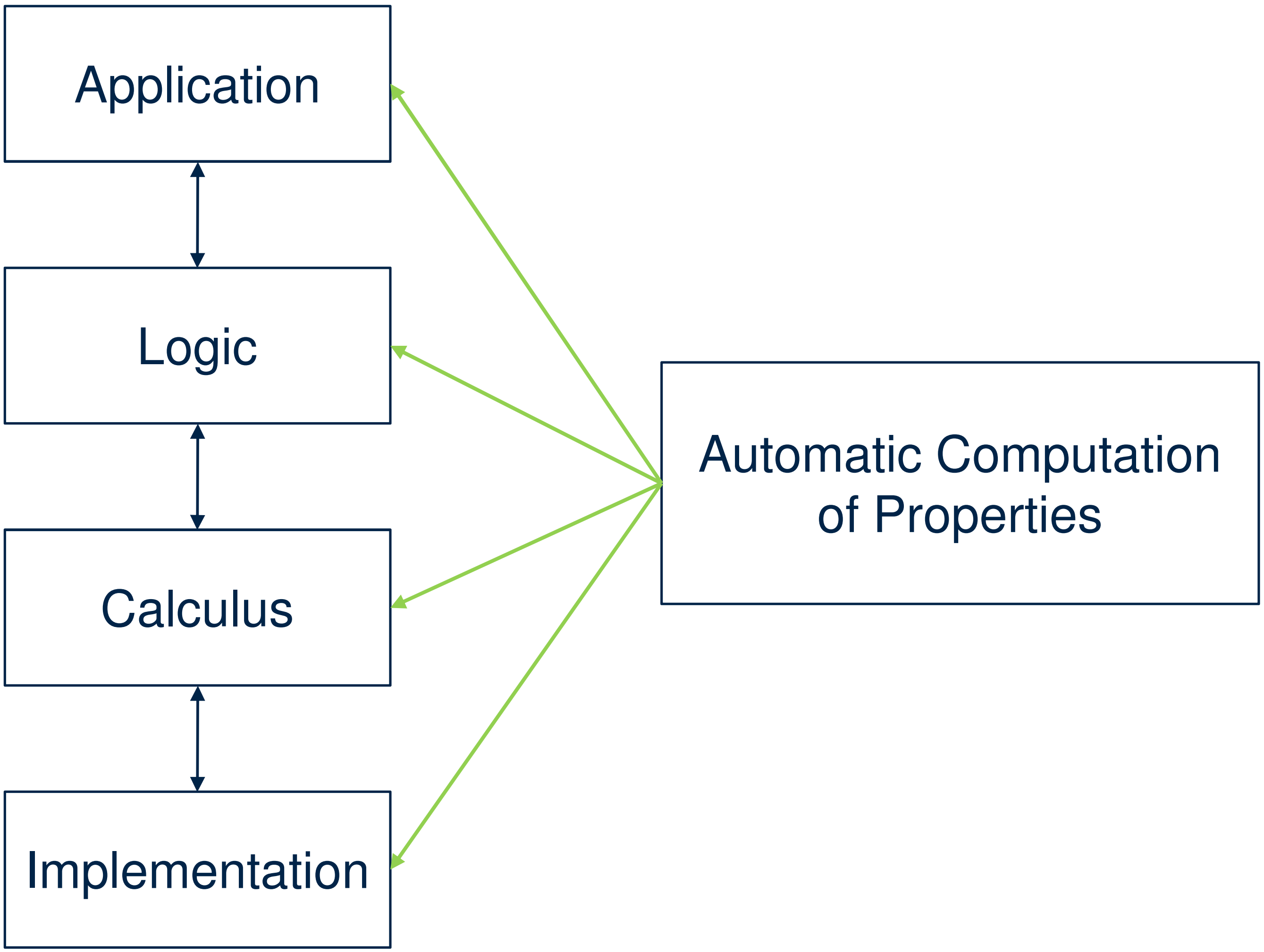
max planck institut
informatik

Variability Management

Prof. Dr. Christoph Weidenbach

Automation of Logic



Examples by PhD Thesis

- Dr. Matthias Horbach: second-order logic decidability
- Dr. Carsten Ihlemann: local theory extensions
- Tinxiang Lu: verifying correctness of PASTRY
- Arnaud Fietzke: combining first-order and prob. reasoning
- Patrick Wischnewski: reasoning in large ontologies
- Ching Hoo Tang: variability management for steel factories (Siemens)

