VeriFIT
Automatizovaná analýza a verifikace

M. Češka K. Dudka J. Fiedor L. Holík V. Hrubá
L. Charvát B. Křena **O. Lengál** Z. Letko
P. Müller P. Peringer A. Rogalewicz A. Smrčka **T. Vojnar**

Ústav inteligentních systémů, FIT VUT

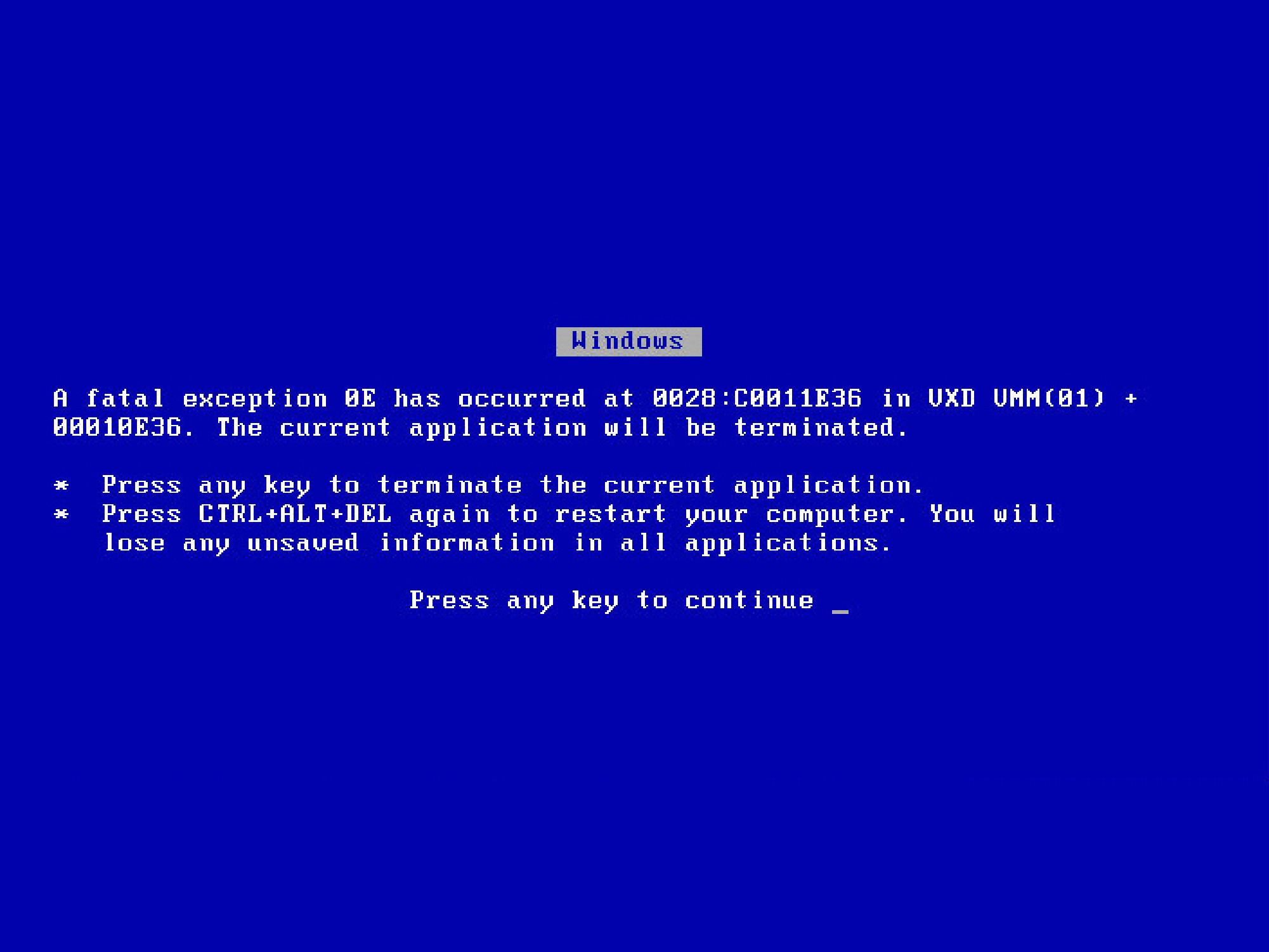
10. 4. 2013

Úvod

Analýza a verifikace

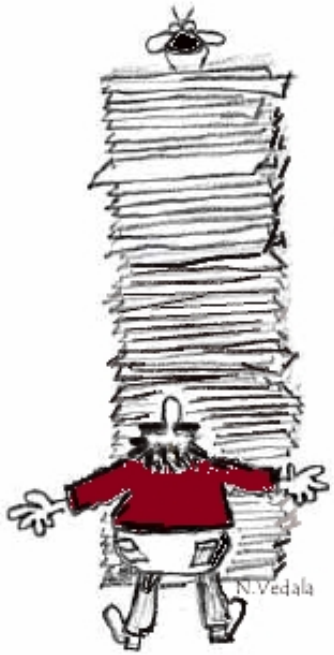
Řešení fundamentálních otázek informatiky:

- Co program dělá?
- Dělá program to, co má?
- Dělá to program správně?
- Jsou v programu chyby?


