Vérification de systèmes probabilistes par réduction de l'espace des états

Simon Pinot

LSV, CNRS UMR 8643, ENS Cachan

Plan

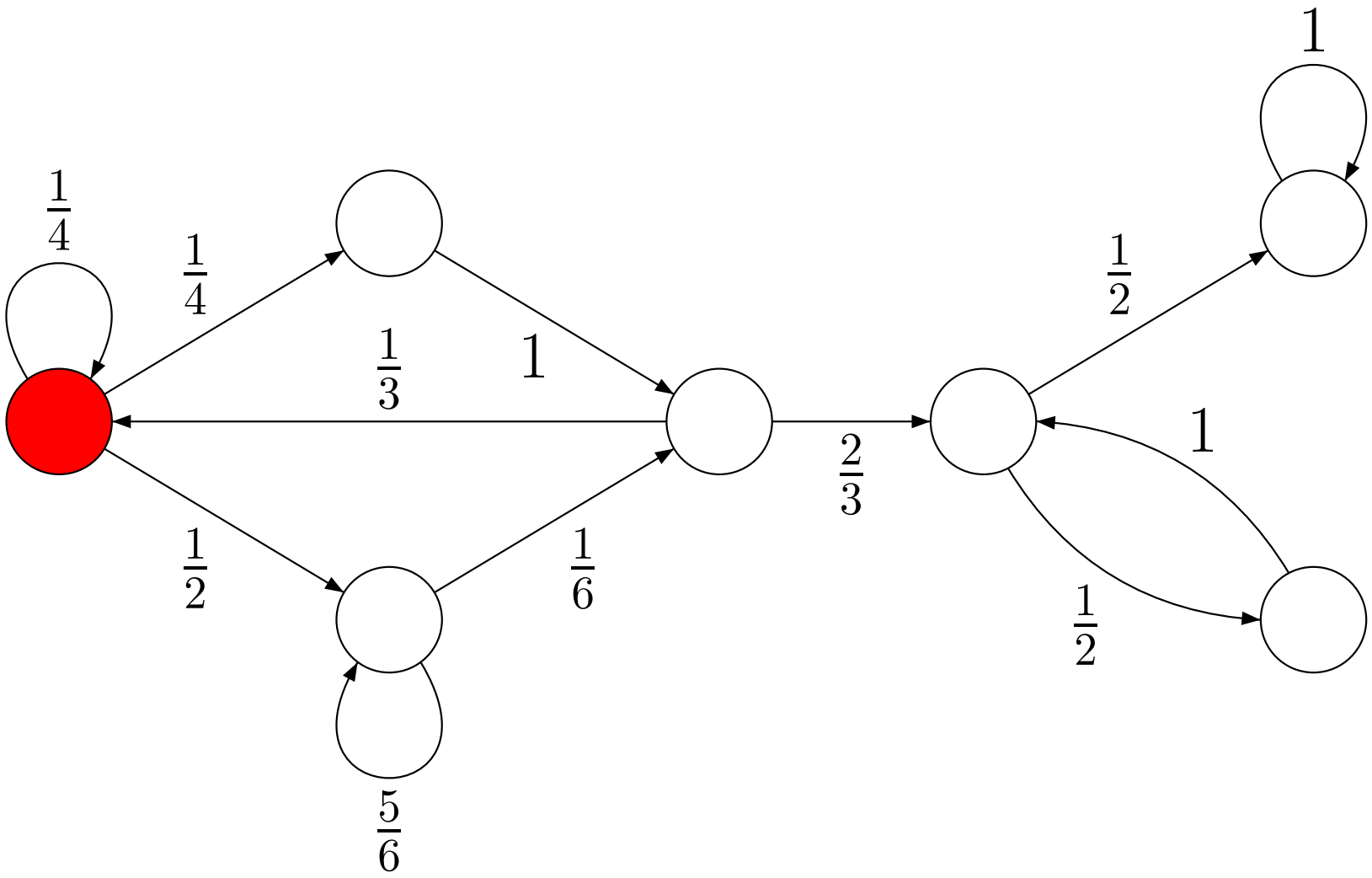
- Processus de décision Markovien
- PRISM & APMC
- CSMA/CA : manipulation de backoff
- Réduction de la taille du modèle
- Perspective

Probabilité

- Chaîne de Markov en temps discret :
  - Un ensemble d'états
  - Un état initial
  - Une matrice de transition
- $\{X_t, t \in \mathbb{N}\}$ est une chaîne de Markov si la valeur de X_t ne dépend que de X_{t-1}.
- L'ensemble des probabilités $Prob(X_t = x | X_{t-1} = y)$ est suffisant pour connaître le comportement du système.

