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
H. Pluháčková A. Rogalewicz A. Smrčka **T. Vojnar**

Ústav inteligentních systémů, FIT VUT

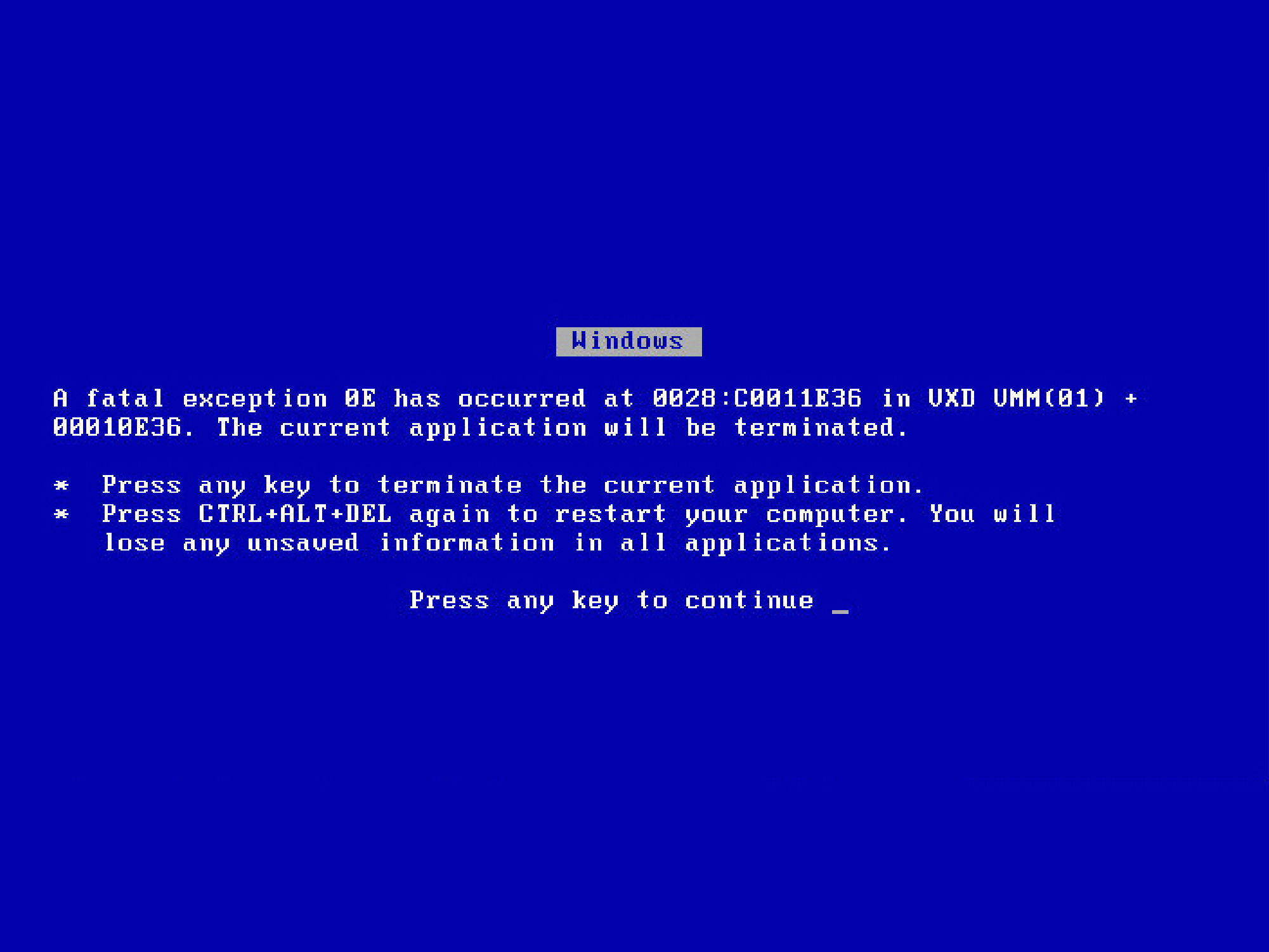
26. 3. 2014

Úvod

Analýza a verifikace

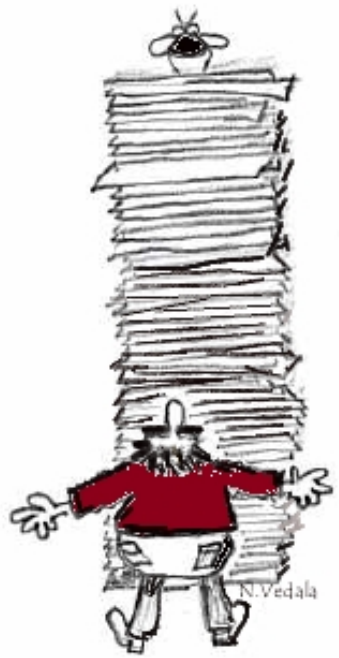
Řešení fundamentálních otázek informatiky:

- 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
- Formální analýza a verifikace



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 (Google, IBM, Intel, Microsoft, NASA, . . .),
  - řada úspěšných spin-off firem (Coverity, GrammaTech, Monoidics, . . .).

