Nie wieder Absturz - science.ORF.at

Der Bildschirm ist blau, nichts geht mehr: Computerabstürze auf dem Schreibtisch sind bestenfalls ärgerlich - in Autos oder Flugzeugen können sie sogar sehr gefährlich werden. Informatiker um Helmut Veith von der TU Wien sagen dem Absturz nun mit logischer Gründlichkeit den Kampf an.

Kategorie: Informatik | Erstellt am 30.04.2010.

Anstatt wie bisher üblich Software bei der Anwendung zu testen, verfolgen Veith und Kollegen einen neuen Ansatz: Sie durchleuchten die Programme bereits im Vorhinein - und zwar mit Hilfe von Prüfprogrammen, die beweisen: Diese Software läuft fehlerlos.

science.ORF.at: Was passiert bei einem Computerabsturz?

Helmut Veith: Das kann natürlich viele Ursachen haben, aber ein klassischer Fall ist: Sie installieren ein neues Gerät für ihren Computer - eine Maus, einen USB-Stick. Mit diesem Gerät kommt ein kleines, zusätzliches Programm in diesen Computer, ein sogenannter Gerätetreiber. Und dieses Programm kann einen Fehler machen.

Es kann zum Beispiel passieren, dass dieses Programm zu arbeiten beginnt, aber nie mehr aufhört. Durch diese Endlosschleife können all die anderen Programme nicht ausgeführt werden. Dann tut sich gar nichts mehr, der Computer hängt - und dann sagt man: Er ist abgestürzt.

Das Problem ist vermutlich nicht nur auf PCs beschränkt.

Ein ganz großer Teil der Computer steht nicht auf unseren Schreibtischen, sondern in den Backoffices von Banken oder an Bord von Flugzeugen oder in Autos - ein modernes Auto hat 70 bis 200 Mikroprozessoren. Und wenn es da zu Fehlern kommt, kann das natürlich dramatischere Folgen haben, als dass sie nur einen Patch installieren müssen.

Zum Beispiel?

In den USA sind beispielsweise schon größere Teile des Telefonnetzes durch kleine Computerfehler ausgefallen. 2003 gab es im Nordosten der USA einen Stromausfall, der die ganze Region für viele Stunden lahmgelegt hat. Ein Experte des Pentagon hat letztes Jahr beim Forum Alpbach berichtet, dass der Blackout von 2003 unter Umständen von einer Cyberattacke ausgelöst wurde.

Wie funktioniert so eine Attacke?

Sicherheitsprobleme mit Computersoftware lassen sich immer auf Fehler in Programmen zurückführen. Es gibt eine klassische Art von Fehlern namens **"Buffer Overflow"**

Helmut Veith

<http://www.forsyte.tuwien.ac.at/~veith/> ist seit Dezember letzten Jahres Professor für Formal Methods in Systems Engineering an

<http://de.wikipedia.org/wiki/Puffer%C3%BCberlauf> - die kann man verwenden, um die Kontrolle über einen Computer zu erlangen. Konsequenz dessen ist, dass es mittlerweile einen Schwarzmarkt für Computerfehler gibt.

der TU Wien. Davor war er unter anderem an der **Carnegie Mellon University** <http://www.cs.cmu.edu> tätig.

Wenn Sie einen Fehler finden, dann können sie ihn entweder veröffentlichen, damit die entsprechende Firma das korrigieren kann. Oder Sie können auch versuchen, das an Kriminelle weiterzuverkaufen, weil die das ausnützen können.

Was ist ein Buffer Overflow?

Ein Buffer Overflow ist ein Fehler, der typischerweise in Programmiersprachen wie C auftritt. Er ermöglicht, dass man einen fremden Computer Befehle ausführen lassen kann, die so nicht vorgesehen waren. Man kann gewissermaßen über vorgetäuschte Eingaben einen Befehl in ein Programm hineinschmuggeln.

Das verhindert man wie?

Traditionell legt man bei der Entwicklung von Computersoftware sehr viel Wert auf die Prozesse. Letztlich wird der Ingenieur überprüft - die Art wie er arbeitet, wie er dokumentiert, wie er systematisch vorgeht, aber nicht das Produkt selbst. Das Produkt selbst, also das Programm, wird vielleicht getestet. Aber es wird keiner systematischen und mathematischen Analyse unterzogen.

Warum reicht das Testen nicht aus?

Wenn wir ein Programm testen, dann bedeutet das: Wir führen für dieses Programm für unterschiedliche Eingaben für dieses Programm aus und schauen, ob dabei Fehler auftreten. Das Problem ist, dass es so viele unterschiedliche Eingaben geben kann, sodass es absolut unmöglich ist, all diese Eingaben zu überprüfen.

Sendungshinweis:
Mit Computerabstürzen und möglichen Gegenstrategien beschäftigt sich auch ein Beitrag im aktuellen **"Dimensionen"-Magazin** <http://oe1.orf.at/programm/236137> :
Ö1, 30.4. um 19.06 Uhr.

Man kann das immer nur für einen ganz kleinen Teil machen - und deswegen sind die Testergebnisse immer nur Erfahrungswerte. Nach dem Motto: "Ich weiß halt, dass bis jetzt nichts passiert ist." Das unterscheidet sich aber sehr stark von einem mathematischen Beweis, der mir zusichert, dass nichts passieren *kann*.

