Vérification Classique et Statistique pour les Systèmes Probabilistes

Benoît Delahaye

Université de Nantes, LS2N UMR 6004

ETR 2017

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

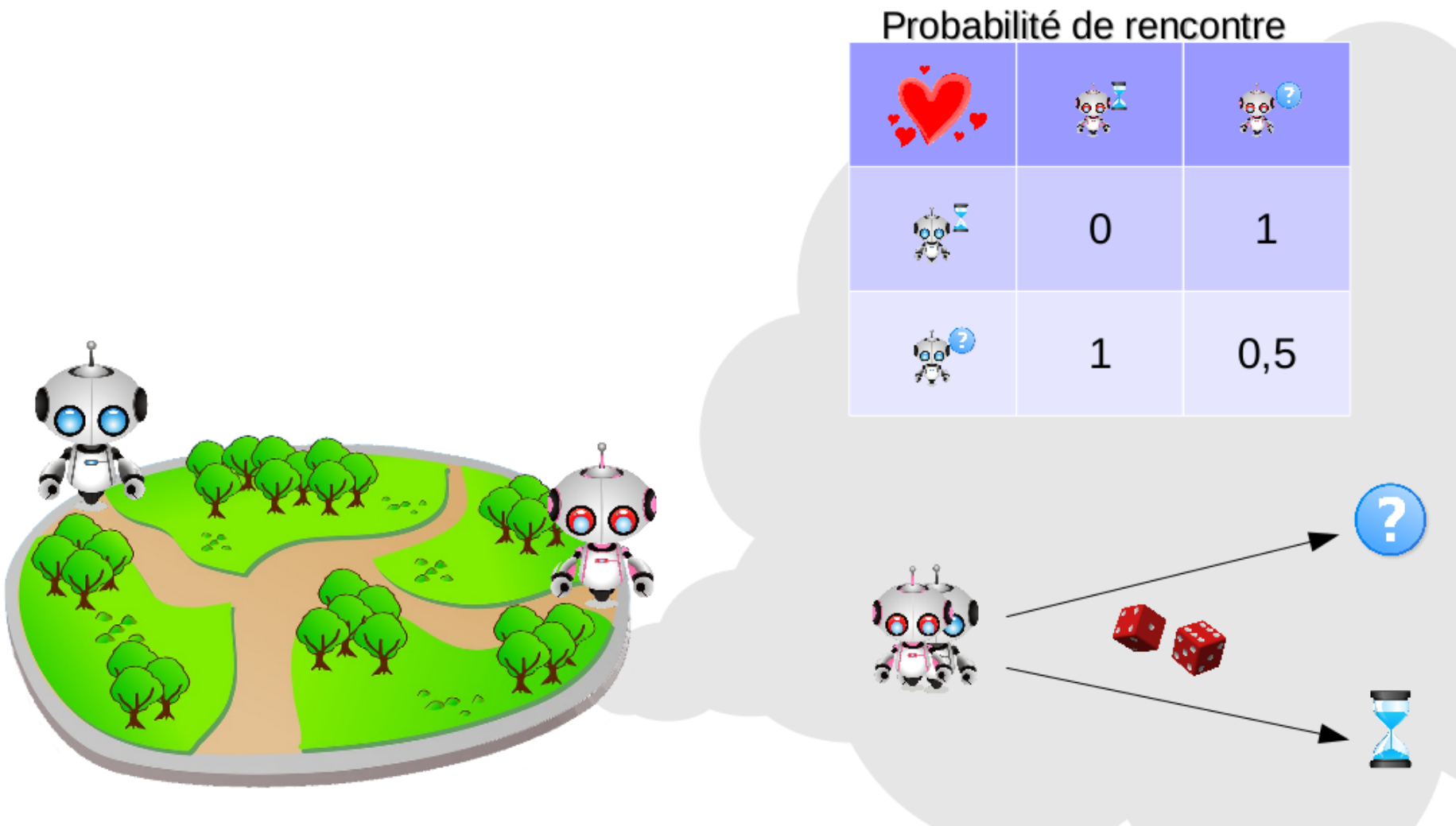
Nécessité des probabilités

- 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

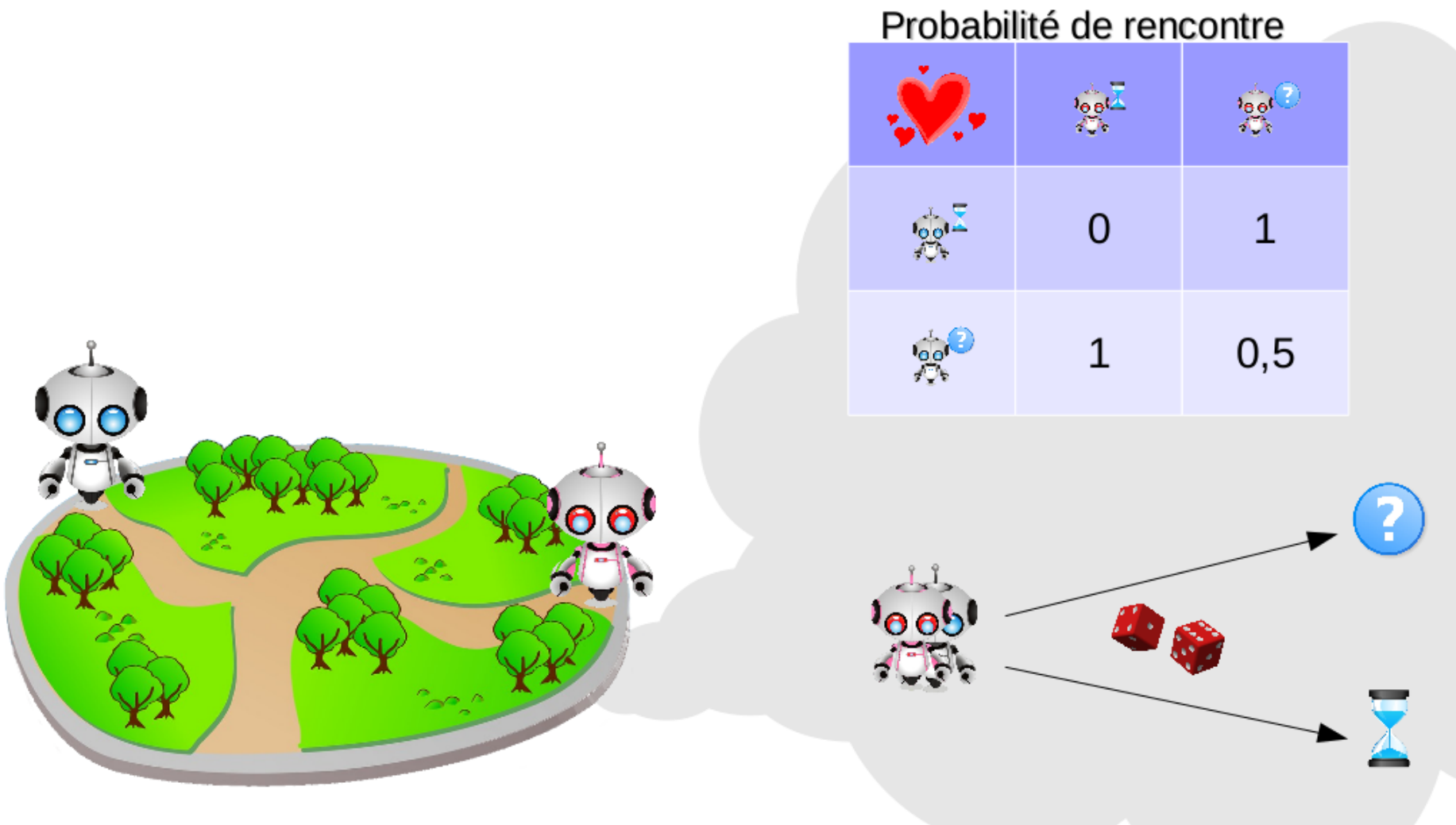
Nécessité des probabilités

- 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

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 ?

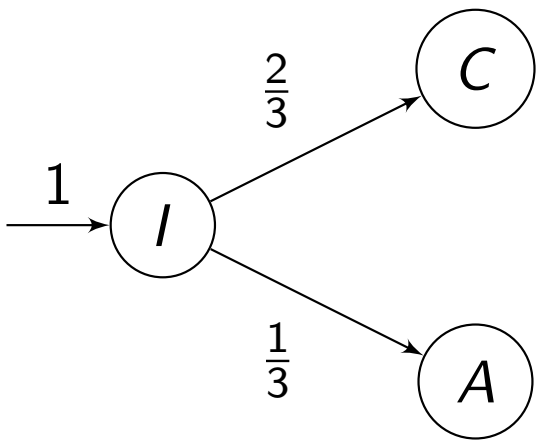
Cela dépend du système à modéliser... le choix se fait en fonction des caractéristiques particulières du systèmes et du type de propriétés à vérifier.

- 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.



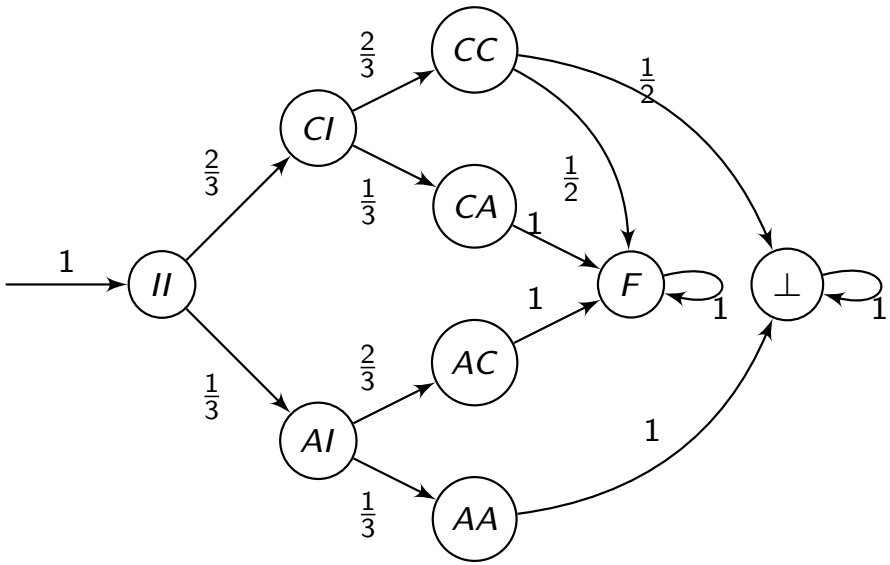
DTMC : Définition

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



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

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

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)

