Modélisation Probabiliste et Apprentissage par Renforcement

Benoît Delahaye

Nantes Université, LS2N UMR 6004

Chapitre 1 : Modélisation Probabiliste

Outline

Mon Parcours

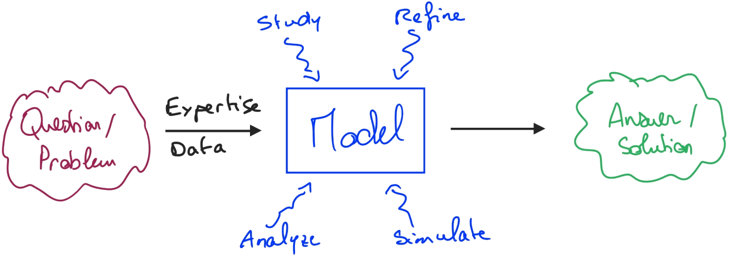
- Bac 2000
- ENS Cachan 2003-2007
- Doctorat Univ Rennes 1 2007-2010
  Spécification Modulaire et Analyse Compositionnelle de Systèmes Stochastiques
- Postdoc Aalborg Univ. Danemark 2011-2012
- Maître de Conférences Univ. Nantes 2013
- Habilitation à Diriger des Recherches 2020
  Modélisation et vérification de systèmes incertains

Science

Science [1]

Science is a systematic enterprise that builds and organizes knowledge in the form of testable explanations and predictions about the universe. [...]

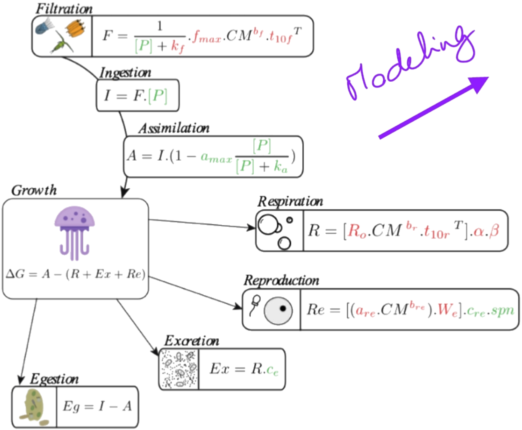
- Natural sciences
  Study nature
- Social sciences
  Study society
- Formal sciences
  Study abstract concepts



1. https://en.wikipedia.org/wiki/Science

Example : Prediction of the growth of a jellyfish

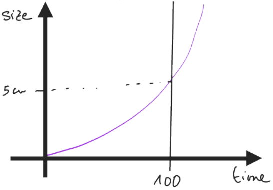
What is the size after 100 days ?



Modeling

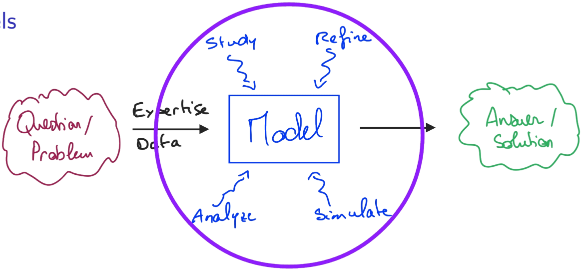
```
# Masse initiale en poids humide (en g)
        CW=0.0036*WM;
        cw=CW
        for i in Temps_D:
            if score_degrowth<lim_degrowth:
            # Application du modèle
                X0=[CW, T, 0]
                [Clear, Ingest, Assim, Respi, Excre, Prod, Egest, Scope_grow
th] = pelagia_feed(X0,param1,param2,0);
                # Stockage des variables
                CW=CW+ Scope_growth*dt;
                wm=CW*(100/0.36);
                pelsize=(wm/a)**(1/b);
                if not pelsize>0:
                    print(pelsize)
                cw=np.append(cw,CW) ;
                WM=np.append(WM,wm);
                Size1=np.append(Size1,pelsize);

                val=vec_D[int(i*10)]
                val=val[0]

                if (val):
                    d=vec_D[int(i*10)]
                    d=d[1]
                    d=d[0]
                    if not (degrowth[d,1]-degrowth[d,2]<Size1[x_d[d]]<degrow
th[d,1]+degrowth[d,2]):
                        score_degrowth=score_degrowth+1

            else:
                bool_break1=True
                break

U(DOS)---  modele2017-02-20-1.py   43% L241   (Python)
```

Verification



Theoretical Computer Science

Science of Models



Motivation

Providing **theories**, **tools** and **techniques** for building and analyzing models

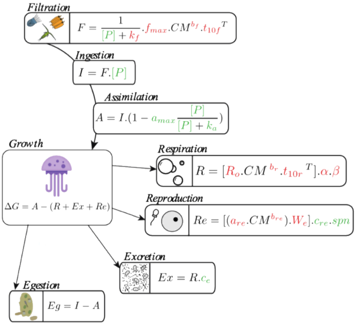
- **Modeling :** How to build models

  Language (formalism) – Expressivity – Abstraction – Manipulation

- **Verification :** How to analyze models

  (Semi-)Automation – Complexity/Efficiency – Diagnosis

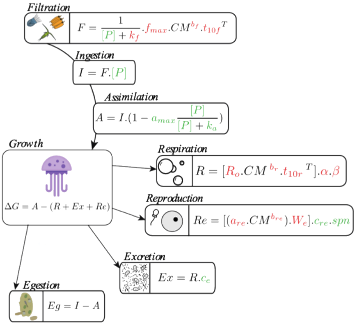
Inconvenience of deterministic modeling



Choosing the value of unknown variables

- Guess ?
- Experimental data ?
- Mean value ?

Inconvenience of deterministic modeling



Choosing the value of unknown variables

- Guess ?
- Experimental data ?
- Mean value ?

The **deterministic** model is **not** representative of a familly of variable systems

Modeling with uncertainties

Deterministic modeling requires perfect knowledge of the systems

Uncertainties can come from several sources

- Incomplete knowledge of the system or environment (ex : unknown variables)
- Abstraction : the model is too complex to be studied in its entirety
- Take into account choices in the system's behaviour
- ...