Nach solchen Beweisen suchen Sie in ihrer Forschung?

Ja, das Ganze nennt sich **"Model Checking"** <http://de.wikipedia.org/wiki/Model_Checking> . Das ist eine Methode, die in der theoretischen Informatik Anfang der 80er Jahre von Kollegen von uns entwickelt wurde. Sie wurden dafür kürzlich mit dem Turing-Preis ausgezeichnet - das ist der wichtigste Preis in der Informatik, vergleichbar dem Nobelpreis.

Berechnungen zeigen: Die Zahl möglicher Zustände selbst einfacher Computer übersteigt die Zahl der Elementarteilchen im Universum bei weitem. Auch Beweise müssen diese kombinatorische Explosion irgendwie bändigen.

Das ist das Hauptproblem des Model Checking. Die Entwicklungsschübe dieser Disziplin sieht man genau daran, wie gut man mit dieser kombinatorischen Explosion umgehen kann. Ein ganz entscheidender Faktor dabei ist, dass Programme letztlich Texte bzw. Formeln sind, die von Menschen geschrieben wurden.

Sie sind nicht zufällig, sie haben eine Struktur, die ihnen der Programmierer gegeben hat. Die Herausforderung ist, die Struktur in mathematische Methoden zu fassen, damit es eben nicht zu dieser Explosion kommt.

Und wenn mehrere Softwaresysteme zusammenkommen?

Dann verwendet man eine spezielle Strategie, die wir "assume guarantee reasoning" nennen: Man trifft vereinfachende Annahmen für einen Teil des Systems, um den anderen Teil zu überprüfen - und umgekehrt. Wenn man das ganze logisch geschickt auspendelt, kann man einen Schluss bauen, der nicht-zirkulär und korrekt ist.

Ganz vereinfacht gesprochen: Was tut ein Model Checker?

Model Checking bedeutet im Grunde, dass wir ein Computerprogramm entwickeln, das ein anderes Computerprogramm liest, um dort Fehler zu finden.

Ein Programm prüft ein Programm. Klingt paradox.

Ist es auch. Der britische Mathematiker Alan Turing hat schon **in den 30er Jahren gezeigt** <http://de.wikipedia.org/wiki/Halteproblem> , dass unser Ansinnen eigentlich mathematisch-logisch betrachtet unmöglich ist. Das heißt, kein Computer wird jemals in der Lage sein, *alle* Fehler in einem anderen Computer zu finden. Das führt zu einem logischen Paradox und ist deshalb nicht möglich.

Und wie entkommen Sie dem Turing'schen Beweis?

Wir entkommen ihm auf pragmatische Art und Weise: Indem wir uns einerseits auf Programme konzentrieren, wie sie im wirklichen Leben auftauchen. Zum Beispiel auf Gerätetreiber, typische Arten von Software.

Und zum anderen, indem wir einfach auf den Anspruch verzichten, dass wir *alle* Fehler finden müssen. Es reicht uns schon, wenn wir viele Fehler finden und dadurch die Sicherheit und Qualität der Programme erhöhen.

Sie gründen nun mit Fachkollegen aus Österreich einen Verein namens "ARiSE".

Wir befinden uns in der glücklichen Situation, dass sich in den letzten Jahren eine starke Konzentration in diesem Forschungsgebiet ergeben hat. Die Leute sitzen an ganz unterschiedlichen Universitäten über ganz Österreich verteilt.

Wir werden gemeinsam eine wissenschaftliche Gesellschaft gründen: "Austrian Rigorous Systems Engineering". Sie dient der Koordination unserer Forschung, der Ausbildung unserer Studenten, der Ausrichtung von Kongressen usw.

Die Gesellschaft hat durchaus den Anspruch wissenschaftlich in der ersten Liga mitzuspielen. Wenn Sie sich mit den österreichischen Quantenphysikern oder Molekularbiologen vergleichen: Sehen Sie sich auf dem gleichen Niveau?

Veranstaltung:
Das Forschungszentrum IST Austria und der Verein ARiSE veranstalten nächste Woche einen zweitägigen Fachkongress: **"Reactive Modeling in Science and Engineering"** <http://www.ist.ac.at/nc/news-events/news-detail/article/computer-science-symposium/>
Ort: IST Campus Klosterneuburg, Lecture Hall
Beginn: 6. Mai, 9:00 Uhr

Zum einen ist der Verein eine Plattform für unsere Tätigkeiten. Selbstverständlich arbeiten wir für unsere Universitäten, an denen wir angestellt sind. Was wir möchten, ist, unsere Kräfte in Österreich zu bündeln. Wenn das gelingt, wird viel möglich sein. Sagen wir es so: Im

internationalen Vergleich sind wir sicher sehr herzeigbar.

Interview: Robert Czepel

Mehr zu diesem Thema:

- **"Ein burlesker Fall" wird Realität** <http://science.orf.at/stories/1645033/>
- **Wiener Super-Computer auf Platz 156** <http://science.orf.at/stories/1632841/>
- **Spielen als Technik des Selbst** <http://science.orf.at/stories/1632851/>
- **Konjunktiv im neuronalen Netz** <http://science.orf.at/stories/1640677/>