

Vérification des Processus Décisionnels de Markov pondérés

Florent Delgrange

UMONS

Faculté des Sciences

Mab2 Science Informatique

Table des matières

1. Préliminaires

1.1 Système de transition

1.2 Chemins et Traces de TS

2. LTL

2.1 Intuition

2.2 Syntaxe

2.3 Sémantique

3. CTL

3.1 Intuition

3.2 Syntaxe

3.3 Sémantique

3.4 LTL vs CTL

4. PCTL

4.1 MC

4.2 Intuition

4.3 Syntaxe

4.4 Sémantique

4.5 Comparaison de logiques

temporelles en branchements

5. PRCTL

5.1 WMC

5.2 Intuition

5.3 Syntaxe

5.4 Sémantique

5.5 MDP et stratégies

5.6 PRCTL pour les MDPs

5.7 PRCTL dans Storm

Système de transition

Definition (Système de transition)

Un *système de transition* (noté TS, pour *transition system*) est un tuple $\mathcal{T} = (S, A, \rightarrow, AP, L)$ où

- S est un ensemble d'états,
- A est un ensemble d'actions,
- $\rightarrow \subseteq S \times A \times S$ est une relation de transition,
- AP est un ensemble de propositions atomiques et
- $L : S \rightarrow 2^{AP}$ est une fonction d'étiquetage.

Système de Transition

- **Idée** : Graphe orienté
 - noeuds : états du système
 - arcs : transitions du système
- **État** : décrit les informations d'un système à un certain moment de son comportement.
- **Transition** : si un état a plus d'une transition sortante, alors le comportement du système est **non-déterministe**, i.e., l'évolution du système requiert la sélection d'une transition.
- **Étiquetage** : $L(s)$ est l'ensemble des étiquettes $a \in AP$ de l'état s .
- **Pas d'états terminaux !**

Système de Transition

Exemple

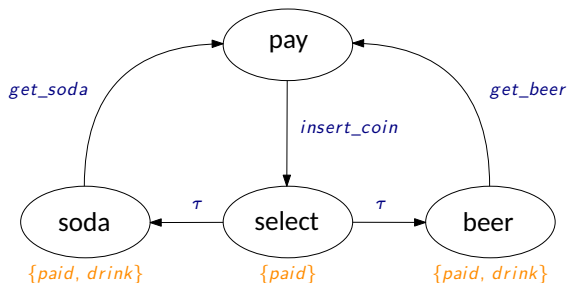


Figure – Distributeur de boissons [1]

- $S = \{pay, select, beer, soda\}$
- $A = \{insert_coin, \tau, get_soda, get_beer\}$
- $AP = \{paid, drink\}$

Chemins

Un chemin d'un système de transition est une succession d'état possible résultant de l'exécution de ce système.

- **Idée** : pas d'états terminaux \implies chemins infinis.

Definition (Chemin d'un TS)

Soit $\mathcal{T} = (S, A, \rightarrow, AP, L)$, un TS.

$\pi = s_0s_1s_2s_3\dots$ est un *chemin* (infini) de \mathcal{T} ssi pour tout $i \in \mathbb{N}$, il existe une action $\alpha \in A$ telle que $s_i \xrightarrow{\alpha} s_{i+1}$, avec $s_i, s_{i+1} \in S$.

L'ensemble des chemins (infinis) $\pi = s_0s_1\dots$ commençant en l'état s (i.e., tels que $s_0 = s$) est dénoté par $Paths(s)$.

Traces

Les traces d'un système de transition sont des mots infinis sur l'alphabet 2^{AP} formés lors de l'exécution du système.

Definition (Traces)

Soit $\mathcal{T} = (S, A, \rightarrow, AP, L)$, un TS. La trace du chemin $\pi = s_0 s_1 \dots$ est donné par

$$trace(\pi) = L(s_0)L(s_1)\dots$$

Dès lors, soit $s \in S$, un état de \mathcal{T} , les traces du système provenant de l'état s est donné par

$$Traces(s) = \{trace(\pi) \mid \pi \in Paths(s)\}$$

Chemins et Traces

Exemple

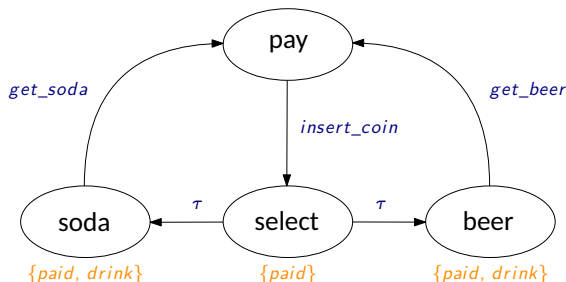


Figure – Distributeur de boissons [1]

- $\pi = \text{pay select soda pay select beer} \dots \in \text{Paths}(\text{pay})$
- $\emptyset\{\text{paid}\}\{\text{paid, drink}\}\emptyset\{\text{paid}\}\{\text{paid, drink}\}\dots = \text{trace}(\pi) \in \text{Traces}(\text{paid})$