Modeling with uncertainties

Deterministic modeling requires perfect knowledge of the systems

Uncertainties can come from several sources

- Incomplete knowledge of the system or environment (ex : unknown variables)
- Abstraction : the model is too complex to be studied in its entirety
- Take into account choices in the system's behaviour
- ...

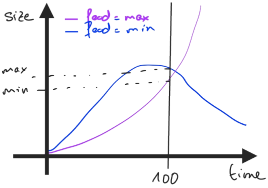
How are uncertainties accounted for ?

- Non-determinism
- Probabilities
- Parameters
- ...

Example : Jellyfish size ctd.

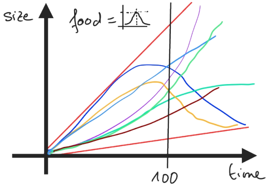
- **Non-determinism :** No further information

  $\Rightarrow$ Study all potential outputs



- **Probabilities :** Imperfect information

  $\Rightarrow$ Study the distribution of outputs



- **Parameters :** Building more information

  $\Rightarrow$ The outcome is a function of p

size = f(food)
$\Rightarrow$ Optimization

Plan du cours

- Chapitre 1 : Modélisation probabiliste **10H**
  - Chaînes de Markov à Temps Discret (DTMC)
  - Processus de décision Markoviens (MDP)
  - Chaînes de Markov à Temps Continu (CTMC)
  - Automates Probabilistes (Temporisés)
- Chapitre 2 : Vérification de systèmes probabilistes **10H**
  - Model Checking Probabiliste
    - Propriétés d'accessibilité
    - Logiques LTL / CTL / PCTL
    - Vérification de propriétés qualitatives et quantitatives
  - Model Checking Statistique
    - Vérification quantitative (Monte Carlo)
    - Vérification qualitative (Test d'hypothèses)
- Chapitre 3 : Apprentissage par renforcement (RL) **10H**
  - Fondements du RL
  - Application au model checking statistique

Evaluation

Examen sur table — 50%

A la fin des cours, sur tout le module, le **29/03/23**.

Projet de TP (en binôme) — 50%

Construction d'un model-checker probabiliste et statistique pour les modèles probabilistes discrets. 2 notes :

- Note 1 : Interprétation et simulation de chaînes de Markov et MDP. Rendu et rapport noté pour le **01/03/23**.
- Note 2 : Model checking et SMC de chaînes de Markov. Extension aux MDP (besoin de RL pour le model checking statistique). Rendu et rapport noté pour le **01/04/23**.

Outline

Nécessité des probabilités

- Abstraction
- Processus physiques
- Incertitude
- Briser la symétrie

Nécessité des probabilités

- Abstraction
  - Changement d'échelle
- Processus physiques
- Incertitude
- Briser la symétrie

Nécessité des probabilités

- Abstraction
  - Changement d'échelle
  - Utilisation des probabilités pour remplacer un processus complexe
- Processus physiques
- Incertitude
- Briser la symétrie

Nécessité des probabilités

- Abstraction
  - Changement d'échelle
  - Utilisation des probabilités pour remplacer un processus complexe
- Processus physiques
  - Environnement, Fautes, Variabilité
- Incertitude
- Briser la symétrie

Nécessité des probabilités

- Abstraction
  - Changement d'échelle
  - Utilisation des probabilités pour remplacer un processus complexe
- Processus physiques
  - Environnement, Fautes, Variabilité
  - Mécanismes naturels (mécanique quantique)
- Incertitude
- Briser la symétrie

Nécessité des probabilités

- Abstraction
  - Changement d'échelle
  - Utilisation des probabilités pour remplacer un processus complexe
- Processus physiques
  - Environnement, Fautes, Variabilité
  - Mécanismes naturels (mécanique quantique)
- Incertitude
  - Protocoles expérimentaux
- Briser la symétrie

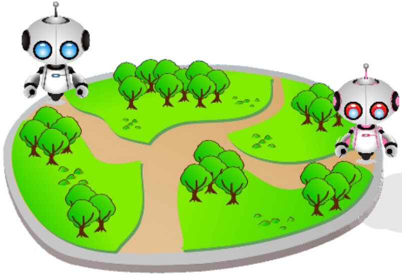
Nécessité des probabilités

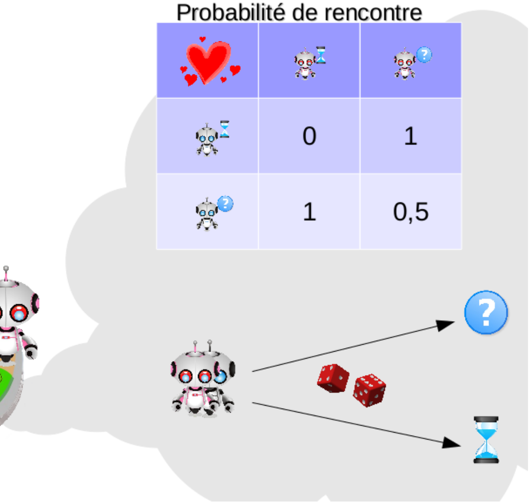
- Abstraction
  - Changement d'échelle
  - Utilisation des probabilités pour remplacer un processus complexe
- Processus physiques
  - Environnement, Fautes, Variabilité
  - Mécanismes naturels (mécanique quantique)
- Incertitude
  - Protocoles expérimentaux
  - Matériel
- Briser la symétrie

