Aussi sûr que 2 et 2 font 4

Dans les programmes informatiques modernes, de plus en plus complexes, se faufilent souvent des erreurs de programmation, les tristement célèbres bugs qui peuplent les cauchemars des programmeurs. Nous avons appris à vivre avec ces erreurs parfois exaspérantes, appliquant des rustines informatiques sur les programmes défaillants.

Mais il est des domaines où la moindre erreur peut conduire à la catastrophe, comme le piratage d'un système informatique. Même quand il n'y a pas malice, les conséquences d'une erreur de programmation peuvent être économiquement désastreuses ; ainsi, le célèbre épisode du « Pentium buggé » de 1994, commercialisé par Intel : le remplacement de ces puces mal calibrées, qui effectuaient parfois des calculs erronés, arrache à son fabricant près d'un demi-milliard de dollars. Deux ans plus tard, un autre bug coûte encore plus cher : à la suite d'une petite erreur de programmation, le prototype de la fusée Ariane-5 explose quarante secondes après le décollage !

Quand les programmes informatiques mettent en jeu des vies humaines, on ne peut tolérer un tel risque ; il faut donc garantir la fiabilité des programmes. C'est en particulier le cas pour les commandes électroniques de vol des avions modernes, ces programmes informatiques qui font l'interface entre les commandes des pilotes et les réponses de l'avion.

Actuellement ces systèmes, dits embarqués, sont vérifiés par de longues procédures de tests que l'on espère exhaustifs ; on imagine sans mal l'intérêt que représenterait un programme informatique sûr, capable de les vérifier automatiquement. Un programme qui vérifie d'autres programmes ! Alan Turing, le père de l'informatique moderne, avait déjà anticipé ce besoin dans les années 1940.

La tâche est d'autant moins simple que les programmes ne sont jamais écrits directement dans le langage binaire que parlent les ordinateurs : on commence par les écrire dans un langage qui ressemble à une langue humaine, avant de les traduire au moyen d'un programme complexe, le compilateur, en un programme fait d'une suite monotone de 0 et de 1. Pour éliminer les risques d'erreur, il faut donc également vérifier le compilateur...

En 2004, Xavier Leroy et son équipe de l'Institut national de recherche en informatique et automatique (Inria) se lancent de ce fait dans l'ambitieux projet « Compcert » : écrire un compilateur C – langage couramment utilisé pour réaliser des logiciels embarqués – et le certifier fiable au moyen du langage Coq, conçu pour vérifier automatiquement les preuves mathématiques. Après tout, un programme informatique ressemble dans sa structure logique à une preuve mathématique !

Ce projet vient d'entrer dans la phase finale, avec des tests réalisés en collaboration avec Airbus. Pour la première fois, on pourra valider un compilateur complexe sans l'ombre d'un doute ! Dans le même temps, une équipe australienne annonce un résultat complémentaire impressionnant : la certification d'un système d'exploitation entier. Connaîtrons-nous un monde où les bugs auront été éradiqués ? ■