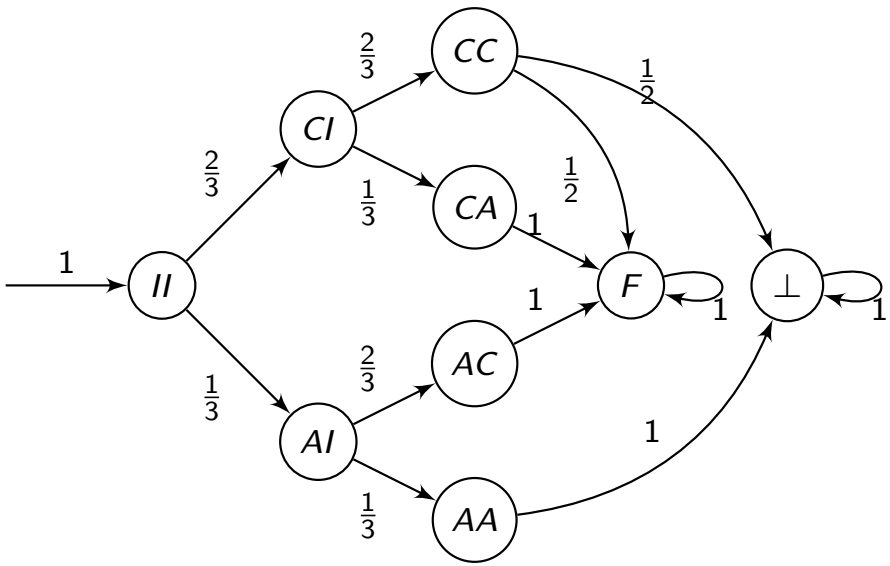
- Un *chemin* est une séquence (infinie) d'états $\pi = s_0 s_1 \cdots \in S^\omega$ telle que $P(s_i, s_{i+1}) > 0$ pour tout $i \geq 0$.
- Etant donné un chemin fini $\widehat{\pi} = s_0, \ldots s_n \in S^*$, le *cylindre* engendré par $\widehat{\pi}$

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

Mesure de probabilité définie sur les cylindres :

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

Exemple



Cylindre

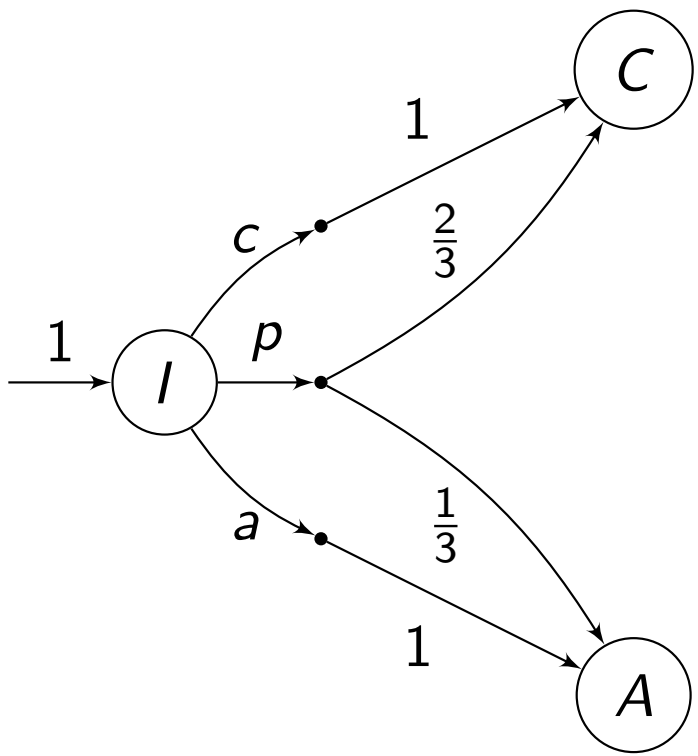
$$\mathrm{Cyl}(II.CI.CC) = \{II.CI.CC.F.F\ldots., II.CI.CC.\bot.\bot\ldots.\}$$

Probabilité

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

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.



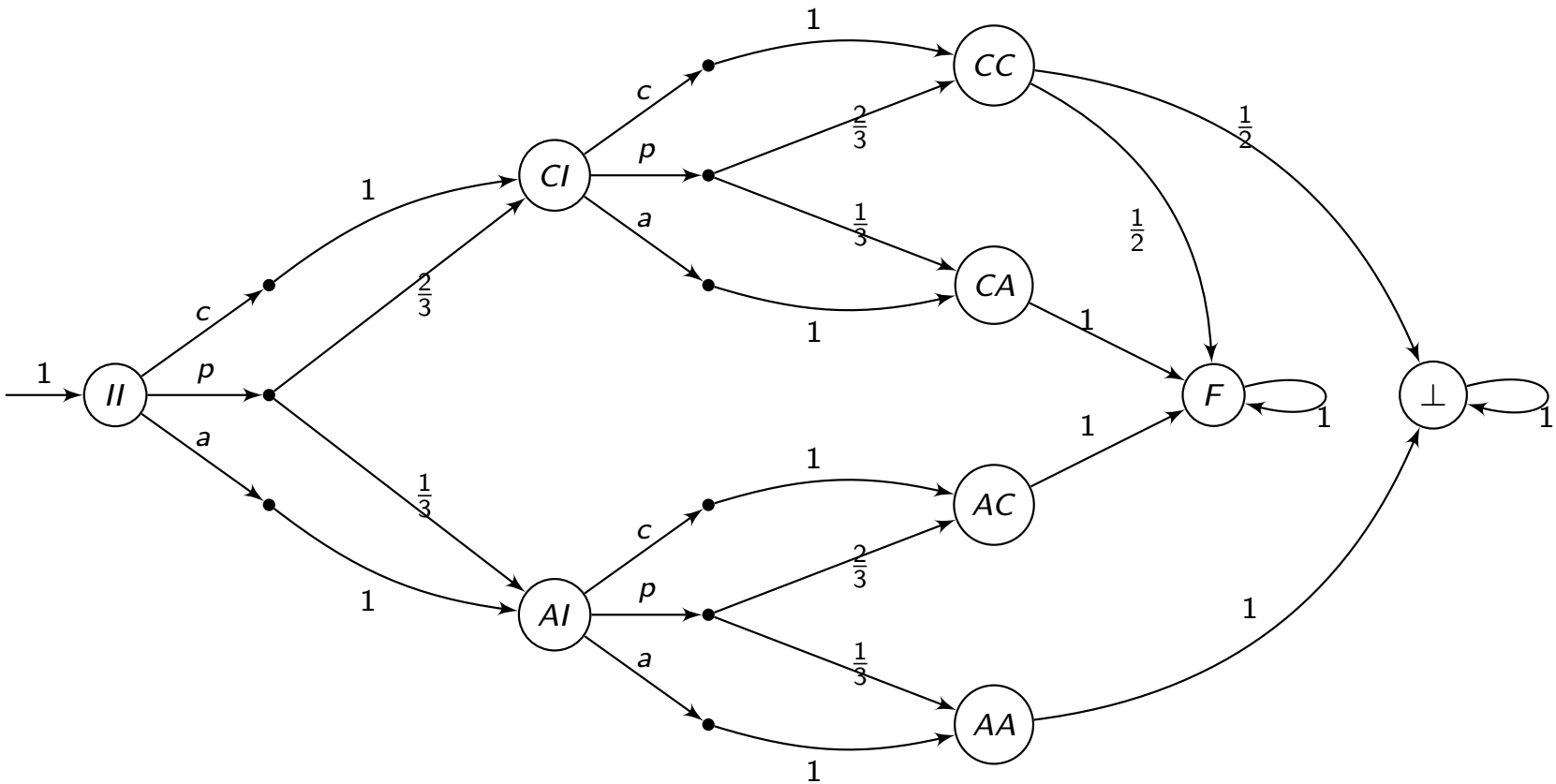
MDP : Définition

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

- S, ι_{init}, L et AP sont définis comme pour les DTMC,
- Act est un ensemble d'actions, et
- $P : S \times \text{Act} \times S \to [0, 1]$ 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



Interprétation transitoire : Chemins ?

Les chemins dépendent des choix non-déterministes effectués :

Definition (Chemin)

Un chemin est une séquence infinie $\pi = s_0\alpha_1 s_1\alpha_2 \ldots \in (S \times \text{Act})^\omega$ 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 $\mathfrak{S} : S^+ \to \text{Act}$ telle que, pour tout $s_0 s_1 \ldots s_n \in S^+$, on a $\mathfrak{S}(s_0 s_1 \ldots s_n) \in \text{Act}(s_n)$.

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)

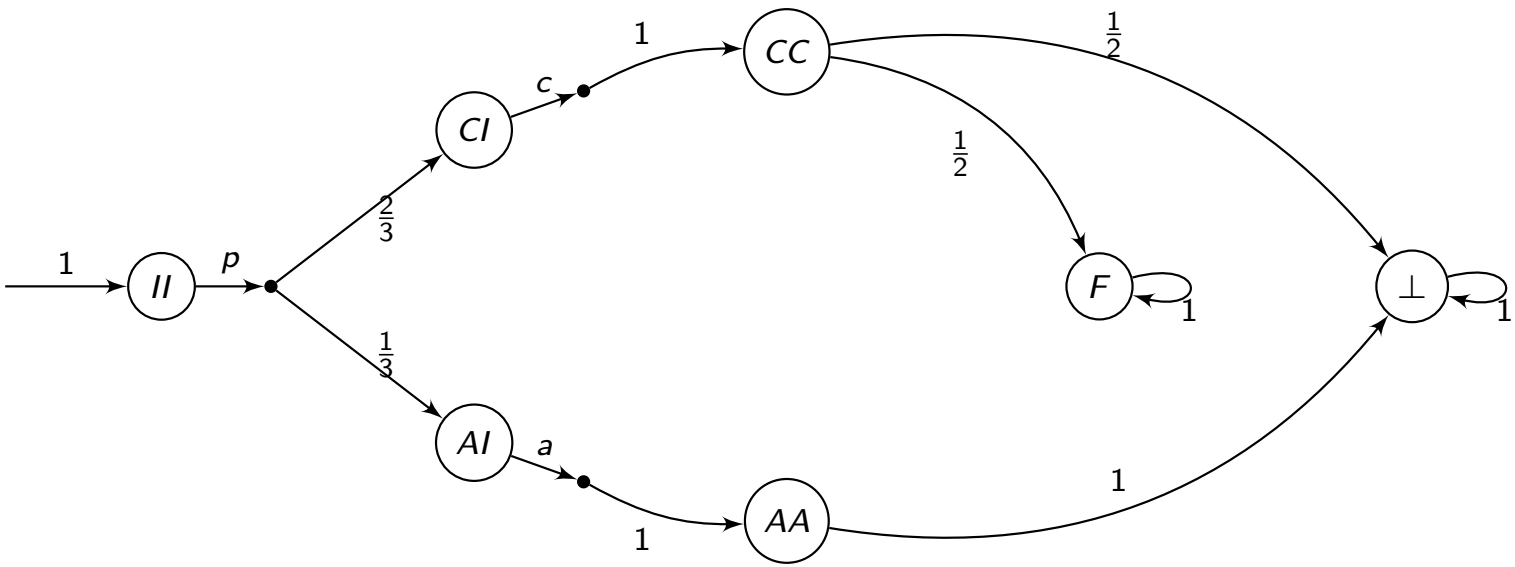
$$\mathcal{M}_{\mathfrak{S}} = (S^+, P_{\mathfrak{S}}, \iota_{\text{init}}, AP, L'),$$

où pour tout $\sigma = s_0 s_1 \dots s_n$, $P_{\mathfrak{S}}(\sigma, \sigma s_{n+1}) = P(s_n, \mathfrak{S}(s_n), s_{n+1})$, 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



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

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.