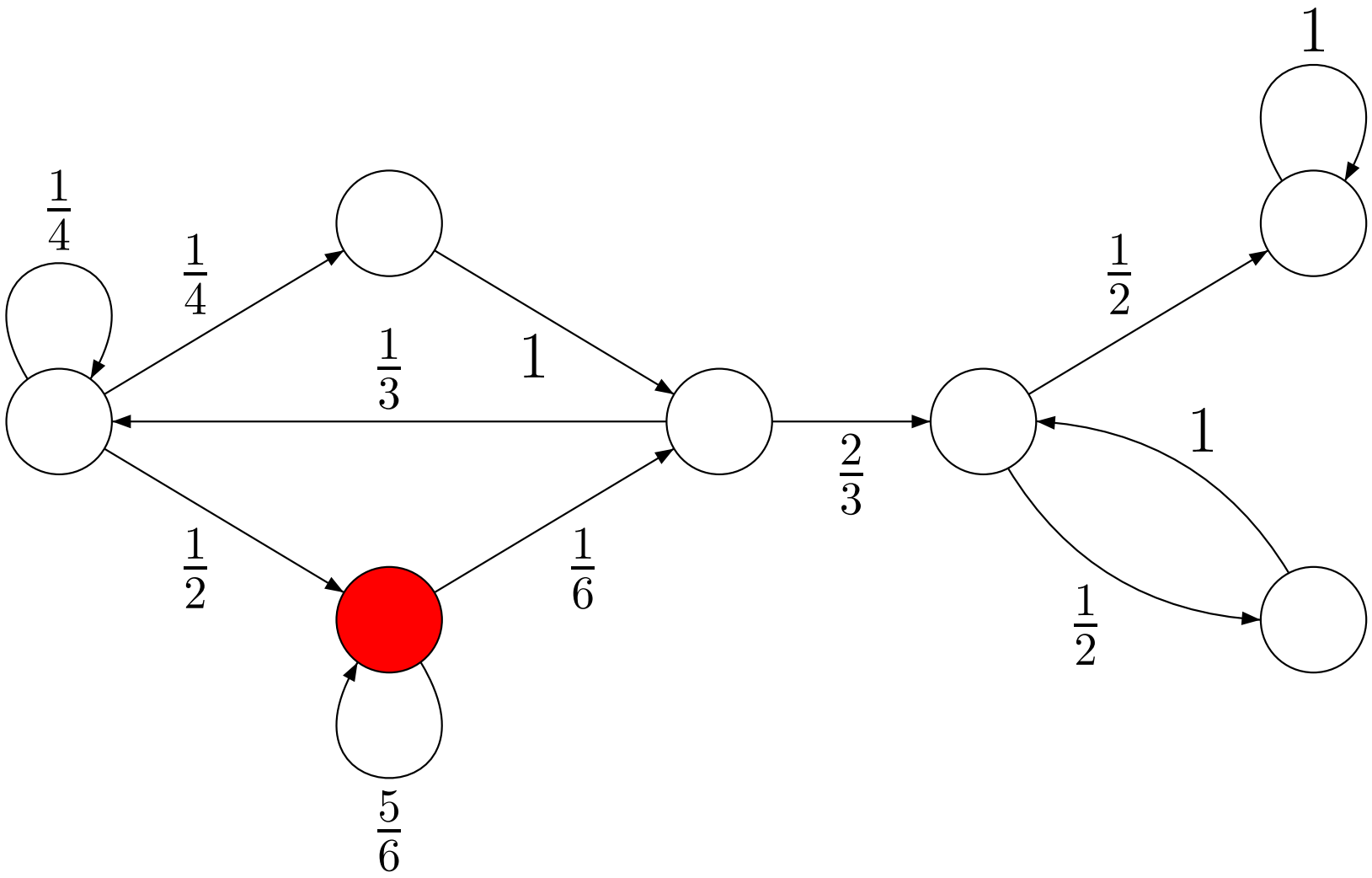
Probabilité

- Exemple



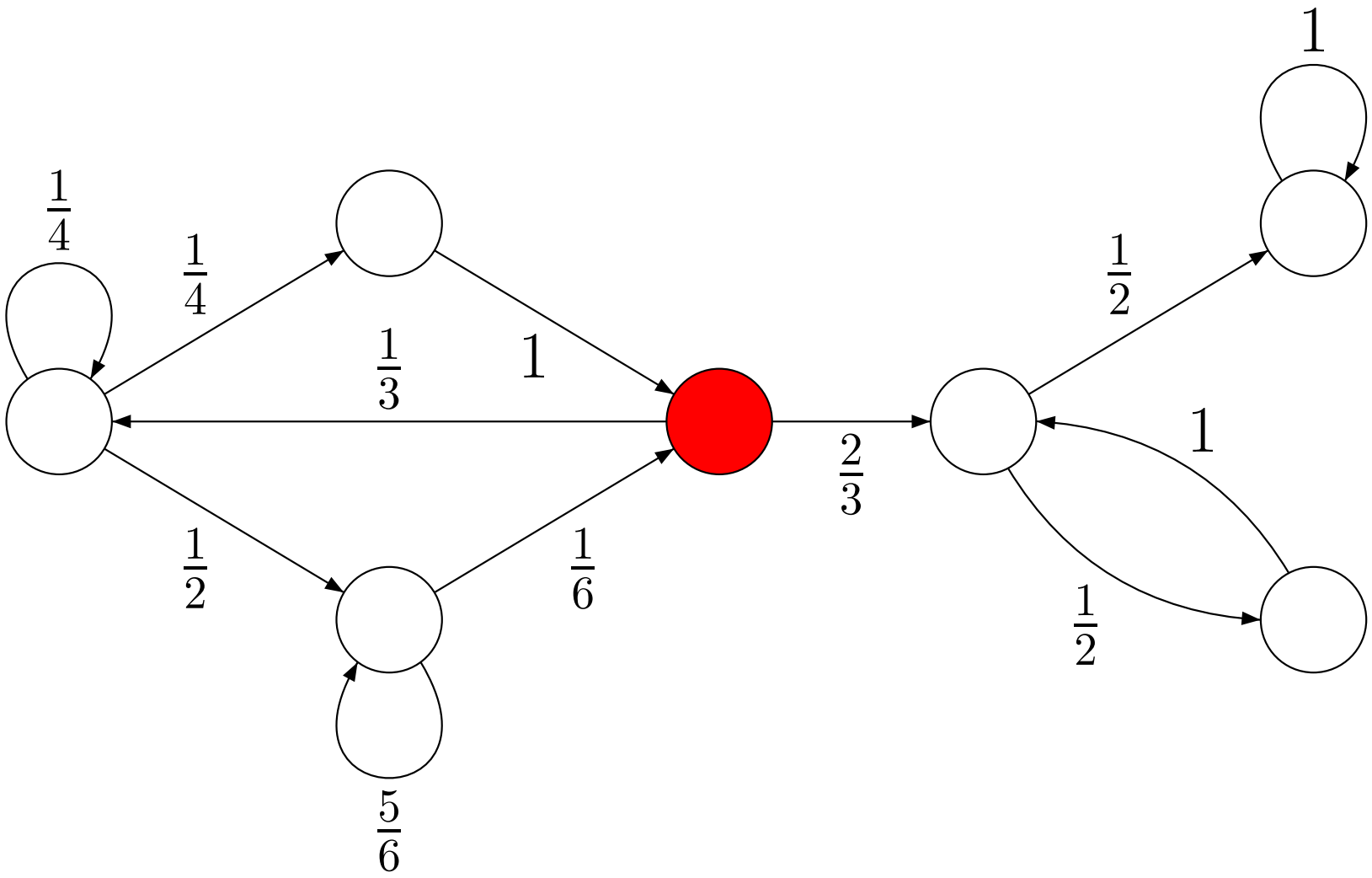
Probabilité

- Exemple



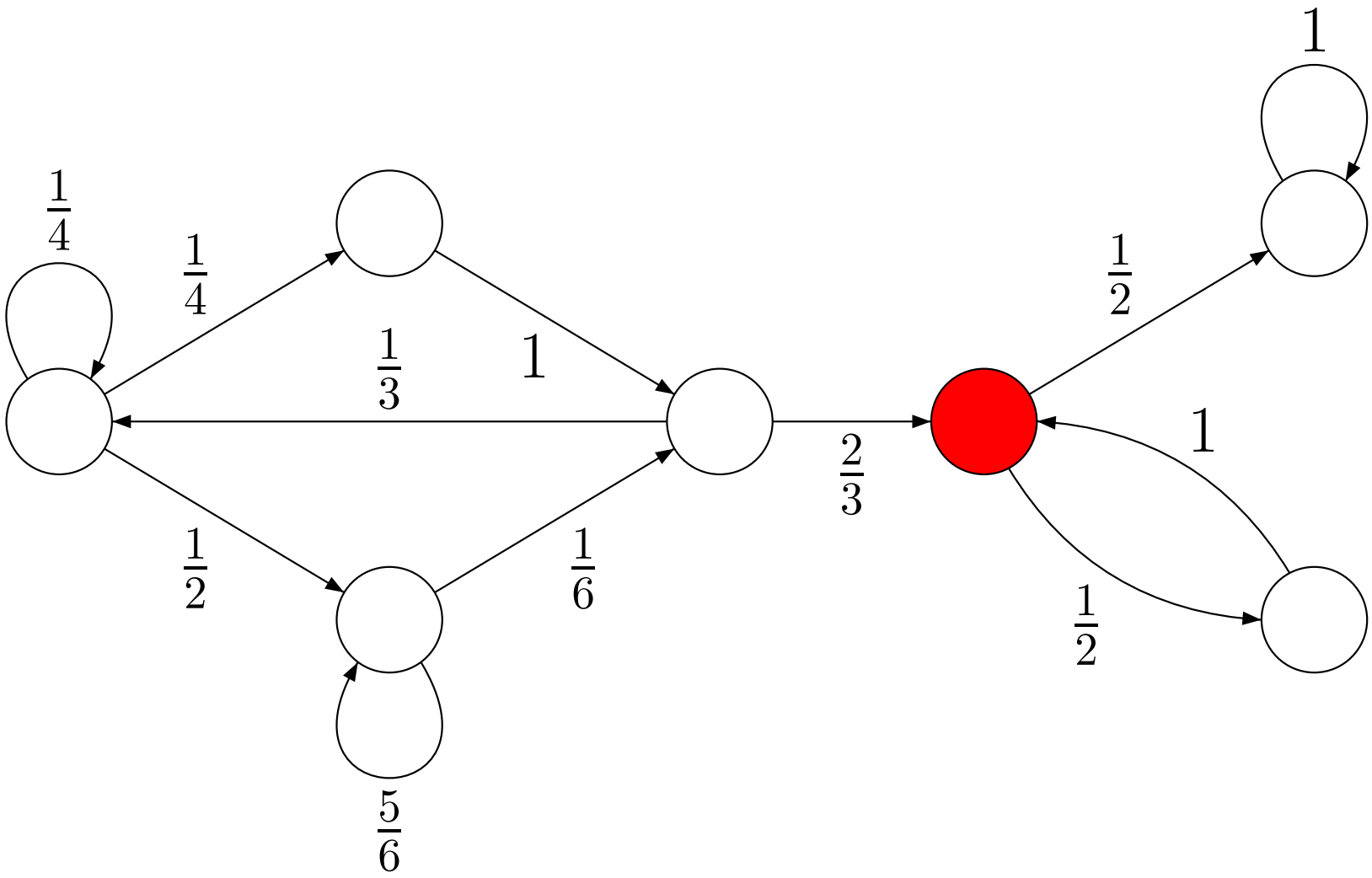
Probabilité

- Exemple



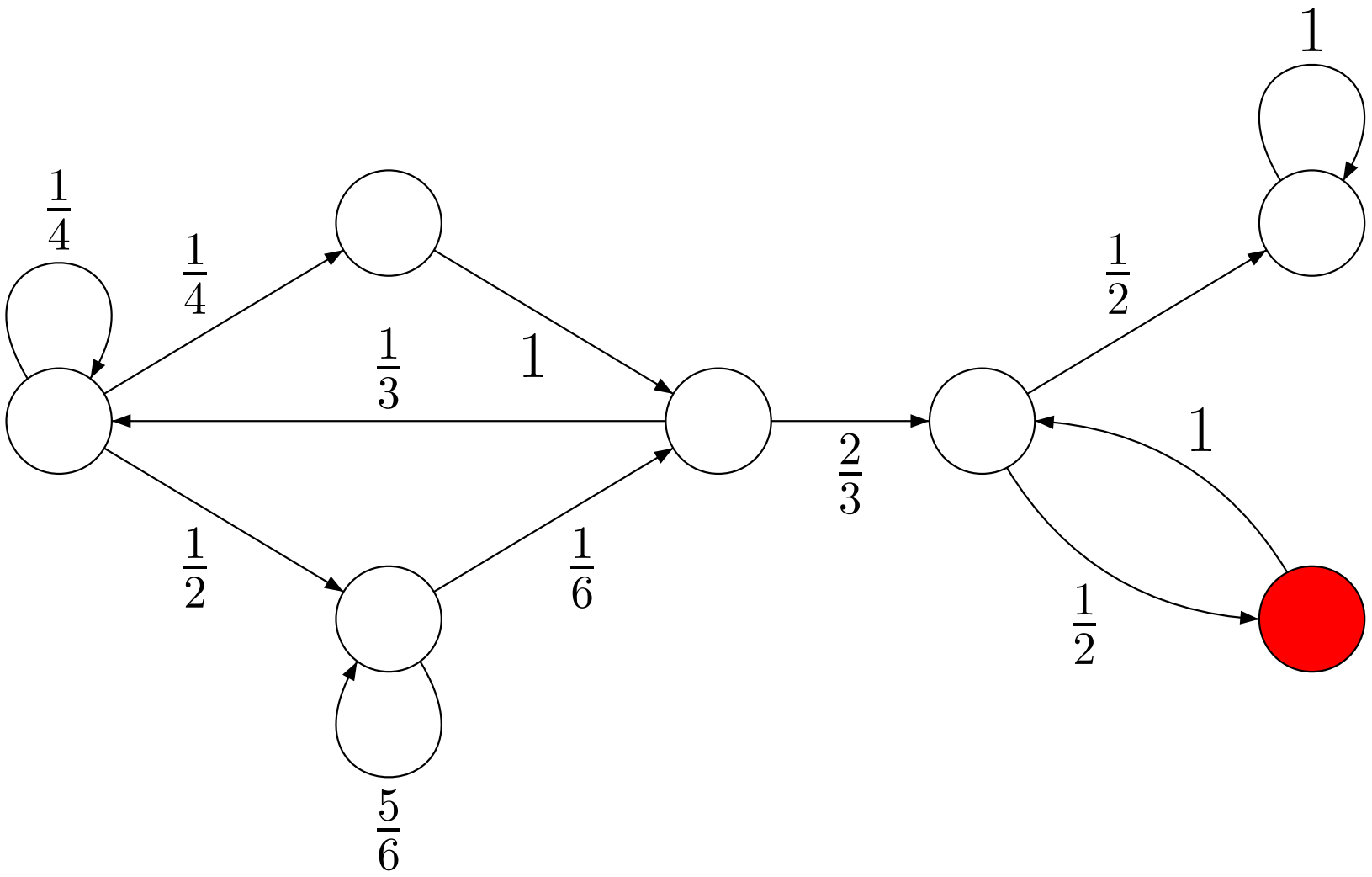
Probabilité

- Exemple



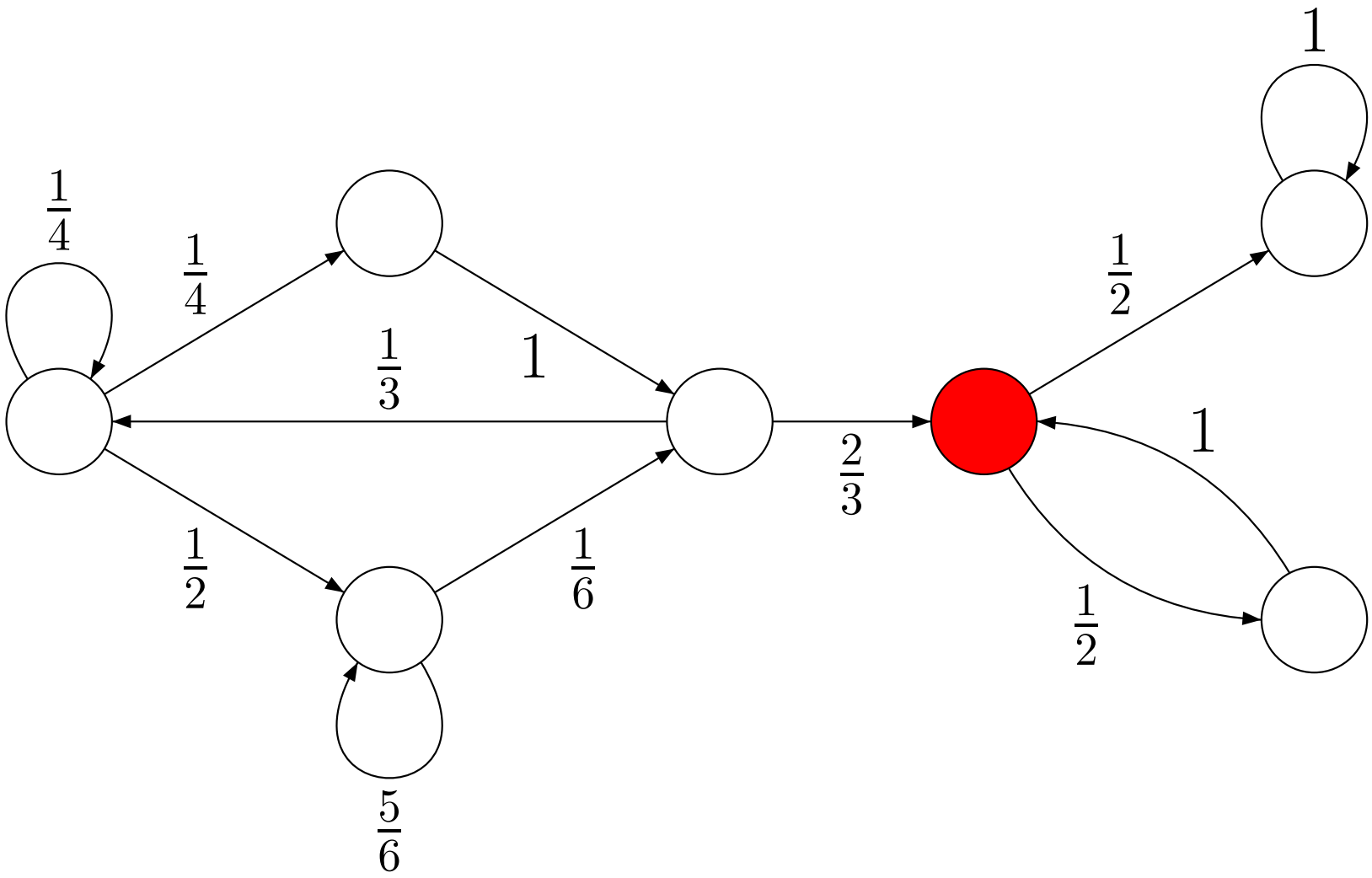
Probabilité

- Exemple



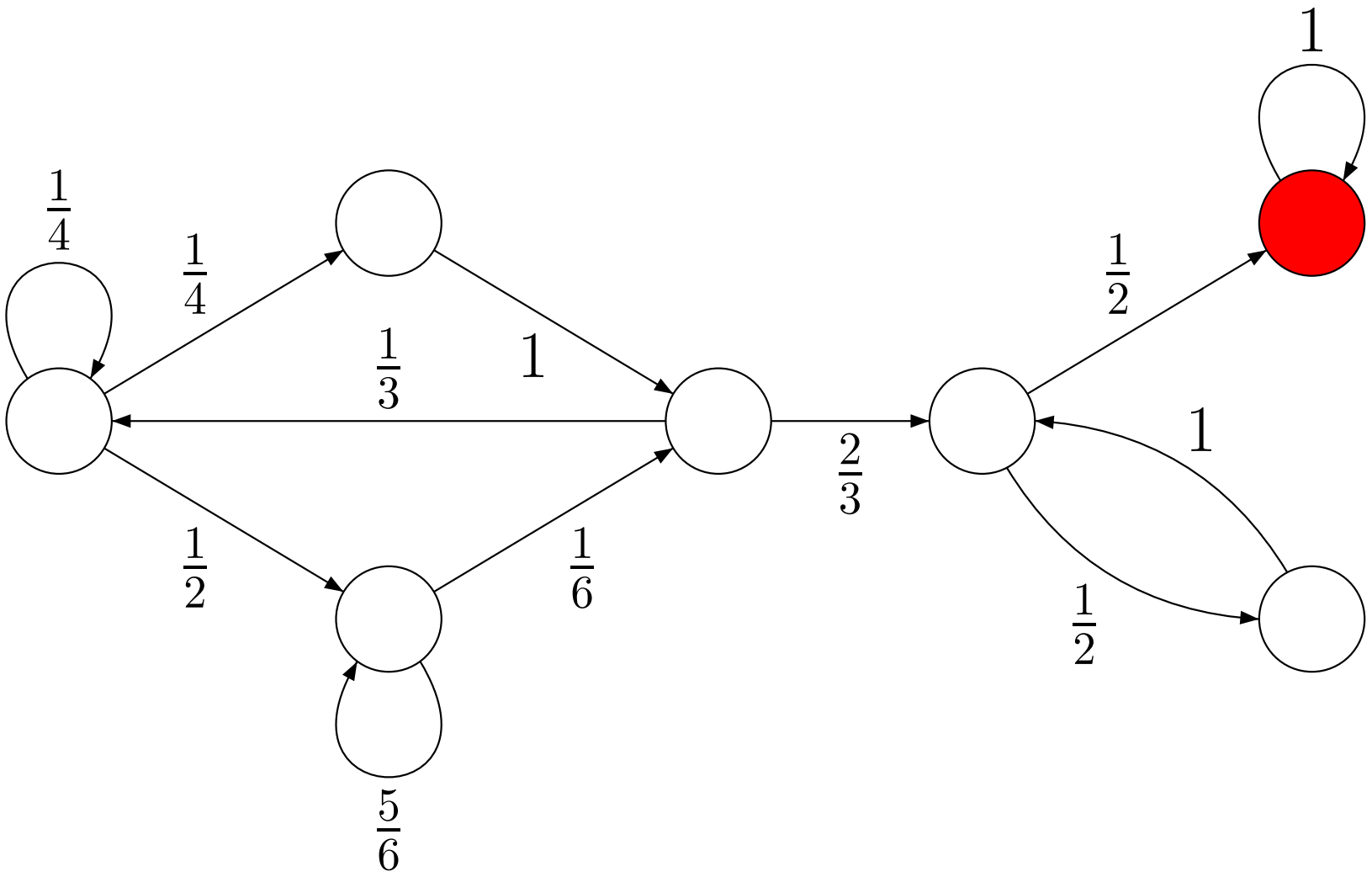
Probabilité

- Exemple



Probabilité

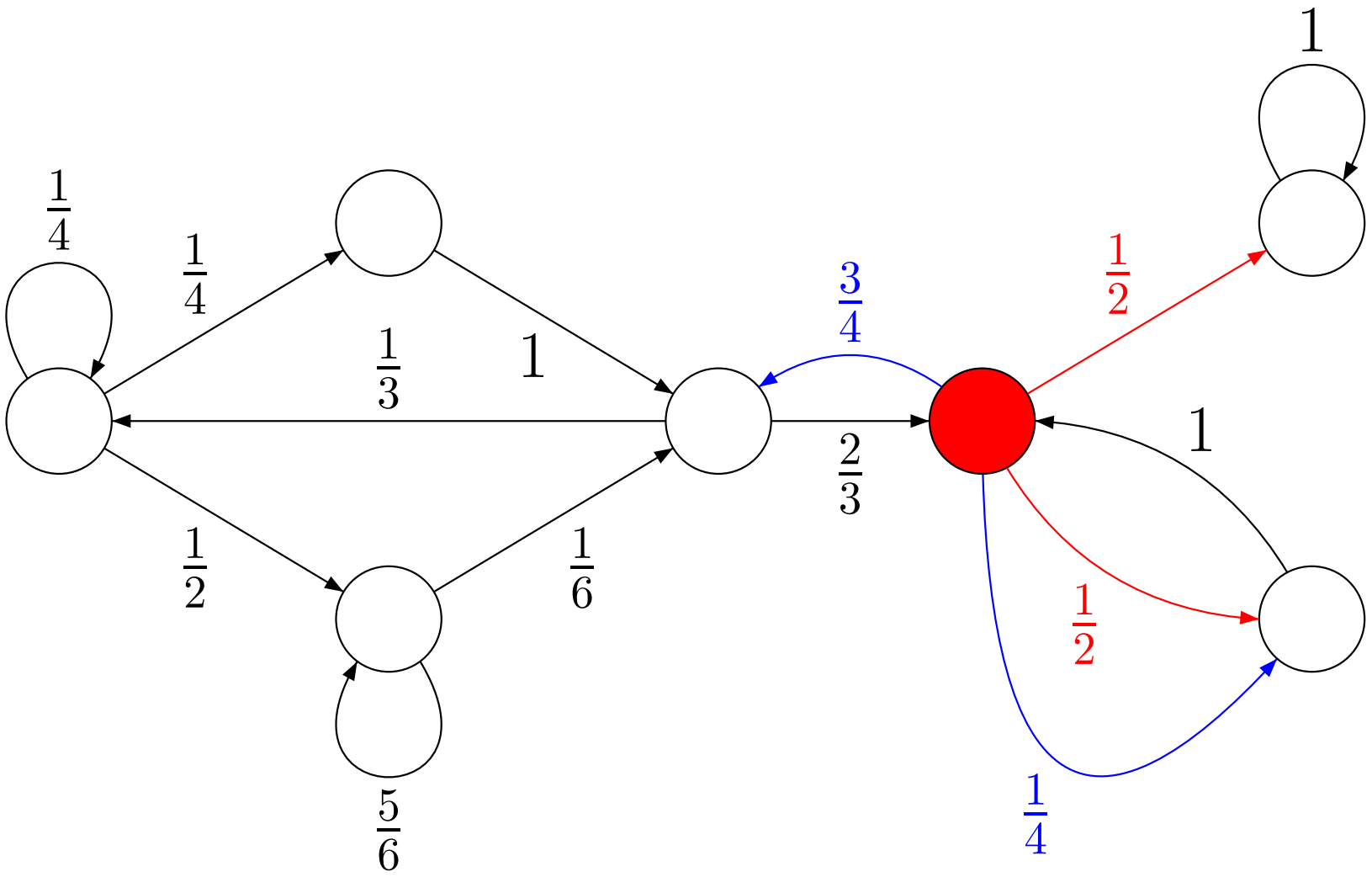
- Exemple



Probabilité

- Processus de décision Markovien :
  - Un ensemble d'états
  - Un état initial
  - Un ensemble d'actions pour chaque état
  - Une distribution de probabilité pour chaque action
- Dans chaque état, le système peut choisir une action et choisir son prochain état suivant la distribution associée.
