Vérification algorithmique de systèmes probabilistes.

Processus stochastique

- Un processus stochastique est une famille $\{X_t | t \in T\}$ de variables aléatoires.
  - Exemple 1. $T = \mathbb{N}$: temps discret.
  - Exemple 2. $T = IR$: temps continu.
- La suite X_t peut prendre ses valeurs dans un ensemble X
  - continu (exemple $X = IR$),
  - ou discret
    - fini (exemple $X = \{1, 2, \cdots, n\}$),
    - ou infini (exemple $X = \mathbb{N}$).

Propriété de Markov

- $\{X_n | n \in \mathbb{N}\}$ est Markovienne si

$$Prob(X_{n+1} = s | X_0 = s_0, X_1 = s_1, \cdots, X_n = s_n) = Prob(X_{n+1} = s | X_n =$$

pour tout $n, s, s_0, \cdots, s_n$

- $\{X_t | t \in IR\}$ est Markovienne si

$$Prob(X_{t+t'} = s | X_{t'} = s', X_{t''} = s(t''), t'' < t') = Prob(X_{t+t'} = s | X_{t'} = s$$

pour tout $t, t', s, s', s(t'')$

- Homogène si de plus ces probabilités ne dépendent pas du temps, c'est-à-dire sont indépendantes de n ou de t'.

Chaine de Markov

- Une chaine de Markov (MC, DTMC) est un processus stochastique Markovien à temps discret dans un espace discret fini.
- Une chaine de Markov homogène (HMC) correspond à (S, p_0, p)
  - S: espace fini.
  - $p : S \times S \to [0, 1]$ fonction de probabilités de transitions

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

  - $p_0 : S \to [0, 1]$ une fonction décrivant la probabilité initiale.

$$\sum_{s} p_0(s) = 1$$

Exemple

- Le temps de Belfast en tant que HMC.
<figure>

Notion de chemin

- Un chemin est une suite $s_0 s_1 \cdots s_n$ telle que, $s_i \in S$, $p_0(s_0) > 0$, $p(s_i, s_{i+1}) > 0$ pour tout i.
- La fonction de probabilités de transitions donne une distribution de probabilités sur les chemins.

$$Prob(s_0) = p_0(s_0),$$

$$Prob(s_0 s_1 \ldots s_n) = p(s_{n-1}, s_n) Prob(s_0 s_1 \cdots s_{n-1}).$$

Chaine de Markov Concurrente

- Une chaine de Markov concurrente (CMC) (aussi connu sour le nom de processus de décision de Markov (MDP)) est $M = (S, Steps, p_0)$
  - S est un ensemble fini d'états.
  - $Steps : S \to \mathcal{P}_{<\infty}(Dist(S))$, où $Dist(S)$ désigne les distributions de probabilités sur S,

$$\mu \in Dist(S), \text{ si } \mu : S \to [0,1], \sum_{s \in S} \mu(s) = 1,$$

    $\mathcal{P}_{<\infty}(Dist(S))$ désigne les parties finies de $Dist(S)$.
  - $p_0 \in Dist(S)$ une fonction décrivant la probabilité initiale.

Notion d'ordonnanceur

- Un ordonnanceur est une fonction qui à chaque histoire $s_0 s_1 \cdots s_n$ associe une distribution $\mu \in Steps(s_n)$. Formellement, $u : S^* \rightarrow Dist(S)$, avec $u(s_0 s_1 \cdots s_n) \in Steps(s_n)$.
- Une chaine de Markov concurrente + Un ordonnanceur = Une chaine de Markov.
  - à chaque étape, dans l'état s_n, le prochain état est choisi selon la probabilité $\mu = u(s_0 s_1 \cdots s_n)$.
- Un ordonnanceur permet donc d'éliminer le non-déterminisme.

Exemple

- Problème des philosophes.
  <figure>
  (synchronisation sur les étiquettes, $\neg take_i$ est une étiquette particulière).