Nécessité des probabilités

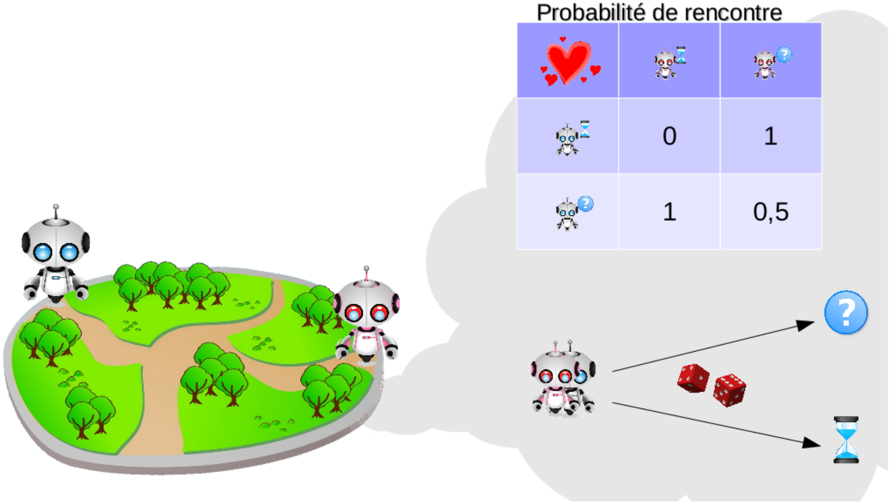
- Abstraction
  - Changement d'échelle
  - Utilisation des probabilités pour remplacer un processus complexe
- Processus physiques
  - Environnement, Fautes, Variabilité
  - Mécanismes naturels (mécanique quantique)
- Incertitude
  - Protocoles expérimentaux
  - Matériel
- Briser la symétrie
  - Introduites artificiellement
    Ex : Protocoles réseau

Briser la symétrie





Briser la symétrie



Sans probabilité : $\frac{1}{2}$ Avec probabilités : $\frac{2}{3}$

Modèles et Vérification probabilistes

Il est donc impératif de développer des modèles prenant en compte les probabilités et des méthodes pour vérifier ces modèles.

Le but de cet exposé est de présenter rapidement

- Des formalismes de modélisation probabilistes
  - Discrets
  - Continus
- Les principes de deux techniques de vérification probabilistes
  - Le model checking Probabiliste
  - Le model checking statistique

Outline

Modèles Probabilistes

Il existe pléthore de modèles permettant de prendre en compte les probabilités pour tous types de systèmes.

Ici, on ne présentera que certains d'entre eux :

- Chaînes de Markov (Discretes)
- Processus de Décision Markoviens / Automates Probabilistes (Discrets)
- Chaînes de Markov continues
- Automates Temporisés Probabilistes (continus)

Quel formalisme utiliser ?

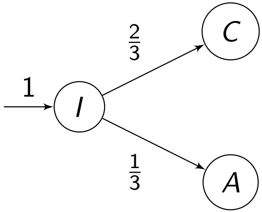
Cela dépend du système à modéliser... le choix se fait en fonction des caractéristiques particulières du systèmes et du type de propriétés à vérifier.

- Discret / Continu (/ Hybride)
- Pûrement probabiliste / Non-déterministe
- Système ouvert / fermé
- Temps explicite / Temps implicite
- Paramètres
- ...

Outline

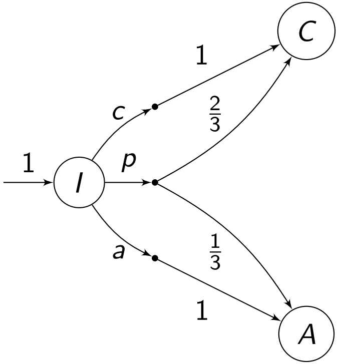
Chaînes de Markov à temps Discret (DTMC)

Une *chaîne de Markov à temps discret* (DTMC) est un système de transitions dont les transitions sont étiquetées par des probabilités discrètes.



Processus de Décision Markoviens (MDP)

Dans les Processus de Décision Markoviens (MDP), on autorise des choix non-déterministes entre plusieurs distributions de probabilités dans chaque état.



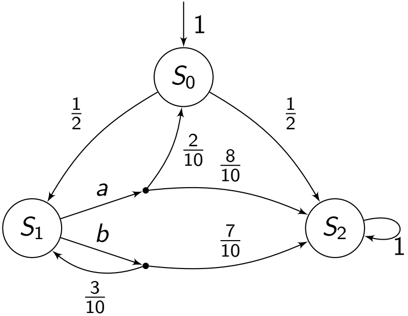
Présentation du Projet noté – Partie 1

Construction d'un model-checker probabiliste et statistique pour les chaînes de Markov et les MDP, en Python.
`https://uncloud.univ-nantes.fr/index.php/s/9Ec2cXfKxoxCp8y`

- Etape 1 : Structure de données et lecture d'un modèle

Modèle 1

```
States S0, S1, S2;
Actions a,b;
S0 -> 5:S1 + 5:S2;
S1 [a] -> 2:S0 + 8:S2;
S1 [b] -> 3:S1 + 7:S2;
S2 -> 1:S2;
```



- Etape 2 : Interprétation et simulation de Chaînes de Markov et MDP
- Etape 3 : Rendu et évaluation pour le **01/03/23**

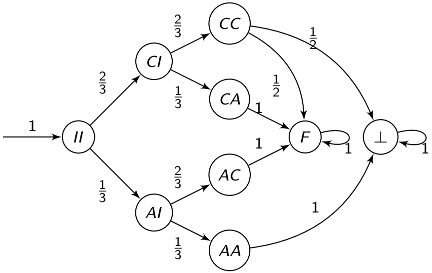
DTMC : Définition

Definition (DTMC)

Une DTMC est un tuple $\mathcal{M} = (S, P, \iota_{\text{init}}, AP, L)$, où

- S est un ensemble non-vide et dénombrable d'états,
- $P : S \times S \to [0,1]$ est la fonction de transition probabiliste, telle que pour tout état s,
$$\sum_{s' \in S} P(s, s') = 1,$$
- $\iota_{\text{init}} : S \to [0,1]$ est la distribution initiale, telle que $\sum_{s \in S} \iota_{\text{init}}(s) = 1$,
- AP est un ensemble de propositions atomiques et $L : S \to 2^{AP}$ est une fonction d'étiquetage.

