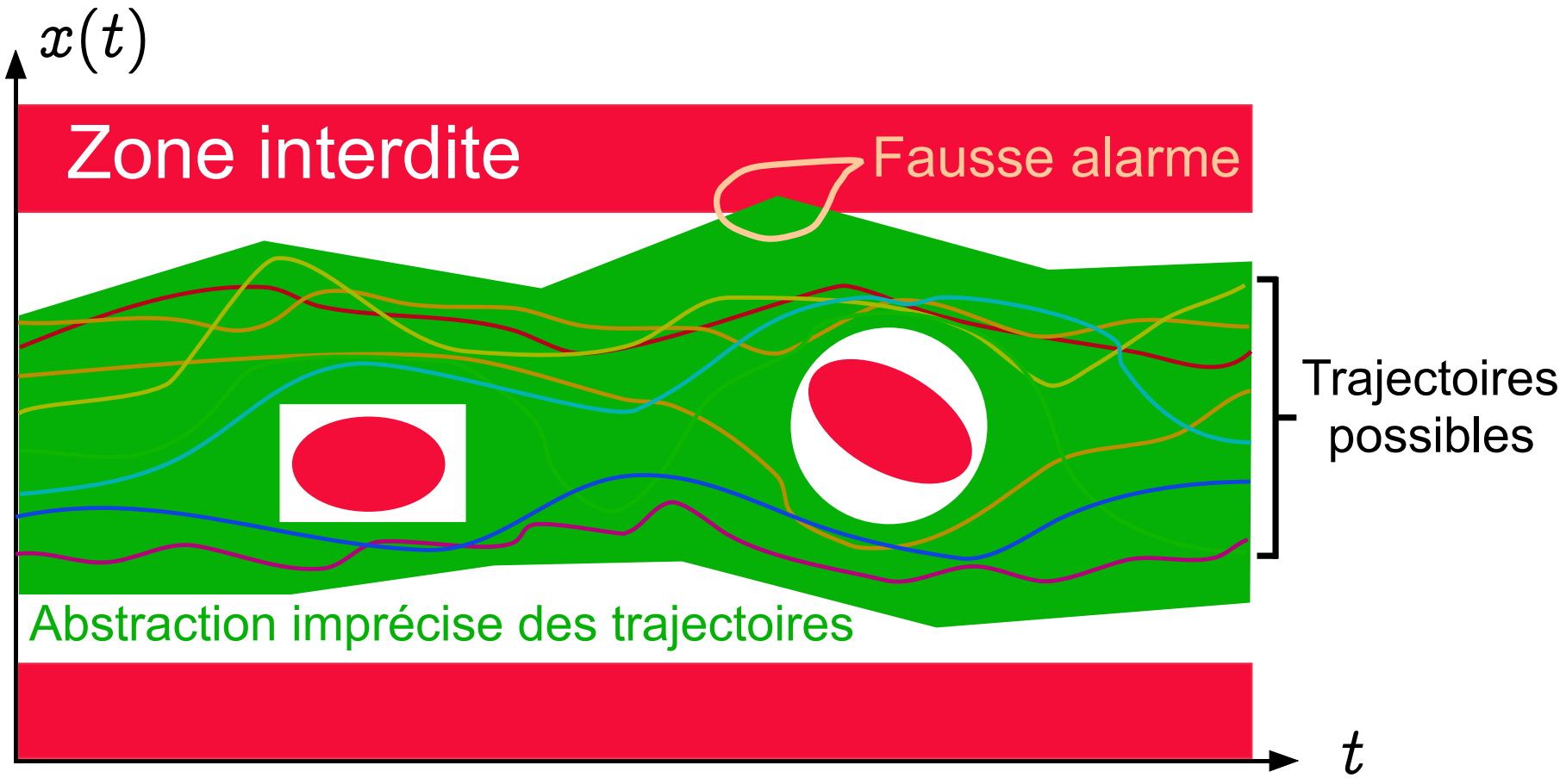
Exemple intuitif : Abstraction Erronée — I