Logique LTL

- LTL (PTL) est la logique temporelle

$$\phi ::= p|\phi_1 \wedge \phi_2|\neg\phi_1|\circ\phi_1|\phi_1 U \phi_2.$$

- Quelques problèmes:
  - "Satisfaction":
    - Pour DTMC: $M \models \phi$ ssi $P_M(L(\phi)) > 0$
    - Pour CMC: $M \models \phi$ ssi $\exists u\ P_{M,u}(L(\phi)) > 0$
  - "Universalité":
    - Pour DTMC: M génère des chemins qui vérifient ϕ avec probabilité 1.
  - "Calculs":
    - Pour DTMC: Calculer $P_M(L(\phi))$.

Résultats

Théorème:

- Pour une DTMC,
  - “Satisfaction”: algorithme en temps linéaire en la taille de M et exponentiel en la taille de ϕ.
  - “Calcul”: algorithme en temps polynomial en la taille de M et exponentiel en la taille de ϕ.
- Pour une CMC,
  - “Satisfaction”: algorithme en temps doublement exponentiel (complet pour cette classe).

Un algorithme pour LTL & DTMC

- Algorithme qui, étant donnés ϕ, M, détermine $P_M(L(\phi))$.
  - donne une solution au problème du calcul.
  - donne une solution au problème de la satisfaction.
- Il y a deux transformations correspondantes aux opérateurs $\circ\phi_1$, et $\phi_1 U \phi_2$.
- On va décrire celle de $\phi_1 U \phi_2$.
- On suppose que $\phi_1 U \phi_2$ innermost: ϕ_1, ϕ_2 sont des combinaisons booléennes de propositions atomiques.

Transformation pour $\phi = \phi_1 U \phi_2$

- On partitionne les états S de M en $S^{yes}, S^{no}, S^{?}$ tels que
  - Un état $s \in S^{yes}$ est tel qu'une trajectoire partant de s satisfait ϕ avec probabilité 1.
  - Un état $s \in S^{no}$ est tel qu'une trajectoire partant de s ne satisfait pas ϕ avec probabilité 1.
  - Un état $s \in S^{?}$ est tel que les deux évenements précédent (satifaction de ϕ ou de sa négation) peuvent se produire chacun avec une probabilité non-nulle.

Algorithme

- On met dans S^{yes} tous les états qui satisfont ϕ_2.
- On met dans S^{no} tous les états qui satisfont $\neg\phi_1 \wedge \neg\phi_2$.
- En voyant M comme un graphe (il y a une arête entre s et s' ssi $p(s, s') > 0$), soit M' le sous-graphe induit par les états restants (qui satisfont ϕ_1 et $\neg\phi_2$).
  - On met dans S^{no} tous les états s' de M' tels qu'il y aie aucun chemin dans M' vers un état s'' de M' avec une arête de s'' vers un état s de S^{yes} dans M.
  - On met dans S^{yes} tous les états s' de M' tel qu'il aie aucun chemin dans M' vers un état s'' avec $s'' \in S^{no}$ (mis à jour par règle précédent), ou tel qu'il y a une arête de s'' vers un état s satisfaisant $\neg\phi_2$ dans M.
- On met dans $S^{?}$ tous les états restants.

- Propriété: ces trois ensembles vérifient bien la propriété désirée.

- Propriété: La probabilité q_s qu'un état s satisfasse $\phi = \phi_1 U \phi_2$ peut se calculer par l'ensemble d'équations suivantes:

$$\begin{aligned} q_s &= 1 && \text{if } s \in S^{yes} \\ q_s &= 0 && \text{if } s \in S^{no} \\ q_s &= \textstyle\sum_{s'} p(s,s') q_{s'} && \text{if } s \in S^{?} \end{aligned}$$

  Cet ensemble d'équations possède une solution unique.

- Conséquence: un algorithme de résolution de systèmes linéaires (comme l'élimination de Gauss) donne tous les q_s, $s \in S$.

- Soit ζ une nouvelle proposition atomique, $\zeta \notin AP$
- (motivation ζ en un état sera vrai ssi $\phi_1 U \phi_2$ l'est).
- Soit ϕ^ζ la formule obtenue en remplaçant $\phi_1 U \phi_2$ par ζ dans ϕ.
- Soit M^ζ la nouvelle chaine de Markov (S', p', p'_0) définie par:
  - Etats S':
    - pour $s \in S^{yes}$, S' contient un état s avec $\zeta \in L(s')$.
    - pour $s \in S^{no}$, S' contient un état s avec $\zeta \notin L(s')$.
    - pour $s \in S^{?}$, S' contient un état s^+ avec $\zeta \in L(s+)$, et un état s^- avec $\zeta \notin L(s-)$.

