

# Les jeux à la rescousse de l'informatique

James Campbell Alexander Main

UMONS – Université de Mons and F.R.S.-FNRS, Belgium

The logo for the University of Mons (UMONS) features the word "UMONS" in a bold, sans-serif font. The letter "U" is grey, while "MONS" is red. A horizontal red line is positioned below the "U".

The logo for the National Research Fund (FNRS) consists of the lowercase letters "fnrs" in a bold, purple, sans-serif font. Below the letters, the tagline "LA LIBERTÉ DE CHERCHER" is written in a smaller, green, sans-serif font.

Exposé pour le mois du doctorant – 10 mars 2022

# Un tour d'horizon de cet exposé

On discute dans cet exposé des **thématiques de recherche** qui m'intéressent en tant que doctorant.

- Motivations pour l'étude des **méthodes formelles**.
- Présentation de la **synthèse réactive** : conception automatique de **systèmes réactifs** à partir d'une spécification.
- Présentation des **jeux** et de leur intérêt dans la synthèse réactive.

# Un tour d'horizon de cet exposé

On discute dans cet exposé des **thématiques de recherche** qui m'intéressent en tant que doctorant.

- Motivations pour l'étude des **méthodes formelles**.
- Présentation de la **synthèse réactive** : conception automatique de **systèmes réactifs** à partir d'une spécification.
- Présentation des **jeux** et de leur intérêt dans la synthèse réactive.

## Objet central dans mon travail

Les **stratégies** dans les jeux, leurs **représentations** en pratique.

# Plan de l'exposé

1 Motivations

2 Jeux

3 Stratégies

4 Stratégies et représentations efficaces

# Plan de l'exposé

1 Motivations

2 Jeux

3 Stratégies

4 Stratégies et représentations efficaces

# Le coût des bugs informatiques

- Les bugs informatiques peuvent avoir un **coût important**.
  - Un bug de division dans les processeurs Pentium II d'Intel a coûté 475 millions de dollars US en frais de remplacement.
  - Une erreur de software dans la machine de radiothérapie a causé au moins six incidents où des patients reçurent des doses massives de radiation.

# Le coût des bugs informatiques

- Les bugs informatiques peuvent avoir un **coût important**.
  - Un bug de division dans les processeurs Pentium II d'Intel a coûté 475 millions de dollars US en frais de remplacement.
  - Une erreur de software dans la machine de radiothérapie a causé au moins six incidents où des patients reçurent des doses massives de radiation.
- Il est impératif que les **systemes critiques** soient **corrects**.

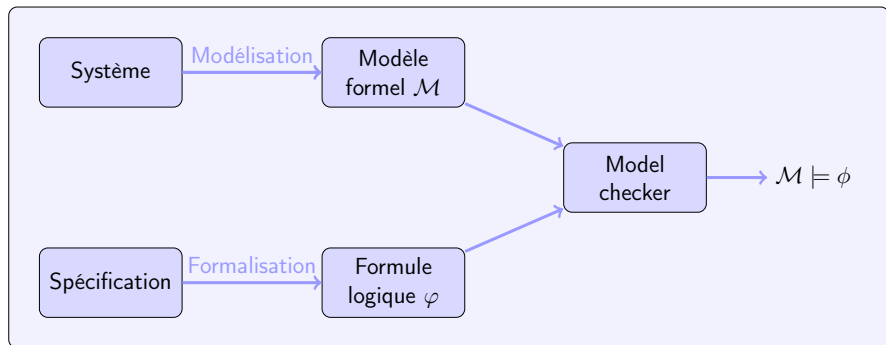
# Le coût des bugs informatiques

- Les bugs informatiques peuvent avoir un **coût important**.
  - Un bug de division dans les processeurs Pentium II d'Intel a coûté 475 millions de dollars US en frais de remplacement.
  - Une erreur de software dans la machine de radiothérapie a causé au moins six incidents où des patients reçurent des doses massives de radiation.
- Il est impératif que les **systèmes critiques** soient **corrects**.
- Les approches comme les **tests** ou **l'examen du code par des pairs** ont comme désavantage qu'ils interviennent **tard** dans l'implémentation d'un système  $\rightsquigarrow$  les correctifs sont souvent **coûteux** à implémenter.