- Politique déterministe : choix d'une action dans chaque état.

- Exemple : action a ou action b



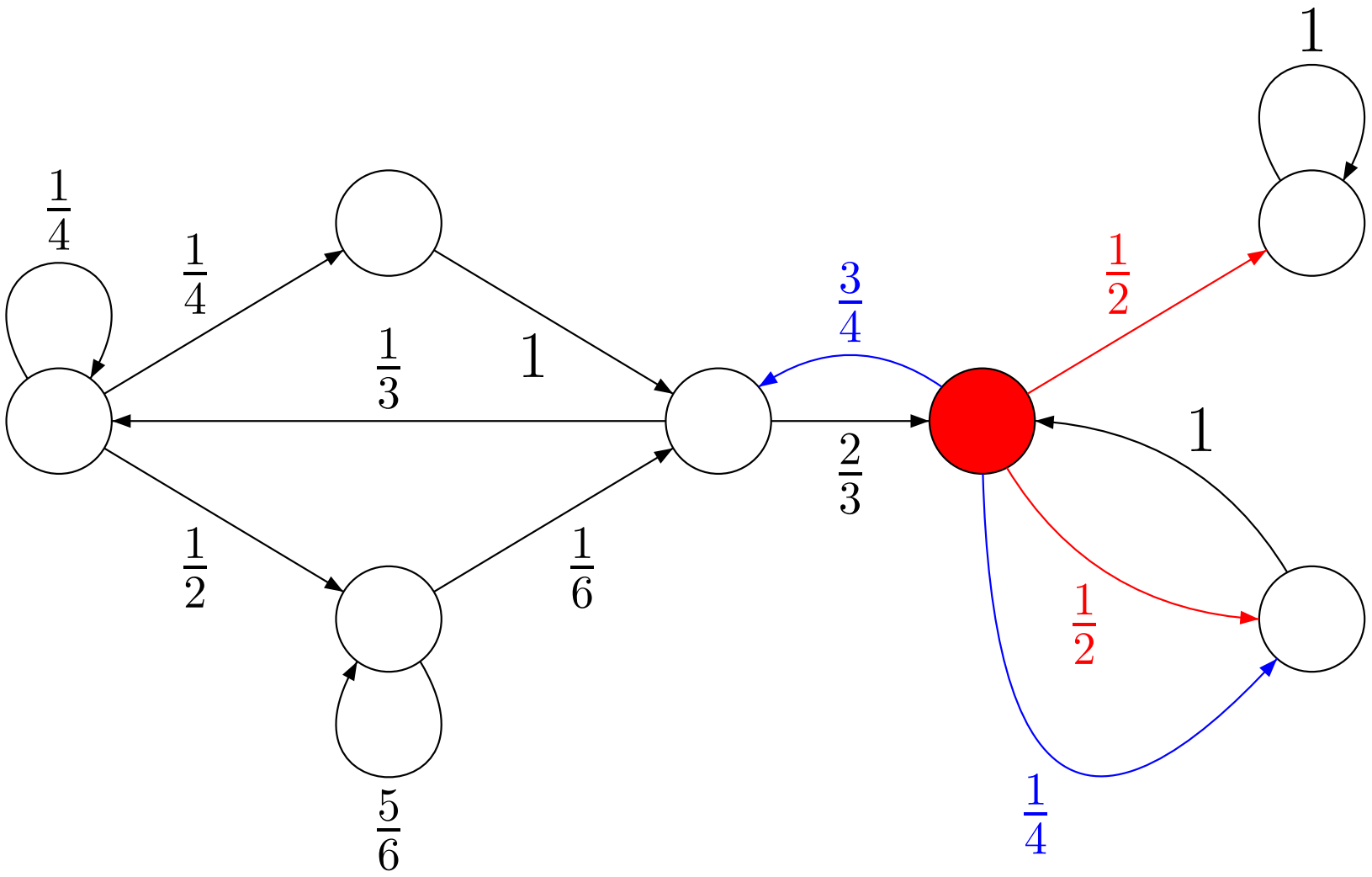
Probabilité

- On introduit une fonction de coût :

$$(etat, action) \rightarrow cout_{etat}^{action}$$

- Le coût d'une exécution $(s_0, a_0), \cdots, (s_n, a_n)$ est la somme $\sum_{k=0..n} cout_{s_k}^{a_k}$.
- Politique optimale : une politique pour laquelle le coût moyen est minimal (ou maximal).
- Objectifs : trouver une politique optimale et le coût optimal correspondant.

- Exemple : action a ou action b



Vérification de propriété d'accessibilité

- Formule de PCTL
  - Exemple : $P[\ true\ U\ goal] > 0.9$
  - Calcul de probabilités : $P[\ true\ U\ goal] = ?$
- Deux outils principaux :
  - PRISM
  - APMC

PRISM : méthode théorique

- PRISM : PRobabilistIc Symbolic Model checker
- Traite les :
  - ➤ Chaînes de Markov en temps discret et continu
  - ➤ Processus de décision Markovien
- On sait théoriquement calculer le coût et une politique optimale.
- PRISM construit l'ensemble des états accessibles à partir d'un code décrivant le système et implémente les algorithmes connus.

APMC : méthode statistique

- APMC : Approximate Probabilistic Model Checking.
- Même syntaxe que PRISM.
- Traite les chaînes de Markov en temps discret.
- APMC génère des chemins d'une longueur fixé et compte ceux qui vérifie la formule F pour retourner une approximation de $Prob(F)$.

