

"Intelligenz ohne Moral ist unlogisch"

INTERVIEW | MARKUS BÖHM

31. Jänner 2012, 18:45



foto: standard/corn

Setzen sich mit intelligenten Programmen und Fehlersuche in ebendiesen auseinander: Helmut Veith (li.) und Stefan Szeider.

Computern das logische Denken beizubringen ist weniger schwierig, als ihnen menschliche Alltagslogik zu lehren, sagen Helmut Veith und Stefan Szeider von der TU Wien

STANDARD: Was wird ein Computer niemals logisch erfassen können?

Szeider: Es gibt Dinge, wo man Alltagswissen braucht. Ein Programm, das eine bestimmte Entscheidung treffen soll, kann das nicht, wenn es ihm an Erfahrung fehlt. Es ist aber möglich, dem Computer dieses Wissen beizubringen, zumindest für einzelne Anwendungsbereiche.

STANDARD: Wie bringt man einem Computer das logische Denken bei?

Szeider: Wir brauchen dafür zwei Dinge: einerseits die Gesetze der Logik selbst, und andererseits Computerprogramme, die mit dieser Logik umgehen können, also geeignete Algorithmen. Aber die Logik des Alltags ist oft eine andere als die formale Logik der Mathematik.

STANDARD: Was heißt das?

Szeider: Es braucht eine Logik, die auch mit Wahrscheinlichkeiten umgehen kann. Ein Beispiel: Als Menschen nehmen wir an, dass, wenn wir Kinder sehen, diese auch Eltern haben. Das ist aber nicht hundertprozentig sicher im Sinne einer einfachen Ja-Nein-Logik. Das macht schließlich das menschliche

Denken aus: Wir treffen ständig Entscheidungen, indem wir unzählige Dinge abwägen. Ein Computer, der sinnvoll in der Wirklichkeit agieren soll, sollte das können.

Veith: Hier gibt es zwei Stoßrichtungen. Eine, die versucht, einem Computer menschliche Handlungsweisen und Alltagslogik beizubringen. Zumindest so weit, dass er Annahmen treffen und auf dieser Grundlage beispielsweise im Internet recherchieren kann. Und eine, die den Computer einsetzt, um die menschliche Logik zu verstärken.

STANDARD: Als Werkzeug, so wie einen Taschenrechner?

Veith: Genau. Der Taschenrechner hilft mir beim Rechnen, der Computer beim Programmieren. Voraussetzung dafür: Logik und Algorithmen. Unsere Alltagslogik reicht nicht aus, um ein komplexes Programm zu schreiben. Auch im Bereich Constraint-Satisfaction geht es darum, schwierige logische Probleme mithilfe des Computers zu lösen. Logik und Algorithmen sind der Werkzeugkoffer für den

Informatiker. Wir sind gerade dabei, von Hammer und Schraubenzieher zu komplexeren Werkzeugen überzugehen.

STANDARD: Wie sehr ist das in der Praxis relevant?

Szeider: Wir machen theoretische Grundlagenforschung, die aber direkt in praktischen Anwendungen mündet. So wird Constraint-Satisfaction etwa auch in Linux-Distributionen verwendet, um Abhängigkeiten zwischen einzelnen Programmpaketen aufzulösen.

Veith: Verifikationstools wiederum werden bei Betriebssystemen wie Linux oder Windows eingesetzt, damit diese reibungslos funktionieren. Vieles ist für den Endanwender versteckt, erhöht aber die Funktionalität der Software erheblich.

STANDARD: Wenn sich ein PC aufhängt, liegt das oft an einem Fehler in der Logik des Programms?

Veith: Im Grunde ist ein Computerprogramm nichts anderes als eine sehr lange logische Formel. An dieser arbeiten zahlreiche Programmierer über den ganzen Globus verstreut, manchmal schon über Generationen. Die Werkzeuge, die wir entwickeln, helfen ihnen dabei, Fehler zu vermeiden.

STANDARD: Ist die Logik eines Computers demnach stärker als die menschliche?

Veith: Wir sehen hier folgenden Widerspruch: Ein Computer kann viele Sachen besser und schneller als ein Mensch. Das sind Dinge, die man durch einfache Rechenvorschriften angeben kann. Der Computer kann sicher viel besser multiplizieren. Allerdings sind Computer mit Aufgaben wie dem Lesen von Texten im Internet oft überfordert.

STANDARD: Was tun Sie, um diesen Widerspruch aufzulösen?

Veith: Wir wollen das, was Computer können, verstärken, um das auf besonders schwierige Fragen anzuwenden. Auf der anderen Seite wollen wir die Bereiche, in denen der Computer schwach ist, verbessern. Vor allem in der Wissensrepräsentation.

Szeider: Das funktioniert wiederum nur über möglichst effiziente Algorithmen. Im Prinzip kann man einem Computer alle Lösungsmöglichkeiten, die es gibt, durchprobieren lassen. Im schlimmsten Fall dauert das aber Jahre. Daher ist unser Ziel, möglichst effiziente Algorithmen zu entwickeln, die den Lösungsweg abkürzen. Man kann damit auch Energie sparen. Ein effizienter Algorithmus braucht keinen Hochleistungschip, und die Batterie des Smartphones hält dann länger durch. Andererseits kann man mit effizienten Algorithmen auch größere und schwierigere Probleme lösen.

STANDARD: Das klingt wie ein kreativer Prozess.

Szeider: Den nimmt uns der Computer noch nicht ab.

Veith: Aber auch das ändert sich schon. Programme werden dafür eingesetzt, andere Programme zu synthetisieren. Weil Programme schwierige Routinearbeiten oft besser bewältigen als menschliche Programmierer.

STANDARD: Der englische Dramatiker John Osborne meinte, der Computer sei die logische Weiterentwicklung des Menschen: Intelligenz ohne Moral. Was halten Sie davon?

Veith: Intelligenz ohne Moral ist unlogisch. Ich denke da an den Science-Fiction-Autor Isaac Asimov und seine Robotergesetze. Das erste Gesetz lautet sinngemäß, dass ein Roboter keinen Menschen verletzen darf. Das heißt, man könnte durchaus moralische Prinzipien in der Programmierung abbilden, aber sie werden nicht von allein aus dem Programm entstehen. Die Entscheidung darüber trifft immer der Mensch. (DER STANDARD, Printausgabe, 01.02.2012)

=> Wissen: Algorithmen

Stefan Szeider promovierte 2001 an der Universität Wien in Mathematik. 2009 gewann er einen Starting Grant vom European Research Council (ERC). Seit Jänner 2010 ist er Professor für Discrete Reasoning Methods an der TU Wien.

Helmut Veith studierte Computationale Logik und Informatik an der TU Wien. Nach Forschungs- und Lehraufenthalten im Ausland ist er seit 2009 Professor an der TU Wien.

© derStandard.at GmbH 2014

Alle Rechte vorbehalten. Nutzung ausschließlich für den privaten Eigenbedarf.
Eine Weiterverwendung und Reproduktion über den persönlichen Gebrauch hinaus ist nicht gestattet.

.