Exemple



$$P = \begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\iota_{\text{init}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Exercice

Exercice 1

Modéliser à l'aide d'une chaîne de Markov le protocole de communication simplifié qui suit. Le protocole procède en 4 étapes :

1. Sans communication, on est dans un état d'attente ("iddle"). De cet état, on peut aller avec probabilité 1 vers un état d'envoi ("Send").
2. Lorsque l'on envoie un message (depuis l'état "send"), on a une probabilité $\frac{1}{10}$ que le message soit perdu. Si c'est le cas, on va dans un état "lost". Si le message n'est pas perdu, on va dans l'état "delivered".
3. Depuis l'état "lost", on revient avec probabilité 1 vers l'état "send" pour renvoyer le message.
4. Depuis l'état "delivered", on revient avec probabilité 1 vers l'état "iddle".

Construire la chaîne de Markov représentant ce protocole et identifiez la matrice de transitions et la distribution initiale.

Interprétation(s)

2 visions principales :

- **transitoire** : Dans cette interprétation, la distribution initiale permet de choisir un état de départ, puis la matrice de transition permet de choisir un état successeur, etc.

- **à l'équilibre** : Dans cette vision, à un instant donné, la chaîne ne se trouve pas dans un état singulier mais dans une distribution d'état. A l'origine, elle se trouve dans la distribution initiale. La matrice de transition permet alors de faire évoluer la distribution d'états de la chaîne à chaque transition.

Interprétation(s)

2 visions principales :

- **transitoire** : Dans cette interprétation, la distribution initiale permet de choisir un état de départ, puis la matrice de transition permet de choisir un état successeur, etc.
  - 1 état courant
  - Distribution sur les chemins
  - Propriétés sur les chemins
- **à l'équilibre** : Dans cette vision, à un instant donné, la chaîne ne se trouve pas dans un état singulier mais dans une distribution d'état. A l'origine, elle se trouve dans la distribution initiale. La matrice de transition permet alors de faire évoluer la distribution d'états de la chaîne à chaque transition.

Interprétation(s)

2 visions principales :

- **transitoire** : Dans cette interprétation, la distribution initiale permet de choisir un état de départ, puis la matrice de transition permet de choisir un état successeur, etc.
  - 1 état courant
  - Distribution sur les chemins
  - Propriétés sur les chemins
- **à l'équilibre** : Dans cette vision, à un instant donné, la chaîne ne se trouve pas dans un état singulier mais dans une distribution d'état. A l'origine, elle se trouve dans la distribution initiale. La matrice de transition permet alors de faire évoluer la distribution d'états de la chaîne à chaque transition.
  - "Etat" courant = distribution sur les états
  - 1 seul chemin
  - Propriétés sur "états" visités

Interprétation transitoire : Chemins, Cylindres et Mesure

Definition (Chemin, cylindre)

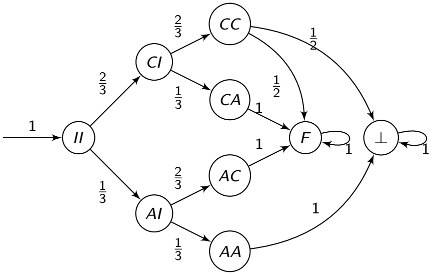
- Un *chemin* est une séquence (infinie) d'états $\pi = s_0 s_1 \cdots \in S^\omega$ telle que $P(s_i, s_{i+1}) > 0$ pour tout $i \geq 0$.
- Etant donné un chemin fini $\widehat{\pi} = s_0, \ldots s_n \in S^*$, le *cylindre* engendré par $\widehat{\pi}$

$$\mathrm{Cyl}(\widehat{\pi}) = \{w = w_0, \ldots, w_k \ldots \in S^\omega \mid w_i = s_i, 0 \leq i \leq n\}$$

Mesure de probabilité définie sur les cylindres :

$$\mathrm{Pr}^{\mathcal{M}}(\mathrm{Cyl}(s_0 \ldots s_n)) = \iota_{\mathrm{init}}(s_0) \cdot \prod_{0 \leq i < n} P(s_i, s_{i+1})$$

Exemple



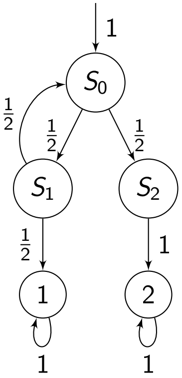
Cylindre

$\mathrm{Cyl}(II.CI.CC) = \{II.CI.CC.F.F.\dots, \; II.CI.CC.\perp.\perp.\dots\}$

Probabilité

$\mathbb{P}(\mathrm{Cyl}(II.CI.CC)) = \frac{4}{9}$

Exercice



Exercice 2

1. Calculer la probabilité des chemins suivants :
   1.1 $S_0 S_2 2$
   1.2 $S_0 S_1 S_0 S_2 2$
   1.3 $S_0 S_1 S_0 S_1 S_0 S_2 2$
2. Quelle est la probabilité globale d'arriver dans l'état 2 ?
3. Quelle est la probabilité globale d'arriver dans l'état 1 ?
4. Proposer une chaîne de Markov permettant de simuler un dé à 3 faces avec des pièces de monnaie.
5. Proposer une chaîne de Markov permettant de simuler un dé à 6 faces avec des pièces de monnaie.

Exercice

Exercice 3

Le Craps est un jeu de dés, qui fonctionne par jets successifs d'une paire de dés à 6 faces. Lors du premier lancer, on gagne si on obtient un total de 7 ou 11 et on perd si on effectue un total de 2, 3 ou 12. Dans les autres cas, on enregistre le score effecuté (qu'on appelle le "point") et on relance les dés. Lors du 2e lancer, et des suivants, on gagne si on obtient un total égal au point, on perd si on obtient un total de 7, et on relance les dés à nouveau dans les autres cas, jusqu'à ce que le point ou le 7 soit obtenu.

