Titre du sujet de TER :

Minimisation de systèmes de transitions par rapport à la bisimulation

Encadrantes :
Serenella Cerrito (serenella.cerrito@univ-evry.fr),
Sophie Paillocher (sophie.paillocher@hotmail.fr)
(équipe COSMO du laboratoire IBISC)

Description du sujet

But général du TER
Le but de ce TER est de définir et implémenter un algorithme qui minimise un système de transitions par rapport à la relation de bisimulation. De facto, cet algorithme sera très semblable à l'algorithme de minimisation pour les automates finis, sur des mots finis, étudié dans n'importe quel cours de théorie des langages formels.

Explications Préliminaires.
Un système de transitions *M*, dit aussi « *structure de Kripke* » est un graphe permettant de modéliser les exécutions d'un système informatique dynamique *S*, dont le comportement évolue selon le temps. Les sommets de *M* sont dits *états*, et on passe d'un état à un autre par le biais d'une *relation de transition*. Une structure de données de ce type est étudiée dans le cours de spécifications et vérifications formelles du second semestre du M1 SA enseigné par S. Cerrito. Chaque chemin dans *M* (à partir d'un état initial fixé, disons s_0), permet d'évaluer des formules de la logique LTL (*Linear Temporal Logic*) exprimant des propriétés du système *S* qui décrivent son comportement selon le passage du temps (par exemple : « *S* ne rentre jamais en panne »). La logique LTL aussi est étudiée dans le cours de S. Cerrito.

Plus précisément : dans le contexte de ce TER, un système de transitions *M* est un 5-plet $<P, S, I, T, L>$ où :

- *P* est un ensemble de variables booléennes
- *S* est un ensemble d'états,
- *I* est un sous-ensemble de S, dit « ensemble des états initiaux »
- *T*, la relation de transition, est un sous-ensemble de S×S,
- *L* (*labelling)* est une fonction d'étiquette, qui associe à chaque sommet *s* un sous-ensemble de *P* (intuitivement : l'ensemble des variables booléennes vraies à s).

On supposera aussi que *T* est totale, c'est à dire définie pour chaque état (chaque état a au moins un successeur, il n'y a pas de *deadlocks*).

L'idée intuitive de la bisimulation entre deux systèmes de transition M1 et M2 différents c'est que les deux, même si différents, et ayant même, éventuellement, des ensembles d'états différents, montrent, *de facto*, «le même comportement ». Une définition plus technique est donnée ci dessous, et *elle sera expliquée, par le biais d'exemples, dans les premiers RDV du TER.*

Bisimulation.

Soient $M1 = <P, S1, I1, T1, L1>$ et $M2 = <P, S2, I2, T2, L2>$ deux systèmes de transition (remarquer que P est le même pour M1 et *M2).*

Bisimulation entre états.

On appelle *bisimulation entre états* une relation *R* dans S1× S2 telle que **si *s1 R s2*, alors** :

1. *L (s1) = L (s2)* (les deux états rendent vraies exactement les mêmes variables booléennes)

2. Quelque soit l'état *s'* de *S1*, si *s1 T s'* alors il existe un *s"* de *S2* tel que *s2 T s"* et *s' R s"*

3. Quelque soit l'état *s"* de S2, si *s2 T s"* alors il existe un *s'* de S1 tel que s T s" et s' R s"

Pour *s1* e *S1* et *s2* e *S2*, on dit que *s1* et *s2* sont *bisimilaires*, et on note *s1 ~ s2*, si et seulement s'il existe une bisimulation *R* telle que *s1 R s2*. **La relation ~ est une relation d'équivalence** (elle est réflexive, symétrique et transitive).

Bisimulation entre systèmes de transition.

Une relation *R* est une *bisimulation entre les deux systèmes de transitions M1 et M2* si, pour tout *s1* élément de *I1* , il existe au moins un *s2* élément de *I2* tel que *s1 R s2* et, symétriquement, pour tout *s2* de *I2* , il existe au moins un *s1* de *I1* tel que *s1 R s2*.
De même, on dit que *M1* et *M2* sont bisimilaires, et on note $M1 \sim_{syst} M2$, si et seulement s'il existe une bisimulation *R* entre *M1* et *M2*. Cette relation aussi est une équivalence (entre systèmes de transitions).

Ces notions seront illustrées par le biais d'exemples simples et concrets dans les premières séances du TER. Dans ces séances (deux ou trois, à voir) on expliquera aussi les articles donnés en bibliographie.

Quelques indications de plus sur l'algorithme de minimisation.
Il s'agit de calculer, par raffinements successifs, le quotient de l'ensemble des états du système de transitions input par rapport à la relation d'équivalence $\sim_{\text{états}}$. Si le système de transition input est *M1*, à la fin du processus on aura calculé un nouveau système *M2* tel que : $M1 \sim_{syst} M2$, et *M2* contient un nombre d'états qui peut être strictement inférieur à celui de M1. On le répète, d*e facto,* cet algorithme sera très semblable à l'algorithme de minimisation pour les automates finis étudié dans n'importe quel cours de théorie des langages formels.

Il est possible que, dans le TER, on étudie aussi des variations de cet algorithme, par rapports à des variantes de la notion de bisimulation.

Déroulement du TER, et prérequis.
Les étudiants travailleront de façon autonome, mais rencontreront les encadrantes une fois par semaine pour faire le point sur l'avancement des travaux. Les premiers rencontres (deux ou trois, en principe) seront dédiés à l'explication, par les encadrantes, des notions dont la compréhension est nécessaire pour bien démarrer le travail, et qui ont été ici présentées de façon forcement rapide ; plusieurs exemples en facilitant la compréhension seront fournis. Les seuls prérequis demandés de la part des étudiants sont une certaine aisance avec la programmation et une disponibilité à bien comprendre des notions théoriques comme les structures de Kripke et la logique LTL, qui sont d'ailleurs illustrées par le cours du M1. *De facto*, effectuer ce TER peut aider à bien réussir ce cours, puisque les notions impliquées seront comprises plus en profondeur.

BIBLIOGRAPHIE

- *Advanced Topics in Bisimulation and Coinduction* edited by David Sangiorgi and Jan Rutten, Cambridge University Press, Chapitre 3.

- R. Paige and R. E. Tarjan, *"Three Partition Refinement Algorithms,"* SIAM Journal on Computing, Society for Industrial and Applied Mathématisme (SIAM), Jan 1987.
  The definitive version is available at https://doi.org/10.1137/0216062