Table des matières

1. Préliminaires

1.1 Système de transition

1.2 Chemins et Traces de TS

2. LTL

2.1 Intuition

2.2 Syntaxe

2.3 Sémantique

3. CTL

3.1 Intuition

3.2 Syntaxe

3.3 Sémantique

3.4 LTL vs CTL

4. PCTL

4.1 MC

4.2 Intuition

4.3 Syntaxe

4.4 Sémantique

4.5 Comparaison de logiques

temporelles en branchements

5. PRCTL

5.1 WMC

5.2 Intuition

5.3 Syntaxe

5.4 Sémantique

5.5 MDP et stratégies

5.6 PRCTL pour les MDPs

5.7 PRCTL dans Storm

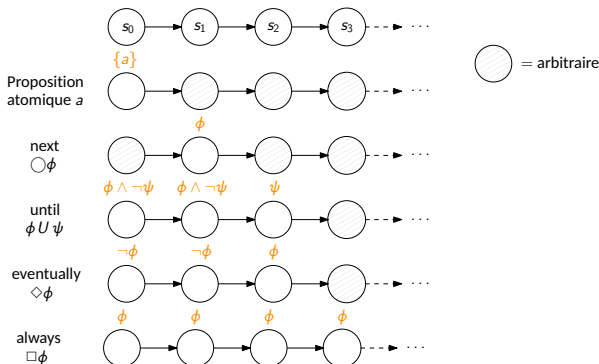
Logique temporelle linéaire (LTL)

- L'exactitude des systèmes réactifs dépend des exécutions + de l'équité du système
 - La logique **temporelle** permet de traiter ces aspects
 - *temps "réel"* (discret !)
 - Temps linéaire \implies logique basée sur les **chemins du système**
 - à chaque étape, un seul successeur est possible
- **LTL** \approx langage qui a pour but de vérifier des propriétés sur les exécutions d'un système

Logique temporelle linéaire (LTL) : intuition

Soit $\mathcal{T} = (S, A, \rightarrow, AP, L)$, LTL est formée par ...

1. des propositions atomiques $a \in AP$,
2. des combinaisons booléennes de formules : $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$ et
3. des opérateurs temporels : soit $\pi = s_0s_1s_2s_3 \cdots \in Paths(\mathcal{T})$



Syntaxe

Soit AP , un ensemble de propositions atomiques, les *formules* LTL sont formées selon la *grammaire* suivante :

$$\phi ::= true \mid a \mid \phi \wedge \psi \mid \neg \phi \mid \bigcirc \phi \mid \phi U \psi$$

où $a \in AP$

Note : $\phi U \psi$ requiert l'apparition de ψ dans le chemin ; ϕ indéfiniment n'est pas suffisant !

Syntaxe

Opérateurs dérivés :

eventually

$$\Diamond\phi \equiv \text{true} \, U \, \phi$$

always

$$\Box\phi \equiv \neg\Diamond\neg\phi$$

Ordre de précedence :

1. parenthèses
2. opérations unaires (\neg , \bigcirc)
3. opérations binaires :

3.1 U (associatif par la droite, e.g., $\phi_1 U \phi_2 U \phi_3 \equiv \phi_1 U (\phi_2 U \phi_3)$)

3.2 \wedge

Combinaisons de modalités temporelles :

- $\Box\Diamond\phi$ “infiniment souvent ϕ ”
- $\Diamond\Box\phi$ “éventuellement toujours ϕ ”

Combinaisons de modalités temporelles

Exemple (infiniment souvent)

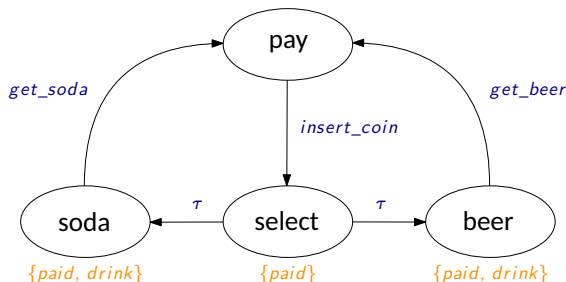


Figure – Distributeur de boissons [1]

- Pour toute trace de l'exécution du système depuis *pay*, i.e., $\forall \sigma \in \text{Traces}(\text{pay})$, pour toute position dans σ , le label *drink* doit apparaître dans le futur.
 $\rightsquigarrow \Box \Diamond \text{drink}$

Combinaisons de modalités temporelles

Exemple (éventuellement toujours)

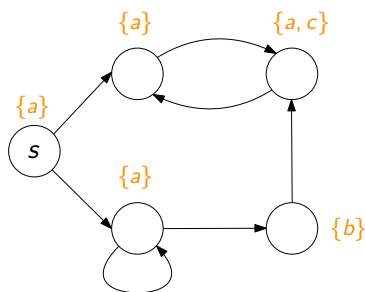


Figure - [2]

- $\forall \sigma \in \text{Traces}(s)$, il y a toujours un moment où on voit toujours a , mais plus b
 $\rightsquigarrow \Diamond \Box (a \wedge \neg b)$

Sémantique

Soient $\mathcal{T} = (S, A, \rightarrow, AP, L)$ et ϕ , une formule LTL sur AP , la propriété LT induite par ϕ est le langage de mots

$$Words(\phi) = \{\sigma = A_0A_1A_2\cdots \in (2^{AP})^\omega \mid \sigma \models \phi\}$$

où \models est la plus petite relation satisfaisant

$$\sigma \models true$$

$$\sigma \models a \quad \text{ssi } a \in A_0$$

$$\sigma \models \phi \wedge \psi \quad \text{ssi } \sigma \models \phi \text{ et } \sigma \models \psi$$

$$\sigma \models \neg\phi \quad \text{ssi } \sigma \not\models \phi$$

$$\sigma \models \bigcirc\phi \quad \text{ssi } \sigma[1:] = A_1A_2\ldots \models \phi$$

$$\sigma \models \phi U \psi \quad \text{ssi } \exists j \geq 0, \sigma[j:] \models \psi \text{ et } \forall 0 \leq i < j, \sigma[i:] \models \phi$$