Windows
A fatal exception 0E has occurred at 0028:C0011E36 in VXD VMM(01) +
00010E36. The current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will
lose any unsaved information in all applications.
Press any key to continue _

VeriFIT — Automatizovaná analýza a verifikace

- Testování a dynamická analýza:
  - široké spektrum technik testování,
  - analýza pokrytí testy,
  - dynamická analýza.
- Formální analýza a verifikace:
  - model checking,
  - statická analýza,
  - theorem proving.



Dr. Dmitry Debugalov
Finds the bug on top
of the Stack.

- Velmi aktuální témata:
  - řada výzkumných projektů na univerzitách a ve výzkumných centrech,
  - podpora významných firem (Microsoft, IBM, Intel, NASA, . . .),
  - řada úspěšných spin-off firem (Coverity, GrammaTech, AbsInt, . . .).

Testování a dynamická analýza

Analýza paralelních programů se sdílenou pamětí (Java, C/C++):

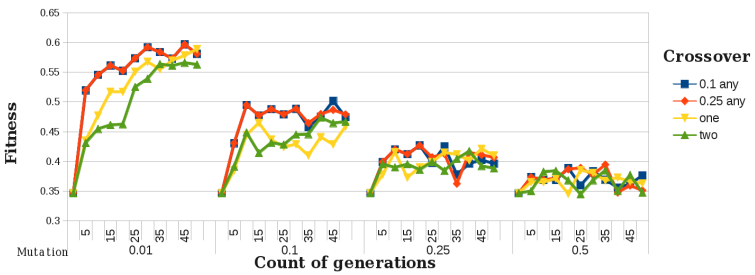
- Systematické testování:
  - využití **noise generátorů**,
  - snaha o **maximalizaci pokrytí testy** pomocí umělé inteligence.
- Dynamická analýza:
  - sledování chování programu za jeho běhu, snaha extrapolovat,
  - **detekce chyb**, i když se neprojeví.
- Detekce chyb typu:
  - **data race** (nekonzistence dat při souběžném přístupu více vláken),
  - **deadlock** (uváznutí programu).

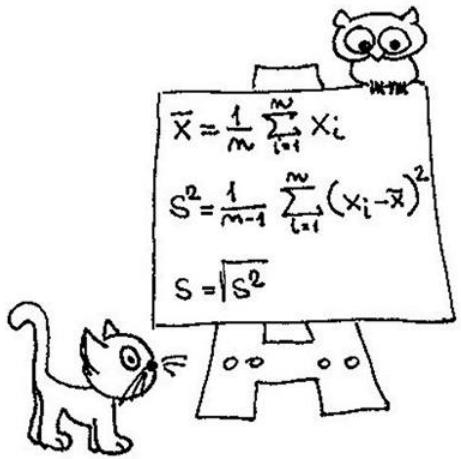
Testování a dynamická analýza

- Automatické opravování programů (self-healing):
  - zamezení projevu detekované chyby.

- Testování programů využívajících transakční paměti
  - komunikace procesů připomínající databázové transakce:
  - **start** . . . **commit** (event. **rollback**).