Reasoning in Large Ontologies

max planck institut informatik

Develop “semantic” GOOGLE:

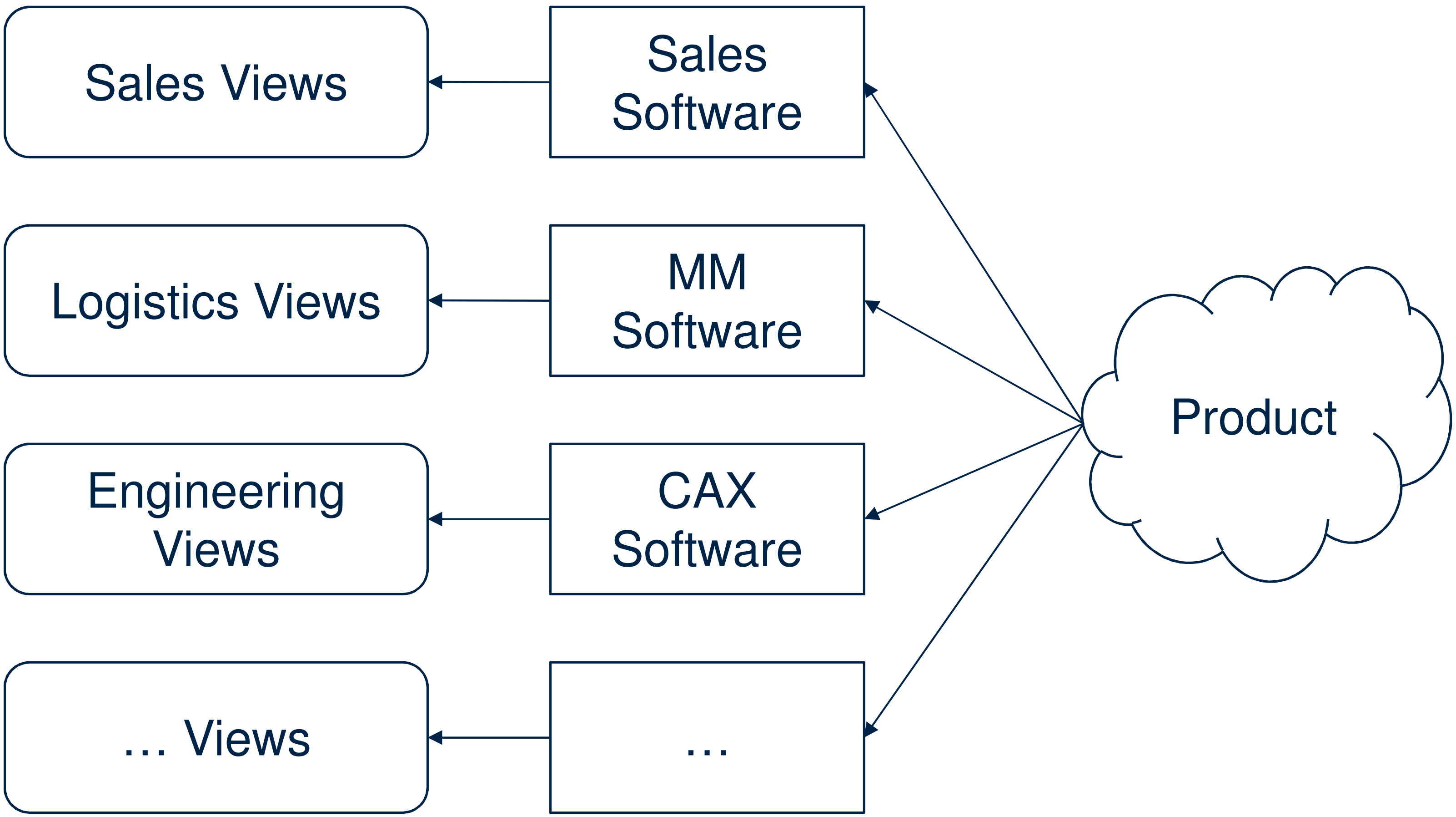
SPASS YAGO

Configuration Today

The car industry:

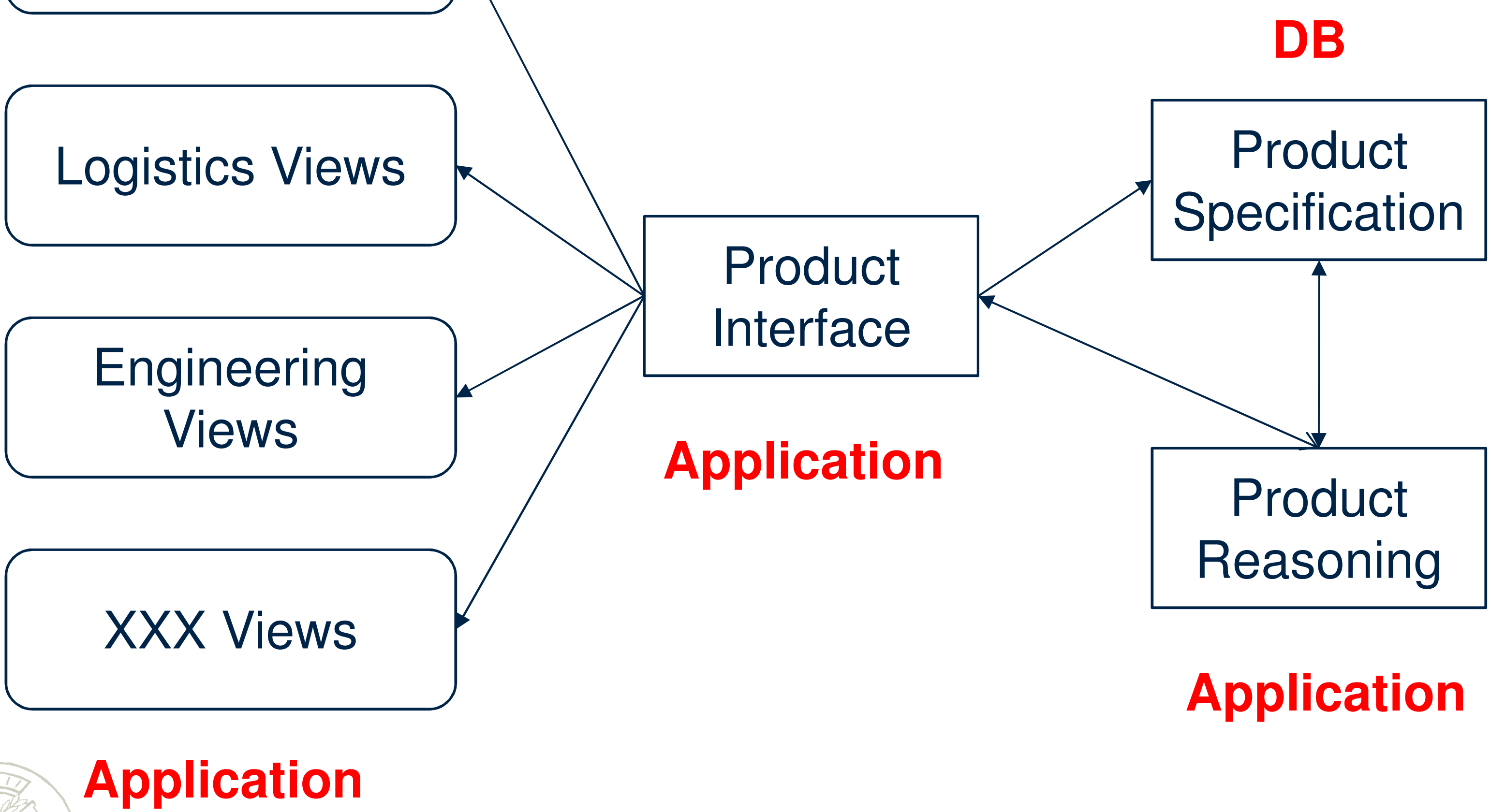
Opel Configuration

Today's Architecture

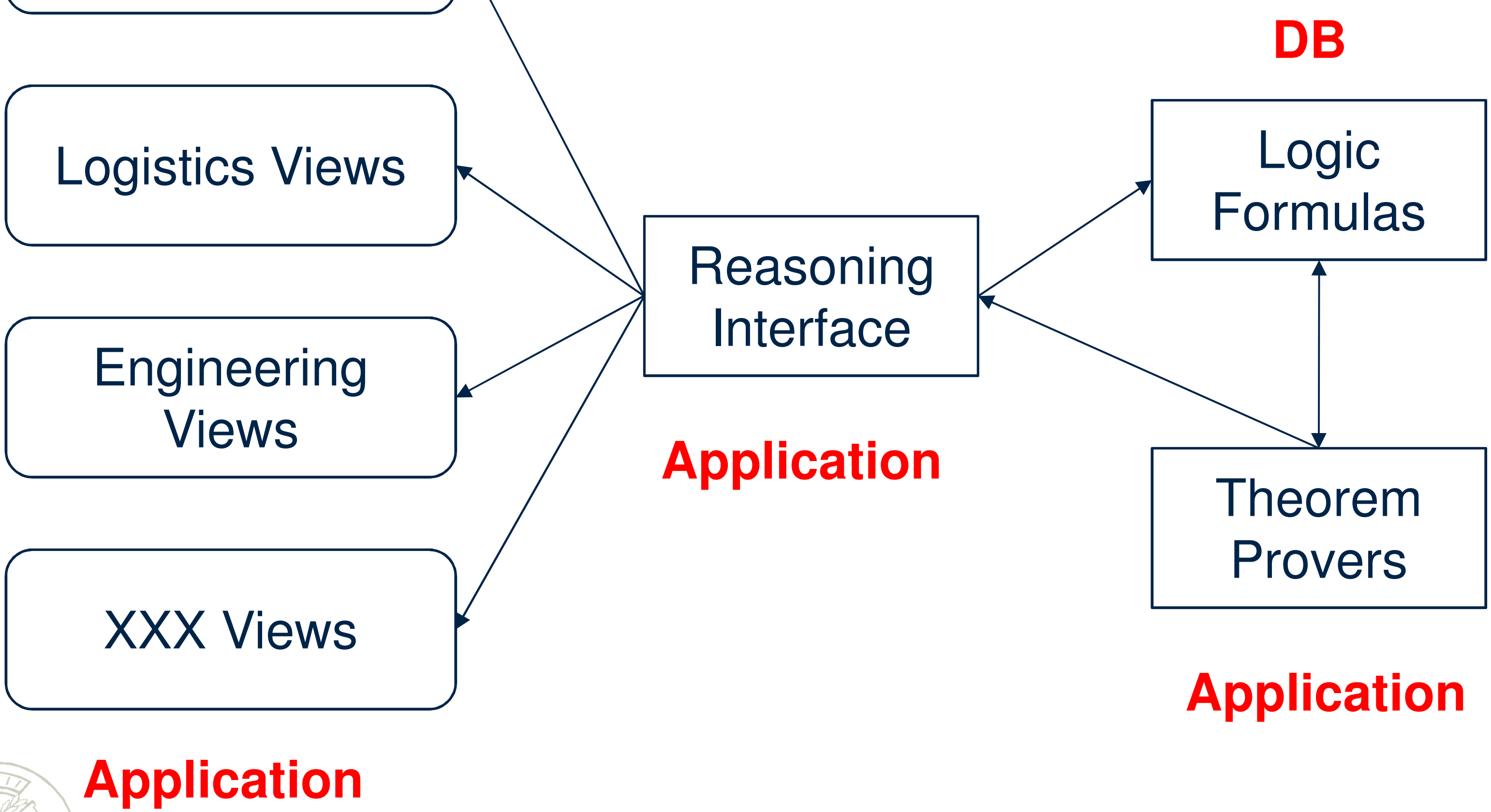


Application **Application + DB** **Paper, People**

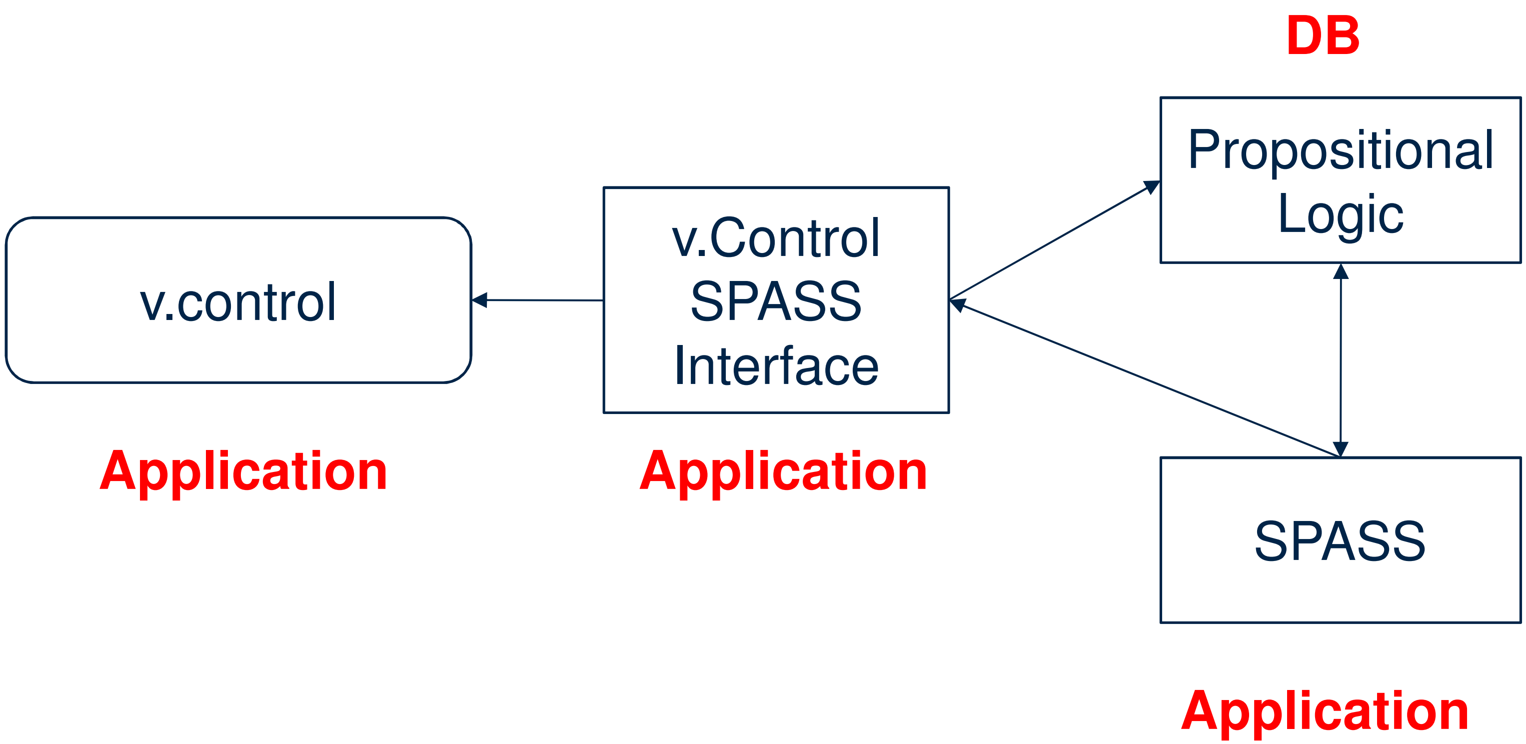
Future's Architecture



Scientific View



Concrete Example



V.CONTROL & SPASS

in cooperation with

Prof. Dr. Georg Rock, Uni App Sc Trier, PROSTEP IMP

Karsten Theis, PROSTEP IMP

Patrick Wischewski, MPI-INF

Propositional Logic

max planck institut informatik

- Language: propositional variables can be true (1) or false (0)
- Connectives: $\Rightarrow$ implication, $\neg$ negation, $\vee$ disjunction, $\wedge$ conjunction
- Clause: disjunction of variables or their negations (literal)
- Validity: a formula is valid iff it is true for all possible assignments