Testování a dynamická analýza

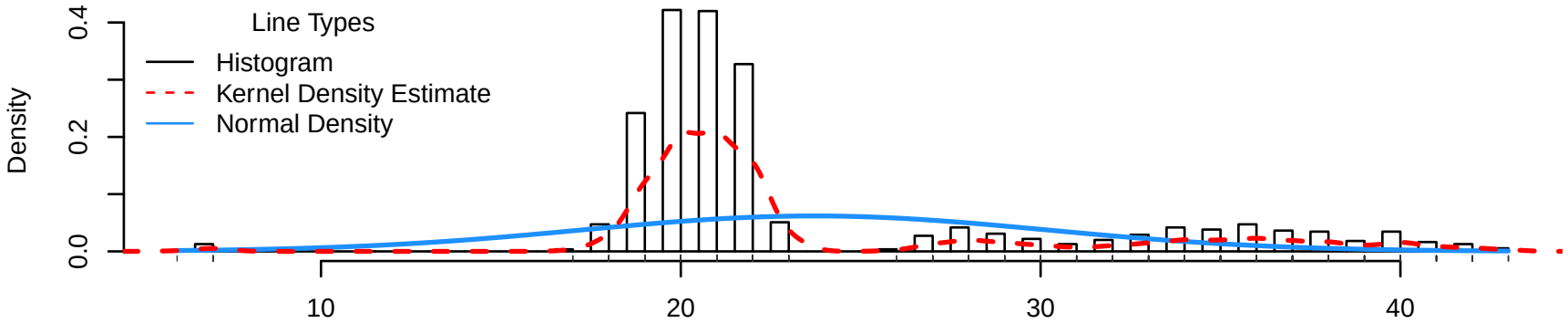
Analýza paralelních programů se sdílenou pamětí (Java, C/C++):

- 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

- 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**.



Automatizované testování GUI

Automatizované testování GUI

- nyní převážně ruční činnost
  - náročné na čas, lidské zdroje, chyby z opomenutí,
- automatizace žádoucí — nutnost řešit
  - ovládání **virtuálního stroje** — rychlý snapshot, restore,
  - **emulace uživatelských vstupů**,
  - **vyhledávání zajímavých oblastí** na obrazovce, . . .

