# Cryptographie asymétrique

19 septembre 2023

## 1 CTL

syntaxe:

- $E$ il existe un chemin..
- $A$ pour tout chemin..
- $\mu$ jusqu'à..
- $X$ a la prochaine étape..
- $F$ chemin ou a un moment..
- $G$ chemin ou on a toujours..
- $\wedge$, &.

structure de Kripke:

**Definition 1.0.1.** $\S = (S, s_o, \rightarrow, l)$, ou $S$ est un ensemble fini d'état, $\rightarrow \subset S \times S$, $l : S \rightarrow 2^{Al}$: associe un ens de prop atomiques à tout état de $\S$.

*remarque* 1.1. Plusieurs anomalies, sans successeurs $AX\phi \equiv T$. Si $\rightarrow$ est "totale", $AX\phi = \neg tX\neg\phi$ sinon $\neg EX\phi = AX\neg\phi$.

## 1.2 model-checking

Question, étant donné une structure de Kripke § et une formule $\phi$. Est-ce qu'il existe un algorithme qui renvoie $\S, s_0 \vDash \phi$. Oui ce qu'on fait c'est qu'on découpe la formule en sous formule puis récursion, et on vérifie les formules atomiques. On marque chaque sous formules puis on monte petit à petit.

**Algorithme, Cas $\phi = A\phi_1\mu\phi_2$:**

- Marquage($\phi_1$)
- Marquage($\phi_2$)
- Pour tout $s \in S$:
  - $s.\phi := false$
  - $s.nbsucc := deg(s)$(on est sur un graphe)
  - si $s.\phi_2 = T$ alors $L = L \cup \{s\}$
- Tant que $L \neq \emptyset$:
  - Piocher $s$ dans $L$
  - $s.\phi := T$
  - Pour tout $s' \to s$:
    - $s'.nbsucc- = 1$
    - si $s'.nbsucc = 0s'.\phi_1 = Ts'.\phi_2 \neq T$: $L := L \cup \{s'\}$

**Proposition 1.2.1.** Décider si $\phi \in CTL$ est vraie pour § se fait en temps $\mathcal{O}(|\phi||\S|)$, $(|\S| = |S| + | \to |)$. (polynomial)

Le model checking de LTL est un pb PSPACE-complet $(2^{|\phi|}|\S|)$.

*remarque* 1.3. $A\phi_1\mu\phi_2 \equiv AF\phi_2\neg E(\neg\phi_2)\mu(\neg\phi_1\neg\phi_2)$ veut simplement dire, on peut pas atteindre $\phi_2$ en croisant un état ou on a ni $\phi_1$ ni $\phi_2$.

# 2 PCTL

**Definition 2.0.1** (Discrete Time Markov Chain)**.** Une chaine de Markov: $M = (S, P, s_{init}, l)$ consiste en, $S$ un ensemble d'états (dénombrable), $s_{init}$ l'état de départ, $P : S \times S \to [0, 1]$ une matrice de probabilités, $l : S \to 2^{Al}$ l'étiquetage des états des props atomiques.

Si $M$ est finie (i.e. $S$ est fini), $|M| = |S| + \{(s, s')|P(s, s') > 0\}$.

**Definition 2.0.2.** Une chaine de Markov $M$ induit une structure de Kripke $K_M = (S, s_{init}, \rightarrow, l)$ par $(s, s') \in \rightarrow \Leftrightarrow P(s, s') > 0$.

...defs a rajouter

## 2.1 Probabilités

**Definition 2.1.1** (Tribu, $\sigma$-algèbre sur $@W$)**.** Ensemble de partie stable par complémentaire, union dénombrable et contenant le vide.

**Definition 2.1.2** (mesure de Probabilité)**.** Mesure $\mu$ tq $\mu(@W) = 1$.

**Definition 2.1.3.** On définit $Path^F(M)$ les chemins finis.

Soit $M = (S, s_0, P, l)$ une chaine de Markov. Soit $\pi_0$ un prefixe de $\pi \in Path(M)$.

**Definition 2.1.4.** $Cyl(\pi_0) := \{\text{Chemins tq}\pi_0\text{en est un prefixe}\}$.

Pour nous, $@W$ est l'ens des chemins et Å la tribu des cylindres de $M$.

**Definition 2.1.5.** La mesure de probabilité sur Å est déf par la proba sur le préfixe.(produit des transitions)

## 2.2 Propriétés d'accessibilité

$M$ une chaine de Markov et $A, B \subset S$ des ensembles d'états.

- 3 propriétés d'accessibilités:
  - $FB = \{\text{chemin qui croise eventuellement B}\}$
  - $A\mu B = \{\text{chemin dans A jusqu'a croiser B, + croise B eventuellement}\}$
  - $GFB = \{\text{croise B une infinité de fois}\}$.

