ÉLÉMENT DE PORTFOLIO 03

Logiciel ou bibliothèque logicielle

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Outil de vérification formelle ITS-Tools

URL de l'élément : `https://ddd.lip6.fr`

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

L'outil ITS-Tools est un outil de vérification formelle et un environnement de modélisation qui permet l'analyse du comportement de systèmes a fortiori concurrents. Il permet en particulier la vérification de propriétés de logique temporelle sur des systèmes exprimés dans des langages divers (réseaux de Petri, langages d'automates communicants, automates temporisés…), à l'aide de plusieurs stratégies de décision. Développé au sein de l'équipe MoVe depuis 2008, cet outil de référence a connu des évolutions importantes sur la période 2017-2022, lui permettant de s'imposer notamment au Model-Checking Contest [6] (MCC) avec de nombreuses médailles d'or depuis 2020.

Son développement continu amène des collaborations applicatives dans divers domaines comme la sécurité avec les arbres d'attaque [8], la modélisation et l'analyse d'algorithmes auto-stabilisants [9], l'utilisation pour analyser les systèmes de manufacture flexibles [5], ou l'analyse de configurations d'un noyau linux. Les interactions logicielles fortes entre ITS-Tools et l'outil Spot [4] de l'EPITA renforcent les collaborations long terme (depuis 2005) entre nos deux équipes.

3 PRÉSENTATION DE CET ÉLÉMENT

3.1 Intégration multi-solution

Une des forces de ITS-Tools est la multiplicité des procédures de décision qu'il est capable d'appliquer. Organisées en portfolio, ces stratégies sont complémentaires et coopérent pour prouver une propriété donnée. En particulier, ITS-tools possède :

- Des solutions basées sur des diagrammes de décision réduits particuliers [2] (et aussi développés dans l'équipe), qui permettent de faire face à de très grands espaces d'états
- Des solutions basées sur les stratégies de réduction d'ordre partiel, appuyé par les technologies de LTSMin [1], un outil développé à Twente (NL).
- Des solutions qui exploitent la puissance des solvers SMT modernes, en appui sur le solver Z3 de Microsoft [3] en particulier, et permettent de surapproximer les comportements d'un sytème par des équations et contraintes
- Des stratégies d'explorations sans mémoire ou avec occupation mémoire bornée, basés sur des runs pseudo-aléatoires ou guidés heuristiquement [11], qui permettent par "sampling" de sous approximer les comportement d'un système
- Des stratégies de réduction structurelles [11], qui vont réecrire le modèle initial tout en préservant la propriété de façon à permettre une analyse sur un modèle plus simple
- Pour l'analyse de propriétés linéaires comme LTL ou PSL, ITS-Tools dispose de stratégies particulièrement avancés, issues de collaborations avec l'EPITA autour de l'outil Spot [4] qui est spécialisé dans la manipulation d'automates de Büchi. Ces stratégies permettent notamment de profiter de réductions (structurelles ou réductions d'ordre partiel en particulier) même si le langage de la propriété est (partiellement) sensible au bégaiements [7].

L'ensemble de l'outil est adapté aux architectures multi-core modernes, est fortement optimisé, et entièrement open source (FOSS, principalement sous licences GPL/EPL).

3.2 Participation au MCC

ITS-Tools participe régulièrement au MCC depuis ses premières éditions ; ces participations ont permis d'améliorer sensiblement l'outil, corrigeant les fautes éventuelles et améliorant les performances continuellement.

L'outil est en général sur les podiums du MCC :

- en 2017, il remporte 1 médaille d'argent et 4 bronze,
- en 2018, il remporte 4 médailles de bronze,
- en 2019, il remporte 2 médailles d'argent et 4 bronze,
- en 2020, il remporte 3 médailles d'or, 1 médaille d'argent et 1 médaille de bronze,
- en 2021, il remporte 3 médailles d'or, 2 médailles d'argent et 1 médaille de bronze,
- en 2022, il remporte 2 médailles d'or, 3 médailles d'argent et 1 médaille de bronze.