- Assignment: setting all propositional variables 1 or 0, can also be expressed by showing the true literals
- we write $M \models C$ if the clause C is true by assignment M
- SAT: propositional satisfiability, find an assignment such that for a set of clauses all clauses are valid in the assignment

Unit Propagation

max planck institut
informatik

UProp(N,M)
  while (there is a clause $C' \lor L \in N$ such that
      $M \models \neg C'$ and $L \notin M$ and $\neg L \notin M$)
    $M := M \cup \{L\}$;
  return M;

UProp($\{\neg A \lor \neg B \lor E, \quad \neg A \lor B, \quad \neg E, \quad D, \quad A\}$,$\emptyset$)

$\rightarrow M = \emptyset$

$\rightarrow M = \{\neg E\}$

$\rightarrow M = \{\neg E, D\}$

$\rightarrow M = \{\neg E, D, A\}$

$\rightarrow M = \{\neg E, D, A, B\}$

DPLL Procedure

DPLL(N,M)
  if for all $C \in N$ we have $M \models C$ return true;
  if there is some $C \in N$ with $M \models \neg C$ return false;
  select a variable P occuring in N but not in M;
  if (DPLL(N, UProp(N, $M \cup \{P\}$))) then
    return true;
  else
    return DPLL(N, UProp(N, $M \cup \{\neg P\}$));

$\neg A \lor \neg B \lor E$
$\neg A \lor B$
$\neg E$
$A \lor D$

DPLL(N, $\emptyset$)

DPLL(N, UProp(N,$\{A\}$)) DPLL(N, UProp(N,$\{\neg A\}$))

DPLL(N, $\{A, B, \neg E\}$) DPLL(N, $\{\neg A, D, \neg E\}$)

DPLL is sound and complete and terminating for SAT.

Propositional Logic Formulas

$$\text{Corsa} \Rightarrow \text{Wheels} \wedge \text{Engines}$$

$$\text{4-Holes} \Rightarrow \text{Wheels}$$
$$\text{5-Holes} \Rightarrow \text{Wheels}$$
$$\text{4-Holes} \Rightarrow \neg\text{5-Holes}$$
$$\text{5-Holes} \Rightarrow \neg\text{4-Holes}$$

$$\text{Diesel} \Rightarrow \text{Engines}$$
$$\text{Gasoline} \Rightarrow \text{Engines}$$
$$\text{Diesel} \Rightarrow \neg\text{Gasoline}$$
$$\text{Gasoline} \Rightarrow \neg\text{Diesel}$$

$$\text{Diesel} \Rightarrow \neg\text{4-Holes}$$

Reasoning: $\text{Corsa} \rightarrow \text{Wheels}, \text{Engines}$

$\text{4-Holes} \rightarrow \neg\text{5-Holes}, \neg\text{Diesel}, \text{Gasoline}$

$\text{Gasoline} \rightarrow \neg\text{Diesel}$

Challenge: Scalability

- worst case SAT searches 2^n nodes
- before 2009: approx. 1500 nodes
- in 2011: v.control + SPASS approx. 6000 nodes
- in x years: for a reasonable product approx. 60000 nodes

max planck institut informatik

- Automated Reasoning Lecture:

  http://www.mpi-inf.mpg.de/departments/rg1/teaching/

- contact us on student assistant jobs, bachelor-master-PhD thesis

Thank you for your attention