- Distribution Initiale p'_0:
  - $p'_0(s) = p_0(s)$ pour $s \in S^{yes} \cup S^{no}$.
  - $p'_0(s^+) = q_s p_0(s)$, $p'_0(s^-) = (1 - q_s)p_0(s)$
- Transitions:
  - $p'(s_1, s_2) = p(s_1, s_2)$ si $s_1, s_2 \in S^{yes} \cup S^{no}$
  - $p'(s_1, s_2^+) = p(s_1, s_2)q_{s_2}$, $p'(s_1, s_2^-) = p(s_1, s_2)(1 - q_{s_2})$ si $s_1 \in S^{yes}$, $s_2 \in S^?$
  - $p'(s_1^+, s_2) = p(s_1, s_2)/q_{s_1}$ si $s_1 \in S^?$, $s_2 \in S^{yes}$
  - $p'(s_1^-, s_2) = p(s_1, s_2)/(1 - q_{s_1})$ si $s_1 \in S^?$, $s_2 \in S^{no}$
  - $p'(s_1^+, s_2^+) = p(s_1, s_2)q_{s_2}/q_{s_1}$,
    $p'(s_1^-, s_2^-) = p(s_1, s_2)(1 - q_{s_2})/(1 - q_{s_1})$ si $s_1, s_2 \in S^?$.
  - (autres probabilités à 0).

Propriété fondamentale

- Propriété:

$$P_M(L(\phi)) = P_{M^\zeta}(L(\phi^\zeta)).$$

- Il y a une transformation similaire pour le cas d'une sous-formule $\circ\phi_1$.

Conséquence

- Algorithme global:
  - Tant que ϕ a des sous-formules $\circ\phi_1$ ou $\phi_1 U \phi_2$, on applique la transformation correspondante.
  - On se ramène alors à un problème sur une chaine de Markov M_n avec une formule ϕ_n combinaison booléenne de propositions atomiques.
  - $P_{M_n}(L(\phi_n))$ est alors donné par la somme des probabilitiés intiales des états satisfaisant ϕ_n.

PCTL

- $PCTL = CTL+$ quantifications sur les probabilités.
- ($PCTL^* = CTL^*+$ quantifications sur les probabilités existe aussi).

- Exemple:
  - $try_1 \rightarrow P_{\geq 1}[trueUcrit_1]$: avec probabilité 1, une tentative d'entrée en session critique réussit.
  - $P_{<0.5}(\neg(crit_2 \vee crit_3)Ucrit_1))$: la probabilité que 1 rentre en session critique avant 2 et 3 est inférieure à 1/2.

$PCTL$ pour DTMC

- Syntaxe:

$$\phi ::= p|\phi_1 \wedge \phi_2|\neg\phi_1|P_{\approx p}(\circ\phi_1)|P_{\approx p}(\phi_1 U \phi_2)$$

avec $\approx \in \{\leq, <, \geq, >\}$, $p \in [0, 1]$.

- Sémantique pour DTMC:

$s \models P_{\approx p}(\psi)$ est vrai en un état s si $P(L_M(\psi)) \approx p$.

$PCTL*$ pour DTMC

- Formules d'états: $p|f_1 \wedge f_2|\neg f_1|P_{\approx p}(g)$, avec $\approx \in \{\leq, <, \geq, >\}$, $p \in [0,1]$, f_1, f_2 formule d'états, g formule de chemin.
- Formules de chemins: $f_1|\neg g_1|g_1 \vee g_2|\circ g_1|g_1 U g_2$ avec f_1 formule d'états, g_1, g_2 formules de chemins.

Algorithme pour $PCTL^*$

- Vraiment similaire à celui pour LTL: pour tester si $s \models P_{[b_1,b_2]}(\psi)$, on calcule q la probabilitié de satisfaire ψ, puis on teste si $b_1 \leq q \leq b_2$.
- Plus précisément, pour une formule innermost $\phi_1 = \exists_{[b_1,b_2]}(\psi)$, avec ψ dans LTL.
  - on calcule les q_s, $s \in S$, comme dans l'algorithme précédent.
  - On ajoute une nouvelle propsition atomique ζ_1 vraie en chaque état s tel que $b_1 \leq q_s \leq b_2$.
  - On transforme la formule ϕ en remplacant ϕ_1 par ζ_1.
  - On continue ainsi jusqu'à ce que la formule devienne une formule propositionnelle, qui peut alors s'évaluer sur M.