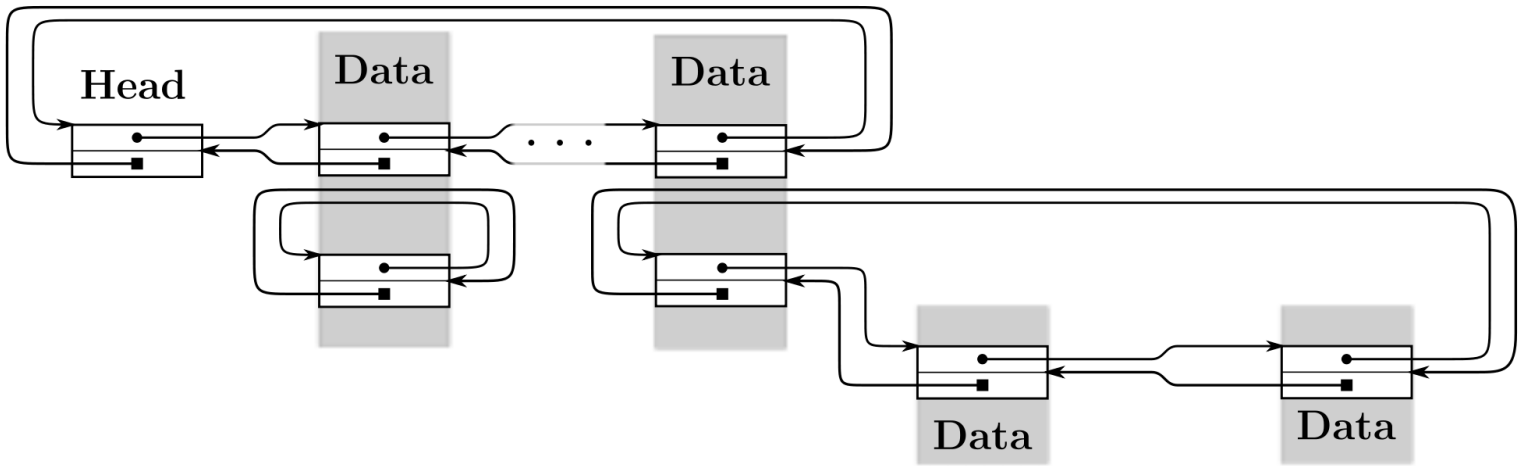
Formální analýza a verifikace

Formální analýza a verifikace:

- schopnost dokázat nepřítomnost chyb (oproti testování),
- mnoho různých technik
  - model checking,
  - statická analýza,
  - theorem proving.

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**, implementace fce **malloc()**, . . .

Formální verifikace programů s dynamickou pamětí

Symbolické paměťové grafy inspirované separační logikou:

- **Predator**
  - verifikace programů se seznamy,
  - nejlepší nástroj pro korektní verifikaci programů s dynamickou pamětí,
  - řada medailí z mezinárodní soutěže ve verifikaci SV-COMP.
- **CPAlien**
  - reinkarnace Predatora v prostředí konfigurovatelné analýzy programů.

Formální verifikace programů s dynamickou pamětí

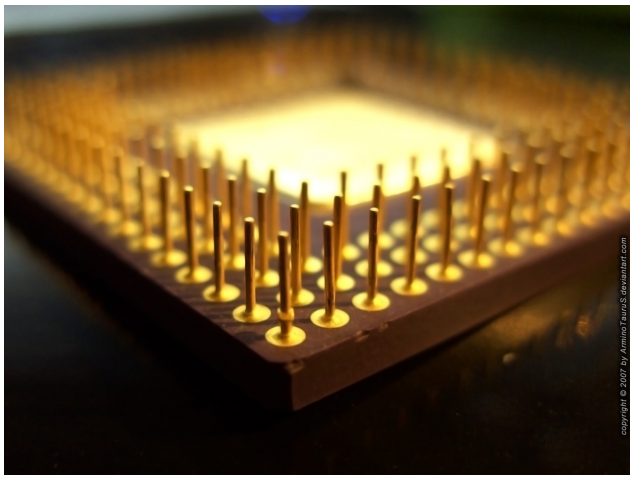
Teorie automatů:

- **Forester**
  - verifikace programů s obecnějšími datovými strukturami,
    - stromy, skip listy, ...
  - reprezentace paměti pomocí konečných automatů.

- základní výzkum v oblasti automatů,
- aplikace automatů v dalších oblastech:
  - rozhodovací procedury pro separační logiku: **SLIDE**, **SPEN**

Verifikace procesorů

- Vývoj hardware: až **80% času** je věnováno **ověřování správnosti**
  - snaha zredukovat tento čas na minimum.
- Verifikované vlastnosti:
  - zaměření na pipeline v procesorech,
  - detekce chyb v hazardech
    1. verifikace každé instrukce odděleně,
    2. verifikace posloupnosti instrukcí.



- Aplikace výzkumu:
  - na reálných procesorech.