PRISM & APMC

- PRISM :
  - prend en compte le non déterminisme,
  - mais la taille des modèles devient vite trop importante.
- APMC :
  - peut effectuer des calculs sur des modèles bien plus gros que PRISM,
  - mais le non déterminisme n'est plus pris en compte.
- PRISM est utilisé tant que les calculs sont possibles.
- Si une politique est choisie, APMC peut calculer des probabilités sur les modèles plus gros.

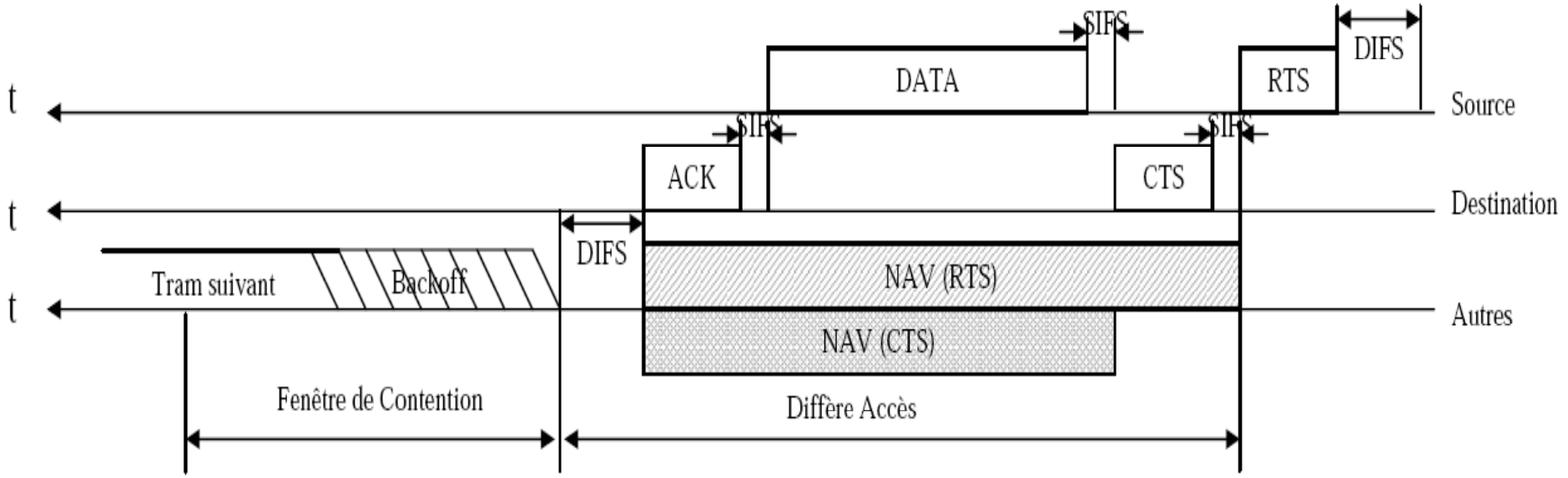
Exemple de système probabiliste

Manipulation de backoff dans CSMA/CA

CSMA/CA

- La technologie sans fil permet à plusieurs utilisateurs d'accéder à une borne fournissant un service.
- Des utilisateurs peuvent tenter d'accéder à la borne en même temps $\rightarrow$ collision.
- Un protocole tente de minimiser le temps perdu par ces collisions tout en prenant en compte les spécificités des réseaux sans fil : CSMA/CA (Carrier Sense Mutiple Access / Collision Avoidance).

Description de CSMA/CA

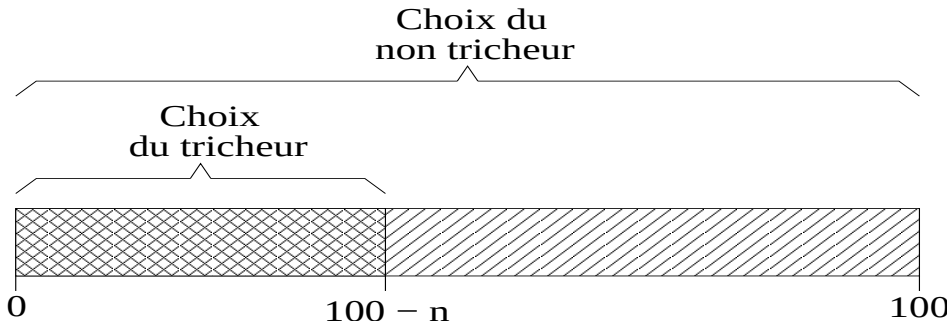


Problème

- CSMA/CA contraint les utilisateurs pour rendre l'accès au médium équitable.
- Les cartes réseaux sont devenues suffisamment modifiables pour qu'un utilisateur puisse modifier le protocole CSMA/CA.
- Possibilité de "tricher" en passant outre les contraintes.
- Objectif :
  - ➤ Détecter les tricheurs.
  - ➤ Ne pas détecter des non tricheurs comme des tricheurs.
  - ➤ Pénaliser les tricheurs en proportion.

Triche

- Ces "tricheurs" respectent les grandes lignes du protocole car sinon ils seraient facilement détectés.
- La triche à $n\%$ consiste à diminuer l'intervalle dans lequel on choisit le backoff.



- Problème : on ne peut trouver un test qui détecte sûrement et uniquement les tricheurs car un innocent a une probabilité non nulle de choisir exactement les même backoff qu'un tricheur.

DOMINO

- DOMINO utilise plusieurs tests pour détecter des tricheurs dans CSMA/CA :
  - Des tests sur le respect du protocole.
  - Deux tests sur le respect du backoff :
    - Maximal Backoff : vérifie que sur une grande période, le backoff peut s'approcher du backoff maximal.
    - Actual Backoff : vérifie que la moyenne du backoff est proche de celle attendue.

DOMINO

- Pour tous les tests, il existe un compteur de triche.

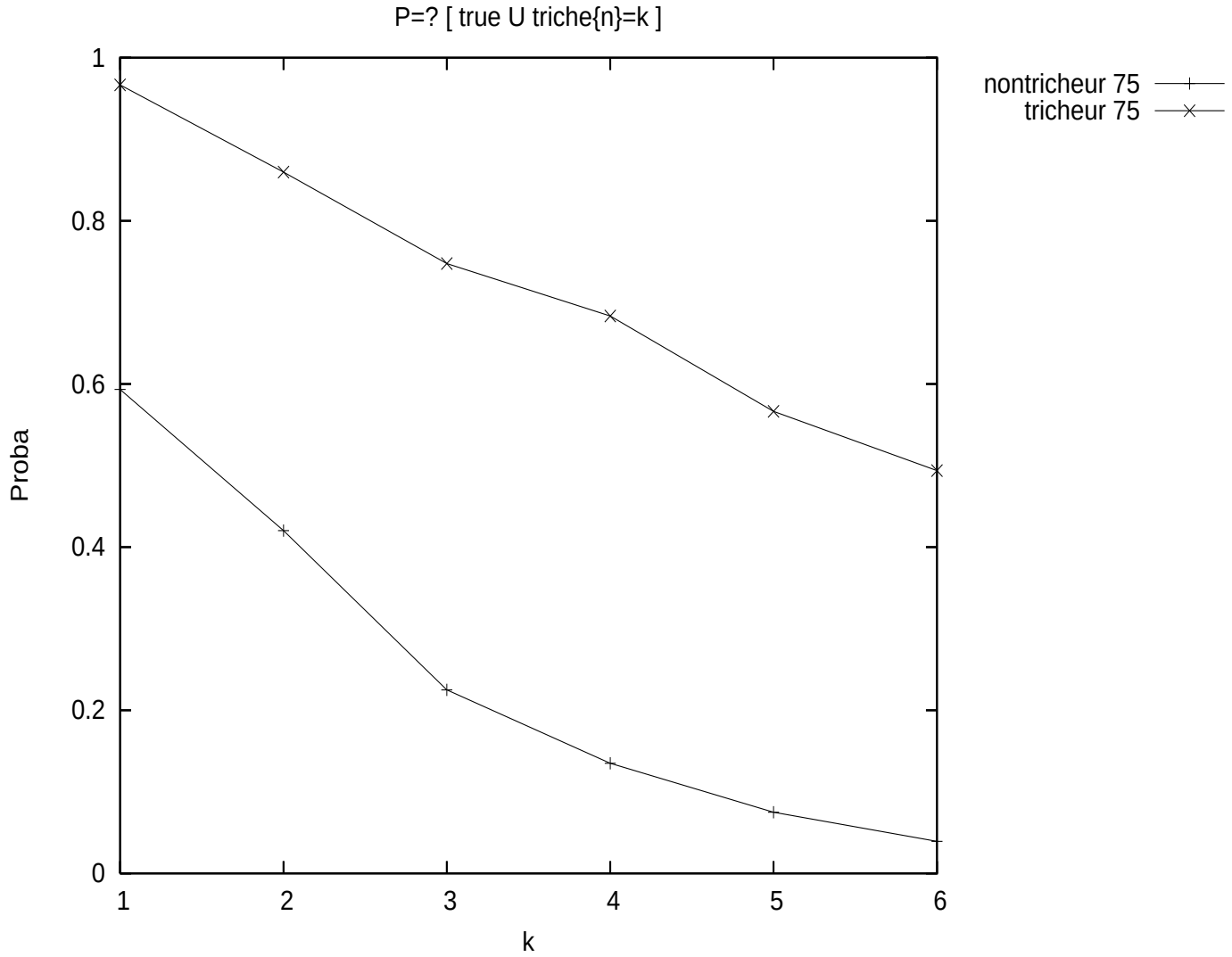
```
if test = true then
    triche := triche + 1
    if triche > K then
        tricheur detecte
        penalisation
else if triche > 0 then
    triche := triche − 1
```

Résultats

- Vérification du gain du tricheur.
- Vérification de l'efficacité de DOMINO.
- Vérification de l'efficacité de la pénalisation.