Etant donné $\phi$ d'un des types décrits avant.

$$P(s \vDash \phi) = P(\{\pi \in Path(M, s)|\pi \vDash \phi\})$$

Faut vérifier que c'est mesurable:

- Pour $FB$ on prend l'union dénombrable des chemins ayant leur bout dans $B$.
- Pour $A\mu B$, pareil que $FB$ mais ou le chemin est d'abord dans $A$.
- Pour $GFB$ on prend l'intersection de $FB$ et $A\mu B$:

$$\cap_n \cup_{m \geq n} \cup_{s_n \in B} Cyl(s_0 \dots s_n)$$

## 2.3 Propriétés d'accessibilité

Pour $s \in S$, **on déf** $x_s = P(s \vDash FB)$:

- $s \in B$, $x_s = 1$.
- $s \nvDash EFB$, alors $x_s = 0$. (exprimable en CTL)
- Pour les autres $s \in S_? := \{s \in S | s \notin B \wedge s \vDash EFB\}$:

$$x_s = \sum_{t \in B} p(s,t) + \sum_{t \in S_?} p(s,t) x_t$$

Si $\overline{x} = (x_s)_{s \in S_?} \to \overline{x} = \overline{b} + M\overline{x}$. $(M = (p(s,t))_{s,t})$

**On déf aussi** $x_s = Pr(s \vDash A\mu B)$:

- $s \in B$, $x_s = 1$.(noté $S_{=1} \subseteq \{Pr(s \vDash A\mu B) = 1\}$, pas d'égalité)
- $s \nvDash\to E(A\mu B)$ (il existe un etat qui atteint B en restand dans A), alors $x_s = 0$. (noté $S_{=0} := \{s \in S | Pr(s \vDash A\mu B) = 0\}$, egalité ici, permet de pas considérer les probas)
- $S_? = S - (S_{=0} \cup S_{=1})$

Soit $\overline{x} = (x_s)_{s \in S_?}$.

**Proposition 2.3.1.** $\overline{x}$ est la solution du système d'équations $\overline{y} = M\overline{y} + \overline{b}$ avec $M$ carrée. ($\overline{b} = (b_s)_{s \in S_?}$ et $b_s = \sum_{t \in B} p(s,t)$)

(**On résoud $M\overline{x} = \overline{x}$, clair+unicité.**)

On peut aussi caractériser par points fixes. On regarde:

$$\Gamma : [0,1]^{S_?} \to [0,1]^{S_?}$$

$$\Gamma(\overline{y} = M\overline{y} + \overline{b})$$

alors $\overline{x} = (x_s)$ avec $x_s = Pr(s \vDash A\mu B)$ est le plus petit point fixe de $\Gamma$. On a

$$\Gamma^n(x_s) = Pr(s \vDash A\mu^{\leq n} B)$$

avec $s \vDash EA\mu^{\leq n}B \equiv$ il existe un chemin depuis $s$, $\pi$, tq $\exists i \leq n$, $\pi(i) \in B$ et $\forall 0 \leq j < i$, $\pi(j) \in A$. En gros on arrive dans $B$ avant $n$ étapes. Si on pose

$$x_s^{(n)} = Pr(s \vDash A\mu^{\leq n}S_{=1})$$

et on a

$$\overline{x}^{(0)} \leq \ldots \leq \overline{x}^{(i)} \leq \ldots \leq \overline{x}$$

(pour $x \leq y$ si $\forall i,\ x_i \leq y_i$) On prouve

$$x_s^{(n)} = Pr(s \vDash A\mu^{\leq n}S_{=1})$$

- récurrence: $x_s^{(n+1)} = \sum_{(s,t) \in S_?} p(s,t)x_t^{(n)} + \sum_{t \in S_{=1}} p(s,t)$
- le premier terme est en degré $n$ et l'autre 1.

Et on prouve $\overline{x}$ est un point fixe, et le plus petit.

- $x_s = \sum_{t \in S_{=0}} p(s,t)x_t + \sum_{t \in S_{=1}} p(s,t)x_t + \sum_{t \in S_?} p(s,t)x_t$

**Enfin on def** $x_s = Pr(s \vDash GFB)$

**Definition 2.3.2.** Un élt $F$ est dit presque sur sous l'hyp d'un evt $D$ ssi $Pr(D) = Pr(D \cap F)$

**Propriété GF:** Pour une chaine de Markov $M$ (possiblement infinie) et $s, t \in S$, alors on :

$$Pr(s \vDash GFt) = Pr(s \vDash \bigwedge_{\pi \in Path^F(t)} GF\pi)$$

(pour tout $\pi$ préfixe fini partant de $t$.)