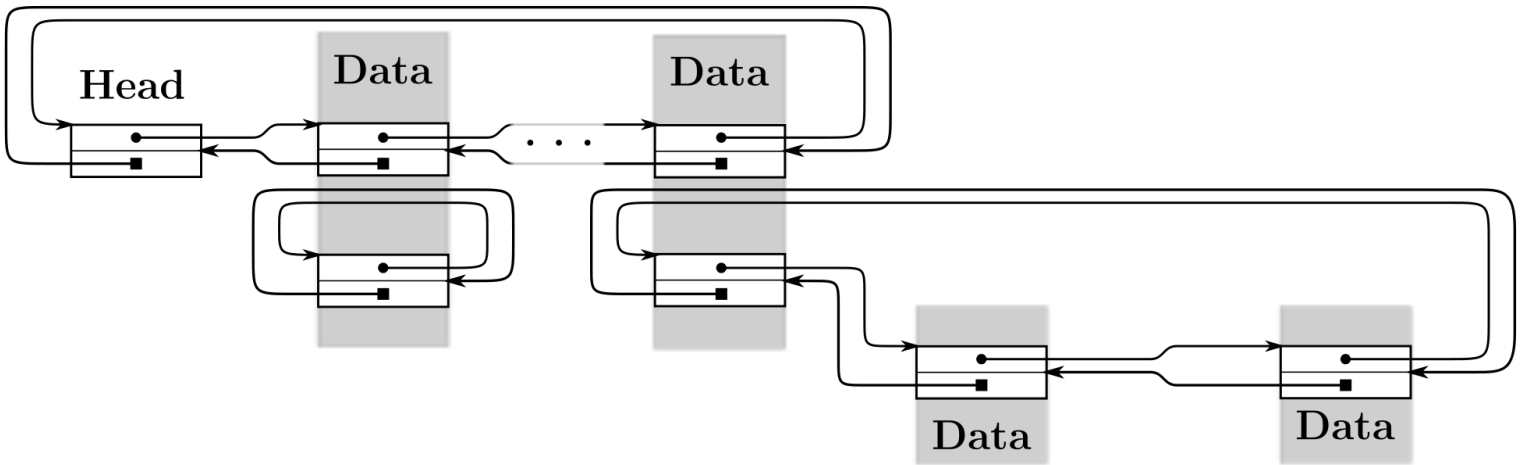
- Dolování znalostí z dat
  - získávání dodatečných informací z nasbíraných dat,
  - **statistika**.





Formální verifikace programů s dynamickou pamětí

- Složité datové struktury založené na ukazatelích:
  - **seznamy** neomezené délky (cyklické, zanořené, skip listy, . . .),
  - **stromy** (binární/n-ární, s parent pointery, RB, AVL, B, B+, . . .),
  - $\Longrightarrow$ typicky nekonečně stavové systémy.



- Aplikace: převážně nízkoúrovňový kód
  - kontejnery ve standardních knihovnách,
  - části ovladačů pro jádro **Linux**,
  - alokátory prohlížeče **Mozilla Firefox**, . . .

Formální verifikace programů s dynamickou pamětí

- Separační logika:
  - **Predator**
    - verifikace programů se seznamy,
    - nejlepší současný nástroj pro verifikaci programů s dynamickou pamětí,
    - vítěz 1 kategorie SV-COMP'12, 3 kategorií SV-COMP'13.
  - **CPAlien**
    - verifikace programů se stromy.

Formální verifikace programů s dynamickou pamětí

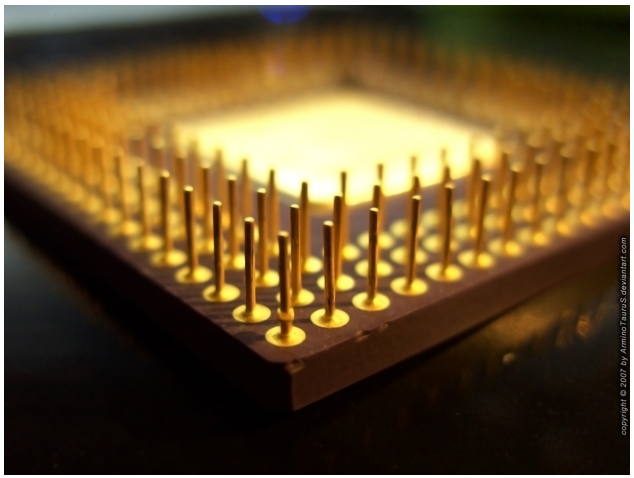
Teorie automatů:

- **Forester**
  - verifikace programů s obecnějšími datovými strukturami,
  - reprezentace paměti pomocí konečných automatů.
- základní výzkum v oblasti automatů,
- aplikace automatů v dalších oblastech (logika, . . .).

Verifikace procesorů

- Vývoj hardware: až **80% času** je věnováno **ověřování správnosti**
  - snaha zredukovat tento čas na minimum.

- Verifikované vlastnosti:
  - verifikace jednotlivých instrukcí,
  - testování posloupnosti instrukcí.
- Aplikace výzkumu:
  - na reálných procesorech.
- Přístupy:
  - automatizované testování: bug-hunting,
  - formální metody: důkaz korektnosti.