1. Proposer un modèle du jeu de Craps sous forme de chaîne de Markov
2. Enumérer les chemins permettant de gagner en moins de 2 lancers
3. Calculer la probabilité de gagner en moins de 2 lancers, en moins de 3 lancers
4. Calculer la probabilité de gagner éventuellement

Exercice

Exercice 4

Le protocole zeroconf est un protocole de communication ad-hoc entre appareils électroniques. Dans ce protocole, chaque appareil doit s'auto-attribuer une adresse IP en évitant autant que possible les collisions. Pour ce faire, un nouvel appareil effectue les étapes suivantes :

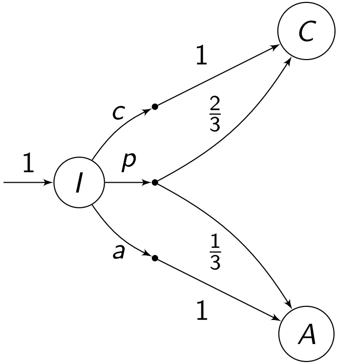
- Choisir une adresse au hasard (sur un total de T adresses).
- Envoyer successivement k messages à tout le monde demandant si l'adresse n'est pas déjà attribuée. Les messages et les réponses peuvent se perdre.
- S'il reçoit une réponse, il reprend à 0. S'il ne reçoit aucune réponse, il considère que son adresse n'est pas attribuée et la conserve.

1. Considérer le cas $T = 10$, $k = 2$, et que la probabilité de "perdre" un message ou sa réponse est de 40%. Supposer qu'il y a déjà 5 appareils installés sur le réseau (donc 5 adresses attribuées). Modéliser à l'aide d'un MDP et calculer la probabilité de s'attribuer une adresse non utilisée.
2. Généraliser le problème à $k = 3$, puis à k quelconque.

Outline

Processus de Décision Markoviens (MDP)

Dans les Processus de Décision Markoviens (MDP), on autorise des choix non-déterministes entre plusieurs distributions de probabilités dans chaque état.



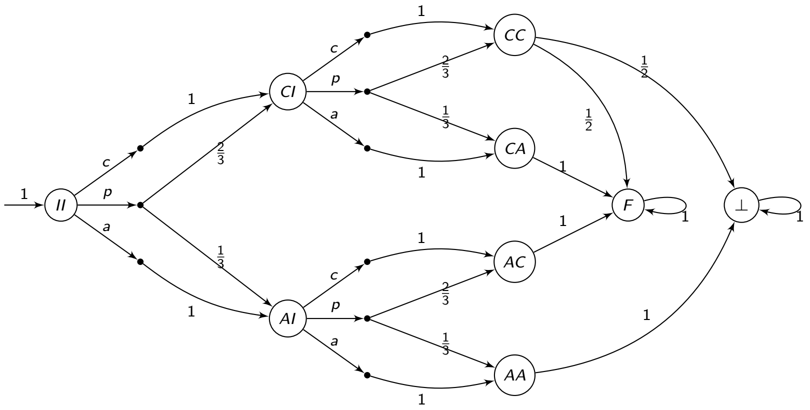
MDP : Définition

Un Processus de Décision Markovien (MDP) est un tuple $\mathcal{M} = (S, \mathrm{Act}, P, \iota_{\mathrm{init}}, AP, L)$, où

- S, ι_{init}, L et AP sont définis comme pour les DTMC,
- Act est un ensemble d'actions, et
- $P : S \times \mathrm{Act} \times S \to [0, 1]$ est la fonction de transition probabiliste, telle que pour tout état s et pour toute action a,

$$\sum_{s' \in S} P(s, a, s') \in \{0, 1\}.$$

Exemple


1
II
c
p
a
1
2/3
1/3
1
CI
AI
c
p
a
1
2/3
1/3
1
c
p
a
1
2/3
1/3
1
CC
CA
AC
AA
1/2
1/2
1
1
1
F
⊥
1
1

Interprétation transitoire : Chemins ?

Les chemins dépendent des choix non-déterministes effectués :

Definition (Chemin)

Un chemin est une séquence infinie $\pi = s_0\alpha_1 s_1\alpha_2 \ldots \in (S \times \mathrm{Act})^\omega$ telle que $P(s_i, \alpha_{i+1}, s_{i+1}) > 0$ pour tout $i \geq 0$.

Afin de définir une mesure de probabilité, il faut donc prendre en compte les choix non-déterministes

Definition (Adversaire)

Un *adversaire* de $\mathcal{M}$ est une fonction $\mathfrak{S} : S^+ \to \mathrm{Act}$ telle que, pour tout $s_0 s_1 \ldots s_n \in S^+$, on a $\mathfrak{S}(s_0 s_1 \ldots s_n) \in \mathrm{Act}(s_n)$.

Un adversaire est **positionnel** s'il ne dépend que de l'état courant (i.e. $\mathfrak{S} : S \to \mathrm{Act}$).

Chaîne induite et mesure de probabilité

Un adversaire permettant de résoudre tous les choix non-déterministes, il transforme le MDP en chaîne de Markov induite :

Definition (MC induite)

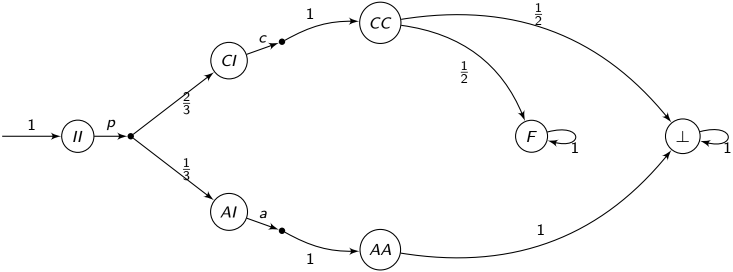
$$\mathcal{M}_{\mathfrak{S}} = (S^+, P_{\mathfrak{S}}, \iota_{\text{init}}, AP, L'),$$

où pour tout $\sigma = s_0 s_1 \ldots s_n$, $P_{\mathfrak{S}}(\sigma, \sigma s_{n+1}) = P(s_n, \mathfrak{S}(s_n), s_{n+1})$, et $L'(\sigma) = L(s_n)$.