**Preuve:** $\pi = ts_1 \ldots s_n$ et on note $p = \prod_i Pr(s_i, s_{i+1})$. On montre les proba

- $GFt \wedge G\neg\pi$ nulle.
- $GFt \wedge FG\neg\pi$ nulle

On déf $E_n(\pi) =$"on visite au moins $n$ fois $t$ et pas $\pi$ avant au moins $n$ étapes. On a

$$Pr(E_n(\pi)) \leq (1-p)^n$$

On pose $E(\pi) = \bigcap E_n(\pi)$, on croise jamais $\pi$. On a $E_{n+1}(\pi) \subseteq E_n(\pi)$ d'ou

$$Pr(E_(\pi)) = \lim_{n \to \infty} Pr(E_n(\pi)) \leq \lim_{n \to \infty} (1-p)^n = 0$$

On déf mtn $F_n(_pi) = GFt \wedge X^n \neg F\pi$ puis $F(\pi) = \bigcup F_n(\pi)$, on a $F_n \subset F_{n+1}$ d'ou:

$$Pr(s \models F(\pi)) = \lim_{n\to\infty} Pr(s \models F_n(\pi))$$

Et on a en fait $Pr(s \models F_n(\pi)) = \sum_{s'\in S} Pr(s \models X^n s') Pr(s' \models E(\pi)) = 0$. Enfin

$$F := \bigcup_\pi F(\pi)$$

et

$$Pr(s \models F) \leq Pr(\sum_\pi F(\pi)) = 0$$

d'ou

$$Pr(s \models GFt) = Pr(s \models GFt \wedge \bigwedge_\pi GF\pi) + Pr(s \models GFt \wedge \bigwedge \vee_\pi FG\neg\pi)$$

et le deuxième terme vaut 0. □

Autrement dit on visite infiniment souvent $t$ si et seulement si on visite tout les préfixes finis sortant de $t$ infiniment souvent.

**Definition 2.3.3.** $CFC(M)$ les composantes fortement connexes (i.e. digraphe ou on peut accéder achaque point de chaque point.).

**Definition 2.3.4.** Une cfc est terminale si $Post^*(C) \subseteq C$ i.e. pas de chemin sortant. On appelle $CFCT$ l'ens.

On note $inf(\pi)$ les états de $\pi$ qui apparaissent infiniment.

**Proposition 2.3.5.** Si $M$ est une chaine finie. Alors

$$Pr(\{\pi / \inf(\pi) \in CFCT(M)\}) = 1$$

**Preuve:**$I(C) := \{\pi / \inf(\pi) \in C\}$,

$$\sum_{C\in CFC(M)} Pr(I(C)) = 1$$

Soit $C \in CFC(M)$ tq $Pr(I(C)) > 0$ et $t \in \inf(\pi)$. On a $Pr(s \models GFt) > 0$ d'ou $\forall \pi \in Path^F(t)$, $Pr(passerpar\pi) > 0$(en fait 1). Si $C$ n'est pas terminale on peut en sortir, contradictoire avec $\inf(t) = C$. Tout les $\pi$ doivent rester dans $C$ $\rightarrow$terminale. □

**Corollaire 2.3.6.** *Si $M$ est une CM finie:*

$$Pr(s \vDash GFt) = \begin{cases} 0 & t \notin C \subset CFCT(M) \\ Pr(s \vDash FC) & sinon \end{cases}$$

Objectifs: On suppose que M est finie.
Calculer $S_{\sim\alpha}(c\mu B)$, états vérifiant $c\mu b$. On a

$$\sim \in \{=, <, >, \leq, \geq\}$$

$$\alpha \in \{0, 1\}$$

$\rightarrow S_{=0}(c\mu B)$...
$\rightarrow S_{=1}(c\mu B)$. $(c, B \subseteq S,\ M(S, P, s_0, l))$

On construit de chaine de Markov $M'$ a partir de $M$ ou les états de $B \cup (S \backslash C)$. Sont absorbantes. (i.e. bouclent sur eux meme avec proba 1)
$M' = (S, P', s_0, l)$, avec :

$$P'(s,t) = \begin{cases} 1 & si\ t = s\ et\ s \in B \cup S \backslash C \\ 0 & si\ t \neq s\ et\ s \in B \cup S \backslash C \\ P'(s,t) & sinon \end{cases}$$

Pour les états de $B \cup S \backslash C$ on connait leur proba de vérifier $c\mu B$.

$$B \rightarrow 1$$

$$S \backslash (C \cup B) \rightarrow 0$$

On a

$$Pr^M(s \vDash c\mu B) = Pr^{M'}(s \vDash FB)$$

ET

$$Pr^M(s \vDash c\mu B) = Pr^{M'}(s \vDash FB) = 1\ si\ s \in B$$
$$Pr^M(s \vDash c\mu B) = Pr^{M'}(s \vDash FB) = 0\ si\ s \in S \backslash (C \cup B)$$

cas général ? Le pb est désormais de calculer

$$S_{=1}(FB)$$

i.e. $\{s | Pr(s \vDash FB) = 1\}$.