# Model checking

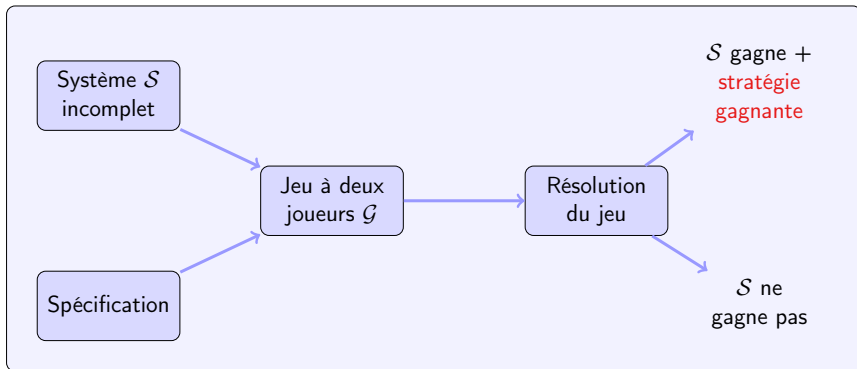
- Utilisé lors de la phase du **conception** du système.
- Un model checker fournit une **garantie formelle** sur le modèle du système.



# Synthèse réactive

Approcher le problème différemment

- La **synthèse réactive** consiste à la **génération automatique** d'un système correct à partir d'une **spécification**.
- Approche basée sur la **théorie des jeux** : on modélise l'interaction du **système** et de son **environnement** comme un jeu.



# Plan de l'exposé

1 Motivations

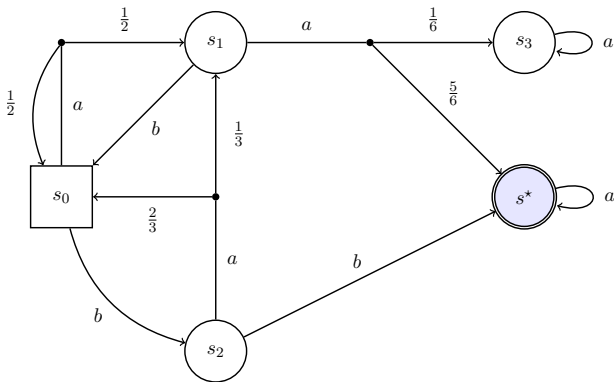
2 Jeux

3 Stratégies

4 Stratégies et représentations efficaces

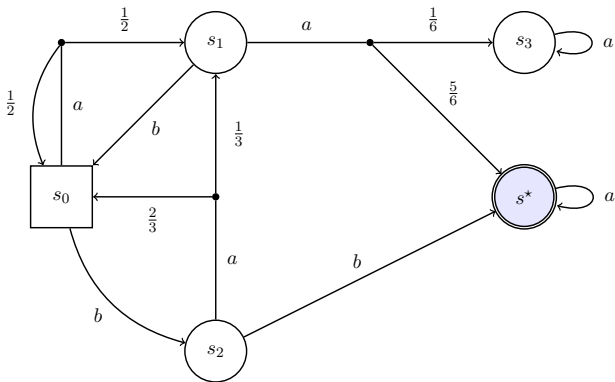
# Qu'est-ce qu'un jeu ?

- On considère des jeux **stochastiques** à deux joueurs.



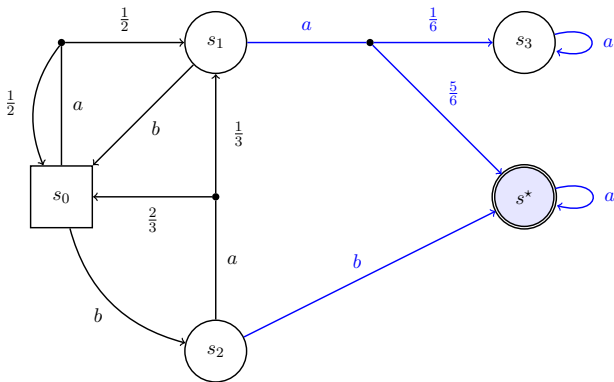
# Qu'est-ce qu'un jeu ?

- On considère des jeux **stochastiques** à deux joueurs.
- Chaque joueur a un **objectif** dont il veut assurer la satisfaction **quelle que soit la stratégie de l'adversaire** : e.g., le but du système est d'atteindre l'état  $s^*$ .



# Qu'est-ce qu'un jeu ?

- On considère des jeux **stochastiques** à deux joueurs.
- Chaque joueur a un **objectif** dont il veut assurer la satisfaction **quelle que soit la stratégie de l'adversaire** : e.g., le but du système est d'atteindre l'état  $s^*$ .