Sémantique

Soit $s \in S$,

- $\forall \pi \in Paths(s), \pi \models \phi$ ssi $trace(\pi) \models \phi$
- $s \models \phi$ ssi $\forall \pi \in Paths(s), \pi \models \phi$

Exemple

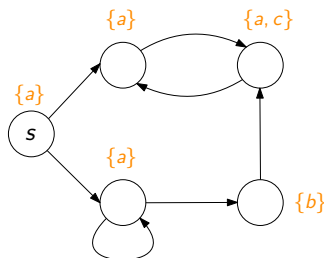


Figure - [2]

- $s \models \Diamond \Box (a \wedge \neg b)$

Table des matières

1. Préliminaires

1.1 Système de transition

1.2 Chemins et Traces de TS

2. LTL

2.1 Intuition

2.2 Syntaxe

2.3 Sémantique

3. CTL

3.1 Intuition

3.2 Syntaxe

3.3 Sémantique

3.4 LTL vs CTL

4. PCTL

4.1 MC

4.2 Intuition

4.3 Syntaxe

4.4 Sémantique

4.5 Comparaison de logiques

temporelles en branchements

5. PRCTL

5.1 WMC

5.2 Intuition

5.3 Syntaxe

5.4 Sémantique

5.5 MDP et stratégies

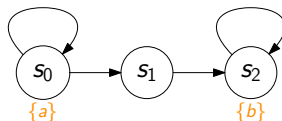
5.6 PRCTL pour les MDPs

5.7 PRCTL dans Storm

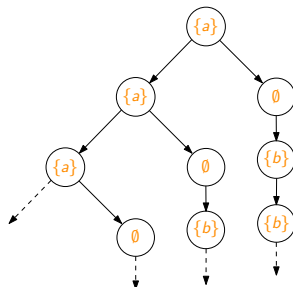
Logique d'arbre de calculs (CTL)

- Notion d'**arbre d'exécution = arbre de calcul**
- ↪ Dépliage infini du système considérant toutes les possibilités de branchement

Arbre de calculs



Arbre de calculs depuis l'état s_0 ?



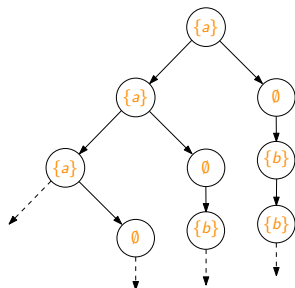
Est-ce que toutes les exécutions ont toujours la possibilité d'atteindre éventuellement $\{b\}$?

Quantificateurs

- **LTL** : $s \models \phi$ signifie que tous les chemins commençant en s satisfont ϕ
 - Quantification explicite !
 - $s \models \forall \phi$
- **CTL** : on peut considérer seulement certains chemins
 - Existe-t-il un chemin satisfaisant ϕ commençant en s ?
 - $s \models \exists \phi \iff \underbrace{s \not\models \forall \neg \phi}_{\text{LTL : } s \not\models \neg \phi}$

Quantificateurs

Motivation

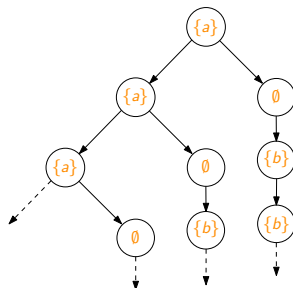


Est-ce que **toutes** les exécutions ont toujours **la possibilité** d'atteindre éventuellement **{b}** ?

- **LTL :**
 - $s_0 \models \Box \Diamond b$ **ne fonctionne pas !**
 - requiert que **tous les chemins** du système de transition atteignent **{b}**
 - ⇒ On ne parle pas de possibilité d'atteindre **{b}**
- **Pas expressible en LTL**

Quantificateurs

Motivation



Est-ce que **toutes** les exécutions ont toujours **la possibilité** d'atteindre éventuellement **{b}** ?

→ Besoin de quantificateurs

• **CTL** :

- $s_0 \models \forall \Box \exists \Diamond b$
- Pour tout chemin commençant en s_0 , à chaque étape, il existe un chemin qui peut atteindre b .

CTL vs LTL

Comparaison intuitive

- **LTL :**

- chemins + traces
- temps linéaire
- chaque point a un seul futur possible

- **CTL :**

- arbre de calculs + comportement des branchements
- temps en branchements
- chaque noeud de l'arbre a plusieurs futurs possibles

CTL

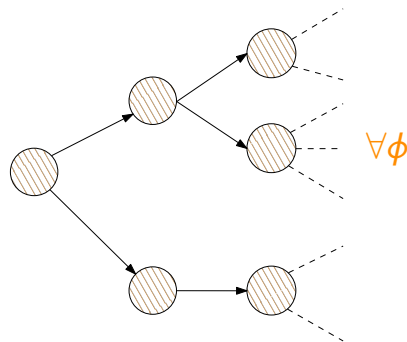
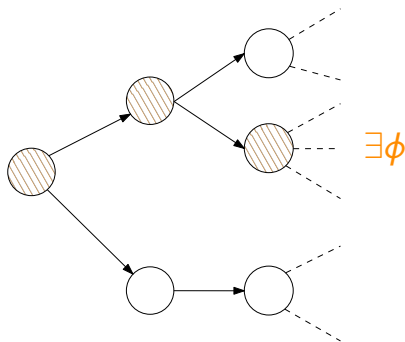
Intuition

- Formules d'états
- = Assertions de propositions atomiques dans des états ainsi que leur structure de branchement
- propositions atomiques $a \in AP$
 - combinaisons booléennes de formules : $\neg\Phi$, $\Phi \wedge \Psi$, $\Phi \vee \Psi$
 - quantification de chemins via des **formules de chemins**

CTL

Intuition

- Formules de chemins



Formules de chemins

Formules LTL \neq formules de chemin CTL !