On note alors $\mathbb{P}_{\mathfrak{S}}$ la mesure de probabilité associée

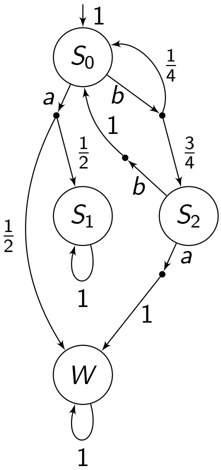
Exemple

Ex : $\mathfrak{S}$ choisit p en II, c en CI et a en AI



$$\mathbb{P}_{\mathfrak{S}}(\mathrm{Cyl}(II.CI.CC)) = \frac{2}{3}$$

Exercice



Exercice 5

1. Définir un adversaire $\mathfrak{S}$ et construire la chaîne de Markov induite $\mathcal{M}_{\mathfrak{S}}$.
2. Quelle est la probabilité $\mathbb{P}_{\mathfrak{S}}$ d'arriver en W ?
3. Quelle est la probabilité $\mathbb{P}_{\mathfrak{S}}$ d'arriver en W en moins de 2 transitions ?
4. Proposer un adversaire $\mathfrak{S}_{max}$ qui maximise la probabiliter d'arriver en W, sans borne sur le nombre de transitions.
5. Proposer un adversaire $\mathfrak{S}^2_{max}$ qui maximise la probabiliter d'arriver en W, en 2 transitions au maximum.
6. Généraliser à k transitions au maximum.

Exercice

Exercice 6

Dans le jeu de Craps, on ajoute une nouvelle possibilité. Lorsque l'on a ni gagné ni perdu au premier lancer (uniquement), on a le choix entre poursuivre normalement ou jeter une pièce de monnaie.

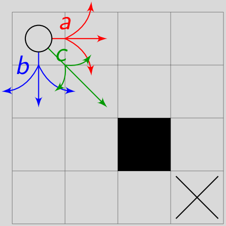
- Si on obtient pile, on poursuit le jeu normalement ;
- Si on obtient face, on reprend à 0.

1. Modéliser ce nouveau jeu à l'aide d'un MDP.
2. Considérer l'adversaire qui choisit systématiquement de jeter la pièce et construire la chaîne de Markov induite.
3. Calculer la probabilité, dans la chaîne induite, de gagner en 1 lancer de dés, en moins de 2 lancers de dés, en moins de 3 lancers de dés.
4. Proposer un adversaire qui maximise la probabilité de gagner.
5. Et si on pouvait choisir de lancer la pièce à chaque fois que l'on doit relancer les dés ?

Exercice

Exercice 7

On considère le déplacement d'une bille sur un plateau (cf dessin ci-dessous). Le but du joueur est d'emmener la bille dans la case marquée d'une croix, en évitant la case noire. Si la bille sort du plateau ou si elle entre dans la case noire, la partie s'arrête.



Dans chaque position, le joueur a 3 possibilités :

- *a* La bille va vers la droite avec probabilité $\frac{3}{5}$. Avec probabilité, $\frac{1}{5}$ elle dévie vers le haut ou le bas.
- *b* La bille va vers le bas avec probabilité $\frac{3}{5}$. Avec probabilité, $\frac{1}{5}$ elle dévie vers la gauche ou la droite.
- *c* La bille va en diagonale avec probabilité $\frac{1}{2}$. Avec probabilité $\frac{1}{4}$ elle dévie vers la gauche ou le haut.

- Modéliser le problème sous forme de processus de décision Markovien.
- Proposer un adversaire et calculer la probabilité de gagner avec cet adversaire.
- Calculer la probabilité maximale de gagner.

Exercice

Exercice 8

Soit $\mathcal{M} = (S, \text{Act}, P, \iota_{\text{init}}, AP, L)$ un processus de décision Markovien.

1. Donner la définition de la chaîne de Markov induite par un adversaire positionnel $\mathfrak{S}$

2. Soit $\mathfrak{S}_{np}$ un adversaire quelconque et soit π un chemin dans la chaîne de Markov induite. Montrer qu'il existe un adversaire positionnel $\mathfrak{S}_p$ et un chemin π' dans la chaîne de Markov induite par $\mathfrak{S}_p$ tels que π_p termine dans le même état que π et $\mathbb{P}_{\mathfrak{S}_p}(\pi_p) \geq \mathbb{P}_{\mathfrak{S}_{np}}(\pi)$.

3. Etant donné un état $s \in S$, montrer qu'il existe toujours un adversaire positionnel qui maximise la probabilité d'atteindre s.

Outline

Récompenses

On peut considérer des fonctions de récompenses dans les modèles Markoviens. Plusieurs définitions possibles

Definition

Fonction de récompense Etant donné un modèle markovien (MC ou MDP) $\mathcal{M} = (S, (A,)P, \iota_{\text{init}} \ldots)$, une fonction de récompense est une fonction rew : $S \rightarrow \mathbb{N}$.

- Non négatif (a priori)
- Peut être associé aux action, aux transitions, plutôt qu'aux états

Markov Reward Models

Definition

Un **Modèle de récompenses Markovien** (MRM) est le couplage $(\mathcal{M}, \text{rew})$ d'un modèle Markovien (MC ou MDP) $\mathcal{M}$ et d'une fonction de récompenses rew : $S \to \mathbb{N}$ portant sur ses états. Définitions pour les MC, mais adaptable aux MDP.

La récompense d'un état est pris en compte lorsqu'on le quitte.
La récompense d'un chemin fini $\widehat{\pi} = s_0, \ldots s_n \in S^*$ est

$$\text{rew}(\widehat{\pi}) = \sum_{i=0}^{n-1} \text{rew}(s_i)$$

Récompense attendue pour l'accessibilité

On note $\lozenge B$ la propriété d'atteindre un ensemble B d'états.