Exemple intuitif : Abstraction Erronée — II



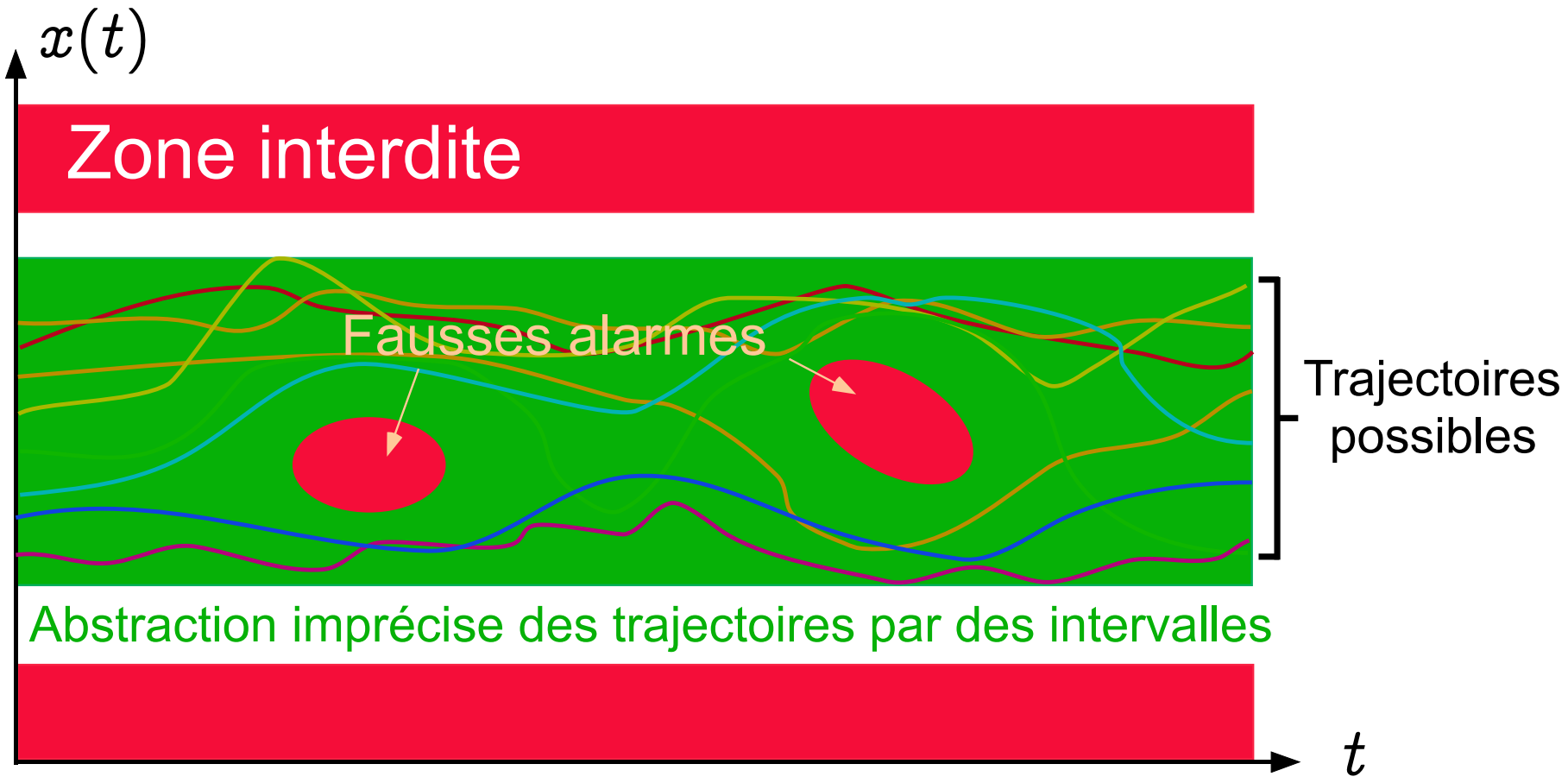
Exemple intuitif : Imprécision $\Rightarrow$ fausses alarmes