En effet, les formules de chemin CTL...

- ne peuvent pas être combinées avec des connecteurs booléens
- ne permettent pas l'imbrication des modalités temporelles

Exemple :

$$s \models \forall \square \exists \diamond b$$

correct

$$s \models \forall \square \diamond b$$

incorrect

Syntaxe

Soit AP , un ensemble de propositions atomiques.

- Les *formules d'états* CTL sont formées selon la grammaire suivante :

$$\Phi ::= true \mid a \mid \Phi \wedge \Psi \mid \neg \Phi \mid \exists \phi \mid \forall \phi$$

où $a \in AP$ et ϕ est une formule de chemin.

- Les *formules de chemins* CTL sont formées selon la grammaire suivante :

$$\phi ::= \bigcirc \Phi \mid \Phi U \Psi$$

où Φ et Ψ sont des formules d'états.

Sémantique

Soient $\mathcal{T} = (S, A, \rightarrow, AP, L)$, un TS et $s \in S$, un **état** de \mathcal{T} .

$s \models \Phi$ ssi la formule Φ tient dans l'état s , i.e.,

$$s \models \text{true}$$

$$s \models a \quad \text{ssi } a \text{ est un label de } s, \text{ i.e., } a \in L(s)$$

$$s \models \Phi \wedge \Psi \quad \text{ssi } s \models \Phi \text{ et } s \models \Psi$$

$$s \models \neg \Phi \quad \text{ssi } s \not\models \Phi$$

$$s \models \exists \phi \quad \text{ssi } \exists \pi \in \text{Paths}(s), \pi \models \phi$$

$$s \models \forall \phi \quad \text{ssi } \forall \pi \in \text{Paths}(s), \pi \models \phi$$

Sémantique

Soient $\mathcal{T} = (S, A, \rightarrow, AP, L)$, un TS et $\pi = s_0 s_1 s_2 \cdots \in Paths(s)$,
un **chemin**.

$\pi \models \phi$ ssi π satisfait ϕ , i.e.,

$$\pi \models \Phi \quad \text{ssi } s_0 \models \Phi$$

$$\pi \models \bigcirc \Phi \quad \text{ssi } s_1 \models \Phi$$

$$\pi \models \Phi U \Psi \quad \text{ssi } \exists j \in \mathbb{N}, s_j \models \Psi \text{ et } \forall 0 \leq i < j, s_i \models \Phi$$

$$\pi \models \Diamond \Phi \quad \text{ssi } \exists j \in \mathbb{N}, s_j \models \Phi$$

$$\pi \models \Box \Phi \quad \text{ssi } \forall j \in \mathbb{N}, s_j \models \Phi$$

Satisfiabilité

Definition (Ensemble de satisfaction)

Soient $\mathcal{T} = (S, A, \rightarrow, AP, L)$, un TS et Φ , une formule d'état CTL sur AP . L'ensemble de satisfaction du TS \mathcal{T} est donné par

$$Sat_{\mathcal{T}}(\Phi) = \{s \in S \mid s \models \Phi\}$$

LTL vs CTL

LTL et CTL sont incomparables !

Exprimable en ...

- CTL mais pas LTL :
 $\forall \Box \exists \Diamond a$ (voir exemple)
- LTL mais pas CTL :
 $\Diamond \Box a$

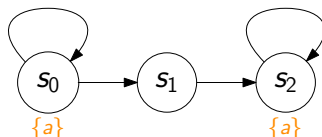
En effet, $\Diamond \Box a \neq \forall \Diamond \forall \Box a$!

- $\Diamond \Box a$ assure que a sera atteint éventuellement en tout point.
- $\forall \Diamond \forall \Box a$ affirme que pour toute exécution, un état S est éventuellement atteint, tel que $S \models \forall \Box a$

LTL vs CTL

Exemple

- $AP = \{a\}$



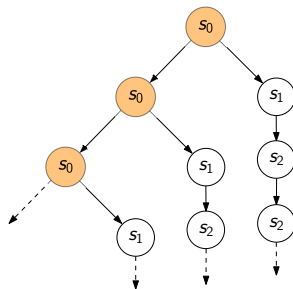
- s_0 satisfait la formule LTL $\Diamond \Box a$ car chaque chemin commençant en s_0 reste éventuellement toujours en s_0 ou en s_2 , tous les deux étiquetés avec a .
- s_0 **ne satisfait pas** la formule CTL $\forall \Diamond \forall \Box a$.
Prenons le chemin s_0^ω .
 $s_0^\omega \not\models \Diamond \forall \Box a$

LTL vs CTL

Exemple

$$s_0^\omega \not\models \Diamond \forall \Box a$$

- $\pi = s_0 s_1 \dots \models \Diamond \Phi \iff \exists j \in \mathbb{N}, s_j \models \Phi$
- $s \models \forall \Box a \iff \forall \pi = s_0 s_1 s_2 \dots \in Paths(s), a \in L(s_i) \forall i \in \mathbb{N}$



Le chemin

$$s_0^* s_1 s_2^\omega$$

passé par un état $\neg a$
(i.e., par s_1).

