

Test de SIMPA

La [description](#) en est donnée en cours (chapitre 2, LBT).

Le logiciel à tester se trouve à l'URL suivante qui vous donne accès à un tar.gz de 23Mo:

<https://ligcloud.imag.fr/public.php?service=files&t=9e1f5a647020261eb0f101c05bf83b0f>

Chargez ce projet dans un espace partagé entre l'équipe, typiquement, une archive git. Cela fait environ 60Mo une fois décompressé, mais le plus gros est constitué par les jar pour s'interfacer à HtmlUnit et à SIP. Il vous est conseillé de mettre votre archive partagée sur le serveur depots.ensimag.fr, accessible par ssh, et de la mettre dans le répertoire /depots/2016.

Ce projet contient deux répertoires:

- simpa-clean, qui est le logiciel simpa lui-même
- simpa-test qui est une plate-forme ("test harness") qui permet de lancer des tests, enregistrer les résultats etc.

Pour les questions que vous vous posez sur ce sujet, deux supports vous sont proposés sur ce cours Chamilo:

- un forum
- un Wiki

N'hésitez pas à vous en servir pour échanger entre vous, ne serait-ce que pour documenter comment on utilise le logiciel, comment on peut le tester, à quoi servent tels ou tels paquetages etc. Les équipes ne sont pas en concurrence, et la contribution d'une équipe à l'intérêt général sera appréciée positivement par vos enseignants.

Les principaux articles scientifiques décrivant les algorithmes sont disponibles dans les Documents de ce cours, dans le dossier [Tp-test-SIMPA](#).

Si vous avez des questions plus spécifiques que vous ne souhaitez pas partager avec vos camarades, envoyez un e-mail à la fois à Roland.Groz@imag.fr et à Lingxiao.Wang@imag.fr.

Recommandation importante: commencez par tester si les algorithmes fonctionnent pour le cas Mealy. Vous n'aborderez les autres aspects (EFSM, Web, systèmes réels) qu'ensuite si vous en avez le temps.

Pour cela, créez des automates sous format .dot de GraphViz, et utilisez le driver transparent FromDotMealyDriver qui simule cet automate.

Par exemple, exécutez la commande:

```
main.simpa.SIMPA drivers.mealy.transparent.FromDotMealyDriver --loadDotFile -- --
```

Les algorithmes applicables au cas Mealy sont:

- `lm`
- `tree` (aussi appelé ZQ: pour les options, voire ZQ)
- `noReset`: dans le cas transparent (`fromDot`), pas besoin des paramètres `stateBound` et `characterizationSeq`, ils sont automatiquement calculés
- `rivestSchapire` (si vous avez le temps)

Sur des automates en `.dot`, comme ils sont connus, l'oracle de test consiste à vérifier qu'on a bien inféré le bon automate, ou un automate équivalent. Vous pouvez reprogrammer un test d'équivalence d'automates en `.dot`, ou utiliser la méthode `getShortestCounterExemple` disponible dans `MealyDriver.java`