La validation d'architectures matérielles de communication

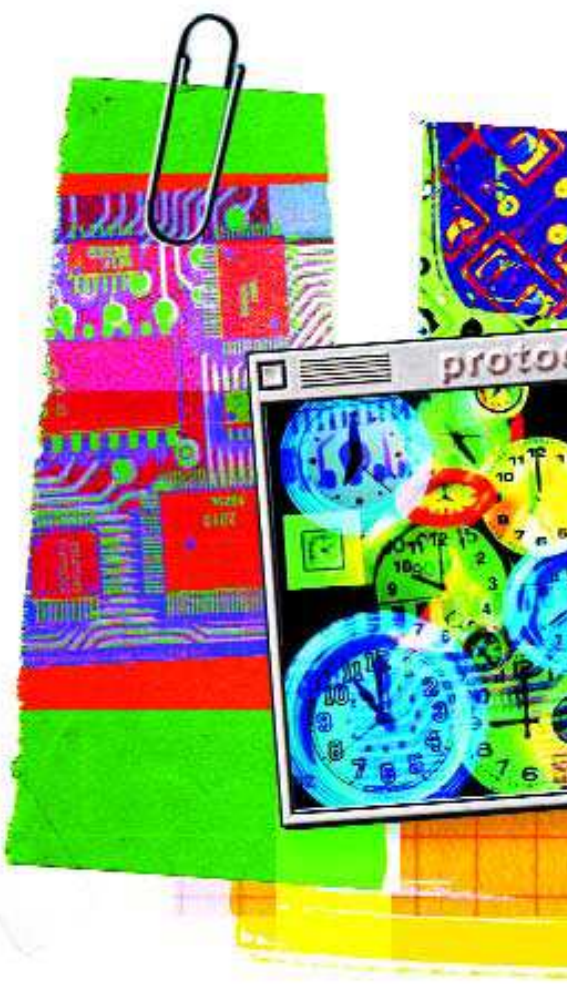
L'apport des méthodes formelles

Ordinateurs pour le calcul scientifique, serveurs de réseaux pour l'Internet, contrôleurs automobiles..., autant d'outils électroniques qui possèdent des architectures matérielles de plus en plus complexes à concevoir. Autant de raisons d'utiliser des outils de validation automatique à base de méthodes formelles.

De nombreuses applications font désormais appel à des architectures électroniques "distribuées" ou "réparties" dont les tâches sont partagées entre plusieurs éléments. Pour mener ces tâches simultanément, "en parallèle", leurs composants matériels (processeurs, mémoires...) sont capables de communiquer, d'échanger des données, et cela de façon réactive, c'est-à-dire en inter-agissant. Lorsque le nombre de processeurs atteint plusieurs centaines, concevoir de telles architectures devient de plus en plus complexe. Or, ces machines gèrent souvent des systèmes critiques où l'erreur n'est pas permise.

Certaines machines travaillent de façon asynchrone (chaque processeur possède sa propre horloge qui doit être synchronisée avec les autres).

Un des points clés de leur communication concerne le protocole de cohérence de caches, qui permet d'optimiser la gestion de la mémoire autonome de chaque processeur et la mémoire du système. Concevoir ces algorithmes impose de prévoir la multitude de comportements possibles du système. La problématique de cette communication est en partie comparable au fonctionnement des réseaux de télécommunication dont les protocoles sont désormais testés en faisant appel à des méthodes formelles mises en œuvre dans des outils automatiques d'aide à la conception et à la validation. Les chercheurs des projets VASY et PAMPA collaborent depuis plusieurs années au sein d'un GIE (dénommé DYADE) avec Bull, pour développer des outils de validation de protocoles adaptés aux architectures matérielles haut de gamme développées par Bull. Jusqu'alors, la validation de ces machines faisait appel à des programmes manuscrits utilisant des langages spécifiques adaptés à la simulation d'une architecture matérielle mais pas à la validation de protocoles complexes comme la cohérence de caches. L'idée innovante des chercheurs a été de faire appel à un langage formel, en l'occurrence LOTOS (normalisé ISO), initialement développé pour la validation de systèmes distribués, pour traiter ce type de problème de façon automatique. Selon les ingénieurs de Bull, cela permettra de diminuer de manière significative le temps de validation d'une architecture, tout en améliorant sa qualité. La première étape, conduite par le projet VASY, consiste à spécifier le système, en d'autres termes à décrire formellement en LOTOS l'architecture matérielle et les protocoles de cohérence de caches. Il s'agit ensuite de vérifier son bon fonctionnement. Pour cela, les chercheurs font appel à un atelier logiciel dédié à l'ingénierie des protocoles, CADP (pour Caesar/Aldebaran), développé depuis 10 ans par le projet VASY (en collaboration avec le projet PAMPA et d'autres partenaires). CADP permet la compilation, la simulation, la vérification et la synthèse de tests de spécifications formelles en LOTOS. À l'heure actuelle, près de 200 sites (industriels ou universitaires) l'utilisent dans le monde.