- Il n'existe pas d'états dans le chemin s_0^ω qui va satisfaire $\forall \Box a$ car $s_0 \not\models \forall \Box a$

Table des matières

1. Préliminaires

1.1 Système de transition

1.2 Chemins et Traces de TS

2. LTL

2.1 Intuition

2.2 Syntaxe

2.3 Sémantique

3. CTL

3.1 Intuition

3.2 Syntaxe

3.3 Sémantique

3.4 LTL vs CTL

4. PCTL

4.1 MC

4.2 Intuition

4.3 Syntaxe

4.4 Sémantique

4.5 Comparaison de logiques temporelles en branchements

5. PRCTL

5.1 WMC

5.2 Intuition

5.3 Syntaxe

5.4 Sémantique

5.5 MDP et stratégies

5.6 PRCTL pour les MDPs

5.7 PRCTL dans Storm

Definition (Chaîne de Markov à temps discret)

Une *chaîne de Markov à temps discret*, notée **MC** (pour *Markov Chain*), est un modèle probabiliste défini par un tuple $\mathcal{M} = (S, \Delta, AP, L)$ où :

- S est un ensemble dénombrable d'états,
- $\Delta : S \times S \rightarrow [0, 1] \cap \mathbb{Q}$ est une *fonction de transition* telle que

$$\forall s \in S, \sum_{s' \in S} \Delta(s, s') = 1$$

où $\Delta(s, s')$ est la probabilité de passer de l'état s à l'état s' ,

- AP est un ensemble de propositions atomiques et
- $L : S \rightarrow 2^{AP}$ est une fonction d'étiquetage.

Chaînes de Markov

- Les MCs sont des modèles **déterministes**
- L'idée des chemins d'une MC est la même que pour les TSs :

Definition (Chemin dans une MC)

Un **chemin** (infini) $\pi = s_0 s_1 s_2 \dots \in S^\omega$ est une séquence d'états de la MC $\mathcal{M} = (S, \Delta, AP, L)$ où $\forall i \in \mathbb{N}, \Delta(s_i, s_{i+1}) > 0$.

$Paths(s)$ est l'ensemble des chemins de \mathcal{M} qui commencent en l'état $s \in S$.

Logique en arbre de calculs probabiliste (PCTL)

- CTL probabiliste
- Logique temporelle en branchements pour exprimer les propriétés d'états des MCs.
- Logique proche de CTL pour les systèmes probabilistes.

CTL vs PCTL

- **CTL :**

Chemins quantifiés en utilisant \forall et \exists

- **PCTL :**

Chemins quantifiés en utilisant leur probabilité, avec $\mathcal{P}_J(\phi)$ où $J \subseteq [0, 1]$ et ϕ est une formule de chemin

$$s \models \mathcal{P}_J(\phi) \text{ ssi } \mathbb{P}_s(\{\pi \in \text{Paths}(s) \mid \pi \models \phi\}) \in J$$

+ PCTL inclus additionally le *until borné* $U^{\leq n}$

Syntaxe

Soit AP , un ensemble de propositions atomiques.

- Les *formules d'états* PCTL sont formées selon la grammaire suivante :

$$\Phi ::= \text{true} \mid a \mid \Phi \wedge \Psi \mid \neg \Phi \mid \mathcal{P}_J(\phi)$$

où $a \in AP$, $J \subseteq [0, 1]$ et ϕ est une formule de chemin.

- Les *formules de chemins* PCTL sont formées selon la grammaire suivante :

$$\phi ::= \bigcirc \Phi \mid \Phi U \Psi \mid \Phi U^{\leq n} \Psi$$

où Φ et Ψ sont des formules d'états et $n \in \mathbb{N}$.

Sémantique

Soient $\mathcal{M} = (S, A, \Delta, AP, L)$, une MC et $s \in S$, un **état** de \mathcal{M} .

$s \models \Phi$ ssi la formule Φ tient dans l'état s , i.e.,

$$s \models \text{true}$$

$$s \models a \quad \text{ssi } a \text{ est un label de } s, \text{ i.e., } a \in L(s)$$

$$s \models \Phi \wedge \Psi \quad \text{ssi } s \models \Phi \text{ et } s \models \Psi$$

$$s \models \neg \Phi \quad \text{ssi } s \not\models \Phi$$

$$s \models \mathcal{P}_J(\phi) \quad \text{ssi } \mathbb{P}_s(\phi) \in J$$

où $\mathbb{P}_s(\phi) = \mathbb{P}_s(\{\pi \in \text{Paths}(s) \mid \pi \models \phi\})$ et \mathbb{P}_s est la mesure de probabilité sur le σ -algèbre dont les résultats sont les chemins commençant en s , i.e., $\text{Paths}(s)$

Sémantique

Soient $\mathcal{M} = (S, A, \Delta, AP, L)$, une MC et

$\pi = s_0 s_1 s_2 \cdots \in Paths(s)$, un **chemin**.