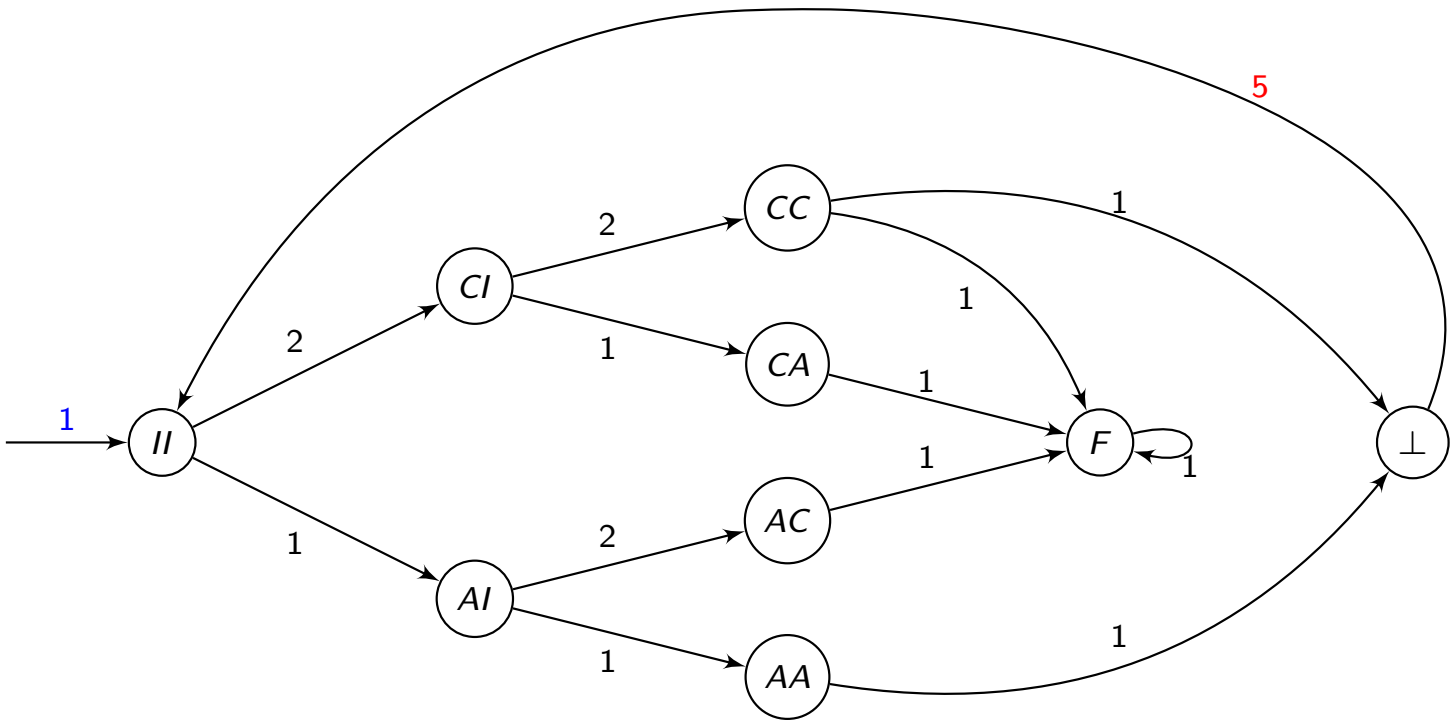
Definition (CTMC)

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.

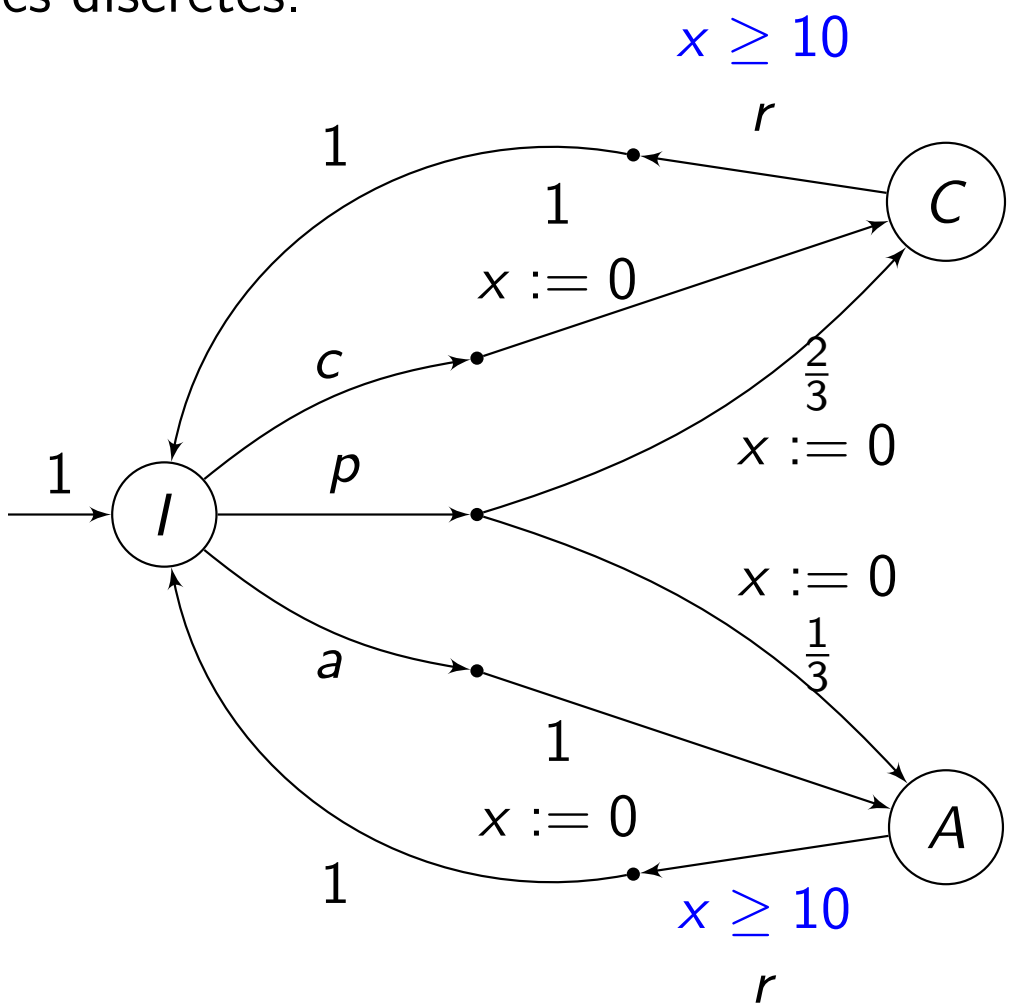
Mesure de probabilité

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 $\mathbb{R}_{\geq 0}$.

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 $\mathcal{T} = (\mathcal{S}, X, \text{Act}, T, \iota_{\text{init}}, I, AP, L)$, où

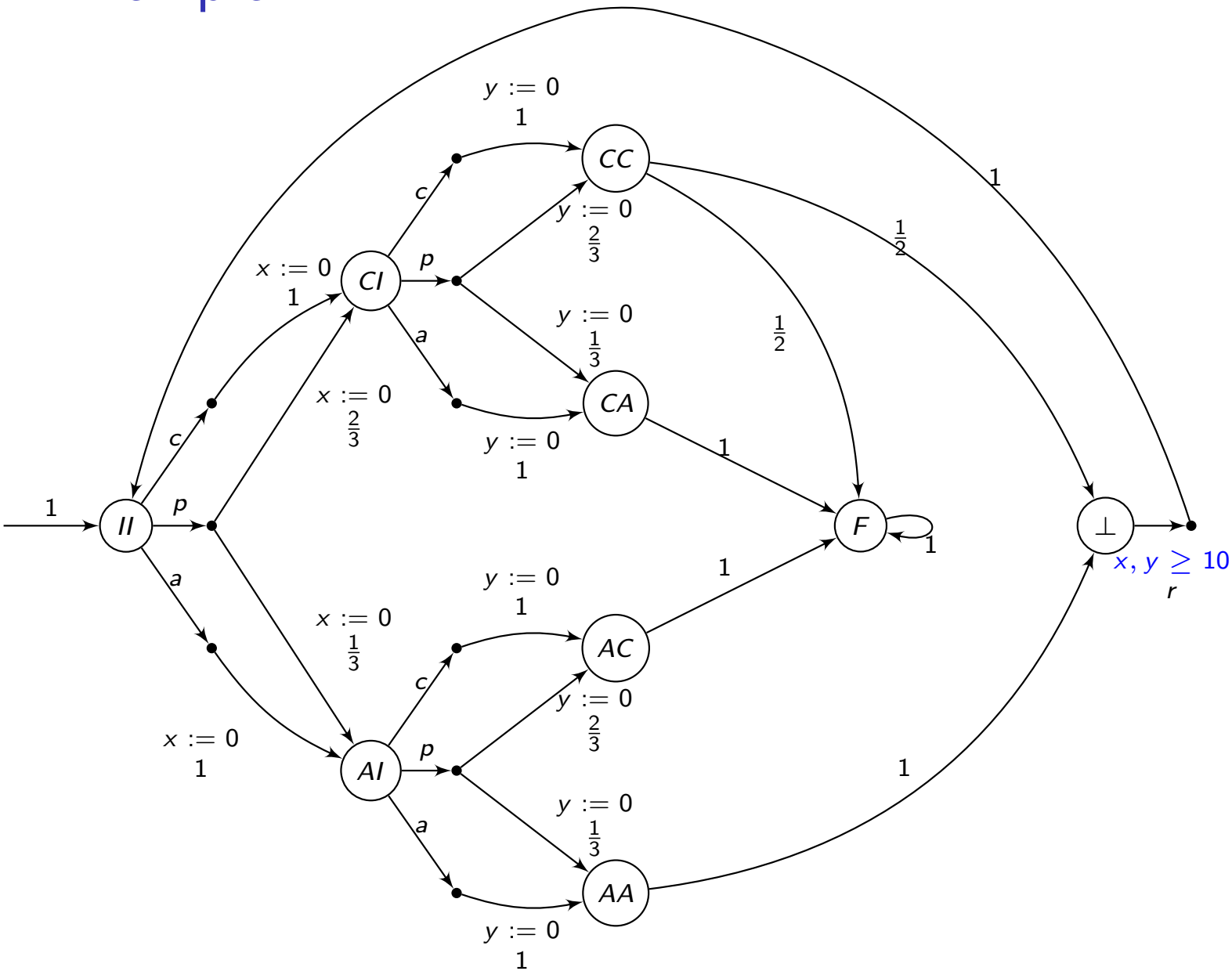
- $\mathcal{S}$ est un ensemble non-vide et dénombrable de localités,
- X est un ensemble fini d'horloges,
- Act est un ensemble d'actions,
- $T \subseteq \mathcal{S} \times C(X) \times \text{Act} \times \text{Dist}(2^X \times \mathcal{S})$ est une relation de transition,
- ι_{init}, AP et L sont définis 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



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]
- ...