# Jeux stochastiques à deux joueurs

## Définition

### Définition

Un **jeu stochastique** à deux joueurs est un tuple  $\mathcal{G} = (S, S_1, S_2, A, \delta)$  où

- $S$  est un ensemble fini d'états ;
- $(S_1, S_2)$  est une partition de  $S$  ;
- $A$  est un ensemble fini d'actions ;
- $\delta: S \times A \rightarrow \mathcal{D}(S)$  est une fonction (partielle) de transition probabiliste.

Pour tout  $s \in S$ , on note  $A(s)$  l'ensemble des actions  $a \in A$  telles que  $\delta(s, a)$  est défini. On suppose qu'il n'y a **pas d'état deadlock**, i.e., on requiert que pour tout  $s \in S$ ,  $A(s) \neq \emptyset$ .

# Jeux stochastiques à deux joueurs

## Parties, objectifs et gains

Soit  $\mathcal{G} = (S, S_1, S_2, A, \delta)$  un jeu stochastique.

- **Partie** : suite  $s_0 a_1 s_1 \dots \in S(AS)^\omega$  telle que pour tout  $n \in \mathbb{N}$ ,  $a_{n+1} \in A(s_n)$  et  $\delta(s_n, a_{n+1})(s_{n+1}) > 0$ .
- **Histoire** : préfixe fini d'une partie s'achevant dans un état. Pour  $i \in \{1, 2\}$ , on note  $\text{Hist}_i$  pour l'ensemble des histoires terminant dans un état de  $S_i$ .



# Jeux stochastiques à deux joueurs

## Parties, objectifs et gains

Soit  $\mathcal{G} = (S, S_1, S_2, A, \delta)$  un jeu stochastique.

- **Partie** : suite  $s_0 a_1 s_1 \dots \in S(AS)^\omega$  telle que pour tout  $n \in \mathbb{N}$ ,  $a_{n+1} \in A(s_n)$  et  $\delta(s_n, a_{n+1})(s_{n+1}) > 0$ .
- **Histoire** : préfixe fini d'une partie s'achevant dans un état. Pour  $i \in \{1, 2\}$ , on note  $\text{Hist}_i$  pour l'ensemble des histoires terminant dans un état de  $S_i$ .

Il existe plusieurs manières de spécifier les buts des joueurs.

- **Objectif** : ensemble de parties gagnantes.
- **Gain** : une fonction de gain assigne à chaque partie un gain réel.

# Plan de l'exposé

1 Motivations

2 Jeux

**3 Stratégies**

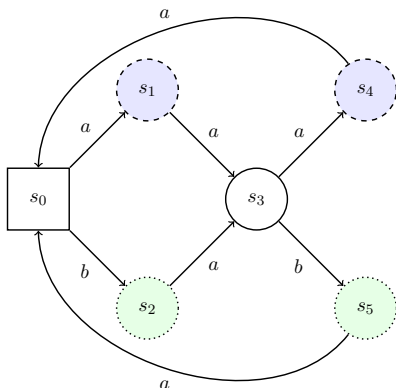
4 Stratégies et représentations efficaces

# Stratégies

- Dans l'exemple précédent, pour jouer de manière optimale, il suffisait de jouer en fonction de l'état courant.

# Stratégies

- Dans l'exemple précédent, pour jouer de manière optimale, il suffisait de jouer en fonction de l'état courant.
- En général, les joueurs ont besoin de mémoire pour jouer de manière optimale.



## Définition

Soit  $i \in \{1, 2\}$ . Une **stratégie** du joueur  $i$  est une fonction  $\sigma: \text{Hist}_i(\mathcal{G}) \rightarrow \mathcal{D}(A)$  telle que pour tout  $h = s_0 a_1 s_1 \dots s_n \in \text{Hist}_i(\mathcal{G})$ , et toute  $a \in A$ ,

$$\sigma(h)(a) > 0 \implies a \in A(s_n).$$

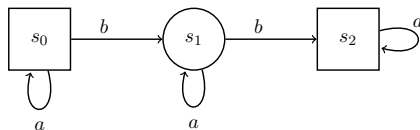
- Une stratégie est l'objet formel qui correspond à un **contrôleur** du système.

# Une stratégie peut-elle être implémentée ?

- Les stratégies qui nous intéressent en **pratique** doivent pouvoir être **implémentées**.
- Or, une stratégie quelconque peut avoir besoin de **mémoire infinie**.