Autre vision. Cas de $PCTL$

- On note $S = \{1, 2, \cdots, n\}$.
- On appelle $Sat(\phi)$ l'ensemble des états satisfaisant ϕ.
- Une DTMC correspond à une matrice P.
- A $Sat(\phi)$ on associe le vecteur colonne b^{ϕ}, dont la $i$ème entrée vaut 1 si l'état i satisfait ϕ, 0 sinon.

- On peut écrire $Sat(P_{\approx p}(\circ\phi_1)) = \{s \in S | x_s \approx p\}$ où le vecteur $x = (x_1, \ldots, x_n)$ est donné par

$$x = Pb^{\phi_1}$$

- On peut écrire $Sat(P_{\approx p}(\phi_1 U \phi_2)) = \{s \in S | x_s \approx p\}$ où le vecteur $x = (x_1, \ldots, x_n)$ est donné par le système

$$\begin{array}{lcll} x_s & = & 1 & \text{if } s \in S^{yes} \\ x_s & = & 0 & \text{if } s \in S^{no} \\ x_s & = & \sum_{s'} p(s,s') x_{s'} & \text{if } s \in S^{?} \end{array}$$

- L'algorithme fonctionne alors par induction structurelle sur la formule.

$PCTL$ pour CMC

- Syntaxe (identique)

$$\phi ::= p|\phi_1 \wedge \phi_2|\neg\phi_1|P_{\approx p}(\circ\phi_1)|P_{\approx p}(\phi_1 U \phi_2)$$

avec $\approx \in \{\leq, <, \geq, >\}$, $p \in [0, 1]$.

- Sémantique pour CMC:
  - On fixe un ensemble A d'ordonnanceur.
  - $s \models P_{\approx p}(\psi)$ est vrai en un état s si $P(L_{M,U}(\psi)) \approx p$ pour tout $u \in A$.

Algorithme

- Similaire à $PCTL$ pour DTMC, sauf que l'on calcule q_s^{min} et q_s^{max} au lieu de seulement q_s.
- Exemple: $\approx \in \{\geq, >\}$ correspond à calculer q_s^{min} ($\approx \in \{\leq, <\}$ correspond à calculer q_s^{max} de façon complètement duale)

- $Sat(P_{\approx p}(\circ\phi_1)) = \{s \in S | q_s^{min}(\circ\phi_1) \approx p\}$ où

$$q_s^{min}(\circ\phi_1) = \min_{\mu \in Steps(s)} \sum_{s' \in Sat(\phi_1)} \mu(s')$$

- $Sat(P_{\approx p}(\phi_1 U \phi_2)) = \{s \in S | q_s^{min}(\phi_1 U \phi_2) \approx p\}$ où

$$\begin{array}{lcll} q_s^{min} & = & 1 & \text{if } s \in S^{yes} \\ q_s^{min} & = & 0 & \text{if } s \in S^{no} \\ q_s^{min} & = & \min_{\mu \in Steps(s)} (\sum_{s'} \mu(s') q_{s'})^{min} & \text{if } s \in S^{?} \end{array}$$

- Avant on devait résoudre un système linéaire, maintenant on a un problème d'optimisation linéaire.

Compléments

- $PCTL$ n'admet que des opérateurs booléens sur les formules d'états.
- $PCTL^*$ plus expressive, mais sa vérification nécessite des transformations couteuses pour stocker des informations sur l'historique.

Chaines de Markov a Temps Continu

- On aurait aussi pu parler de Chaines de Markov a Temps Continu
- Et de la logique CSL, permetant de parler des états stationnaires (long run behaviours).

- Il y a aussi des automates temporisés probabilistes.

Exemple de Système

- PRISM: outil logiciel pour la vérification de systèmes probabilistes.
  - Entrées:
    - DTMC,
    - MDP= CMC,
    - CTMC
  - Propriétés:
    - PCTL
    - CSL