Outline

Vérification Probabiliste

Comme la vérification standard,

- Basé sur les traces
- Propriétés d'états / de chemins
- Exhaustif
- Basé sur l'exploration

Mais

- Mesure sur les ensembles de chemins
- Quantification $\rightarrow$ Mesure
- Propriétés *Qualitatives*
  - $\forall \rightarrow \mathbb{P}_{=1}$
  - $\exists \rightarrow \mathbb{P}_{>0}$
- Propriétés *Quantitatives*
  - $\mathbb{P}_{\sim b},\ b \neq 0, 1$

Outline

Probabilistic Computation Tree Logic

Definition (PCTL [Hansson and Jonsson, 1994])

Formules d'états :

$$\Psi ::= \text{true} \mid a \mid \Psi_1 \wedge \Psi_2 \mid \neg\Psi \mid \mathbb{P}_J(\varphi),$$

avec $a \in AP$, φ formule de chemins et $J \subseteq [0,1]$ un intervalle à bornes rationnelles.

Formules de chemins :

$$\varphi ::= \bigcirc\Psi \mid \Psi_1 \,\mathsf{U}\, \Psi_2 \mid \Psi_1 \,\mathsf{U}^{\leq n}\, \Psi_2,$$

avec Ψ, Ψ_1 et Ψ_2 des formules d'états, et $n \in \mathbb{N}$.

Vérification de PCTL [Baier and Katoen, 2008]

Algorithme inductif sur la formule. Soit s un état.

Pour les cas de base $\Psi \equiv \text{true} \mid a \mid \Psi_1 \wedge \Psi_2 \mid \neg\Psi$, identique à CTL.

Pour $\Psi \equiv \mathbb{P}_J(\varphi)$, on a $s \models \Psi$ ssi

$$\Pr(s \models \varphi) = \mu(\{\omega = s \ldots \in S^\omega \mid \omega \models \varphi\}) \in J$$

Nécéssité d'avoir une mesure unique μ sur les chemins.

Vérification de PCTL [Baier and Katoen, 2008]

Algorithme inductif sur la formule. Soit s un état.

Pour les cas de base $\Psi \equiv \text{true} \mid a \mid \Psi_1 \wedge \Psi_2 \mid \neg\Psi$, identique à CTL.

Pour $\Psi \equiv \mathbb{P}_J(\varphi)$, on a $s \models \Psi$ ssi

$$\Pr(s \models \varphi) = \mu(\{\omega = s \ldots \in S^\omega \mid \omega \models \varphi\}) \in J$$

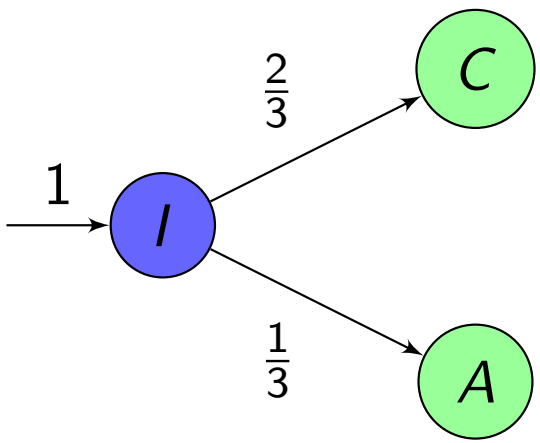
Nécéssité d'avoir une mesure unique μ sur les chemins.

Complexité : Linéaire dans la taille de la formule et polynomial dans la taille du modèle (états)

Mesure du Next

Si $\varphi = \bigcirc \Psi'$, alors

$$\Pr(s \models \bigcirc \Psi') = \sum_{s' \in \mathsf{Sat}(\Psi')} P(s, s').$$



Mesure du Until

Système d'équations linéaires.

1. Identification des états "simples" ($\Pr(s) = 0$ ou 1) $\Rightarrow S_0, S_1$.
2. Réduction du problème aux états d'intérêt $S_?$ $\Rightarrow$ Matrice A.

$$A = (P(s,t))_{s,t \in S_?}$$

3. Probabilité de gagner en 1 coup depuis $S_?$: vecteur b.

$$b = (P(s, S_1))_{s \in S_?}$$

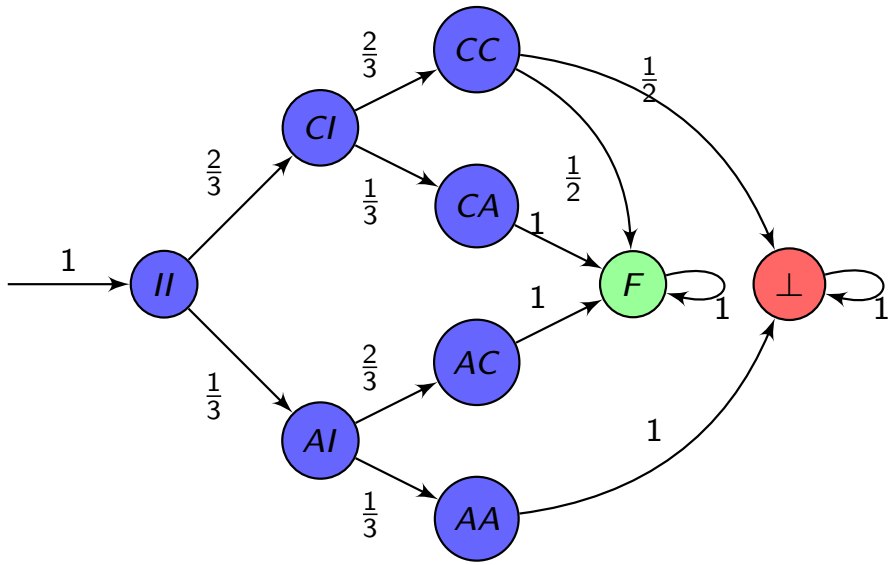
$$\Pr(s \models \Psi_1 \,\mathsf{U}\, \Psi_2) = x(s),$$

avec x le plus petit point fixe de $\Gamma(y) = A \cdot y + b$.

$$\Pr(s \models \Psi_1 \,\mathsf{U}^{\leq n}\, \Psi_2) = x_n(s),$$

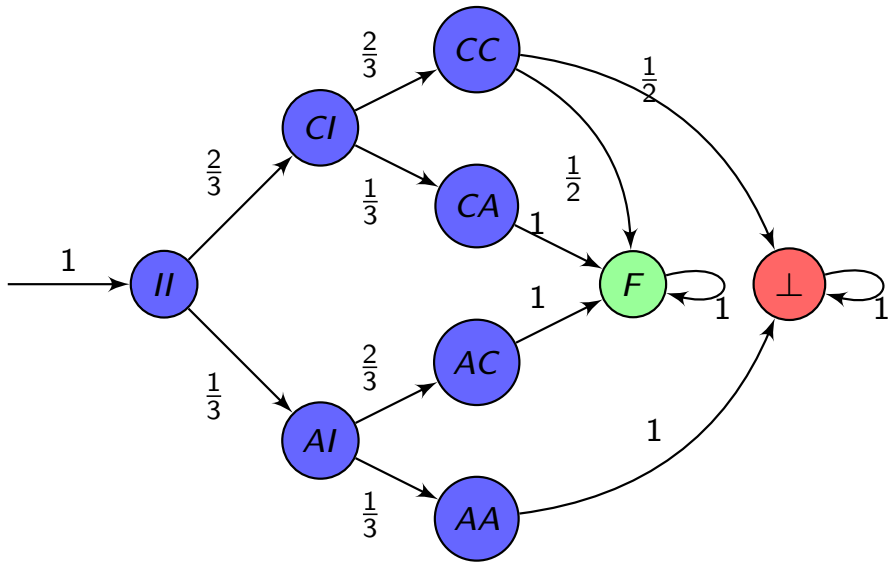
avec $x_n = \Gamma^n(x_0)$ et $x_0 = (0, \ldots 0)$.