$\pi \models \phi$ ssi π satisfait ϕ , i.e.,

$$\pi \models \Phi \quad \text{ssi } s_0 \models \Phi$$

$$\pi \models \bigcirc \Phi \quad \text{ssi } s_1 \models \Phi$$

$$\pi \models \Phi U \Psi \quad \text{ssi } \exists j \in \mathbb{N}, s_j \models \Psi \text{ et } \forall 0 \leq i < j, s_i \models \Phi$$

$$\pi \models \Phi U^{\leq n} \Psi \quad \text{ssi } \exists 0 \leq j \leq n, s_j \models \Psi \text{ et } \forall 0 \leq i < j, s_i \models \Phi$$

$$\pi \models \Diamond \Phi \quad \text{ssi } \exists j \in \mathbb{N}, s_j \models \Phi$$

$$\pi \models \Box \Phi \quad \text{ssi } \forall j \in \mathbb{N}, s_j \models \Phi$$

Sémantique

Remarque

Soient $\mathcal{M} = (S, \Delta, AP, L)$, une MC et $T \subseteq S$, un sous-ensemble d'états de \mathcal{M} .

La pseudo-formule de chemin $\Diamond T$ est équivalente à la formule de chemin $\Diamond \Phi$ telle que :

- $\exists T_{AP} \subseteq AP$
- $\Phi ::= \bigwedge_{a \in T_{AP}} a$
- $\forall t \in T, \forall a \in T_{AP}, a \in L(t)$
- $\forall s \notin T, \exists b \in T_{AP}$ telle que $b \notin L(s)$

Satisfiabilité

L'ensemble de satisfaction d'une MC est essentiellement défini de la même façon que pour les TSs.

Definition (Ensemble de satisfaction)

Soient $\mathcal{M} = (S, A, \Delta, AP, L)$, une MC et Φ , une formule d'état PCTL sur AP . L'ensemble de satisfaction de la MC \mathcal{M} est donné par

$$Sat_{\mathcal{M}}(\Phi) = \{s \in S \mid s \models \Phi\}$$

PCTL vs CTL

$$s \models \mathcal{P}_{=1}(\Diamond\Phi) \not\Rightarrow s \models \forall\Diamond\Phi$$

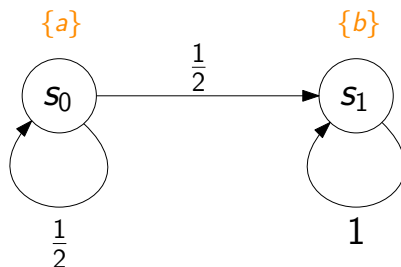
La probabilité que tous les chemins satisfassent une formule de chemin PCTL avec une probabilité de 1 ne signifie pas que tous les chemins satisfassent la formule de chemin CTL correspondante !

$$s \models \mathcal{P}_{>0}(\Box\Phi) \not\Leftrightarrow s \models \exists\Box\Phi$$

Le fait qu'un chemin satisfasse une formule de chemin CTL n'implique pas forcément que la probabilité des chemins satisfaisant la formule PCTL correspondante soit non-nulle !

PCTL vs CTL

Exemple



- $s_0 \models \mathcal{P}_{=1}(\Diamond b)$, mais $s \not\models \forall \Diamond b$
- $s_0 \models \exists \Box a$, mais $s \not\models \mathcal{P}_{>0}(\Box a)$

Table des matières

1. Préliminaires

1.1 Système de transition

1.2 Chemins et Traces de TS

2. LTL

2.1 Intuition

2.2 Syntaxe

2.3 Sémantique

3. CTL

3.1 Intuition

3.2 Syntaxe

3.3 Sémantique

3.4 LTL vs CTL

4. PCTL

4.1 MC

4.2 Intuition

4.3 Syntaxe

4.4 Sémantique

4.5 Comparaison de logiques temporelles en branchements

5. PRCTL

5.1 WMC

5.2 Intuition

5.3 Syntaxe

5.4 Sémantique

5.5 MDP et stratégies

5.6 PRCTL pour les MDPs

5.7 PRCTL dans Storm

Chaînes de Markov pondérées

Definition (Chaîne de Markov pondérée)

Une *chaîne de Markov pondérée* (WMC, pour *weighted Markov chain*) \mathcal{M} est une chaîne de Markov enrichie par une fonction de poids. \mathcal{M} est définie par le tuple (S, Δ, AP, L, w) tel que :

- S, Δ, AP et L sont définis comme pour une MC classique et
- $w : S \times S \rightarrow \mathbb{N}^{>0}$ est la fonction de poids associant à chaque transition un coût strictement positif.

Chaînes de Markov pondérées

Exemple

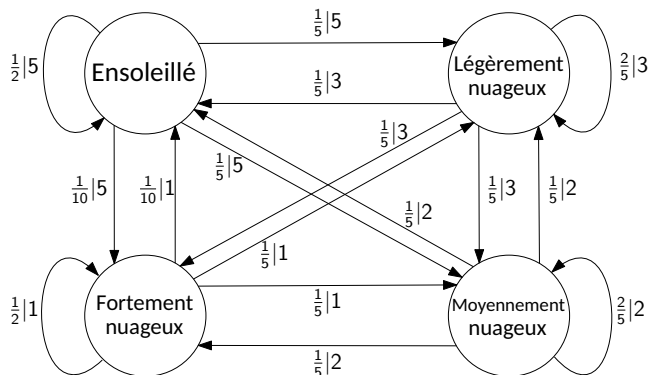


Figure – Système équipé de panneaux solaires produisant de l'énergie en fonction du climat.

