La modélisation et la conception d'algorithmes en contrôle aérien

Gilles Dowek

Que sont les méthodes formelles ?

Que nous apprennent-elles sur l'informatique ?

I. L'âge de la complexité

De 10^2 / 10^3 composants (machine à vapeur, radio...) à 10^8 (programme, ordinateur...)

Les Humains ne peuvent pas éviter de faire des erreurs

Les bugs ne sont pas toujours des catastrophes (mais : transports, santé, énergie)

Méthodes formelles : Toutes les méthodes inventées pour éviter les (réduire le nombre d') erreurs

Empiriques (test), *a priori* (vérification, preuves)

Reformulation

Introduire une forme de redondance : deux algorithmes pour résoudre le même problème, démontrer qu'il font la même chose
même idée que les codes correcteurs

Donner moins d'information : spécification
Par exemple, soustraction : $y + x_2 = x_1$

Cet exposé

Un exemple : le contrôle aérien
Plus précisément le Small aircraft transportation system (avec Muñoz et Carreño)

Le futur des transports : sans Humains
Ascenseur, métro, drone — train, avion de ligne — voiture

Plus efficace, plus sûr

Un paradoxe : plus sûr, mais perçu comme moins sûr

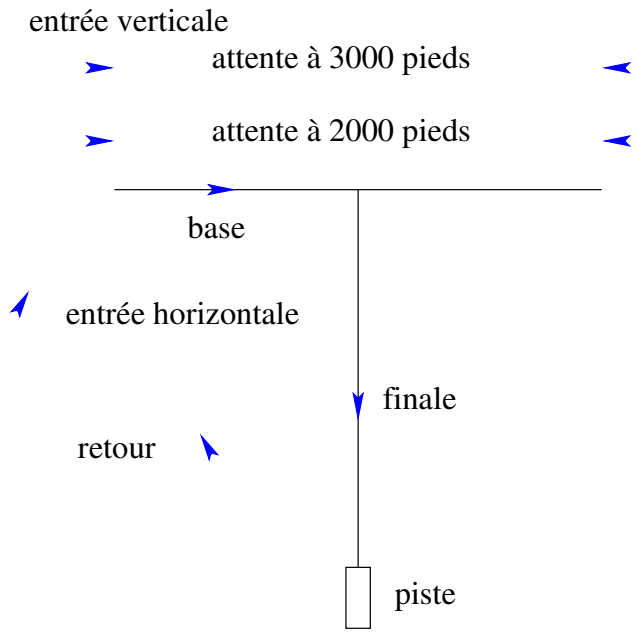
II. Small aircraft transportation system

De très nombreux petits aéroports dans le monde

Quelques avions chaque jour

Pas de personnel dans la tour de contrôle

Un protocole pour atterrir et décoller sans intervention humaine


entrée verticale
attente à 3000 pieds
attente à 2000 pieds
base
entrée horizontale
finale
retour
piste

Un exemple de règle

Un avion peut entrer dans la zone d'attente à 3000 pieds à droite si

- aucun avion dans cette zone
- aucun avion dans la zone d'approche interrompue à droite
- aucun avion dans la zone d'approche horizontale à droite
- au plus un avion dans une zone à droite ou dans avec un point de replis à droite

Questions

Peut-il y avoir un conflit (deux avions dans la même zone) ?,
Interblocage ?...

Essayer et voir (mais beaucoup d'accidents)
Modéliser et simuler (mais combien d'heures de vol ?)
Modéliser et démontrer

Expliquer les règles à un ordinateur

Un système à états et transitions

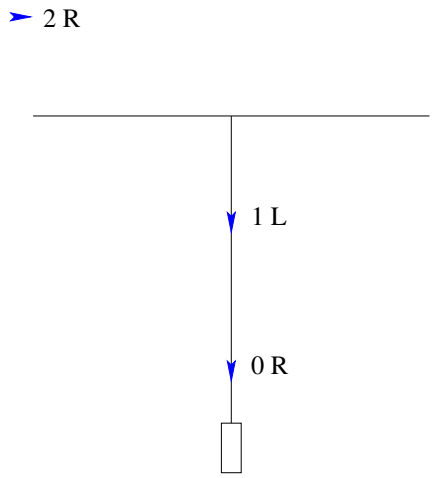
Comme le jeu d'échec

État : position de chacune des pièces sur l'échiquier (+ ε)
Transition : bouger une pièce