Exemple : $\varphi \equiv \mathit{blue}\,\mathsf{U}\,\mathit{green}$



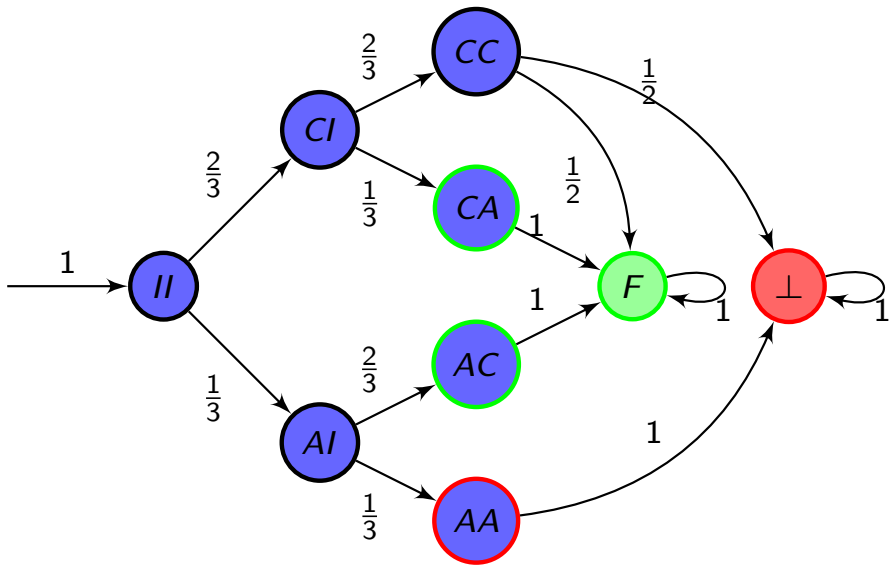
Exemple : $\varphi \equiv \mathit{blue}\ \mathsf{U}\ \mathit{green}$

1. Identifier S_0, S_1 et $S_?$



Exemple : $\varphi \equiv \mathit{blue}\ \mathsf{U}\ \mathit{green}$

1. Identifier S_0, S_1 et $S_?$

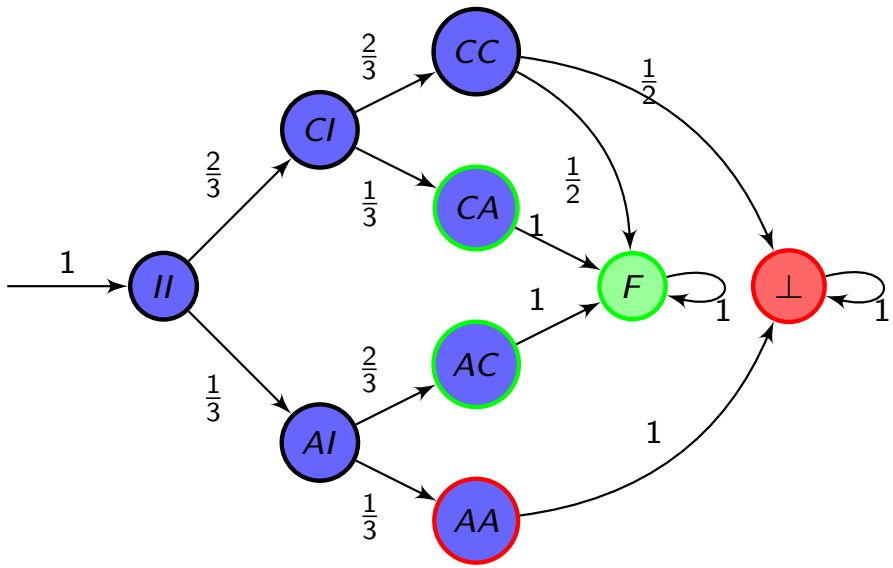


Exemple : $\varphi \equiv \mathit{blue}\ \mathsf{U}\ \mathit{green}$

1. Identifier S_0, S_1 et $S_?$
2. Matrice

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

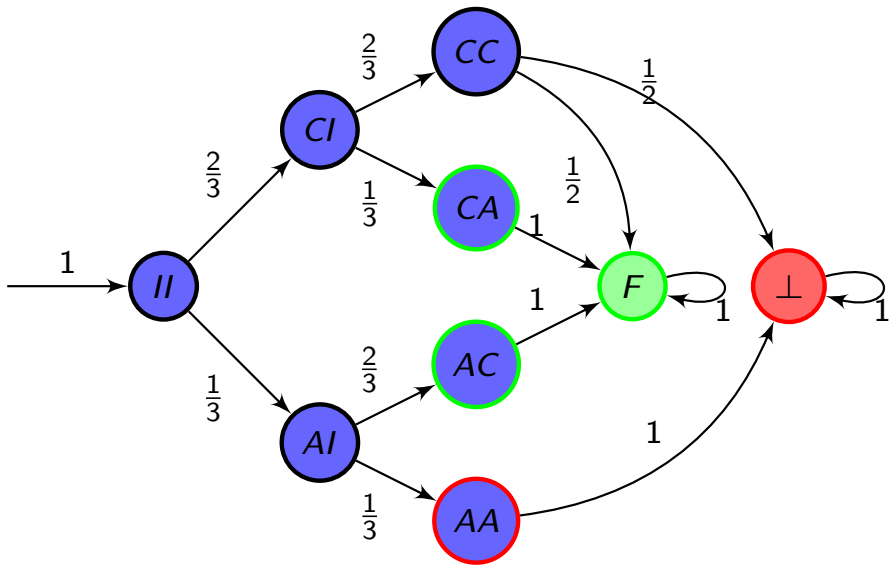
3. Vecteur
   $b = \begin{pmatrix} 0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}$



Exemple : $\varphi \equiv \mathit{blue}\ \mathsf{U}\ \mathit{green}$

1. Identifier S_0, S_1 et $S_?$
2. Matrice

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



3. Vecteur
$b = \begin{pmatrix} 0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{2} \end{pmatrix}$

4. Résolution de $\Gamma(y) = A \cdot y + b$:

$$\mathbb{P}(\mathit{II}) = \frac{2}{3}, \mathbb{P}(\mathit{CI}) = \frac{2}{3}, \mathbb{P}(\mathit{AI}) = \frac{2}{3}, \mathbb{P}(\mathit{CC}) = \frac{1}{2}$$

PCTL pour les MDP/PA [Baier and Katoen, 2008]

Plusieurs mesures de probabilité ⇒ Nécéssité de filtrer sur les adversaires.
On a alors

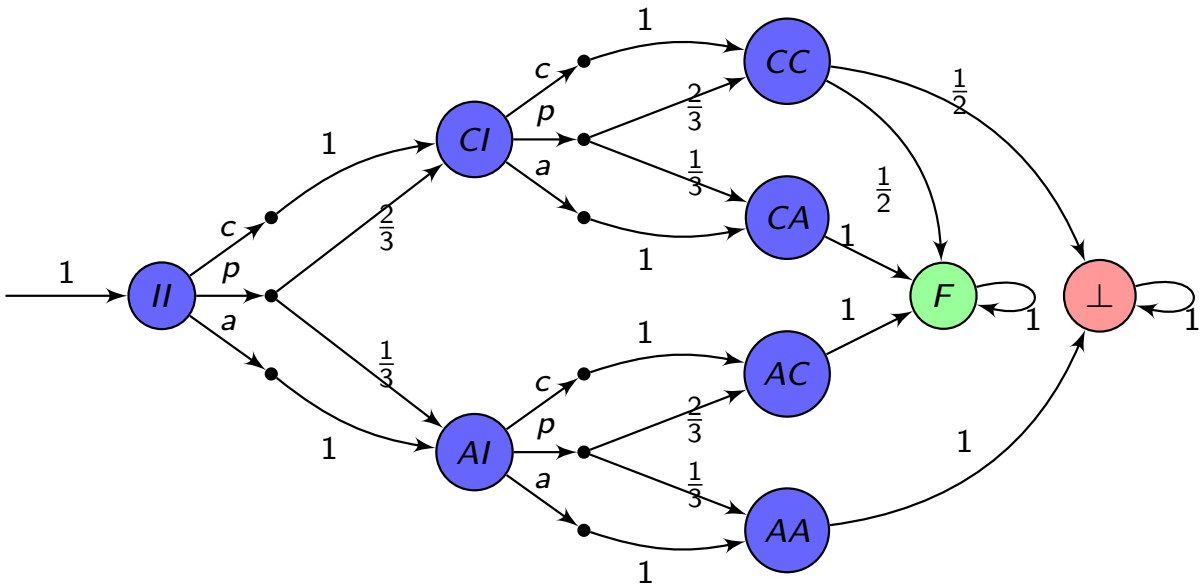
$$\begin{aligned} s \models \mathbb{P}_J^{\max}(\varphi) \Leftrightarrow & \quad \sup_{\mathfrak{S} \in \mathsf{adv}(\mathcal{M})} \Pr^{\mathcal{M}_{\mathfrak{S}}}(s \models \varphi) \in J \\ s \models \mathbb{P}_J^{\min}(\varphi) \Leftrightarrow & \quad \inf_{\mathfrak{S} \in \mathsf{adv}(\mathcal{M})} \Pr^{\mathcal{M}_{\mathfrak{S}}}(s \models \varphi) \in J \end{aligned}$$

Propriété

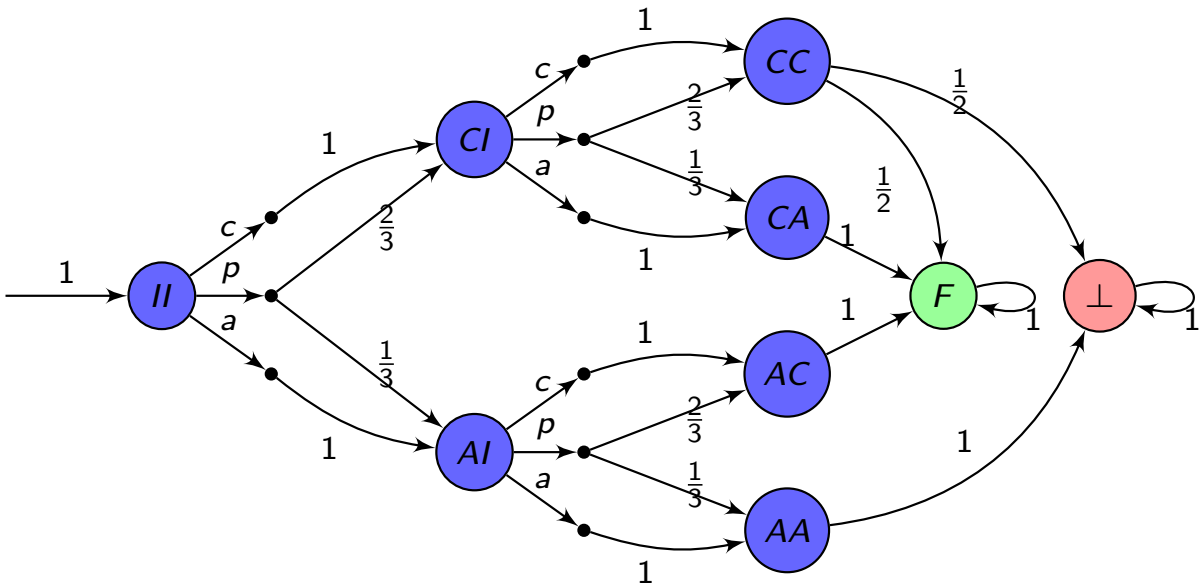
Il suffit de considérer les adversaires **positionnels**