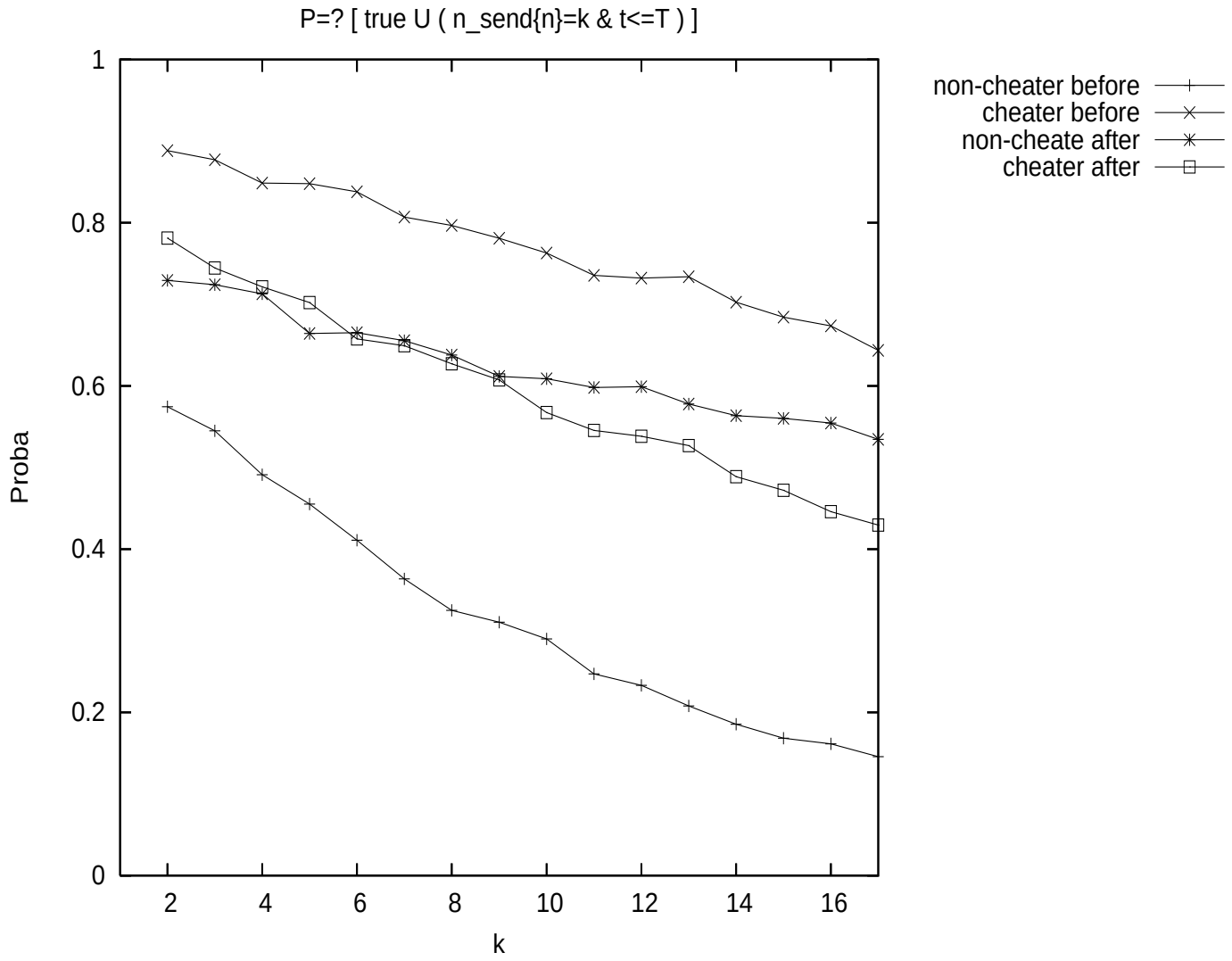
Résultats

$$P =? [\ true\ U\ (triche = K\ \&\ t <= T)\ ]$$



Résultats

$$P =? [\ true\ U\ (n\_send = K\ \&\ t <= T)\ ]$$



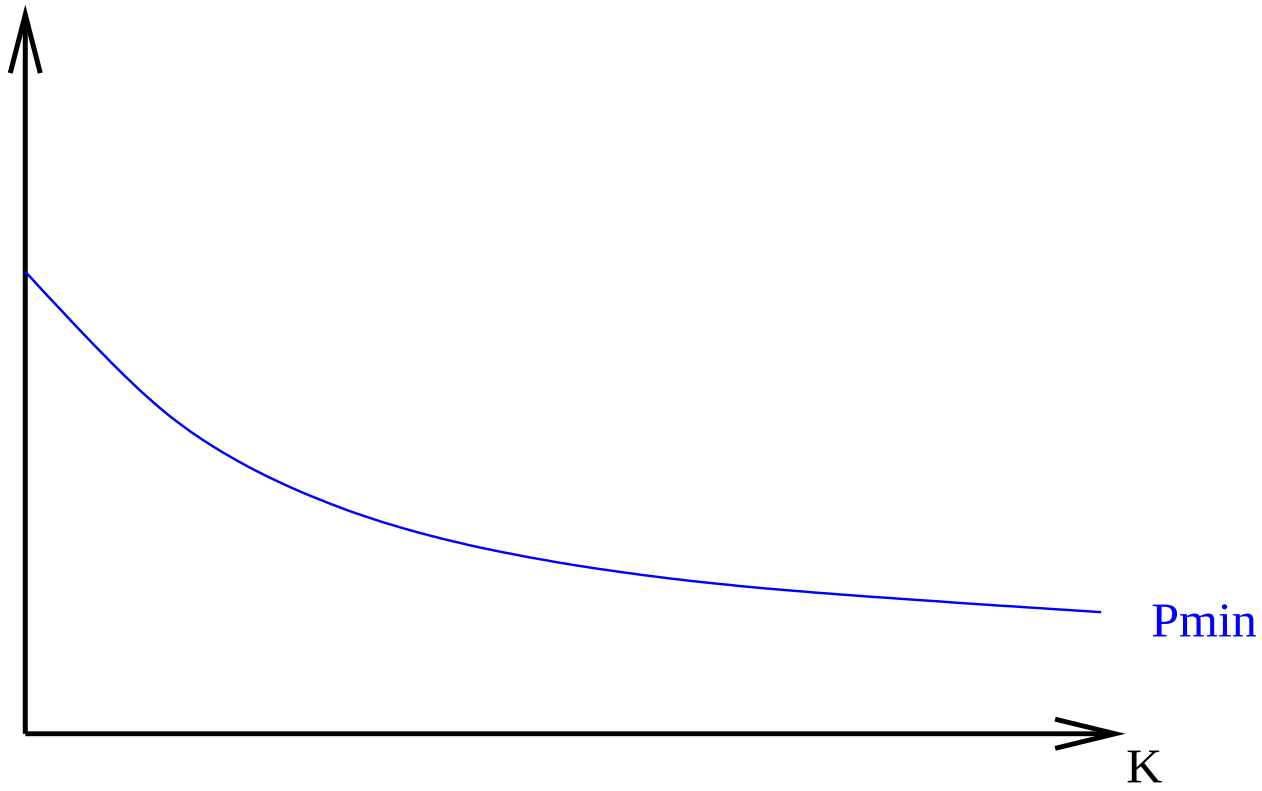
Problèmes

- Avec PRISM :
  - Modélisation du tricheur.
  - DOMINO a besoin de plusieurs envois pour détecter les tricheurs : taille du modèle trop importante.
- Courbes obtenues par APMC.