# Une stratégie peut-elle être implémentée ?

- Les stratégies qui nous intéressent en pratique doivent pouvoir être implémentées.
- Or, une stratégie quelconque peut avoir besoin de mémoire infinie.



- Objectif :  $\{(s_0a)^\omega\} \cup \{(s_0a)^k s_0b(s_1a)^k s_1b(s_2a)^\omega \mid k \in \mathbb{N}\}$ .

# Stratégies à mémoire finie

Le modèle classique utilisé pour représenter un contrôleur est basé sur des automates.

## Définition

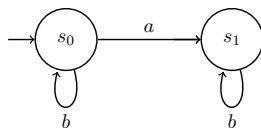
Une stratégie de  $\mathcal{P}_i$  est dite à **mémoire finie** si elle peut être encodée par une **machine de Mealy** stochastique  $\mathcal{M} = (M, \mu_{\text{init}}, \alpha_{\text{up}}, \alpha_{\text{act}})$  où :

- $M$  est un ensemble fini d'états de mémoire ;
- $\mu_{\text{init}} \in \mathcal{D}(M)$  est une distribution initiale ;
- $\alpha_{\text{up}}: M \times S \times A \rightarrow \mathcal{D}(M)$  est une fonction de mise à jour de la mémoire ;
- $\alpha_{\text{act}}: M \times S_i \rightarrow \mathcal{D}(A)$  est une fonction d'action.

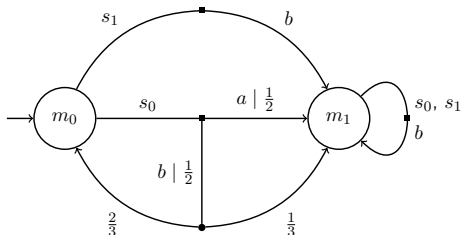


# Comment jouer avec une stratégie à mémoire finie ?

- On considère le jeu suivant.



- Un exemple de **machine de Mealy** pour ce jeu peut être représentée comme suit.



# Plan de l'exposé

1 Motivations

2 Jeux

3 Stratégies

4 Stratégies et représentations efficaces

# Qu'est-ce qu'une bonne stratégie ?

En pratique, pour une classe donnée de spécifications, on recherche des algorithmes **efficaces** pour synthétiser des stratégies **optimales**, **simples** et **petites**.

↪ Les questions de **taille** et de **simplicité** sont dépendantes du **modèle** choisi pour représenter les stratégies.

# Qu'est-ce qu'une bonne stratégie ?

En pratique, pour une classe donnée de spécifications, on recherche des algorithmes **efficaces** pour synthétiser des stratégies **optimales**, **simples** et **petites**.

↪ Les questions de **taille** et de **simplicité** sont dépendantes du **modèle** choisi pour représenter les stratégies.

Le point central de ma thèse porte sur ces **modèles de stratégie**.

- Comprendre les **liens** entre les différents modèles.
- Identifier la **complexité** des stratégies indépendamment des modèles.

# Classification des stratégies à mémoire finie

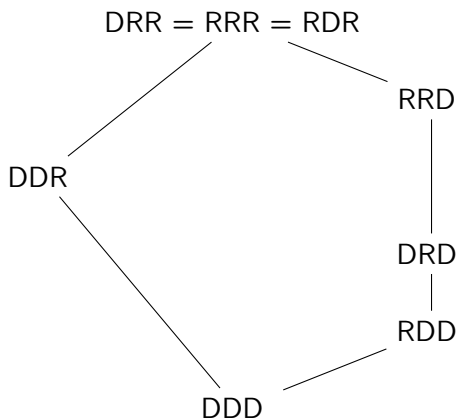
- **Résultats préliminaires** : classification des machines de Mealy en termes de **richesse des comportements possibles**.
- Les machines de Mealy ont en généralité une **initialisation**, des **sorties** et des **mises à jour** stochastiques.

## Question préliminaire

Comment est affectée la **richesse des comportements** si on rend certains de ces aspects **déterministes** ?

- **Notation** pour classifier les machines de Mealy : acronymes XYZ où X, Y,  $Z \in \{D, R\}$  où D signifie déterministe, R aléatoire (random) et
  - X correspond à l'initialisation,
  - Y correspond aux sorties,
  - Z correspond aux mises à jour.

# Classification des stratégies à mémoire finie



Merci pour votre attention !  
Des questions ?

# Références I