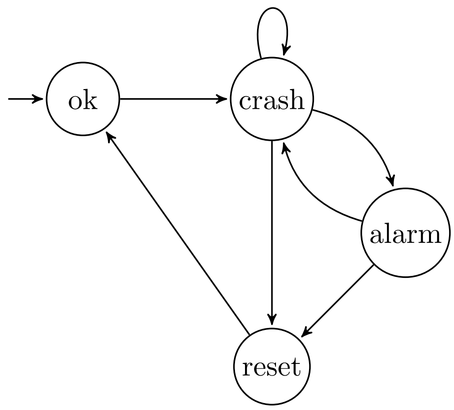
TD 4 : formules du passé en LTL

Rappel de la sémantique des modalités du passé : le mot infini $\sigma = s_0 s_1 s_2 \dots$ vérifie les formules en la position i selon la sémantique

$\sigma, i \models \mathsf{Y}\varphi$	ssi $i > 0$ et $\sigma, i-1 \models \varphi$	(yesterday, X^{-1})
$\sigma, i \models \mathsf{P}\varphi$	ssi $\exists k, 0 \leq k \leq i \;.\; \sigma, k \models \varphi$	(in the past, F^{-1})
$\sigma, i \models \varphi \,\mathsf{S}\, \psi$	ssi $\exists k, 0 \leq k \leq i \;.\; \sigma, k \models \psi$ et $\forall j, k < j \leq i \;.\; \sigma, j \models \varphi$	(since, U^{-1})

1 Spécification avec le passé

On considère un système d'alarme modélisé par l'automate suivant :



Donner des spécifications en LTL avec et sans passé des énoncés :

1. « Chaque fois que l'alarme sonne, il y a forcément eu un crash à l'instant précédent. »
2. « Chaque fois que l'alarme sonne, il y a forcément eu un crash auparavant, mais il n'y a pas eu de reset depuis le dernier crash. »

2 LTL avec ou sans passé

2.1 Variables d'histoire

Une technique d'élimination de formules qui ne contiennent que des modalités passées dans une spécification consiste à modifier le modèle et la spécification, en ajoutant des variables d'histoire au modèle et en remplaçant les formules du passé par des propositions portant sur ces variables. Par exemple, une formule $\mathsf{Y}\varphi (= \mathsf{F}^{-1}\varphi)$ sera remplacée par une variable $h_{\mathsf{Y}\varphi}$ dans la spécification, et le modèle se chargera de mettre à jour cette variable selon que φ soit vérifiée ou non dans l'état précédent.

1. En utilisant cette méthode, donner un modèle de système d'alarme et une spécification de bon fonctionnement qui traduit vos réponses à l'exercice 1.
2. Quel est le coût de cette manipulation du modèle ?

2.2 Taille des formules

On considère un ensemble de propositions atomiques $\mathrm{AP}_{n+1} = \{p_0, \ldots, p_n\} = \mathrm{AP}_n \cup \{p_n\}$ qui définit l'alphabet $\Sigma_{n+1} = 2^{\mathrm{AP}_{n+1}}$ et les mots infinis sur Σ_{n+1}^ω. Le but de l'exercice est d'établir qu'il existe une formule LTL avec passé de taille $O(n)$ dont la traduction en LTL est de taille $\Omega(2^n)$.

On considère la formule de LTL suivante :

$$\bigwedge_{S \subseteq \mathrm{AP}_n} \left(\begin{array}{l} (\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j \wedge p_n) \Rightarrow \mathsf{G}((\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j) \Rightarrow p_n) \\ \wedge(\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j \wedge \neg p_n) \Rightarrow \mathsf{G}((\bigwedge_{p_i \in S} p_i \wedge \bigwedge_{p_j \notin S} \neg p_j) \Rightarrow \neg p_n) \end{array} \right) \qquad (\varphi_n)$$

1. Quels mots de Σ_{n+1}^ω sont modèles de φ_n ?
2. Comment vérifier avec une formule LTL avec passé que la position courante est la position initiale du mot ? Donnez une formule LTL avec passé ψ_n de taille $O(n)$ (initialement) équivalente à φ_n.
3. Soit le langage $L_n = \{\sigma \in \Sigma_{n+1}^\omega \mid \sigma \models \mathsf{G}\varphi_n\}$. On souhaite montrer que tout automate de Büchi généralisé qui reconnaît L_n a au moins 2^{2^n} états. Pour cela on fixe une permutation $a_0 \ldots a_{2^n-1}$ des symboles de Σ_n et on considère différents sous-ensembles K de $\{0, \ldots, 2^n - 1\}$. Pour chaque K, on définit alors le mot $w_K = b_0 \ldots b_{2^n-1}$ de $\Sigma_{n+1}^{2^n}$ avec, pour tout i de 0 à $2^n - 1$, $b_i = a_i$ si $i \in K$ et $b_i = a_i \cup \{p_n\}$ sinon : K désigne l'ensemble des positions de w_K où p_n est vraie.

   En utilisant les mots w_K pour différents ensembles K, montrer que tout automate de Büchi généralisé pour $\mathsf{G}\varphi_n$ requiert au moins 2^{2^n} états.
4. En déduire une borne inférieure à la taille de toute formule LTL équivalente à φ_n.

3 Passé et automates

Soit φ une formule LTL *pure* du passé, qui n'utilise que S, P ou Y comme modalités. Comment construire un automate fini $\mathcal{A}$ qui reconnaisse

$$\{w = s_0 \ldots s_n \in \Sigma^* \mid w, n \models \varphi\}$$