⇒ La vérification de PCTL sur les MDPs à états finis est *décidable*

Exemple

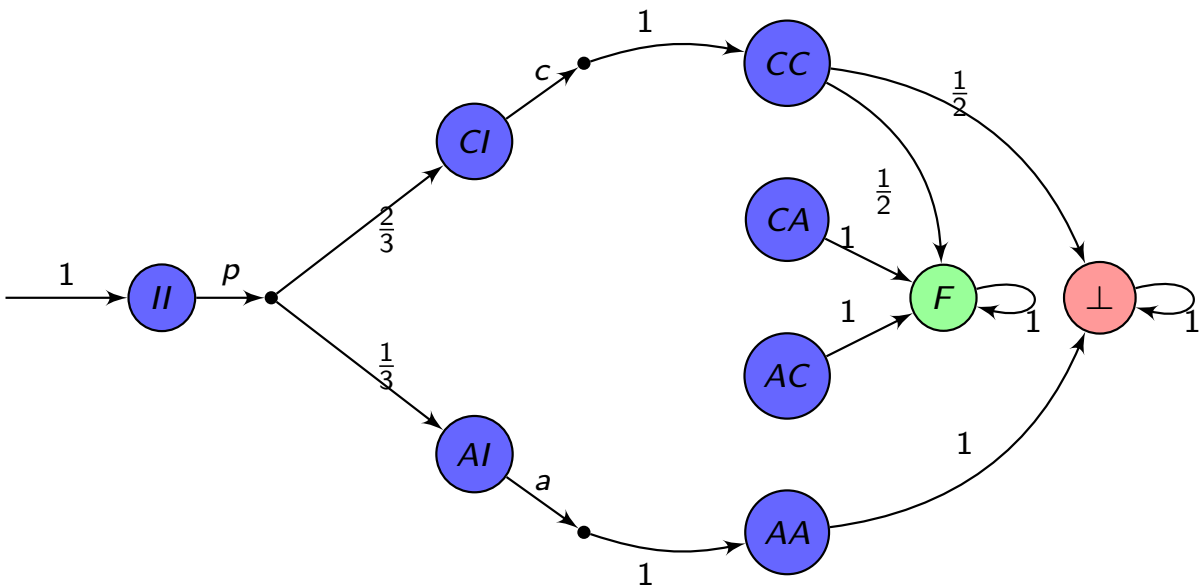


Exemple



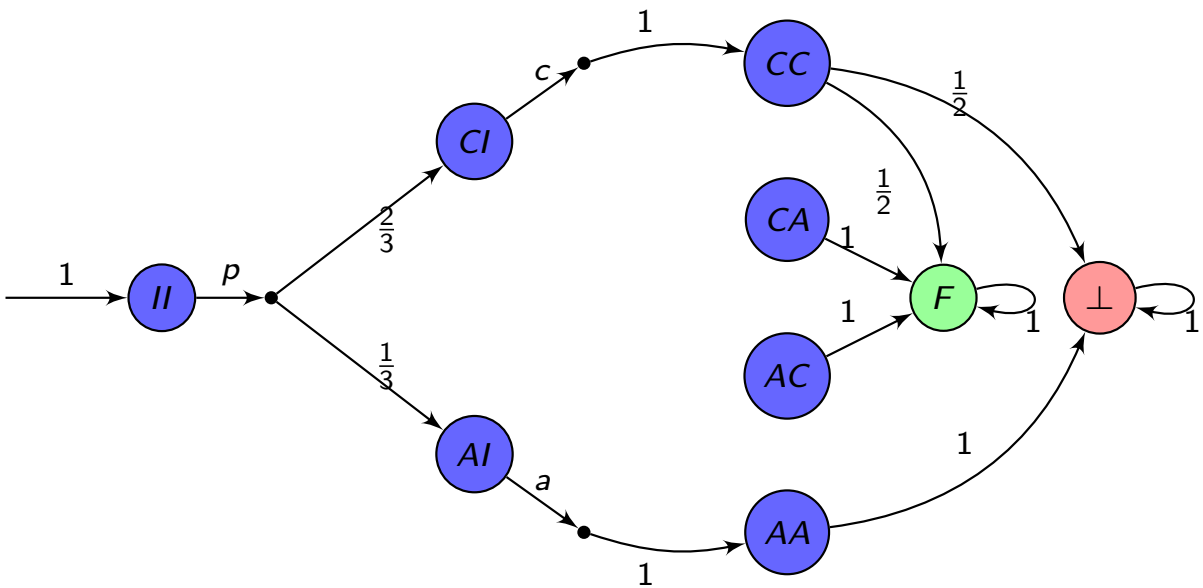
1. Choisir un adversaire $\mathfrak{S}$

Exemple



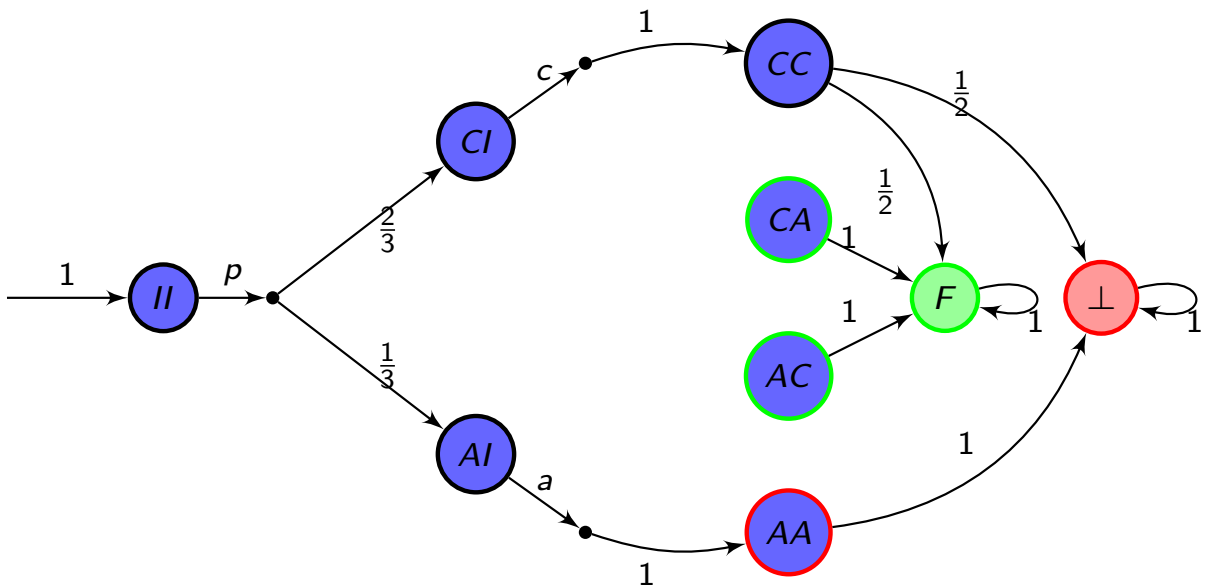
1. Choisir un adversaire $\mathfrak{S}$

Exemple



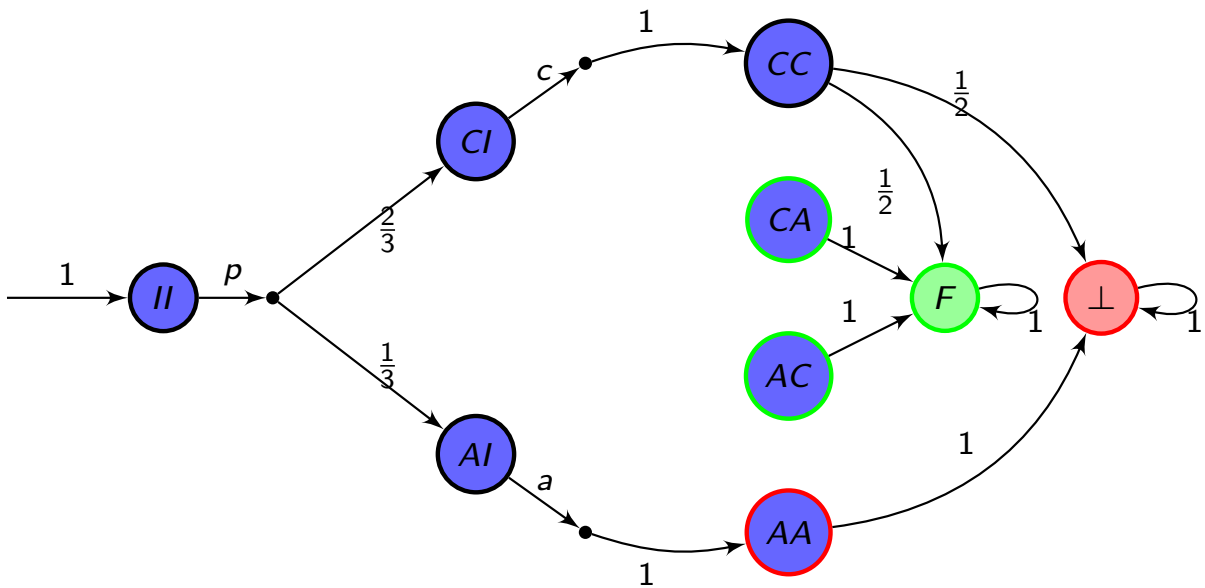
1. Choisir un adversaire $\mathfrak{S}$
2. Identifier S_0, S_1 et $S_?$

Exemple



1. Choisir un adversaire $\mathfrak{S}$
2. Identifier S_0, S_1 et $S_?$

Exemple



1. Choisir un adversaire $\mathfrak{S}$
2. Identifier S_0, S_1 et $S_?$
3. Résoudre le système linéaire

$$\mathbb{P}_{\mathfrak{S}}(II) = \frac{1}{3}, \mathbb{P}_{\mathfrak{S}}(CI) = \frac{1}{2}, \mathbb{P}_{\mathfrak{S}}(AI) = 0, \mathbb{P}_{\mathfrak{S}}(CC) = \frac{1}{2}$$

Outline

Continuous Stochastic Logic [Baier et al., 2003]

Definition (CSL (transitoire))

Formules d'états :

$$\Psi ::= \text{true} \mid a \mid \Psi_1 \wedge \Psi_2 \mid \neg\Psi \mid \mathbb{P}_J(\varphi),$$

avec $a \in AP$, φ formule de chemins et $J \subseteq [0,1]$ un intervalle à bornes rationnelles.

Formules de chemins :

$$\varphi ::= \bigcirc^I \Psi \mid \Psi_1 \, \mathsf{U}^I \, \Psi_2,$$

avec Ψ, Ψ_1 et Ψ_2 des formules d'états, et $I \subseteq \mathbb{R}_+$ un intervalle à bornes rationnelles.