Expliquer ce qu'est un état à un ordinateur

Définir un type de données

Une liste d'avions (numéro d'ordre, point de replis, position)




```
[(0,R,final); (1,L,final); (2,R,holding3r)]
```

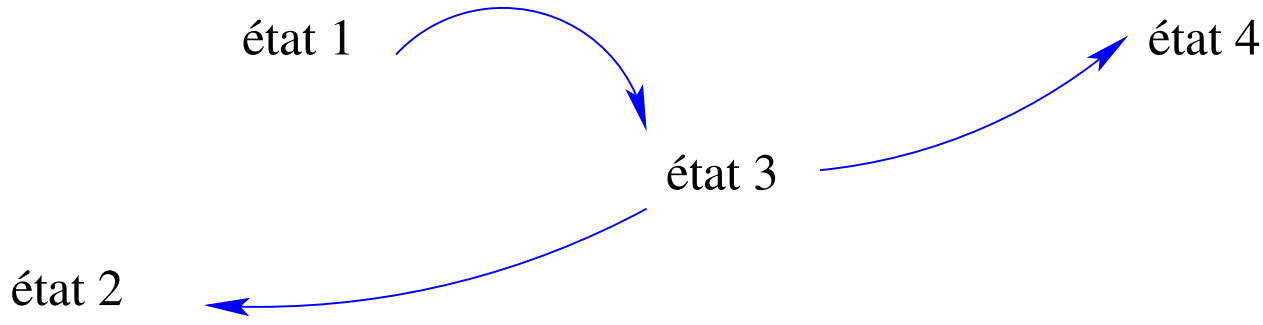
Expliquer ce qu'est une règle à un ordinateur

Définir un algorithme qui à chaque état associe une liste d'états

```
let rule6r s = match s#baser with
| [] -> []
| h::r ->
   let x = order h
   in if x = 0 || mem (x-1) s#final
      then [{<baser = r; final = final@[h]>}]
      else []
```

De la simulation à l'énumération

Simuler au hasard



Mais aussi essayer d'énumérer tous les états accessibles

A priori, une infinité d'états possibles
Mais seulement un nombre fini d'états accessible depuis l'état initial

Quelques résultats

Énumération des états (quelques milliers)
Pas de conflits, mais un interblocage

Un scenario complexe menait à

- 0 (3000 pieds)
- 1 (2000 pieds)

La modification d'une seule règle a fait disparaître l'interblocage

III. Quelques leçons

Sur les bugs

Pas facile de concevoir un protocole de contrôle aérien sans faire d'erreurs (on ne peut penser à tout)

Vrai de n'importe quel algorithme / programme / circuit...

Mais il existe des outils (conceptuels et logiciels) pour modéliser, spécifier et étudier ces objets

Faire connaître ces méthodes : un malentendu (comprenons-nous quelques chose au contrôle aérien ?)

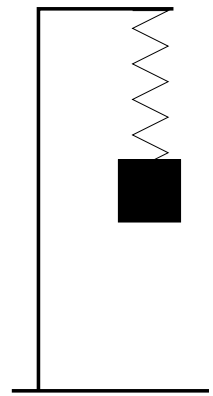
Comparer les descriptions

50 pages en anglais	2 pages de code
De nombreux exemples	Le code se suffit à lui-même
Pas exécutable	Exécutable
Impossible démontrer quoi que ce soit	Énumération exhaustive
Imprécis	Précis

Mathématiser...

... mais pas n'importe comment

Un aéroport n'est pas une masselotte au bout d'un ressort



$$m\ddot{x} = -mg - kx$$

$$-\frac{mg}{k} + a\ cos(\sqrt{\frac{k}{m}}t + \phi)$$

Comment mettre en équation

une langue ?

mais

$$S \longrightarrow GN\ GV$$

$$GN \longrightarrow NP$$

$$GV \longrightarrow VT\ GN$$

$$NP \longrightarrow \text{Alice}$$

$$NP \longrightarrow \text{Bob}$$

$$VT \longrightarrow \text{voit}$$

$S \longrightarrow$ Alice voit Bob

Comment mettre en équation

?

L'informatique

Permet de construire un monde

Mais aussi de décrire le monde (une seconde révolution galiléenne)

Science is what we understand well enough to explain to a computer. Art is everything else we do.