On a l'**équivalence** suivante:

1. $Pr(s \models FB) = 1$
2. $Post^*(t) \cap B \neq \emptyset$, $\forall t \in Post^*(s)$.
3. $s \in S \backslash Pre^*(S \backslash Pre^*(B))$.

**Preuve:** 1. $\implies$ 2. est clair. 2. $\implies$ 1. Une execution depuis $s$ finit avec proba 1 dans une CFCT. Celles ci étant de deux types.

1. Singleton dans B.
2. Cycle d'états dont aucun est dans B.

(faut se rappeler que $Post^*(C) = C$) Pour tout état d'une CFCT, on a $t \in Post^*(C)$ donc $Post^*(t) \cap B \neq 0$. Donc on peut pas avoir une CFCT comme 2. donc la proba d'avoir $G\neg B$ est nulle. 2. $\equiv$ 3.

$$Post^*(t) \cap B \neq \emptyset \quad \forall t \in Post^*(s)$$

$$\Leftrightarrow Post^*(s) \subseteq Pre^*(B)$$
$$\Leftrightarrow Post^*(s) \cap S \backslash Pre^*(B) = \emptyset$$
$$\Leftrightarrow s \notin Pre^*(S \backslash Pre^*(B))$$
$$\Leftrightarrow s \in S \backslash Pre^*(S \backslash Pre^*(B))$$

□

**Corollaire 2.3.7.** *Pour calculer $S_{=1}(c\mu B)$ on construit $M'$ puis on calcule $S \backslash Pre^*(S \backslash Pre^*(B))$. Temps linéaire en $|M|$.*

Maintenant pour l'accessibilité répétée ? $GFB$?
On a:

1. $Pr(s \models GFB) = 1$
2. $C \cap B \neq \emptyset$ pour toute CFCT $C$ atteignable depuis $s$.
3. $s \models AG\ EF\ B$ (CTL).

Pour tous ces ensembles $S_{=1,0}$ on a pas utilisé la valeur de la proba, juste $> 0$! Y s'avére que c'est vrai uniquement parce qu'on regarde des chaines de Markov finies.

Vérifier les props **qualitatives** peut nécessiter de regarder la valeur réelle.

# 3 PCTL, 2

**Syntaxe**:

- $\phi_1, \phi_2 := T|a|\phi_1 \wedge \phi_2|\neg\phi_1 P_J(\phi_l)$, avec $a \in AP$ et $J \subseteq [0,1]$, aux bornes rationelles.
- $\phi_l := X\phi_1|\phi_1\mu\phi_2|\phi_1\mu^{\leq n}\phi_2$

Ou aussi : $X, F = T\mu\phi$. On utilisera l'opérateur $G$. En pratique on se limite à $J = [0,1], [0,p[, [p;1], ]p;1]$. (P est pas une probabilité.)

- Pour $G$: $P_{\leq\alpha}(G\phi) = P_{\geq 1-\alpha}(F\neg\phi)$
- $G^{\leq n}\phi = \phi$ est vraie pour les $n+1$ premier états.

$$P_{\leq\alpha}(G^{\leq n}\phi) = P_{\geq 1-\alpha}(F^{\leq n}\neg\phi)$$

**Sémantique**:

- $M = (S, s_0, P, l)$ une chaine de Markov.
- $s \vDash T$ toujours
- $s \vDash e$ ssi $e \in l(s)$
- $s \vDash \phi_1 \wedge \phi_2$ ssi $(s \vDash \phi_1\ et\ s \vDash \phi_2)$
- $s \vDash \neg\phi_1$ ssi $s \nvDash \phi_1$
- $s \vDash P_J(\phi_l)$ ssi $Pr(s \vDash \phi_l) \in J$ i.e. $Pr\{\phi \in Path(s)|\pi \vDash \phi_l\}$.
- $\pi \vDash X\phi_1$ ssi $\pi(1) \vDash \phi_1$
- $\pi \vDash \phi_1\mu\phi_2$ ssi $\exists i \geq 0\ (\pi(i) \vDash \phi_2 \wedge \forall 0 \leq j < i, \pi(j) \vDash \phi_1)$
- $\pi \vDash \phi_2\mu^{\leq n}\phi_2$ ssi $\exists 0 \leq i \leq n,\ \pi(i) \vDash \phi_2 \wedge (\forall 0 \leq j < i,\ \pi(j) \vDash \phi_1)$

  Equivalence de formules: $\forall M,\ \forall s,\ M, s \vDash \phi_1 \Leftrightarrow M, s \vDash \phi_2$.