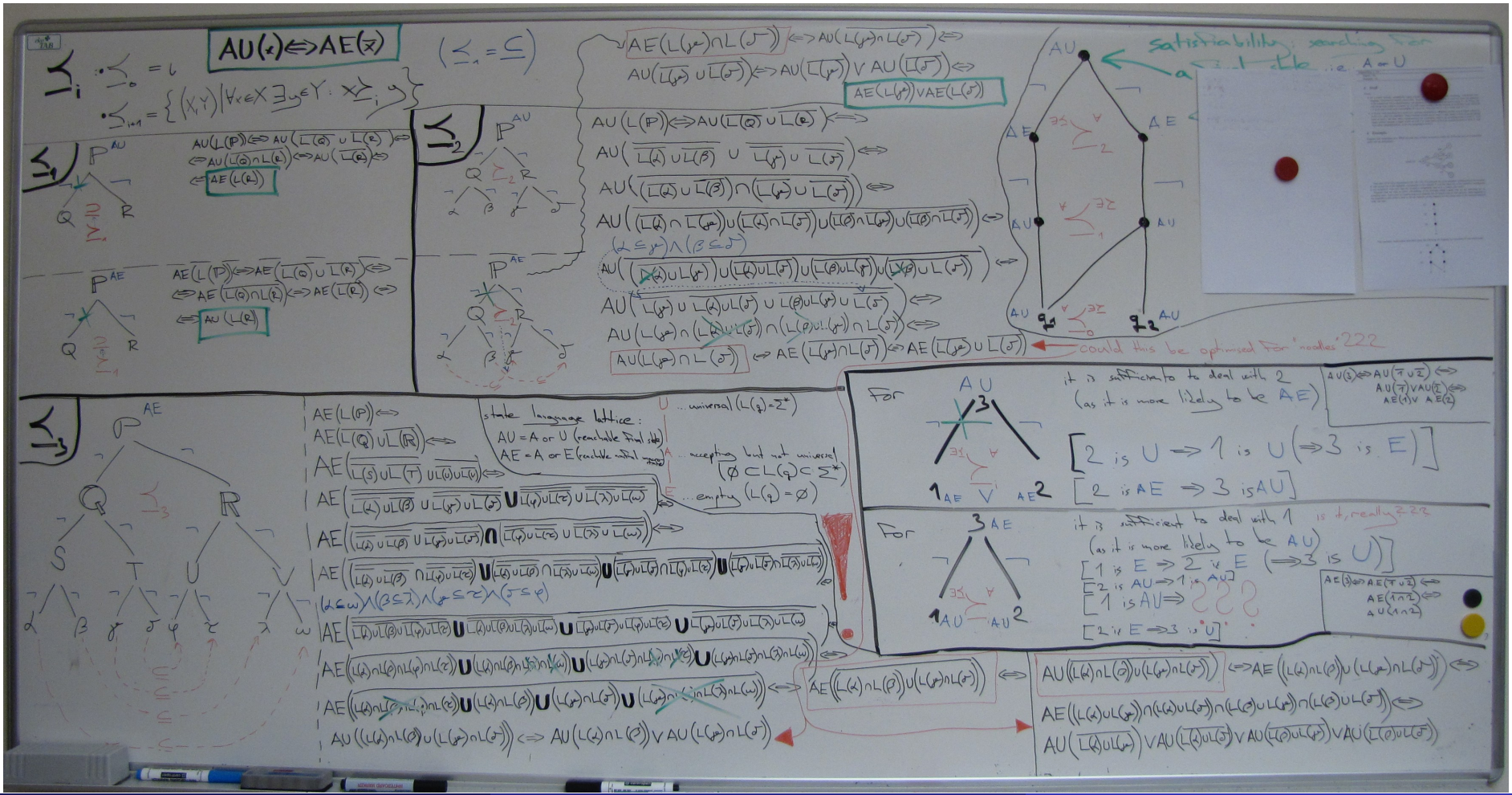
Styl práce VeriFIT

Výzkum zahrnuje:

- **teoretické bádání**: rozhodnutelnost, složitost, ...

  Theorem. *Given an abstract heap* $\widetilde{H}$, $sat(\widetilde{H}) = \sqcap\{\widetilde{H'} \mid \gamma(\widetilde{H'}) = \gamma(\widetilde{H})\}$.

  Proof. By induction ... □



Styl práce VeriFIT

Výzkum zahrnuje:

- **efektivní implementace**: pokročilé datové struktury a algoritmy,
  - BDD, hash tabulky, efektivní cachování, ...
- **aplikace**:
  - paralelní programy v Javě a C/C++,
  - manipulace ukazatelů v C/C++,
  - komunikační protokoly,
  - návrh mikrokontrolerů,
  - ...

Zahraniční spolupráce

- Švédsko (Uppsala),
- Taiwan (Academia Sinica),
- Velká Británie (UCL London, Edinburgh),
- Izrael (IBM Haifa, Shmuel Ur Innovations),
- Francie (Paříž, Grenoble),
- Německo (Pasov).