La récompense attendue d'un chemin fini $\pi = s_0, s_1 \dots$ relative à l'accessibilité de B est

$$\text{rew}(\pi, \lozenge B) = \begin{cases} \text{rew}(s_0, \dots s_n) & \text{si } s_i \notin B \text{ pour } 0 \leq i < n \text{ et } s_n \in B \\ \infty & \text{si } \pi \not\models \lozenge B \end{cases}$$

Definition

Soient s un état et B un ensemble d'états. La **récompense attendue** pour atteindre B depuis s est $\text{ExpRew}(s \models \lozenge B) = \infty$ si $\mathbb{P}(s \models \lozenge B) < 1$ et sinon

$$\text{ExpRew}(s \models \lozenge B) = \sum_{\widehat{\pi} = s_0, \dots s_n \;|\; s_0 = s, s_n \in B \text{ et } s_0, \dots s_{n-1} \notin B} \mathbb{P}(\widehat{\pi}) \cdot \text{rew}(\widehat{\pi})$$

Exercice

Exercice 9

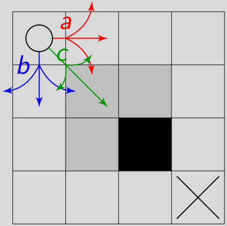
Dans un premier temps, reprendre le modèle de lancers de pièces permettant de simuler un dé à 6 faces.

1. Définir une fonction de récompense rew et un ensemble B permettant de calculer le nombre de lancers attendus avant d'arriver dans un état "final".
2. Calculer cette valeur.
3. Considérer maintenant le jeu de Craps (version chaîne de Markov), et faire de même.
4. Qu'en est-il pour la version "optimale" du Craps avec lancer de pièce ?

Exercice

Exercice 10

On reprend l'exercice de la bille sur le plateau. On considère maintenant que les cases claires ont un coût de 1, les cases foncées un coût de 5 et la case noire un coût de 20.



Dans chaque position, le joueur a 3 possibilités :

- *a* La bille va vers la droite avec probabilité $\frac{3}{5}$. Avec probabilité, $\frac{1}{5}$ elle dévie vers le haut ou le bas.
- *b* La bille va vers le bas avec probabilité $\frac{3}{5}$. Avec probabilité, $\frac{1}{5}$ elle dévie vers la gauche ou la droite.
- *c* La bille va en diagonale avec probabilité $\frac{1}{2}$. Avec probabilité $\frac{1}{4}$ elle dévie vers la gauche ou le haut.

- Modéliser le problème sous forme d'un modèle de récompense Markovien.
- Proposer un adversaire et calculer l'espérance du coût pour gagner avec cet adversaire.
  Les définitions proposées sont-elles correctes ?
- Trouver l'adversaire minimisant cette espérance.

MDP vs Automates Probabilistes

Automates Probabilistes : plus de non-déterminisme d'action

- MDP : 1 distribution par action au plus
- PA : pas de restriction

Attention à l'interprétation des PA

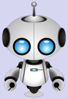
- Segala : Similaire aux MDP
- Rabin : Transducteurs, langages, etc.

Outline

Modèles continus

Dans certains cas, il est nécessaire de quantifier le temps de manière plus précise que par du “pas à pas”. Il est alors nécessaire d’avoir recours aux modèles continus, à temps implicite (Chaînes de Markov continues) ou explicite (automates temporisés probabilistes).

Exemple



Au bout d’un délai donné, le robot peut changer de “mode”

Chaînes de Markov Continues

Représentation similaire aux DTMC, mais transitions temporisées par des délais suivant une distribution exponentielle.

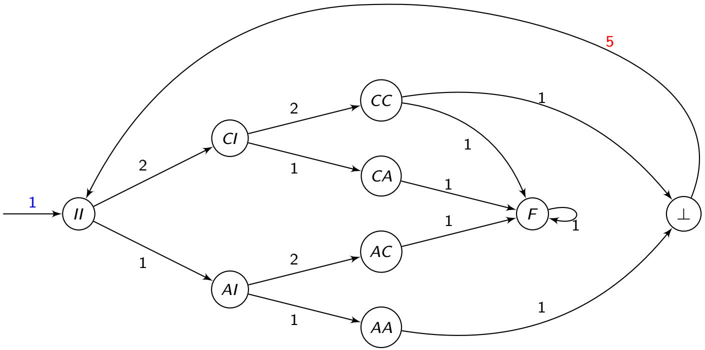
Definition (CTMC)

Une CTMC est un tuple $\mathcal{C} = (S, \iota_{\text{init}}, R, AP, L)$, où

- S, ι_{init}, L et AP sont définis comme pour les DTMC, et
- $R : S \times S \to \mathbb{R}_{\geq 0}$ est la fonction de transition, qui étiquette chaque transition possible par sa vitesse (rate), nécéssairement positive.

On note $E(s) = \sum_{s' \in S} R(s, s')$ la *vitesse totale* de l'état s

Exemple



CTMC : Sémantique

La probabilité de prendre une transition depuis l'état s *avant* t unités de temps est donnée par

$$1 - e^{-E(s)\cdot t}$$

Lorsque plusieurs transitions sont activées dans s ($R(s, s') > 0$), une *course* s'enclenche. La probabilité d'aller en s' est alors

$$\frac{R(s, s')}{E(s)}$$

Simplification

- Les choix des délais sont continus (exponentiels)
- Les choix des successeurs sont discrets (DTMC sous jacente)

CTMC : chemins et mesure de probabilité

Les chemins d'une CTMC comprennent à la fois les états discrets traversés et les délais choisis.

Definition (Chemin)

Un chemin de $\mathcal{C}$ est une séquence alternée $s_0 t_0 s_1 t_1 \ldots$, où $s_i \in S$ sont les états visités, $t_i \in \mathbb{R}_{\geq 0}$ sont les délais d'attente dans chacun de ces états, et où $R(s_i, s_{i+1}) > 0$ pour tout i.

Mesure de probabilité

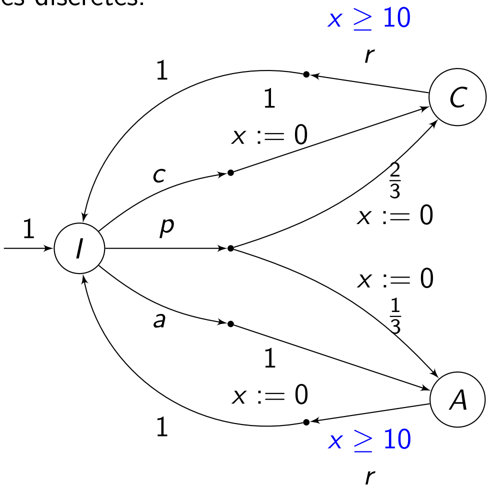
Définie de manière similaire aux DTMCs, mais

Cylindres $= s_0, I_0, \ldots, I_{n-1}, s, n$ avec I_i des intervalles non vides de $\mathbb{R}_{\geq 0}$.