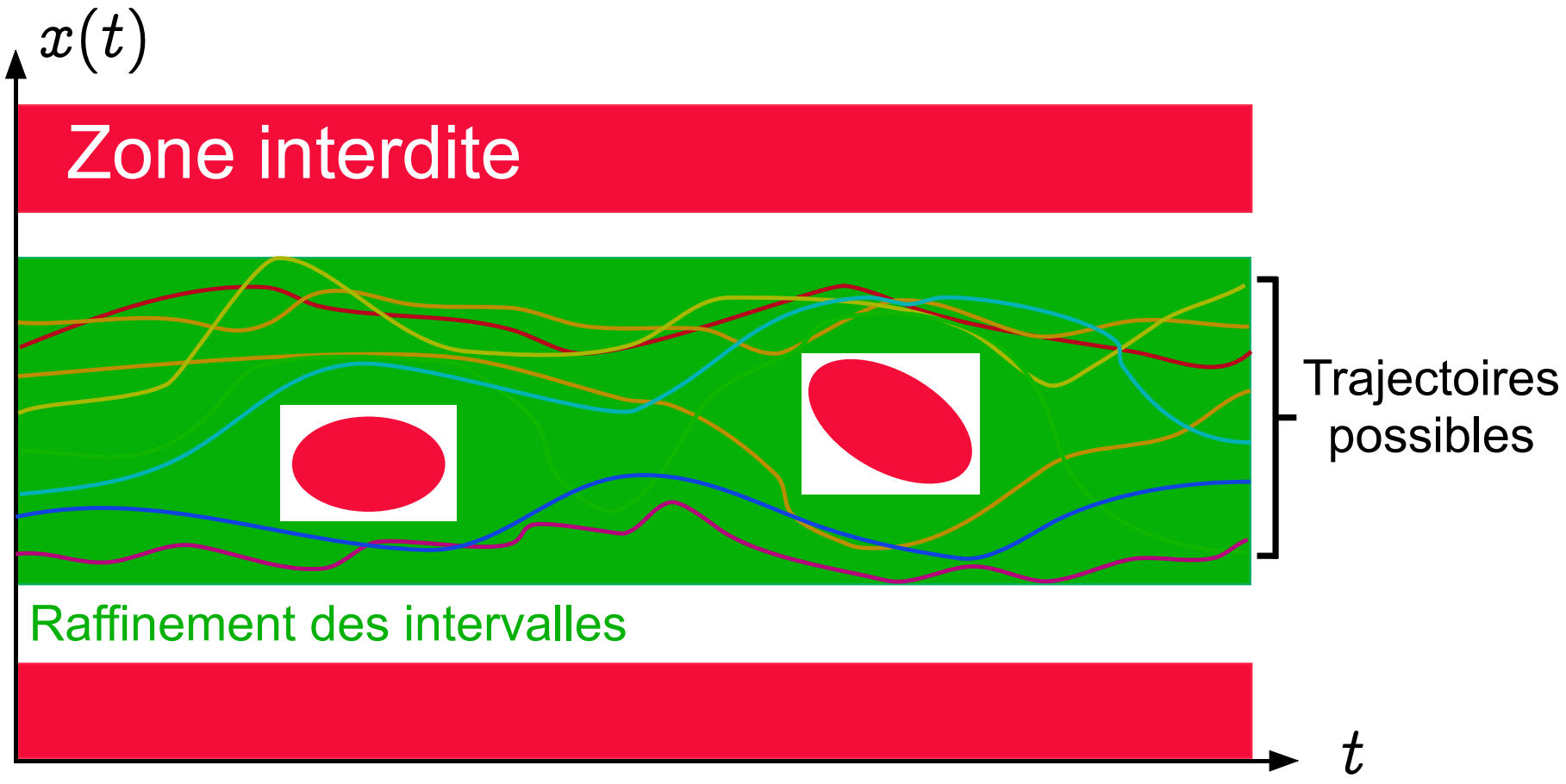
Abstractions “standards”

– servent de base à la conception des analyseurs statiques ;
– un raffinement manuel est nécessaire pour s’adapter au domaine d’application.

Exemple intuitif : Abstraction “standard” par intervalles



Exemple intuitif : Abstraction plus raffinée



Fin, merci de votre attention.