



# Automatic Termination Proofs for Software

Byron Cook

► **To cite this version:**

Byron Cook. Automatic Termination Proofs for Software. Stephan Merz and Tobias Nipkow. Automatic Verification of Critical Systems, Sep 2006, Nancy/France, pp.2, 2006, Automatic Verification of Critical Systems (AVoCS 2006). <inria-00091655>

**HAL Id: inria-00091655**

**<https://hal.inria.fr/inria-00091655>**

Submitted on 8 Sep 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Automatic termination proofs for software

Byron Cook<sup>1</sup>

*Microsoft Research  
Cambridge, U.K.*

---

**Abstract**

In this talk I will describe recent advances in the area of automatic program termination analysis. In particular, I will describe the development of several automatic tools, called TERMINATOR and MUTANT, which implement new termination analysis algorithms. These tools have been used to prove that Windows device driver dispatch routines always return control back to their caller. The tools have also found a number of critical termination bugs in device drivers.

---

---

<sup>1</sup> Email: [bycook@microsoft.com](mailto:bycook@microsoft.com)