**Proposition 3.0.1.** $\alpha \in [0,1]$, $P_{<\alpha}(\phi) \equiv \neg P_{\geq\alpha}(\phi)$

**Model Checking**: Pour $M$ finie, $M \vDash \phi$. On fait comme pour CTL, on vérifie fait un récursion sur les sous formules.

- Changements:
  - $P_{\sim\alpha}(X\phi)$
  - $P_{\sim\alpha}(\phi_1\mu\phi 2)$
  - $P_{\sim\alpha(\phi_1\mu^{\leq n}\phi_2)}$

On déf $Sat(\phi)$ les états de $M$ qui vérifient $\phi$.

- Calcul de: $Sat(P_{\sim\alpha}(X\phi))$
- $Pr(s \models X\phi) = \sum_{s' \in Sat(\phi)} P(s, s')$
- Reste à comparer avec $\sim \alpha$
- Calcul de $Sat(P_{\sim\alpha}(\phi_1\mu\phi_2))$
- Calculer $Sat(\phi_1, 2)$
- Construire $M'$ avc les états de $\neg\phi_1 \wedge \neg\phi_2$ et $\phi_2$
- Reste à calculer les probas d'atteindre $Sat(\phi_2)$ depuis tout état de $M'$. "$FSat(\phi_2)$"
- Calcul de $Sat(P_{\sim\alpha}(\phi_1\mu^{\leq n}\phi_2))$
- Calculer $Sat(\phi_1, 2)$
- Calculer $M'$ avec les états $\neg\phi_1 \wedge \neg\phi_2$ ou $\phi_2$ sont absorants(ou comme union, $Sat(\neg\phi_1 \wedge \neg\phi_2) \cup Sat(\phi_2)$).
- $M' = (S, s_0, P', l)$.
- Calculer $P' * P' \ldots * P'$ avec $n$ termes.

**Conclusion**: Le modèle checking est en temps $\mathcal{O}(poly(|M|).|\phi|.n_{max})$, avec $n_{max}$ le plus grand $n$ de $\mu^n$ apparaissant.

# 4 PCTL, 0/1

On considère que les intervalles: $< 1, > 0, = 1, = 0$. On veut comparer PCTL, 0/1 et CTL. Comme dans le td.

- En général, incomparables. (y'a des trucs qu'on peut faire dans l'un et pas dans l'autre)

- Dans le cas des chaines finies, $PCTL_{0/1} \subset CTL$!

Exemples simples:

$$P_{>0}(a\mu b) = Ea\mu b$$

$$P_{>0}X(a) = EXa$$

$$P_{=1}X(a) = AXa$$

Par contre

$$P_{=1}(a\mu b) = ?$$

ca dépend. Cas général(CM infinies):

1. Il n'existe pas d'équivalent CTL à $P_{=1}F(a)$ ($P_{>0}(Ga)$).
   On peut regarder $\mathbb{N}$ ou $x_n = p * x_{n+1} + (1-p)x_{n-1}$ et $x_0 = (1-p)x_0 + p * x_1$. La structure de Kripke (CTL) sous jacente ne depend pas de $p$ et si $p < 1/2, > 1/2$ on a un comportement différent:

$$P_{=1}F0$$

2. Il n'existe pas d'équivalent $PCTL_{0/1}$ à $AFa, EGa$.

   Dans les chaines de Markov finies, on a:

$$P_{=1}A(EFa)Wa$$

Où $W = Weak\ until$, s'exprime dans CTL (simplement on a pas $a$ sur tout le chemin comme $a\mu b$ je crois).

## 4.1 bisimulation, syst.classiques

- 2 structures de Kripke: $K_1 = (S_1, \rightarrow_1, s_0^1, l_1)$ $K_2 = (S_2, \rightarrow_2, s_0^2, l_2)$.
- Une relation $R \subset S_1 \times S_2$ est une bisimulation ssi $\forall (s_1, s_2) \in R$:
  1. $l_1(s_1) = l_2(s_2)$
  2. $\forall s_1 \rightarrow_1 s_1'$, $\exists s_2 \rightarrow_2 s_2'$ t.q $(s_1', s_2') \in R$
  3. $\forall s_2 \rightarrow_2 s_2'$, $\exists s_1 \rightarrow_1 s_1'$ t.q $(s_1', s_2') \in R$

  Mauvaise def:

**Definition 4.1.1** ($M_1 = (S_1, P_1, s_0^1, l_1)$,$M_2 = \dots$)**.** Une bisimulation probabiliste sur $M_1 \times M_2$ est une relation d'équivalence $R \subseteq S_1 \times S_2$ t.q pour tout $(s_1, s_2) \in R$, on a:

1. $l_1(s_1) = l_2(s_2)$
2. $P_1(s_1, T) = P_2(s_2, T)$

Bonne def:

**Definition 4.1.2** ($M = (S, P, s_0, l)$)**.** Une bisimulation probabiliste sur $M$ est une relation d'équivalence $R \subseteq S \times S$ t.q pour tout $(s_1, s_2) \in R$, on a:

1. $l(s_1) = l(s_2)$
2. $P(s_1, T) = P(s_2, T)$

**Théorème 4.1.3.** *$s_1 \sim s_2 \equiv s_1$ et $s_2$ vérifient les mêmes formules de PCTL!*

(Chaine quotient ! on prend $\overline{P} = \#classe * P$)

# 5 Processus de décision de Markov(MDP)

On a maintenant et des choix non déterministes et des probas. (Mélanges des deux d'avants).

**Definition 5.0.1.** Un MDP, $M = (S, s_0, Act, Pr, l)$:

- $Act$ = ensemble fini d'actions
- $Pr : S \times Act \times S \to [0,1]$ tq $\forall s \forall \alpha \in Act$,

$$\sum_{t \in S} \mathrm{Pr}(s, \alpha, t) \in \{0,1\}$$

**Notation:**

- Une action $\alpha$ est "tirable" ou possible depuis un état $s$ ssi $\sum_{t \in S} Pr(s, \alpha, t) = 1$
- $Act(s)$ = actions possibles depuis $s$
- $Supp(M) = \{s | Act(s) \neq \emptyset\}$
- Si $Pr(s, \alpha, t) > 0$ on dit que $t$ est un $\alpha$-successeur de $s$.

Comme pour les chaines de Markov, on définit la structure de Kripke $K_M$ avec $K_M = (S, s_0, \to, l)$ ou cette fois $s \to s'$ ssi $\exists \alpha \in Act(s)$ tq $Pr(S, \alpha, s') > 0$. De même on définit les chemins avec les actions.

($S^+$ l'ensemble des séquences finies de $S$) (Ici le Scheduler se souvient de la ou il passe pour déterminer les actions d'après)

**Definition 5.0.2.** Un scheduler pour $M$ est une fct $\sigma : S^+ \to Act$ tq $\sigma(s_0 \dots s_n) \to Act(s_n)$

**Proposition 5.0.3.** La donnée d'un MDP et d'un scheduler est une chaine de Markov! $M_\sigma = (S', s_0, P, l')$:

- $S' = S^+$
- $P(s', t) = Pr(s_n, \sigma(s'), t)$

-Les Scheduler sans mémoire:

**Definition 5.0.4.** $\sigma : S \to Act$,

-Les Scheduler à mémoire finie:

**Definition 5.0.5.** $\sigma = (Q, q_0, \Delta, act)$ :

- $\Delta : Q \times S \to Q$
- $act : Q \times S \to Act$ tq

  $act(q, s) =$ Donne l'action à jouer depuis l'état s de la MDP lorsque la mémoire est dans le mod

$$\Delta(q, s) = \text{Le prochain état de la mémoire}$$

-Les Scheduler randomisés: Cette fois on va de $S^+ -> Dist(Act)$ On associe des probabilités d'avoir une action

## 5.1

Approche alternative pour l'analyse des MDP.

- Objectif: obtenir un algo qui permet d'avoir un intervalle de proba.
- Idée: On va remplacer le MDP $M$ par un MDP $M^{min}$ (ou $M^{max}$) ayant une structure particuliere qui permet le calcul des intervalles de probas (et qui conservent les probas min/max).

Décomposition d'un MDP en MEC (Maximal End Component).

**Definition 5.1.1** (End Component)**.** $M = (S, s_0, Act, P, l)$ une MDP,

1. une sous-MDP de $M$ est défini par un $S' \subseteq S$ et $Act'$ t.q $S' \neq \emptyset$ $(\forall s \in S',\ Act'(s) \subseteq Act(s))$
2. $(S', Act')$ est un End Component ssi $\forall s \in S', \forall \alpha \in Act'(s)$ on a $\forall t \in S, P(s, \alpha, t) > 0 \implies t \in S'$.

- On autorise les EC de la forme $(\{s\}, \emptyset)$.
- Etant données deux EC $(S_1, Act_1)$, $(S_2, Act_2)$, d'une MDP $M$:
  1. $(S_1, Act_1) \leq (S_2, Act_2)$ si $S_1 \subseteq S_2$ et $\forall s \in S_1,\ Act_1(s) \subseteq Act_2(s)$
  2. Si $S_1 \cap S_2 \neq \emptyset$, alors
  $$(S_1 \cup S_2, Act_1 \cup Act_2)$$
  est un EC. (y'a de la transitivité)

- On a une notion de Maximal EC (1.).
- deux MEC ne partagent pas d'état (2.). Cela définit une partition de l'ensemble $S$.