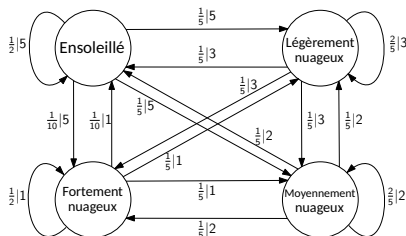
Chaînes de Markov pondérées

Soient $\mathcal{M} = (S, \Delta, AP, L, w)$, une WMC, $s \in S$, un état de \mathcal{M} , $T \subseteq S$, un sous-ensemble d'états cibles et $\pi \in Paths(s)$.

- $TS^T(\pi)$, la *somme tronquée de π* , est le coût du chemin π jusqu'à satisfaire (pour la première fois) $\Diamond T$
- $\mathbb{E}_s(TS^T) = \mathbb{E}_s(\{TS^T(\pi) \mid \pi \in Paths(s)\})$ est l'espérance de la longueur des chemins (en terme de coût) pour que $s \models \mathcal{P}_{=1}(\Diamond T)$ (pour la première fois)
- $\mathbb{P}_s(\Diamond_{\leq l} T) = \mathbb{P}_s(\{\pi \in Paths(s) \mid TS^T(\pi) \leq l\})$ est la probabilité que les chemins $\pi \in Paths(s) \models \Diamond T$ avec un coût (i.e., une somme tronquée) inférieure à $l \in \mathbb{N}$.

Chaînes de Markov pondérées

Exemple



$$TS^{\{Fn\}}(E \cdot Ln \cdot Ln \cdot Mn \cdot Fn \dots)$$

$$= 5 + 3 + 3 + 2 = 13$$

- $\mathbb{E}_E(TS^{\{Fn\}}) = 25Kj$
- $1 - \mathbb{P}_E(\Diamond_{\leq 7}\{Fn\}) = \mathbb{P}_E(\Diamond_{> 8}\{Fn\}) = 1 - 0.14 = 0.86$

PRCTL

Intuition

- PCTL + Espérance des “rewards” (\approx coûts) des chemins.
- Inclus un until borné par le coûts des chemins en terme de somme tronquée.

Syntaxe

Soit AP , un ensemble de propositions atomiques.

- Les *formules d'états* PRCTL sont formées selon la grammaire suivante :

$$\Phi ::= true \mid a \mid \Phi \wedge \Psi \mid \neg \Phi \mid \mathcal{P}_J(\phi) \mid \mathcal{E}_R(\Phi)$$

où $a \in AP$, $J \subseteq [0, 1]$, $R \in [0, +\infty[\cap \mathbb{N}$ (bornes d'espérances du coût des chemins) et ϕ est une formule de chemin.

- Les *formules de chemins* PRCTL sont formées selon la grammaire suivante :

$$\phi ::= \bigcirc \Phi \mid \Phi U \Psi \mid \Phi U^{\leq n} \Psi \mid \Phi U_{\leq r} \Psi$$

où Φ et Ψ sont des formules d'états et $n, r \in \mathbb{N}$.

Sémantique

Soient $\mathcal{M} = (S, A, \Delta, AP, L)$, une MC, $s \in S$, un état de \mathcal{M} et

$\pi = s_0 s_1 s_2 \dots \in Paths(s)$, un chemin de \mathcal{M} .

La sémantique de PRCTL est la même que celle de PCTL, à l'exception que

- $s \models \Phi$ ssi la formule Φ tient dans l'état s , i.e.,

$$s \models \mathcal{E}_R(\Phi) \quad \text{ssi } \mathbb{E}_S(\Diamond \Phi) \in R$$

- $\pi \models \phi$ ssi π satisfait ϕ , i.e.,

$$\pi \models \Phi U_{\leq r} \Psi \quad \text{ssi } \exists j \in \mathbb{N}, s_j \models \Psi, \forall 0 \leq i < j, s_i \models \Phi \text{ et } TS^{Sat_{\mathcal{M}}(\Psi)}(\pi) \leq r$$

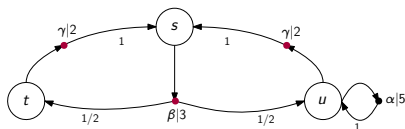
Note : la définition de l'ensemble de satisfaction d'une WMC PRCTL est identique à celle de PCTL

Processus Décisionnel de Markov et Stratégie

- Un *processus décisionnel de Markov* (MDP, pour *Markov decision process*) est un modèle probabiliste **non-déterministe**.
- $\mathcal{M} = (S, A, \Delta, AP, L, w)$
 - Actions : A
 - Fonction de transition : $\Delta : S \times A \times S \rightarrow [0, 1] \cap \mathbb{Q}$
- *Chemins* de \mathcal{M} : $\pi = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} s_3 \dots$
 - $s_i \xrightarrow{\alpha_i} s_{i+1}$ pour tout $i \in \mathbb{N}$
 - $\mathcal{H}(\mathcal{M})$: histoires de \mathcal{M} , i.e., l'ensemble des préfixes des chemins de \mathcal{M}
- *Stratégies* de \mathcal{M} , $\sigma : \mathcal{H}(\mathcal{M}) \rightarrow A$
- Les décisions d'un MDP \mathcal{M} , régulées par une stratégie σ , induisent une MC \mathcal{M}^σ

Processus Décisionnel de Markov

Exemple

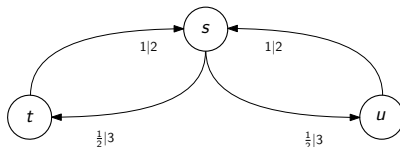


- $\mathcal{M} = (S, A, \Delta, w)$

- $\sigma: S \rightarrow A,$

- $\sigma(s) = \beta$

- $\sigma(t) = \sigma(u) = \gamma$



$$\pi = s \xrightarrow{\beta} t \xrightarrow{\gamma} s \xrightarrow{\beta} u \xrightarrow{\gamma} \dots \in \text{Paths}(s)$$

$\pi = s t s u \dots \in \text{Paths}(s)$ est un chemin de \mathcal{M}^σ

Processus Décisionnel de Markov

Soient $\mathcal{M} = (S, A, \Delta, AP, L, w)$, un MDP, $s \in S$, un état de \mathcal{M} et $T \subseteq S$, un sous-ensemble d'états cibles.