Et aussi $\mathcal{S}_J(\Psi)$ (à l'équilibre)

Satisfaction des opérateurs $\bigcirc^I$ et U^I (CTMC)

Soit $\pi = s_0 t_0 s_1 t_1 \ldots$ un chemin

- $\pi \models \bigcirc^I \Psi$ ssi $s_1 \models \Psi$ et $t_0 \in I$
- $\pi \models \Psi_1 \, \mathrm{U}^I \, \Psi_2$ ssi il existe $t \in I$ et $k \geq 0$ tel que
  - $t_0 + \ldots + t_{k-1} \leq t < t_0 + \ldots + t_k$
  - $s_k \models \Psi_2$
  - $s_i \models \Psi_1$ pour $0 \leq i \leq k-1$ (ou k si $t_0 + \ldots + t_{k-1} < t$)

Vérification de CSL sur les CTMC

1. Transformation de la CTMC en une DTMC *uniformisée*
2. Utilisation des probabilités de Poisson
3. Calcul des probabilités transitoires à l'aide d'une somme infinie
4. Complexité
   - Linéaire par rapport à la formule
   - Polynomial dans le nombre d'états
   - Linéaire par raport à la plus grande constante temporelle de la formule
   - Linéaire par rapport à l'élément le plus grand de la matrice génératrice

Voir [Baier et al., 2003, Katoen et al., 2001, Baier et al., 2000] pour les détails.

Extension aux PTA [Kwiatkowska et al., 2002]

Ajout de non-déterminisme, horloges explicites, syntaxe différente.

CSL n'est pas la logique la plus appropriée (voire PTCTL), mais fonctionne.

Graphe des régions = MDP à états finis

1. Construction du graphe des régions
2. Transformation/Adaptation de la formule (si besoin)
3. Vérification de PCTL dans ce cadre (Adversaires min et max)

Complexité : Algorithmes largement exponentiels mais améliorations possibles

Mais aussi

- Autres logiques (PTCTL, CSLTA, ...)
- Autres modèles (Automates Stochastiques, réseaux d'Automates, ...)
- Vérification des propriétés à l'équilibre
- ...

Outline

Outils

- PRISM [Kwiatkowska et al., 2011]
- UPPAAL [Larsen et al., 1997]
- MRMC [Katoen et al., 2005]
- SPIN [Holzmann, 1997]
- ...

Outline

Motivation

Le model-checking probabiliste souffre des défauts du model-checking standard :

- Indécidabilité pour modèles complexes (Hybrides, paramétrés, etc.)
- Explosion de l'espace d'états
- Ressources nécéssaires (temps, mémoire, etc.)

$\Rightarrow$ Solution approchée : **Model checking statistique (SMC)**
[Younes and Simmons, 2002, Legay et al., 2010]

- Efficace (la complexité ne dépend pas de la taille du modèle)
- Précision contrôlée
- Vérification de modèles indécidables
- Non limité par les logiques
- Non limité par les modèles

SMC : But

But : Estimer la *probabilité* qu'un modèle M *satisfasse une propriété* φ à partir *d'échantillons*.

Echantillons

Traces / Exécutions du modèle

Probabilité

Requiert une mesure de probabilité sur l'espace des échantillons

Satisfaction d'une propriété

- Vérifiable sur un échantillon
  - Borné
  - Linéaire
  - Observable
- Pas nécessairement de logique

$+$ précision, erreur à maîtriser

Attention Uniquement si le modèle est purement probabiliste

Principes du SMC

1. Utiliser le modèle pour générer un échantillon de traces
   $\Rightarrow$ Estimateur de la mesure réelle

2. Mesurer la probabilité de satisfaire la propriété sur cet échantillon
   $\Rightarrow$ Estimation de la probabilité réelle

3. Généraliser l'estimation au modèle initial
   $\Rightarrow$ Précision, taux d'erreur en fonction de la taille de l'échantillon

Types de propriétés

2 types principaux : Propriétés *qualitatives* ou *quantitatives*

- **Quantitatif** : Réponse numérique
  Ex : Calculer $\mathbb{P}(M \models \varphi)$

- **Qualitatif** : Réponse Vrai/Faux
  Ex : $\mathbb{P}(M \models \varphi) > \theta$

Pas besoin de calculer $\mathbb{P}(M \models \varphi)$ pour le problème qualitatif, il existe des algorithmes plus efficaces.

Outline

SMC Quantitatif

Données : Modèle M, propriété φ, précision ε, taux d'erreur δ
But : Estimer la probabilité γ avec laquelle M satisfait φ

Monte Carlo [Robert, 2004]

Expérience : Générer une simulation de M et vérifier si elle satisfait φ

- Variable aléatoire Z, réalisation $z_i = 1$ ssi la i^e expérience est un succès

$\Rightarrow$ variable de Bernouilli de paramètre γ

$$\lim_{N \to \infty} \frac{\sum_{i=1}^{N} z_i}{N} = \gamma$$

Précision et taux d'erreur

Soit $\gamma_N = \frac{\sum_{i=1}^{N} Z_i}{N}$ (moyenne après N essais)

Bornes de Chernoff-Hoeffding [Hoeffding, 1963]

Si $\delta, \varepsilon \in [0, 1]$, alors

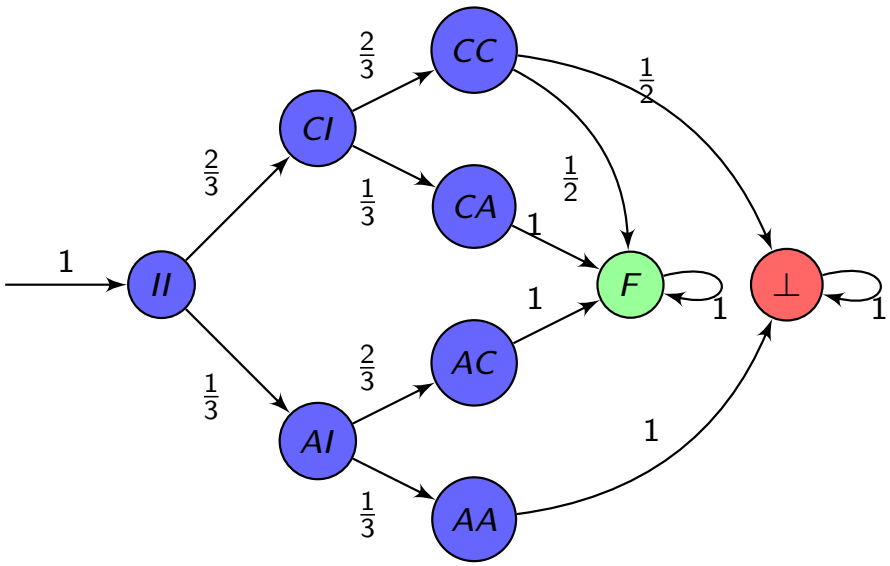
$$N \geq \frac{\ln(2) - \ln(\delta)}{(2\varepsilon)^2} \qquad \Rightarrow \qquad \mathbb{P}(|\gamma_N - \gamma| \geq \varepsilon) \leq \delta$$

Précision ε et taux d'erreur δ guarantis si $N \geq \frac{\ln(2)-\ln(\delta)}{(2\varepsilon)^2}$

Exemple

- $\varphi = \textit{bleu}\ \mathsf{U}^{\leq 5}\ \textit{vert}$
- Précision $\varepsilon = 0.01$
- Erreur $\delta = 0.01$

$\Rightarrow\ N \geq 13246$



Outline

Test d'Hypothèses [Younes, 2005a]

Soit γ la probabilité avec laquelle M satisfait φ) (inconnue)

Le test d'hypothèse a pour but de comparer γ avec une borne θ **sans calculer** γ

Test d'Hypothèse

But : confronter 2 hypothèses

- $H : \gamma \geq \theta$
- $K : \gamma < \theta$

Caractéristique : la **force**, déterminée par 2 paramètres α et β

- α : borne sur l'*erreur de type I* (K acceptée alors que H est vraie)
- β : borne sur l'*erreur de type II* (H acceptée alors que K est vraie)

Problème : Impossible d'optimiser *à la fois* α et β

Région d'indifférence

Afin de garantir des valeurs faibles pour α et β, on utilise une région d'indifférence $[\gamma_1, \gamma_0]$ qui contient θ.
On confronte alors les hypothèses

$$H_0 : \gamma \geq \gamma_0 \qquad \text{VS} \qquad H_1 : \gamma < \gamma_1$$

Si $\gamma \in [\gamma_1, \gamma_0]$ alors le résultat n'a pas d'importance.

En général, on fixe une **précision** ε et on prend $\gamma_1 = \theta - \varepsilon$ et $\gamma_0 = \theta + \varepsilon$.

Résolution du test d'hypothèses

De nombreuses méthodes existent

- Méthodes "statiques" : le nombre d'échantillons est pré-calculé en fonction des paramètres $\alpha, \beta, \varepsilon$
  Ex : Single Sampling Plan $\rightarrow (n, c)$ tel que H_0 est accepté si au moins c échantillons sur n satisfont φ
- Méthodes "dynamiques" : le nombre d'échantillons s'adapte. Critère d'arrêt fixé en fonction des paramètres
  Ex : Sequential Probability Ratio Test

Sequential Probability Ratio Test [Wald, 1945]

Une valeur R_m représentative du ratio de simulations "acceptées" d_m sur le nombre total de simulations m est mise à jour à chaque simulation et comparée à des bornes A et B.

$$R_m = \frac{\gamma_1^{d_m} \cdot (1-\gamma_1)^{m-d_m}}{\gamma_0^{d_m} \cdot (1-\gamma_0)^{m-d_m}},$$

- Dès que $R_m \geq A$, on peut accepter H_0
- Dès que $R_m \leq B$, on peut accepter H_1

Problème : Lier A et B aux paramètres α, β

Sequential Probability Ratio Test [Wald, 1945]

Une valeur R_m représentative du ratio de simulations "acceptées" d_m sur le nombre total de simulations m est mise à jour à chaque simulation et comparée à des bornes A et B.