Comme on le voit, depuis 2020 ITS-tools a sensiblement progressé, s'arrogeant l'or dans trois catégories en 2021 (propriétés globales, accessibilité, bornes) et l'argent en LTL et CTL. Ceci est principalement lié aux nouvelles technologies introduites dans [11] qui lient l'utilisation de solvers de contraintes [3], d'explorations sans mémoire et de réductions structurelles pour traiter efficacement les problèmes les plus difficiles.

3.3 Utilisations notables de l'outil

L'outil est utilisé par d'autres équipes académiques ainsi que par certains industriels. On peut mentionner en particulier :

- Intégration comme principal moteur de solution dans AtSyRa [8], un outil développé à l'IRISA de Rennes pour l'analyse d'arbres d'attaque/défense
- Intégration comme moteur de solution dans l'outil Ecco [12], développé à l'IBISC d'Evry, pour l'analyse du comportement d'écosystèmes et de systèmes écologiques
- Intégration comme moteur d'analyse de système cyber-physiques dans des travaux [5] des universités de Aachen et Stuttgart.

Le papier de référence sur l'outil [10] est cité 72 fois selon Google scholar, donc 52 citations depuis 2019.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Stefan Blom, Jaco van de Pol, and Michael Weber. Ltsmin : Distributed and symbolic reachability. In *CAV*, volume 6174 of *Lecture Notes in Computer Science*, pages 354–359. Springer, 2010.

[2] Maximilien Colange, Souheib Baarir, Fabrice Kordon, and Yann Thierry-Mieg. Towards distributed software model-checking using decision diagrams. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 830–845. Springer, 2013.

[3] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3 : an efficient SMT solver. In *TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[4] Alexandre Duret-Lutz, Etienne Renault, Maximilien Colange, Florian Renkin, Alexandre Gbaguidi Aisse, Philipp Schlehuber-Caissier, Thomas Medioni, Antoine Martin, Jérôme Dubois, Clément Gillard, and Henrich Lauko. From spot 2.0 to spot 2.10 : What's new ? In *CAV (2)*, volume 13372 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 2022.

[5] Marco Grochowski, Hendrik Simon, Dimitri Bohlender, Stefan Kowalewski, Andreas Löcklin, Timo Müller, Nasser Jazdi, Andreas Zeller, and Michael Weyrich. Formale methoden für rekonfigurierbare cyber-physische systeme in der produktion. *Autom.*, 68(1) :3–14, 2020.

[6] Fabrice Kordon, Lom Messan Hillah, Francis Hulin-Hubard, Loïg Jezequel, and Emmanuel Paviot-Adet. Study of the efficiency of model checking techniques using results of the mcc from 2015 to 2019. *International Journal on Software Tools for Technology Transfer*, 2021.

[7] Emmanuel Paviot-Adet, Denis Poitrenaud, Etienne Renault, and Yann Thierry-Mieg. LTL under reductions with weaker conditions than stutter invariance. In *FORTE*, volume 13273 of *Lecture Notes in Computer Science*, pages 170–187. Springer, 2022.

[8] Sophie Pinchinat, Mathieu Acher, and Didier Vojtisek. Atsyra : An integrated environment for synthesizing attack trees - (tool paper). In *GraMSec@CSF*, volume 9390 of *Lecture Notes in Computer Science*, pages 97–101. Springer, 2015.

[9] Arnaud Sangnier, Nathalie Sznajder, Maria Potop-Butucaru, and Sébastien Tixeuil. Parameterized verification of algorithms for oblivious robots on a ring. *Formal Methods Syst. Des.*, 56(1) :55–89, 2020.

[10] Yann Thierry-Mieg. Symbolic model-checking using its-tools. In *TACAS*, volume 9035 of *Lecture Notes in Computer Science*, pages 231–237. Springer, 2015.

[11] Yann Thierry-Mieg. Symbolic and structural model-checking. *Fundam. Informaticae*, 183(3-4) :319–342, 2021.

[12] Colin Thomas, Maximilien Cosme, Cédric Gaucherel, and Franck Pommereau. Model-checking ecological state-transition graphs. *PLoS Comput. Biol.*, 18(6), 2022.