Exemple

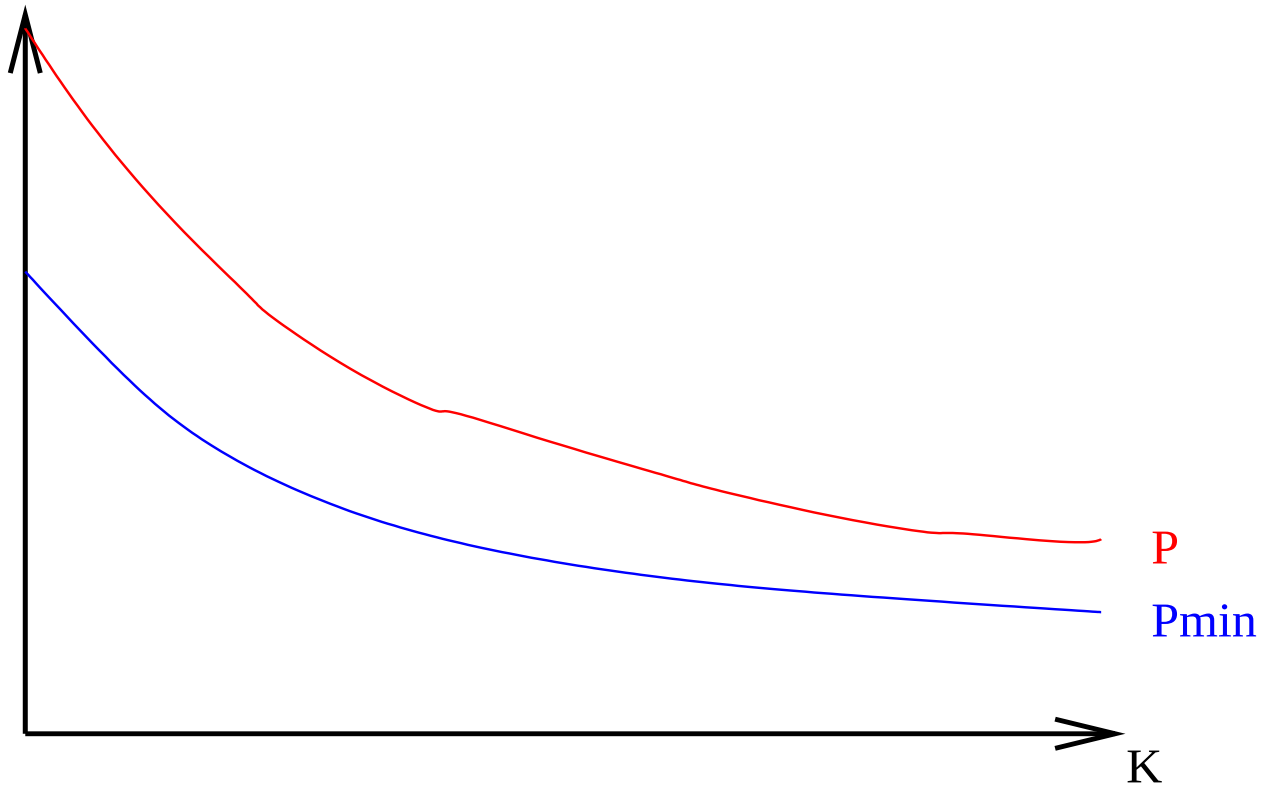
- Le comportement du tricheur est non déterministe (choix du pourcentage de triche, choix d'envoyer...).
- Le tricheur a une politique qui lui permet d'être détecté le moins possible $\rightarrow$ politique optimale.
- On voudrait approcher cette politique.

Problèmes



- La courbe de la probabilité minimale est calculée en prenant en compte le non déterminisme.
- Impossible à obtenir par PRISM.

Problèmes



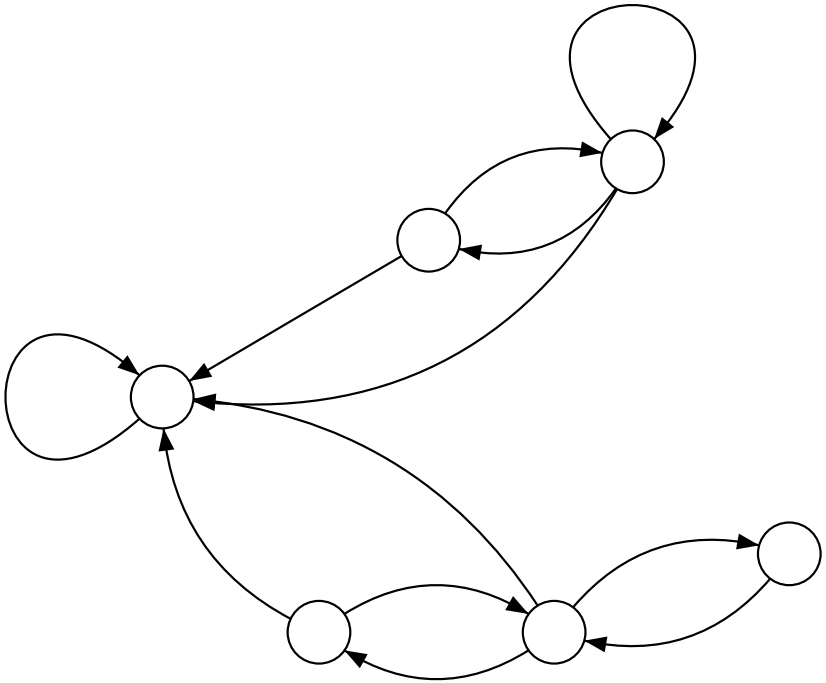
- Choix d'une politique : on obtient facilement avec APMC une courbe au dessus de Pmin.
- Pas d'information précise sur Pmin.

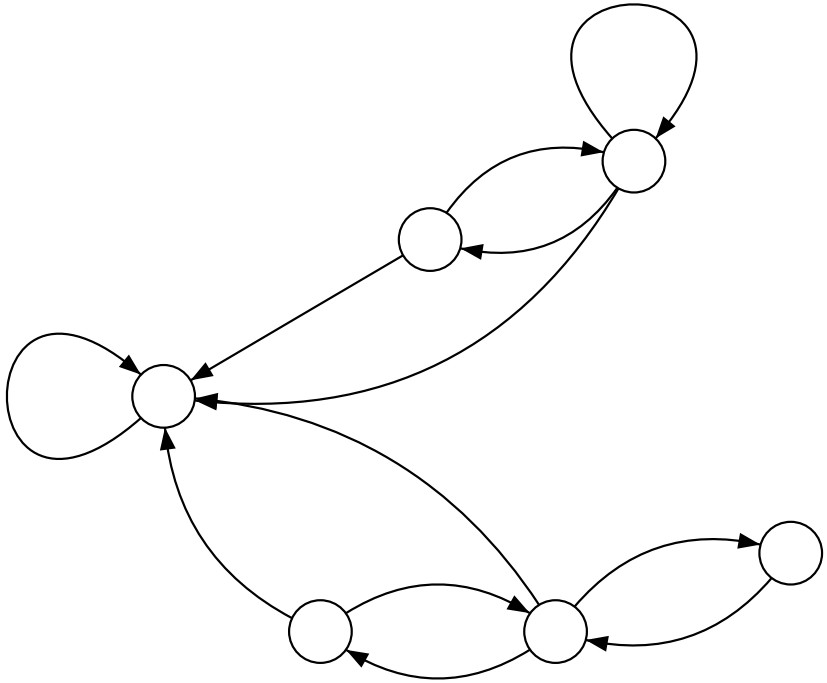
Objectif : réduction de l'espace des états

- On cherche à réduire l'espace des états.
- On regroupe les états de comportement "similaire".
- La taille du système étant réduite, l'application des méthodes théoriques est possible $\rightarrow$ politique optimale du système réduit.
- La politique trouvée induit une politique sur le système initial.