Pour tester les architectures logicielles de protocoles de télécommunication, les chercheurs ont fait appel à des spécifications formelles.

Les chercheurs du projet PAMPA ont utilisé un outil de synthèse automatique de tests initialement baptisé TGV (Test Generation with Verification technology). Comme son nom le suggère, c'est un outil de validation rapide, qualité non négligeable lorsque des centaines, voire des milliers de cas doivent être testés. En outre, fondé sur des descriptions formelles, il améliore la fiabilité des tests. TGV, développé en collaboration avec Vérimag (laboratoire commun au CNRS, à l'INPG et à l'UJF), utilise des bibliothèques de l'atelier CADP et une description formelle des objectifs du test. Il en déduit un cas de test, un graphe où différents cheminements sont à vérifier.

Depuis deux ans, son transfert industriel est en cours avec Vérilog, intégré dans un compilateur du langage SDL (autre langage formel normalisé). La première version vient d'être commercialisée sous le nom de Test Composer. Pour générer automatiquement des tests de hardware, les chercheurs ont adapté TGV d'une part aux spécifications en LOTOS et d'autre part au matériel à tester. Les résultats sont éloquents : grâce à l'atelier CADP associé au testeur TGV, les ingénieurs de Bull ont modélisé leur architecture de protocole de cohérence de caches et ont déjà testé de façon automatique plusieurs centaines de comportements. Une vingtaine de bogues qui n'avaient pas été détectés avec les techniques utilisées auparavant, ont été ainsi mis en évidence dont huit particulièrement sensibles. Au-delà de ces résultats encourageants, c'est la capacité du monde industriel à tirer parti des méthodes formelles qui est satisfaisante pour les chercheurs qui tentent de les promouvoir.

Parmi les langages de programmation adaptés aux systèmes embarqués, l'autre grande famille concerne les langages synchrones ; une seule horloge, globale, régit les différentes tâches à réaliser, un parallélisme local en quelque sorte.

Ces langages sont donc particulièrement adaptés aux systèmes de contrôle en temps réel, comme les contrôleurs dans les voitures, les commutateurs de téléphone ou certains protocoles de communications. Là encore, les méthodes formelles de programmation et de vérification, bien qu'émergentes, sont prometteuses. Une équipe de chercheurs commune à l'INRIA et à l'École des mines de Paris développe depuis plus de dix ans un de ces langages, baptisé Esterel, dans le projet MEIJE. Ce langage possède une syntaxe de nature impérative et une sémantique opérationnelle formelle, à base d'interprétation dans un modèle de représentation comparable aux circuits électroniques. Il est donc adapté à la conception de composants mixtes matériel/logiciel, de plus en plus utilisés. Sa sémantique autorise également la simulation exhaustive, méthode de vérification automatique des programmes en un nombre fini de configurations possibles. Cet environnement de conception, autant dédié au logiciel qu'au matériel, est commercialisé par une PME, Simulog. Il intéresse les grands noms de l'électronique embarquée comme Dassault Aviation (qui a mis en place une équipe de développement pour programmer en Esterel), Texas Instruments (qui désire réécrire ses architectures de processeurs pour téléphones portables en Esterel) ou Intel, Motorola, Thomson...

Contacts chercheurs :

PAMPA (1) : Claude.Jard@inria.fr

MEIJE (2) : Robert.De_Simone@inria.fr

VASY : Radu.Mateescu@inria.fr

(1) projet de l'Irisa commun avec le CNRS, l'Insa de Rennes et l'université de Rennes 1

(2) projet commun avec le Centre de Mathématiques Appliquées (CMA) de l'Ecole des Mines de Paris