On distingue 3 types de MEC:

1. Les TMEC: Trivial MEC$\rightarrow(\{s\}, \emptyset)$.
2. Les BMEC: BottomMEC/TerminalMEC$\rightarrow(S', Act')$ t.q $Act'$ est maximal (donne même actions que $Act$)
3. Les autres.

Le problème qu'on veut: $Fs$? On le résout depuis chaque MEC facilement. On a une partition de $S$ données par les $BMEC, TMEC, autres$(Leurs états).

**Algorithme pour déterminer les MEC de $M$:**

- input: $M$.
- output: ensemble des MEC de $M$.

On fait :

- $P = Pile$
- $P.push((S, Act))$ (init avec M)
- $S_{MEC} = \emptyset$
- Tant que $P \neq \emptyset$:
  - $(S', Act') := P.pop()$
  - Pour $s \in S'$ et $\alpha \in Act'(s)$:
    * Pour $t \in S$ t.q. $P(s, \alpha, t) > 0$:
    * si $t \notin S'$ alors $Act'(s) = Act'(s) \backslash \{\alpha\}$.
  - Calculer les $CFC$ du graphe $(S', Act') \rightarrow (S_1, \ldots, S_k)$.
  - Si $k > 1$ alors:
    * Pour $i = 1$ à $k$:
    * $P.push((S_i, Act'))$
  - Sinon: $SMEC += \{(S', Act')\}$

- Retourne $SMEC$

Pour la MDP de quiver:

- $(S, Act)$
- nettoyage de $Act$, ok.
- $CFC$ du graphe $\{s, s'\}$, $\{u, v\}, \{s+\}, \{t\}$
- $P:\ (\{s, s'\}, Act), (\{u, v\}, Act), (\{s+\}, Act), (\{t\}, Act)$
- $(\{s, s'\}, Act) \to$ supprimer $\beta$ depuis $S'$, $\to$ MEC= $(\{s, s'\}, Act')$

## 5.2 Décomposition pour MIN

**"Min-Réduction"**

(Apparemment probabilité nulle sur les états des BMEC, S_k, ducoup on les fusionne en $s-$)

**Definition 5.2.1.** $M = (S, s_0, Act, P, l)$ et $S = \bigcup_{m=0}^{M} B_m \cup \bigcup_{l=1}^{L}\{t_l\} \cup \bigcup_{k=1}^{K} S_k$:

$$\tilde{S} = \{S+, t_1, \ldots, t_L, s-\}$$

$$\tilde{Act}(s+) = \{loop\} = \tilde{Act}(s-)$$

$$\tilde{Act}(t_i) = Act(t_i)$$

Pour les probas:

$$\tilde{P}(s+, loop, s+) = 1 = \tilde{P}(s-, loop, s-)$$

$$\forall \alpha \in Act(t_i):\ \ \tilde{P}(t_i, \alpha, t_j) = P(t_i, loop, t_j)$$

$$\forall \alpha \in Act(t_i):\ \ \tilde{P}(t_i, \alpha, s+) = P(t_i, loop, s+)$$

$$\tilde{P}(t_i, \alpha, s_i) = \sum_{k=1}^{K} P(t_i, \alpha, s_k) + \sum_{m=1}^{M} P(t_i, \alpha, B_m)$$

(Rappel: $P(t, \alpha, U) = \sum_{u \in U} P(t, \alpha, u)$) C'est la décomposition MIN de celle de tout à l'heure.

**Proposition 5.2.2.** $\forall M, \forall s \in S$, on a

$$Pr_M^{min}(s \models Fs+) = Pr_{M^{min}}^{min}(\tilde{s} \models Fs+)$$

**Preuve:** En gros on a un scheduler $\sigma$ qui réalise la proba min dans $M$. On en déduit un $\tilde{\sigma}$ qui lui correspond. Pour les états fusionnés, la proba min est 0 avec loop, c'est ok. Pour les autres, on joue la meme section dans $M$ et dans $M^{min}$(Le truc c'est que c'est inversible). □

Dans notre cas: $Pr^{min}(t \vDash Fs+) = 0.2$.
Un autre exemple: (En commentaire dans le tex)

Les MEC: $(\{s+\}, loop), (\{s_5\}, loop), (\{s_1\}, \emptyset), (\{s_2\}, \emptyset), (\{s_3, s_4\}, (s_3 \to \alpha, s_4 \to \alpha))$

On obtient:

$$x_0 = min(0, (x_0 + x_1)/2)$$

$$x_1 = (x_1 + x_2)/2 = 3/4$$

$$x_2 = 3/4$$

**Proposition 5.2.3.** Dans $M^{min}$, tout état de $\tilde{S} \backslash \{s+, s-\}$ est une $TMEC$ pour $M^{min}$.(et $s+$ et $s-$ sont des BMEC)

**Proposition 5.2.4.** Dans $M^{min}$, tout chemin "finit" avec proba 1 dans $s-$ ou $s+$. (On peut pas boucler dans les $t$!)