- $\mathbb{E}_s^{\min}(\Diamond T)$ est l'espérance minimale de la longueur des chemins commençant en s (en terme de somme tronquée) de \mathcal{M}
 - i.e., l'espérance de la longueur des chemins commençant en s dans la MC induite par la stratégie qui minimise l'espérance de la longueur des chemins de \mathcal{M} .
- $\mathbb{P}_s^{\max}(\Diamond_{\leq l} T)$ est la probabilité maximale d'atteindre T depuis s avec un coût inférieur à l dans \mathcal{M}
 - i.e., la probabilité d'atteindre T avec un coût inférieure à l dans la MC induite par la stratégie qui maximise cette probabilité dans \mathcal{M} .

PRCTL pour les MDPs

Pour se référer aux MCs induites par ces stratégies, la syntaxe de PRCTL est essentiellement identique, à l'exception des formules d'états suivantes :

- On ne parle plus de $\mathcal{P}_J(\phi)$, mais plutôt de $\mathcal{P}_J^{\max}(\phi)$
- On ne parle plus de $\mathcal{E}_R(\phi)$, mais plutôt de $\mathcal{E}_R^{\min}(\phi)$

où $J \subseteq [0, 1]$, $R \in [0, +\infty[\cap \mathbb{N}$ et ϕ est une formule de chemin.

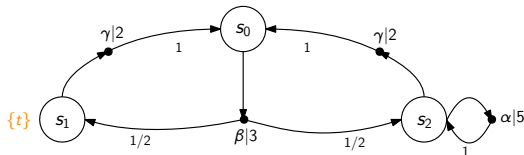
PRCTL dans Storm

Exemple

```

1  mdp
2
3  module classic
4
5  s: [0..2] init 0;
6
7  [beta] s=0 -> 0.5 : (s'=1) +
    0.5 : (s'=2);
8  [gamma] s=1 -> 1 : (s'=0);
9  [alpha] s=2 -> 1 : (s'=2);
10 [gamma] s=2 -> 1 : (s'=0);
11
12 endmodule
13
14 label "t" = s=1;
15
16 rewards "weights"
17   [alpha] true : 5;
18   [beta] true : 3;
19   [gamma] true : 2;
20 endrewards

```



PRCTL dans Storm

Exemple

$$s_0 \models \mathcal{E}_{\leq 10}^{\min}(\Diamond t)$$

```
>> storm --prism resources/simple_mdp.prism --prop "Rmin<=10 [F \"t\"]"
```

Storm 1.2.0

```
-----
Model type: MDP (sparse)
```

```
States: 3
```

```
Transitions: 5
```

```
Choices: 4
```

```
Reward Models: weights
```

```
State Labels: 3 labels
```

```
  * deadlock -> 0 item(s)
```

```
  * init -> 1 item(s)
```

```
  * t -> 1 item(s)
```

```
Choice Labels: none
-----
```

```
Model checking property R[exp]min<=10 [F "t"] ...
```

```
Result (for initial states): true
```

```
Time for model checking: 0.008s.
```

PRCTL dans Storm

Exemple (requête)

$$s_0 \models \mathcal{E}_{=?}^{\min}(\Diamond t)$$

```
>> storm --prism resources/simple_mdp.prism --prop "Rmin=? [F \"t\"]"
```

Storm 1.2.0

```
-----
Model type: MDP (sparse)
```

```
States: 3
```

```
Transitions: 5
```

```
Choices: 4
```

```
Reward Models: weights
```

```
State Labels: 3 labels
```

```
  * deadlock -> 0 item(s)
```

```
  * init -> 1 item(s)
```

```
  * t -> 1 item(s)
```

```
Choice Labels: none
-----
```

```
Model checking property R[exp]min=? [F "t"] ...
```

```
Result (for initial states): 8
```

```
Time for model checking: 0.007s.
```

PRCTL dans Storm

Exemple

$$s_0 \models \mathcal{P}_{\geq 0.7}^{\max}(\Diamond \leq 8 t)$$

```
>> storm --prism resources/simple_mdp.prism --prop "Pmax>=0.7 [F{\"weights\"}<=8 \"t\"]"
```

```
Model checking property Pmax>=7/10 [true Urew{\"weights\"}<=8 "t"] ...
```

```
Result (for initial states): true
```

```
Time for model checking: 0.000s.
```

$$s_0 \models \mathcal{P}_{=?}^{\max}(\Diamond \leq 8 t)$$

```
>> storm --prism resources/simple_mdp.prism --prop "Pmax=? [F{\"weights\"}<=8 \"t\"]"
```

```
Model checking property Pmax=? [true Urew{\"weights\"}<=8 "t"] ...
```

```
Result (for initial states): 0.75
```

```
Time for model checking: 0.010s.s.
```

References I

- [1] Christel Baier et Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008. isbn : 978-0-262-02649-9.
- [2] Mickael Randour. *Formal verification of computer systems*. ULB, 2016.