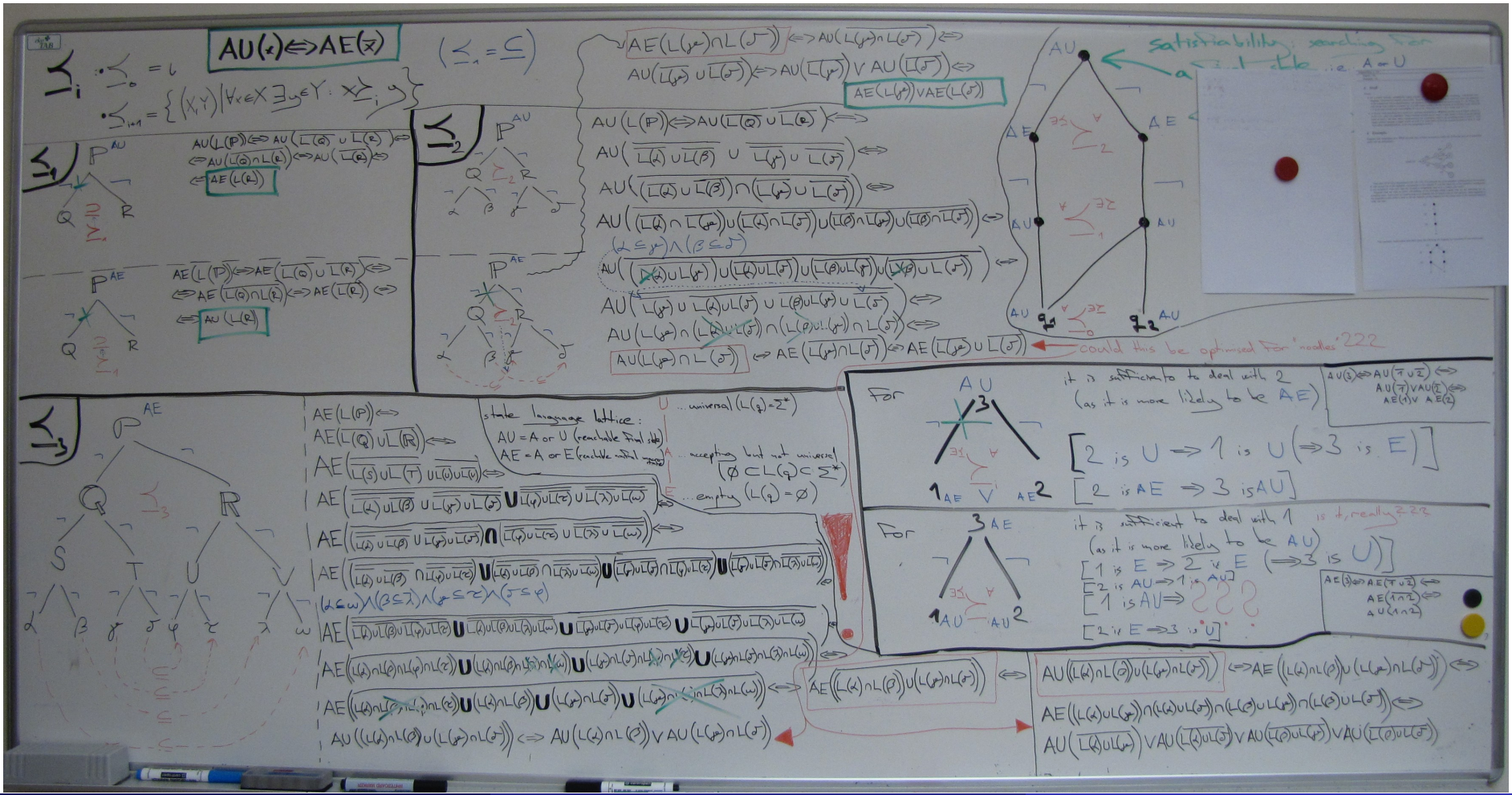
Styl práce VeriFIT

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,
  - ...

Styl práce VeriFIT

Výzkum zahrnuje:

- **cestování**:
  - Švédsko (Uppsala, Linköping),
  - Taiwan (Academia Sinica),
  - Velká Británie (UCL London, Edinburgh),
  - Izrael (IBM Haifa, ORT Braude, Shmuel Ur Innovations),
  - Francie (Paříž, Grenoble, Bordeaux),
  - Portugalsko (Lisabon),
  - Německo (Pasov),
  - Rakousko (Vídeň),
  - konference, ...

Styl práce VeriFIT

Výzkum zahrnuje:

- **odpočinek**:

Styl práce VeriFIT

Výzkum zahrnuje:

- **odpočinek**:

VeriFIT