**Preuve:** $G_0 = \{s-, s+\}$,

$$G_{i+1} = \{s \in \tilde{S} \backslash \bigcup_{j \leq i} G_j | \forall \alpha \in \tilde{Act}(s), \exists s' \in \bigcup_{j \leq i} G_j \ t.q.\ P(s, \alpha, s') > 0\}$$

Les $G_i$ partitionnent $\tilde{S}$. Supp que $s \in \tilde{S} - \bigcup G_i = P$. Pour tout $s \in P$, il existe $\alpha \in Act(s)$, tout les états tq $P(s, \alpha, t) > 0$ sont dans $P$. $P$ définit une sous MDP de $M$. (Bon la preuve est reloue)

**Proposition 5.2.5.**

## 5.3 Décomposition pour MAX

**Definition 5.3.1.** $M = (S \ldots)$, $S = \bigcup_{m=0}^{M} B_m \cup \bigcup_{l=1}^{L} \{t_l\} \cup \bigcup_{k=1}^{K} S_k$. La "max-réduction" $M^{max}$ de $M$ est un MDP=$\bar{M}$:(On veut garantir l'unicité du point fixe)

- $\bar{S} = \{s-, s+, t_1, \ldots, t_K, s_1, \ldots, s_K\}$
- $\bar{Act}(s-) = Act(s+) = \{loop\}$

- $\bar{Act}(t_i)$, $\forall 1 \leq k \leq K$,

$$\bar{Act}(s_k) = \{\alpha | \exists s \in S_k \wedge \alpha \in Act(s) \wedge \exists t \in S - S_k \; tq \; Pr(s, \alpha, t) > 0\}$$

- $\bar{P}(t_i, \alpha, t_j) = P(t_i, \alpha, t_j)$
- Pareil que min pour $s+, s-, t_i$
- Pour $s \in S_k$ tq $\alpha \in Act(s)$: $\bar{P}(s_k, \alpha, s+) = P(s, \alpha, s+) \; pour \; s \in S_k \; tq \; \alpha \in Act(s)$
- $\bar{P}(s_k, \alpha, s-) = P(s, \alpha, \bigcup B_i)$
- Pour $\alpha \in Act(s)$, $s \in S_k$: $\bar{P}(s_k, \alpha, t_i) = P(s, \alpha, t_i)$
- Pour $\alpha \in Act(s)$, $s \in S_k$, $s' \in S_{k'}$: $\bar{P}(s_k, \alpha, s_{k'})$

Le $M^{max}$ obtenu pour la chaine de tout à l'heure: (en commentaire dans le tex)

**Proposition 5.3.2.** • Les probas max sont conservées en passant à $M^{max}$.

- Dans $M^{max}$, tout état de $\bar{S} - \{s+, s-\}$ sont des TMEC.

$$fmax : V \to V; \; fmax(\bar{x}) = (max_{\alpha \in Act(s)} \sum_{s \in \bar{S}} Pr(s, \alpha, t).x+)_{s \in \bar{S}}$$

**Proposition 5.3.3.** Le vecteur $P^{max}_{M^{max}}(Fs+)$ est l'unique point fixe de fmax! (pas vrai pour min?)

**Algo:**(min) On définit deux séquences $x$ (sous approx de $P^{min}$) et $y$ (sur approx de $P^{min}$). $\left\{ \begin{array}{l} x^{(0)} = (1 \; pour \; s+, 0 \; sinon) \\ y^{(0)} = (0 \; pour \; s-, 1 \; sinon) \end{array} \right.$ et $\left\{ \begin{array}{l} x^{(n+1)} = f_{min}(x^{(n)}) \\ y^{(0)} = f_{min}(y^{(n)}) \end{array} \right.$.

(max): $\left\{ \begin{array}{l} x^{(0)} = (1 \; pour \; s+, 0 \; sinon) \\ y^{(0)} = (0 \; pour \; s-, 1 \; sinon) \end{array} \right.$ et $\left\{ \begin{array}{l} x^{(n+1)} = f_{max}(x^{(n)}) \\ y^{(0)} = f_{max}(y^{(n)}) \end{array} \right.$.

**Proposition 5.3.4.** L'algo converge en au plus $n.[\log(\epsilon)/\log(1-\eta^n)]$. ($n$ le nombres des $G_i$ et $\eta$ la proba min $> 0$ dans $M$)