Outline

Automates temporisés probabilistes (PTA)

Les automates temporisés probabilistes sont des automates temporisés dont les transitions sont étiquetées (en plus des guardes et réinitialisations) par des probabilités discrètes.



PTA : Définition

Definition (Automate Probabiliste Temporisé)

Un automate temporisé probabiliste (PTA) est un tuple $\mathcal{T} = (\mathcal{S}, X, \mathrm{Act}, T, \iota_{\mathrm{init}}, I, AP, L)$, où

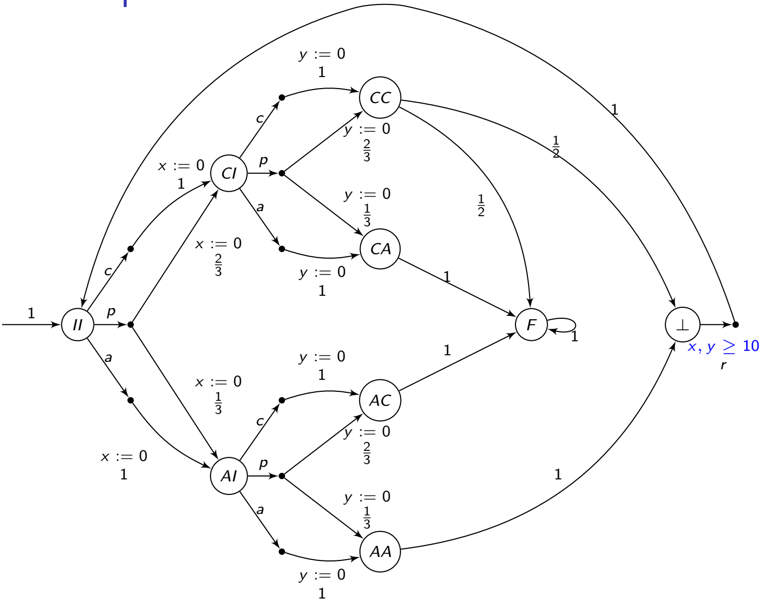
- $\mathcal{S}$ est un ensemble non-vide et dénombrable de localités,
- X est un ensemble fini d'horloges,
- Act est un ensemble d'actions,
- $T \subseteq \mathcal{S} \times C(X) \times \mathrm{Act} \times \mathrm{Dist}(2^X \times \mathcal{S})$ est une relation de transition,
- ι_{init}, AP et L sont définis comme pour les DTMC, et
- $I : \mathcal{S} \to C(X)$ assigne un invariant à chaque localité.

Restrictions

Afin d'éviter les blocages et d'assurer que les distributions de probabilités sont bien définies, on fait les deux hypothèses suivantes :

1. Lorsque, dans une localité atteignable de $\mathcal{T}$, l'écoulement du temps n'est pas permis (i.e. viole l'invariant), la garde d'au moins une transition discrète doit être satisfaite.
2. Lors de l'exécution d'une transition discrète, il n'est pas possible d'atteindre un état non-admissible, i.e. lorsque la garde d'une transition discrète est satisfaite, les invariants de tous les états atteignables avec une probabilité non nulle sont satisfaits.

PTA : Exemple



1
II
c
p
a
x := 0
1
CI
x := 0
2/3
x := 0
1/3
AI
c
p
a
y := 0
1
CC
y := 0
2/3
y := 0
1/3
CA
y := 0
1
y := 0
1
AC
y := 0
2/3
y := 0
1/3
AA
y := 0
1
1/2
1/2
1
1
F
1
1
⊥
x, y ≥ 10
r
1

Sémantique, Chemins

Definition (Chemin)

Un *chemin* dans un PTA est une séquence alternée infinie $\pi = (s_0, d_0)\alpha_1(s_1, d_1)\alpha_2 \ldots \in (\mathcal{S} \times \mathbb{R}_{\geq 0} \times \text{Act})^\omega$ telle que, pour tout i, il est possible d'attendre un délai d_i dans la localité s_i et de prendre ensuite une transition $(s_i, g_i, \alpha_i, \mu)$ telle que : $\exists \rho \in 2^X, \mu(\rho, s_{i+1} > 0)$.

Sémantique : Graphe de zone = MDP

Extensions des PTA

Pour plus de détails :

- CTMC [Baier et al., 2003, Kulkarni, 1995]
- PTA [Sproston, 2004, Baier et al., 2007, Kwiatkowska et al., 2002]

Extensions :

- Entrées / Sorties
- Vitesse des états
- Réseaux [David et al., 2011]
- Hybrides [Alur et al., 1995]
- ...

References I

Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T. A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. (1995).
The algorithmic analysis of hybrid systems.
Theoretical computer science, 138(1) :3–34.

Baier, C., Bertrand, N., Bouyer, P., Brihaye, T., and Größer, M. (2007).
Probabilistic and topological semantics for timed automata.
FSTTCS 2007 : Foundations of Software Technology and Theoretical Computer Science, pages 179–191.

Baier, C., Haverkort, B., Hermanns, H., and Katoen, J.-P. (2003).
Model-checking algorithms for continuous-time markov chains.
IEEE Transactions on software engineering, 29(6) :524–541.

David, A., Larsen, K., Legay, A., Mikučionis, M., Poulsen, D., Van Vliet, J., and Wang, Z. (2011).
Statistical model checking for networks of priced timed automata.
FORMATS, pages 80–96.

Kulkarni, V. G. (1995).
Modeling and analysis of stochastic systems.

References II

Kwiatkowska, M., Norman, G., Segala, R., and Sproston, J. (2002).
Automatic verification of real-time systems with discrete probability distributions.
Theoretical Computer Science, 282(1) :101–150.

Sproston, J. (2004).
Model checking for probabilistic timed systems.
Validation of Stochastic Systems, pages 299–316.