

Nous allons utiliser pour ce TME le logiciel LoTREC, déjà vu lors du premier TME.

L'interface comporte un onglet *Connectors* (« connecteurs »), un onglet *Rules* (« règles »), un onglet *Strategies* (« stratégies »), et un onglet *Predefined Formulas* (« formules prédéfinies »). Les règles indiquent comment les tableaux sont construits (ré-écriture selon les connecteurs, comme les règles α et β , en tenant en plus compte des modalités ici). Les stratégies indiquent comment s'appliquent les règles (en particulier la priorité entre elles).

A l'ouverture, l'application vous demande avec quelle logique vous souhaitez travailler. Nous allons pour commencer définir notre propre logique : choisissez donc **New Logic**.

Echauffement

Nous allons dans un premier temps définir un modèle simple comportant deux mondes. Pour cela, cliquez l'onglet **Rules** et choisissez **Add** pour ajouter une nouvelle règle : cette règle va nous permettre de créer le modèle. Donnez un nom à votre règle, laissez la partie condition vide, puis cliquez sur **Add** dans la partie **Actions**. Un menu vous permet de choisir les actions à exécuter. Pour le moment nous aurons besoin de 3 types d'actions : **createNewNode** qui permet de créer un nouveau monde dans le modèle de Kripke ; **link** qui permet de lier deux mondes par une relation donnée ; et **add** qui permet d'ajouter des variables propositionnelles dont l'interprétation est vraie dans un monde donné.

```
createNewNode w1
createNewNode w2
link w1 w2 R
add w1 P
add w2 P
add w2 Q
```

Cette séquence permet de créer deux mondes w_1 et w_2 , de rendre w_2 accessible depuis w_1 , et de faire en sorte que p soit interprété à vrai dans le monde w_1 , tandis que p et q le sont dans le monde w_2 .

Pour rendre effective votre nouvelle règle, vous devrez ensuite modifier la stratégie de base d'exécution des règles. Pour cela, cliquez sur l'onglet **Strategies**, éditez la stratégie par défaut, et ajoutez votre règle. Vous pouvez ensuite cliquer sur **Build Premodels**, cela affichera votre modèle. Malheureusement votre logique ne possède pour le moment pas de connecteurs particuliers, et aucune autre règle.

Dans ce TME nous allons donc utiliser l'outil LoTREC et la logique minimale K (qui est prédéfinie), avec deux objectifs :

- vérifier si une formule Φ est satisfaite dans un monde w d'un modèle donné M (*model checking*), c'est-à-dire $M, w \models \Phi$
- tester la satisfiabilité, ou la validité, de certaines formules dans la logique K .

Exercice 1 Vérification de modèles dans la logique K

Cliquez sur l'onglet **Logic** puis choisissez **Predefined Logic** et enfin, parmi les différentes logiques prédéfinies, choisissez **Model-Checking mono**. La première règle s'intitule **ExampleOfModelAndFormula**. Nous allons éditer cette règle pour créer le modèle souhaité, comme vu précédemment. Nous ajouterons pour finir une action de la forme : **add w isItTrue ϕ** , où w est le monde à partir duquel vous souhaitez vérifier la formule ϕ .

Attention : Rappelez vous que les formules doivent être écrites en notation préfixe : ainsi, la formule $\Box b \wedge \Box a$ s'écrit **and nec B nec A**. Les différents connecteurs utilisables peuvent être consultés avec l'onglet **Connectors**.

La vérification de modèles s'opère dans LoTREC en deux phases :

- dans la phase descendante (*top-down*), la formule initiale est décomposée, pas à pas, jusqu'à parvenir à des éléments atomiques ;
- dans la phase montante (*bottom-up*), la valeur de vérité est "remontée" jusqu'à la formule initiale.

Pour chaque connecteur, il existe donc deux types de règles spécifiant la ré-écriture effectuée par LoTREC.

1. Reprenez le modèle de Kripke de l'ex. 1 du TD et vérifiez les différentes formules. Observez le résultat donné par le logiciel, et la manière dont le graphe correspondant au modèle est modifié par ré-écriture. Vous pourrez utiliser le mode pas à pas pour suivre plus facilement les différentes étapes de ré-écriture.
- 2* Créez un modèle de Kripke M_0 (en utilisant le moins de mondes possibles) où les deux contraintes suivantes sont vérifiées (conjointement) :
 - au moins un monde w où $M_0, w \models \neg p \wedge \Box \Diamond p$;
 - au moins un monde où $M_0, w \models q \wedge \Diamond^n \neg q$, où \Diamond^n représente un nombre quelconque de modalités \Diamond successives.
3. Le connecteur d'implication n'est pas défini par défaut dans la logique **Model-Checking mono**. Créez un tel opérateur, avec les règles associées. Pour vérifier son comportement, vérifiez la formule $P \rightarrow \Box(Q \vee P)$ dans le monde w du modèle de la question 1.

Exercice 2 *Satisfiabilité dans la logique K.*

On s'intéresse enfin à la satisfiabilité de différentes formules dans la logique K. Cliquez sur l'onglet **Logic** et choisissez maintenant **Predefined Logics** et **K-mono**.

1. Nous allons commencer par tester la satisfiabilité de la formule $\Diamond p \wedge \Box \neg p$. Pour cela, entrez la formule dans le cadre des formules et cliquez sur **Satisfiability Check**. Vous allez constater que le programme construit en partie droite un graphe. Que constatez-vous ? Essayez d'interpréter intuitivement ce graphe, en reprenant le raisonnement que vous avez suivi pour traiter cette même formule en TD. Pour vous aider, recommencez la même opération en utilisant le mode pas à pas (**Step by step**).
2. Essayez à présent les formules suivantes, en essayant d'anticiper les résultats du programme, puis d'interpréter les graphes obtenus.
 - (a) $p \wedge \Diamond(q \wedge \Box \neg p)$
 - (b) $(p \wedge \neg p) \vee \Diamond \Diamond \Diamond p$
 - (c) $(p \rightarrow \Diamond(q \vee \neg p)) \vee q$
- 3* Comment utiliser le programme pour montrer la validité de la formule $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$?