Une algèbre des automates d'acceptation propositionnelle déterministes comme théorie d'interface pour la conception de systèmes cyberphysiques

Aurélien Lamercerie

Une algèbre des automates d'acceptation propositionnelle déterministes comme théorie d'interface pour la conception de systèmes cyberphysiques

Aurélien Lamercerie[1]

Univ Rennes, Inria, IRISA - UMR 6074, F-35000 Rennes, France
aurelien.lamercerie@inria.fr

Résumé

Les systèmes cyberphysiques (CPS) sont des logiciels fonctionnant en interaction étroite avec des périphériques physiques. Leur spécification est une tâche complexe et source d'erreurs. Leur conception résulte de l'assemblage de composants, souvent conçus par des équipes multidisciplinaires indépendantes. Chaque équipe doit non seulement spécifier le comportement attendu du composant en cours de conception, mais également énoncer les hypothèses sous lesquelles le composant doit fonctionner. Au moment de l'intégration, ces hypothèses doivent être vérifiées par rapport au comportement du système, pris dans sa globalité.

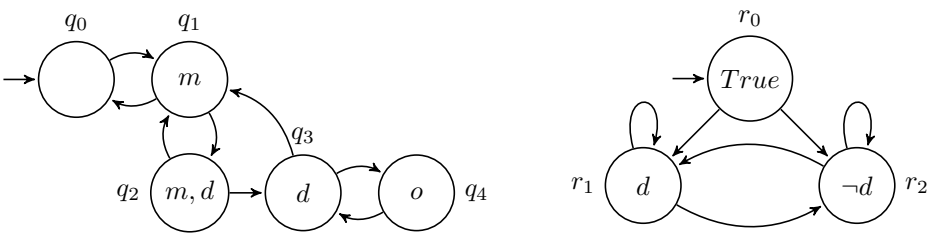
Le raisonnement par contrats [6, 4] est une méthode intégrée à plusieurs langages de programmation [7], inspirés de la logique de Hoare [5] et d'extensions de celles-ci [8]. Détecter les erreurs de spécification aussi tôt que possible permet d'économiser beaucoup de temps, d'efforts et d'argent. Formaliser les besoins, les analyser par des méthodes automatiques ou assistées par ordinateur, ou les vérifier à l'exécution, sont essentielles pour détecter des exigences incohérentes, redondantes ou incomplètes.

Dans le contexte de la conception de CPS, les théories d'interface permettent de décrire en une seule spécification à la fois le comportement attendu d'un composant et les environnements possibles dans lesquels il peut être exécuté. Les exemples typiques de théories d'interface sont les *automates d'interfaces* [3] et les *interfaces modales* [9, 2, 1]. Ces formalismes basés sur la théorie des automates sont capables d'exprimer à la fois la variabilité des conceptions possibles et les incertitudes concernant les environnements possibles d'un composant. Néanmoins, ces théories ne traitent que de la relation entre les entrées et les sorties de composants réactifs. Dans de nombreux cas, il serait souhaitable de relier ces comportements d'entrée/sortie à l'état du composant.

Notre contribution, les automates d'acceptation propositionnelle déterministe (*Deterministic Propositional Acceptance Automata*, DPAA), est une réponse à cette question. Ces automates permettent de spécifier des comportement de temps discret, obligatoire ou interdit. La caractéristique principale de ce formalisme est qu'il permet d'exprimer le comportement attendu lorsque le système est dans un état particulier. Les DPAA combinent des propriétés d'état, exprimées sous forme de formules propositionnelles, et des propriétés temporelles discrètes, exprimées sous forme d'événements obligatoires ou interdits. Ils étendent les systèmes de transition modaux en prenant comme modèles les structures de Kripke plutôt que des systèmes de transition étiquetés.

Ce formalisme forme une algèbre de composition, suivant les principes de la théorie des contrats [1], qui permet de raisonner formellement sur un ensemble de spécifications. Concrètement, les modèles de Kripke sont utilisés pour définir le comportement de composants ou systèmes déterminées. De leur côté, les DPAA capturent la variété des implémentations, exprimées sous

forme d'ensembles d'acceptations. À partir d'un état initial, un automate d'acceptation observe les exécutions d'un modèle et les valide si, pour chaque état du modèle, son ensemble d'états successeurs correspond, d'une manière ou d'une autre, aux ensembles d'acceptation de la spécification. La figure suivante montre un exemple de représentation sur le comportement simplifié des portes d'un bus, à l'arrêt ou en mouvement. Le modèle de Kripke, à gauche, modèle un comportement possible. Le DPAA, à droite, permet de définir les comportements autorisés en fonction de la demande d'un passager pour descendre du bus. Les propositions m, r et o indiquent respectivement que le bus est en mouvement, un passage demande à descendre et la porte est ouverte.



Grâce à une riche algèbre de composition et au fait que la plupart des exigences comportementales exprimées en langage naturel peuvent être traduites en DPAA, la théorie proposée prend en charge les méthodes de raisonnement compositionnel et peut être utilisée pour analyser les exigences comportementales à un stade précoce du processus de conception. La prochaine étape sera la mise en œuvre de cette algèbre et son évaluation sur des études de cas pertinents en ingénierie système.

References

[1] Albert Benveniste, Benoît Caillaud, Dejan Nickovic, Roberto Passerone, Jean-Baptiste Raclet, Philipp Reinkemeier, Alberto Sangiovanni-Vincentelli, Werner Damm, Thomas A. Henzinger, and Kim G. Larsen. Contracts for system design. *Foundations and Trends® in Electronic Design Automation*, 12(2-3):124–400, 2018.

[2] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, and Walter Vogler. Nondeterministic modal interfaces. In *SOFSEM 2015: Theory and Practice of Computer Science - 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec pod Sněžkou, Czech Republic, January 24-29, 2015. Proceedings*, pages 152–163, 2015.

[3] Luca de Alfaro and Thomas A. Henzinger. Interface automata. *SIGSOFT Softw. Eng. Notes*, 26(5):109–120, September 2001.

[4] R. B. Findler, M. Latendresse, and M. Felleisen. Behavioral contracts and behavioral subtyping. In *ACM Conference Foundations of Software Engineering*, 2001.

[5] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM Volume 12 / Number 10 / October, 1969*, 12(10):576–583, 1969.

[6] B. Meyer. Applying "design by contract". *IEEE Computer*, 25(10):40–51, October 1992.

[7] Bertrand Meyer. *Touch of Class: Learning to Program Well Using Object Technology and Design by Contract*. Springer, Software Engineering, 2009.

[8] Peter W. O'Hearn. Separation logic. *Commun. ACM*, 62(2):86–95, 2019.

[9] Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, Axel Legay, and Roberto Passerone. A modal interface theory for component-based design. *Fundamenta Informaticae*, 108(1-2):119–149, 2011.