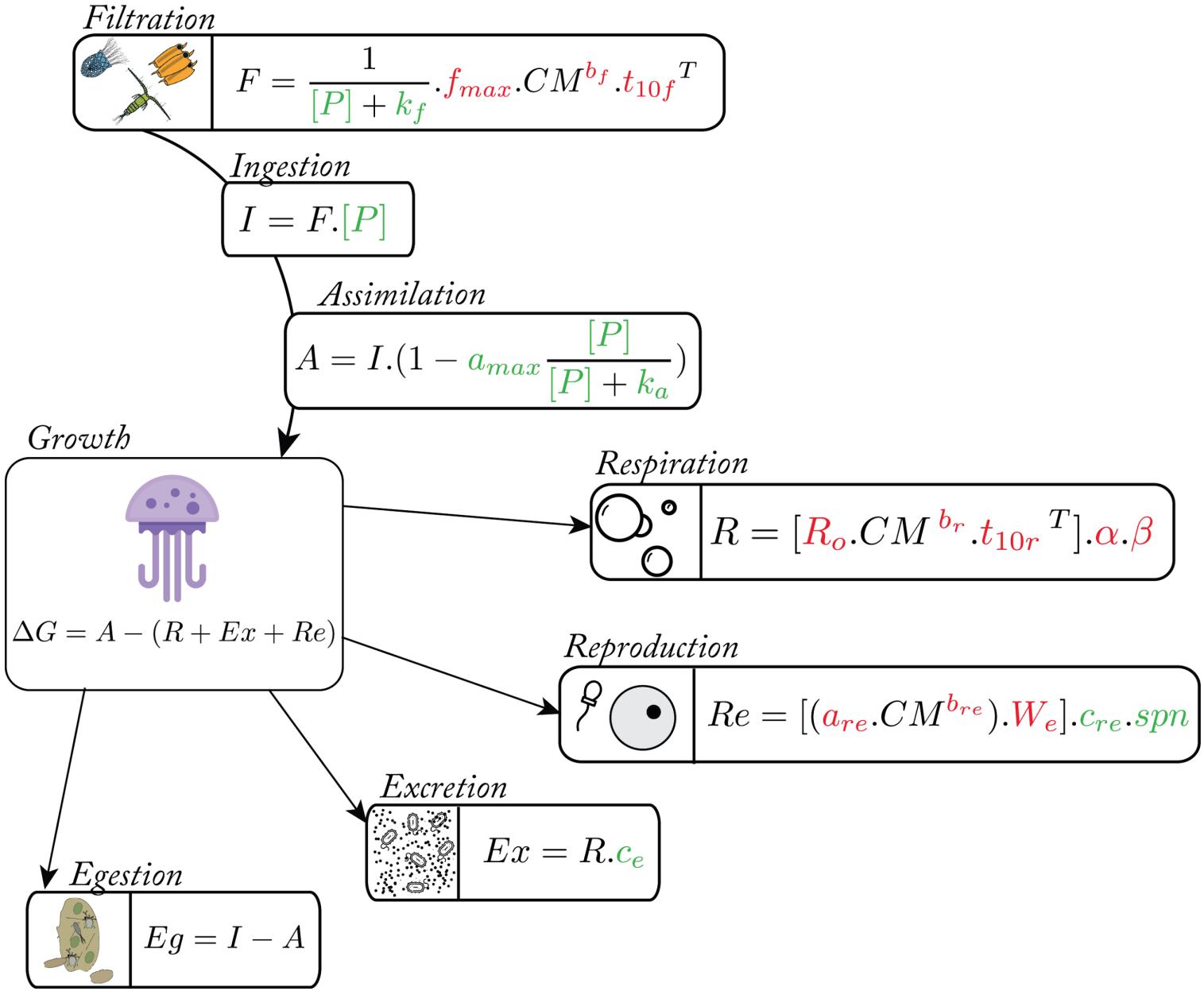
$$R_m = \frac{\gamma_1^{d_m} \cdot (1-\gamma_1)^{m-d_m}}{\gamma_0^{d_m} \cdot (1-\gamma_0)^{m-d_m}},$$

- Dès que $R_m \geq A$, on peut accepter H_0
- Dès que $R_m \leq B$, on peut accepter H_1

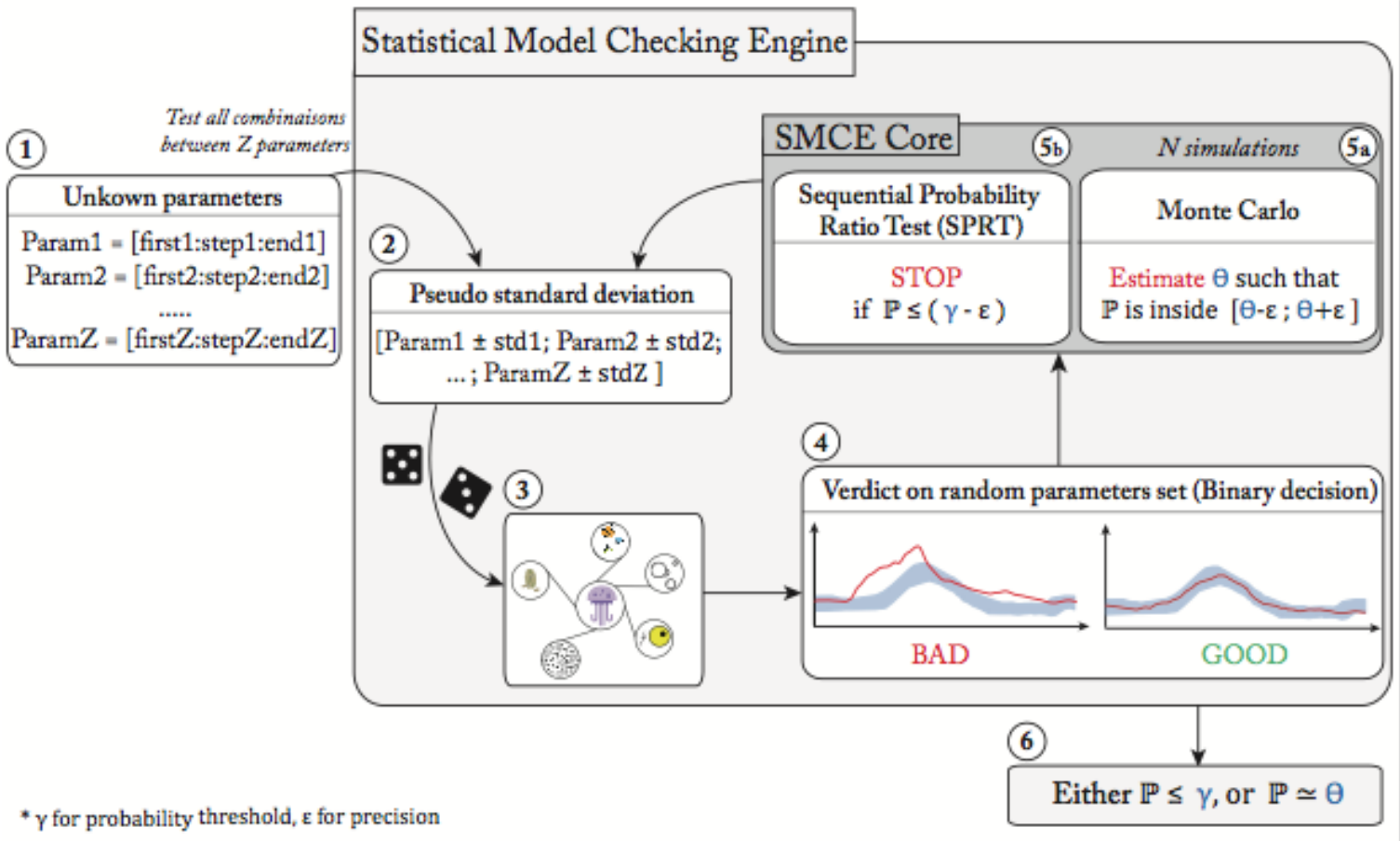
Problème : Lier A et B aux paramètres α, β

En pratique : $A = \frac{(1-\beta)}{\alpha}$ et $B = \frac{\beta}{(1-\alpha)}$ garantissent $\alpha' + \beta' < \alpha + \beta$

Exemple d'application 1/2



Exemple d'application 2/2



Outline

- Uppaal SMC (Networks of PTA) [Larsen et al., 1997]
- PRISM (DTMC, CTMC, PTA) [Kwiatkowska et al., 2011]
- Ymer (Generalized Semi-Markov Processes) [Younes, 2005b]
- COSMOS (Linear Hybrid Automata) [Ballarini et al., 2015]
- VESTA (DTMC, CTMC) [Sen et al., 2005]
- PLASMA-LAB (Générique) [Boyer et al., 2013]

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.

Baier, C., Haverkort, B. R., Hermanns, H., and Katoen, J.-P. (2000).
Model checking continuous-time markov chains by transient analysis.
In *CAV*, volume 1855, pages 358–372. Springer.

Baier, C. and Katoen, J.-P. (2008).
Principles of Model Checking.
The MIT Press.

References II


Ballarini, P., Barbot, B., Duflot, M., Haddad, S., and Pekergin, N. (2015).
HASL : A new approach for performance evaluation and model checking from concepts to experimentation.
Performance Evaluation, 90 :53–77.


Boyer, B., Corre, K., Legay, A., and Sedwards, S. (2013).
Plasma-lab : A flexible, distributable statistical model checking library.
In *International Conference on Quantitative Evaluation of Systems*, pages 160–164. Springer.


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.


Hansson, H. and Jonsson, B. (1994).
A logic for reasoning about time and reliability.
Formal aspects of computing, 6(5).


Hoeffding, W. (1963).
Probability inequalities for sums of bounded random variables.
Journal of the American statistical association, 58(301) :13–30.

References III


Holzmann, G. J. (1997).
The model checker spin.
IEEE Transactions on software engineering, 23(5) :279–295.


Katoen, J.-P., Khattri, M., and Zapreevt, I. (2005).
A markov reward model checker.
In *Quantitative Evaluation of Systems, 2005. Second International Conference on the*, pages 243–244. IEEE.


Katoen, J.-P., Kwiatkowska, M., Norman, G., and Parker, D. (2001).
Faster and symbolic CTMC model checking.
In *Process Algebra and Probabilistic Methods. Performance Modelling and Verification*. Springer.


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


Kwiatkowska, M., Norman, G., and Parker, D. (2011).
Prism 4.0 : Verification of probabilistic real-time systems.
In *Computer aided verification*, pages 585–591. Springer.


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.

References IV


Larsen, K. G., Pettersson, P., and Yi, W. (1997).
Uppaal in a nutshell.
International Journal on Software Tools for Technology Transfer (STTT), 1(1) :134–152.


Legay, A., Delahaye, B., and Bensalem, S. (2010).
Statistical model checking : An overview.
In *Proc. Runtime Verification - First International Conference, RV 2010, St. Julians, Malta, November 1-4, 2010. Proceedings*, volume 6418 of *Lecture Notes in Computer Science*, pages 122–135. Springer.


Robert, C. P. (2004).
Monte carlo methods.
Wiley Online Library.


Sen, K., Viswanathan, M., and Agha, G. (2005).
Vesta : A statistical model-checker and analyzer for probabilistic systems.
In *Quantitative Evaluation of Systems, 2005. Second International Conference on the*, pages 251–252. IEEE.


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

References V


Wald, A. (1945).
Sequential tests of statistical hypotheses.
The Annals of Mathematical Statistics, 16(2) :117–186.


Younes, H. L. (2005a).
Verification and planning for stochastic processes with asynchronous events.
Technical report, Ph.D. thesis, Carnegie Mellon.


Younes, H. L. (2005b).
Ymer : A statistical model checker.
In *CAV*, volume 3576, pages 429–433. Springer.


Younes, H. L. and Simmons, R. G. (2002).
Probabilistic verification of discrete event systems using acceptance sampling.
In *CAV*, volume 2, pages 223–235. Springer.