Stage Master 2 : Logiques quantitatives pour les hyperpropriétés et application à la sécurité

Encadrants : Benjamin Monmege et Jean-Marc Talbot
prenom.nom@univ-amu.fr

Laboratoire d'Informatique et Systèmes (LIS) – Équipe MOVE

La sécurité des logiciels est devenue un enjeu crucial aussi bien pour les entreprises que pour les états. S'assurer que les applications ne disposent pas de failles de sécurité devient un domaine important de la vérification. Parmi ces failles, certaines proviennent de développements incorrects d'applications logicielles, applications qui au travers d'échanges avec d'autres applications livrent des informations (partielles) qui devraient restées secrètes [7]. Une modélisation possible d'un tel phénomène se trouve formalisée dans la notion de non-interférence : les données sont classifiées selon leur nature privée ou publique et seules les valeurs publiques sont observables. La correction stipule alors que les données privées n'interfèrent pas avec les données publiques : l'observation du traitement des données publiques (en entrée et en sortie notamment) ne permet pas d'extraire d'information concernant les données privées [5]. Cette propriété de non-interférence n'est pas une propriété de chacune des traces d'exécution prise individuellement mais de chacune des paires de traces en spécifiant que pour toute paire de traces partageant les mêmes entrées publiques, elles partagent aussi les mêmes sorties publiques. On peut spécifier cela en terme d'hyperpropriété (au lieu de propriétés) qui stipulent des propriétés d'ensembles de traces et non de traces individuelles. Une extension de LTL appelée HyperLTL [2] permet de spécifier des hyperpropriétés, par l'ajout à LTL de quantificateurs sur les traces.

La non-interférence telle que formulée est très sommaire et ne peut pas, par exemple, distinguer la divulgation d'un mot de passe en clair et celle d'un mot de passe *haché* à l'aide d'une valeur aléatoire. Il est cependant évident que connaitre la version en clair d'un mot de passe où sa version hachée ne donne pas la même quantité d'information sur le mot de passe. La non-interférence a alors été étendu par des aspects quantitatifs pour distinguer ces deux situations [6]. On se pose donc la question d'étendre par des aspects quantitatifs la logique HyperLTL comme cela a été fait pour LTL [1] ou d'autres logiques telles que la logique monadique du second ordre [3]. Dans ce cadre, plutôt qu'une valeur booléenne, une valeur numérique sera associée à chaque formule logique.

L'objectif de ce stage sera donc de proposer une définition d'une version quantitative de HyperLTL et d'étudier le problème de model-checking de cette logique, qui consiste à comparer la valeur d'une formule à une constante. On utilisera des outils d'automates quantitatifs pour résoudre cette tâche [4]. La seconde étape du stage consistera à appliquer cet algorithme de model-checking à la problématique de la non-interférence quantitative.

Mots clés : sécurité, non-interférence, flot d'information, logique temporelle, logique quantitative, automate quantitatif

Références

[1] B. Bollig, P. Gastin, and B. Monmege. Weighted specifications over nested words. In *FOSSACS'13*, pages 385–400, 2013.

[2] M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe, and C. Sánchez. Temporal logics for hyperproperties. In *POST'14*, pages 265–284, 2014.

[3] M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380(1-2) :69–86, 2007.

[4] M. Droste, W. Kuich, and H. Vogler. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.

[5] J. A. Goguen and J. Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20, 1982.

[6] G. Smith. On the foundations of quantitative information flow. In *FOSSACS'09*, pages 288–302, 2009.

[7] J.-M. Talbot and S. Fratani. An accurate type system for information flow in presence of arrays. In *FMOODS/FORTE'11*, pages 153–167, 2011.