Réduction de l'espace des états

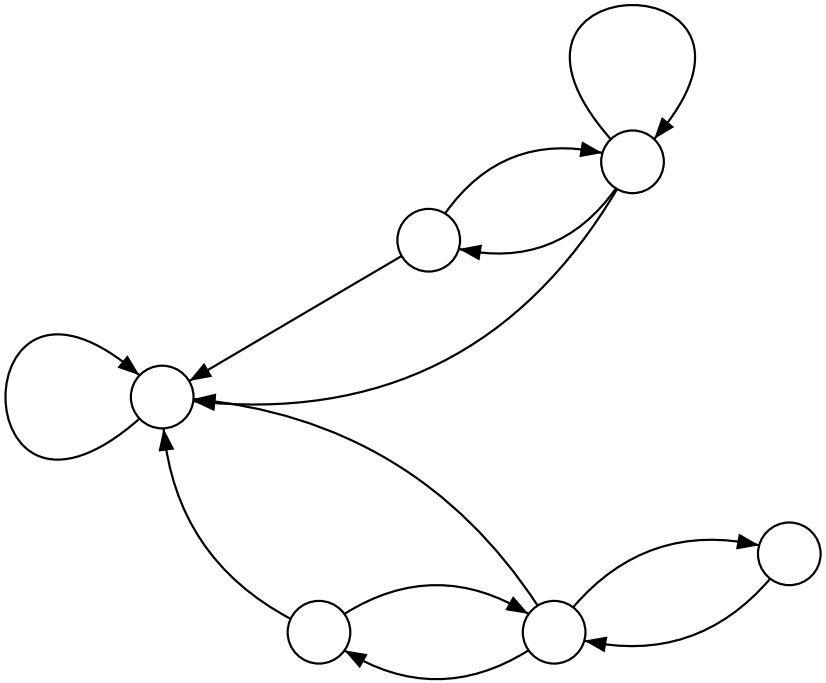
- Exemple

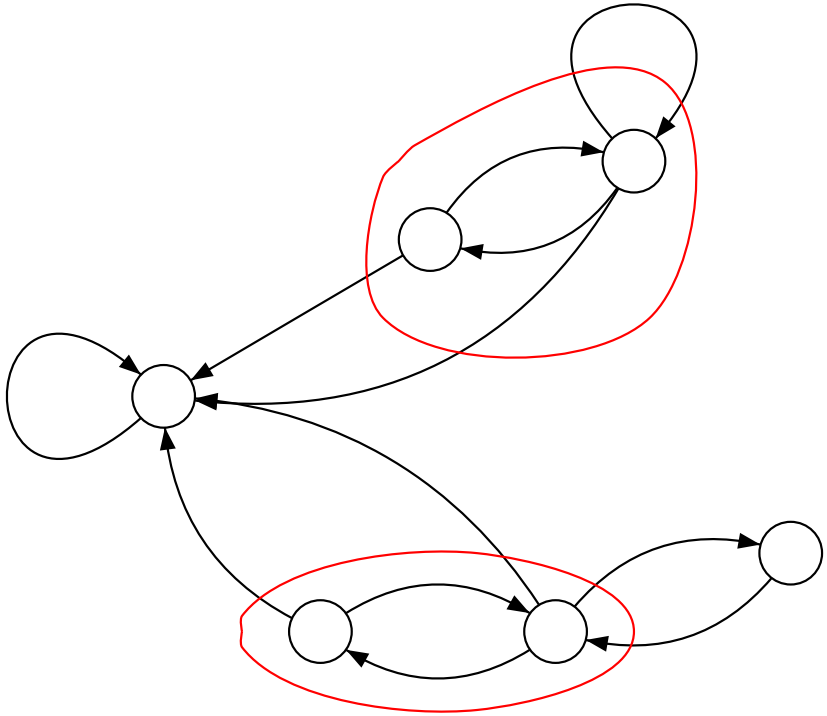




Réduction de l'espace des états

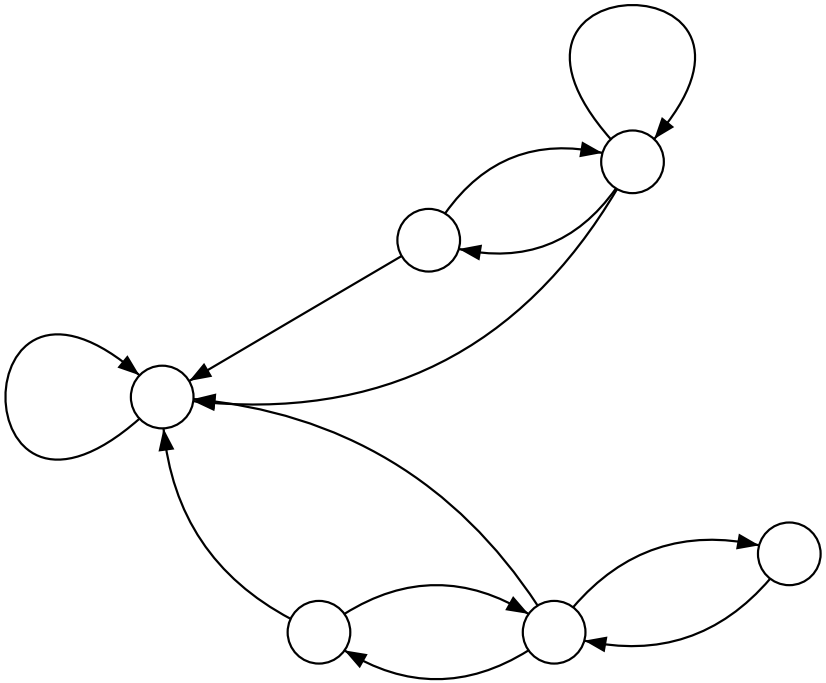
- Exemple

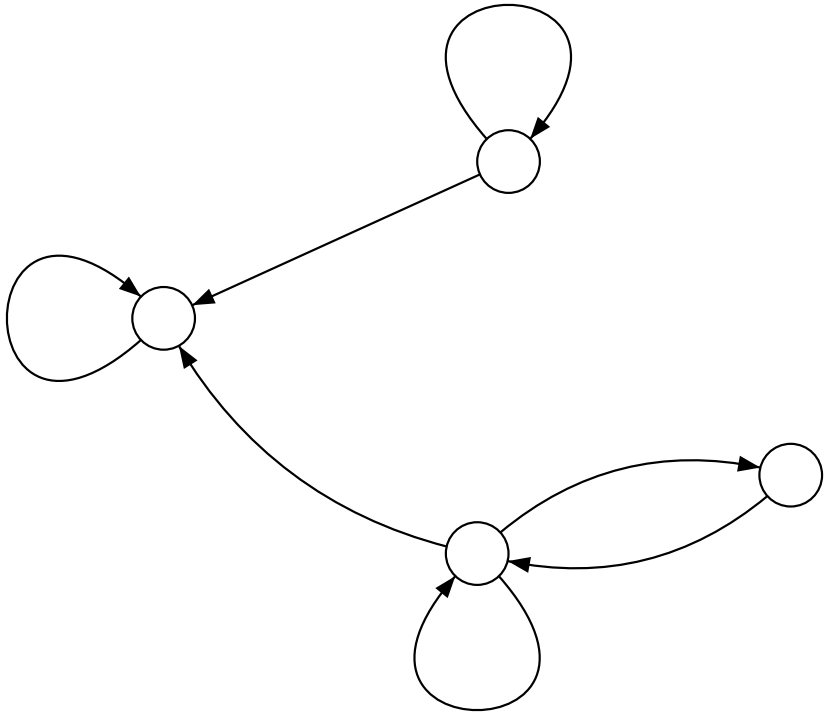




Réduction de l'espace des états

- Exemple





Bisimulation

- La bisimulation probabiliste [Larsen,Skou 91] agglomère les états similaires dans leur comportement probabiliste.
- R est une relation de bisimulation si pour tout états $s, s' \in S$:

$$sRs' \iff \forall a \in Action$$

$$cout_s^a = cout_{s'}^a \; et \; \forall C \in S/R \; (P_s^a(C) = P_{s'}^a(C))$$

Bisimulation

- Deux MDP bisimilaires vérifient les mêmes propriétés.
- La bisimulation est trop stricte car :
  - ➤ elle n'agglomère pas des états très proches
  - ➤ elle n'est pas robuste
- Il est nécessaire d'explorer des solutions approximatives.

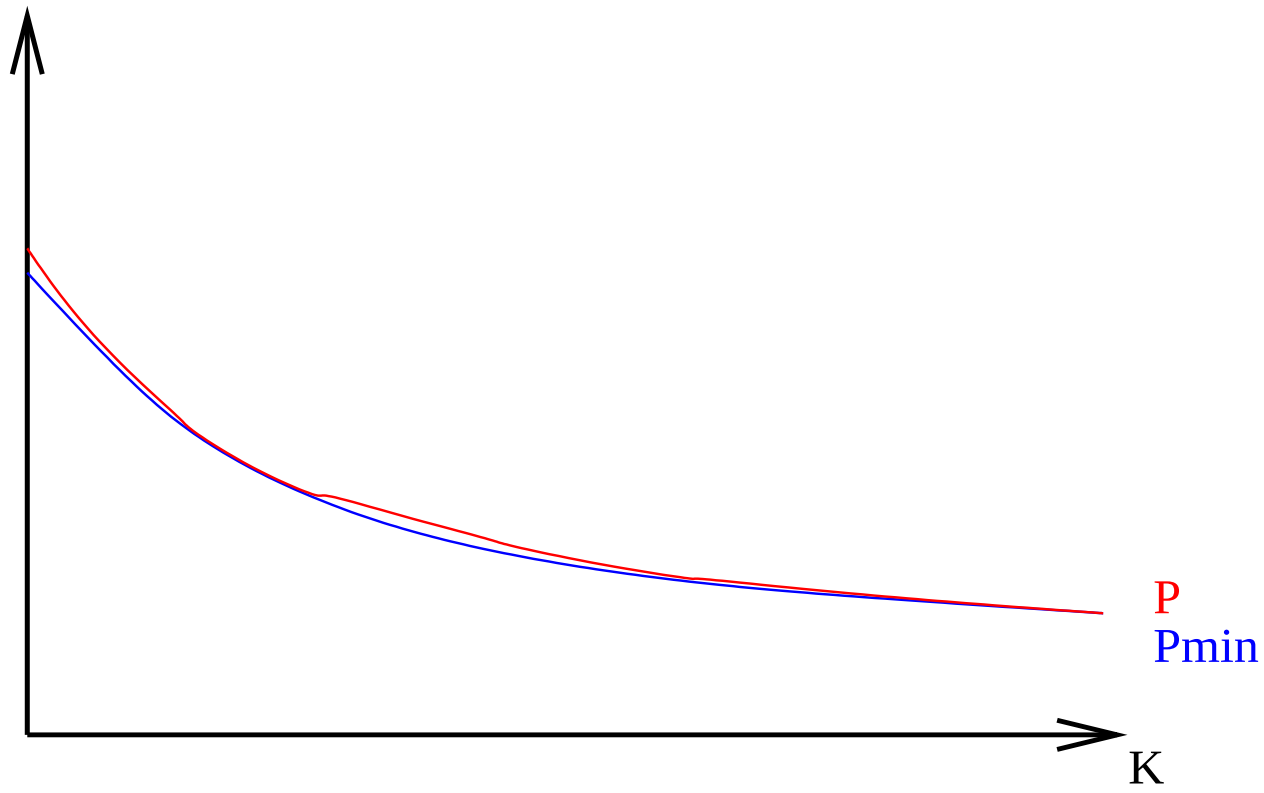
Distance de bisimulation

- Une distance de bisimulation d_{bisim} mesure la différence de deux états dans leur comportement probabiliste.
- On retrouve la bisimulation probabiliste quand la distance vaut 0.
- On peut agglomérer les états à ϵ près.
- La différence du coût optimal de deux états s, s' est inférieur à $O(d_{bisim}(s, s'))$[Ferns03]

Perspective

- Un système bisimilaire à ϵ près ne vérifie plus les mêmes formules.
- On voudrait trouver une agglomération des états telle que la propriété qu'on souhaite vérifier garde sa valeur de vérité dans le système aggloméré.

Perspective



- La politique choisie est proche de l'optimale.