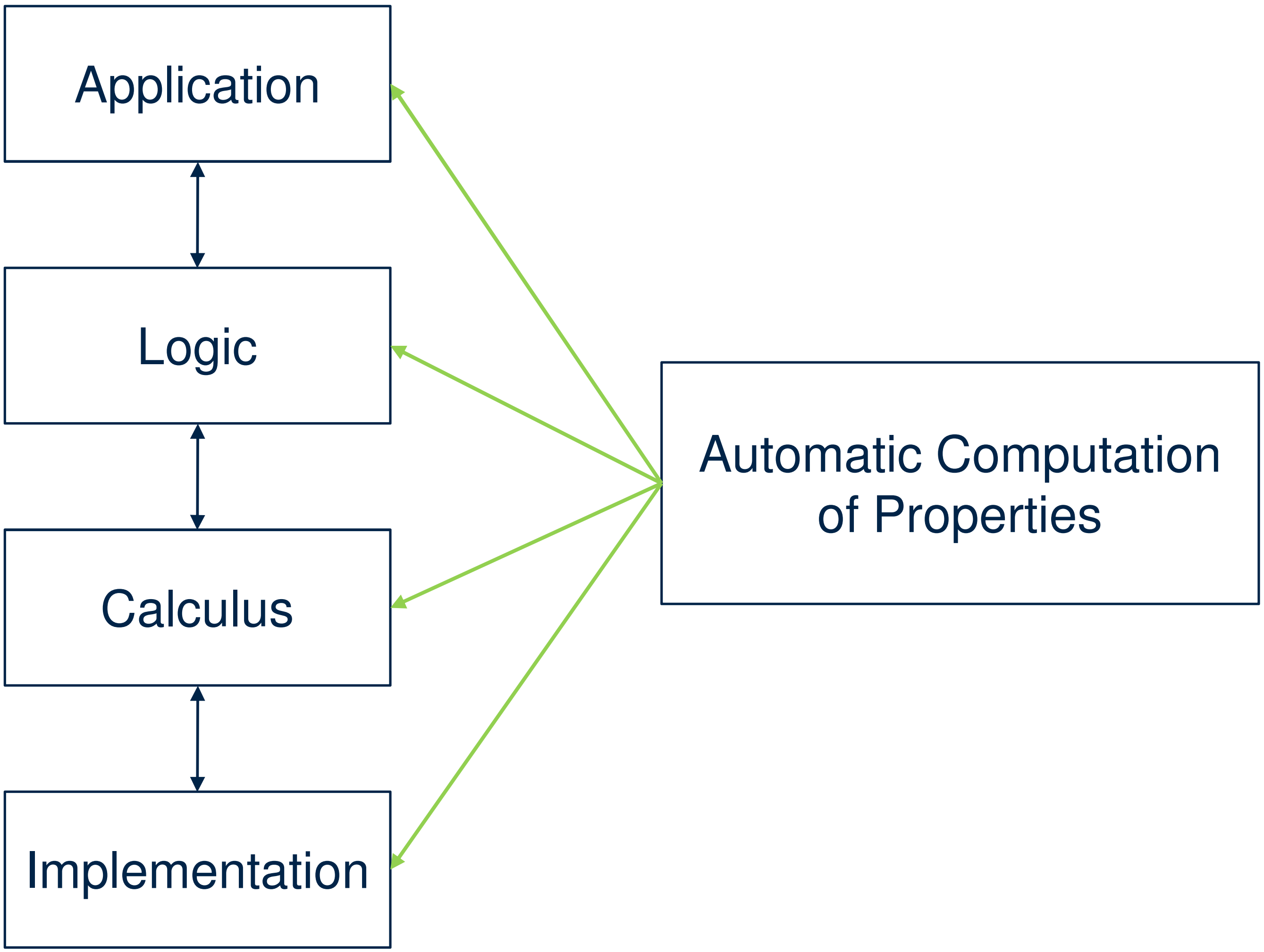
max planck institut
informatik

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
- ?: variability management (PROSTEP, Siemens)

Reasoning in Large Ontologies

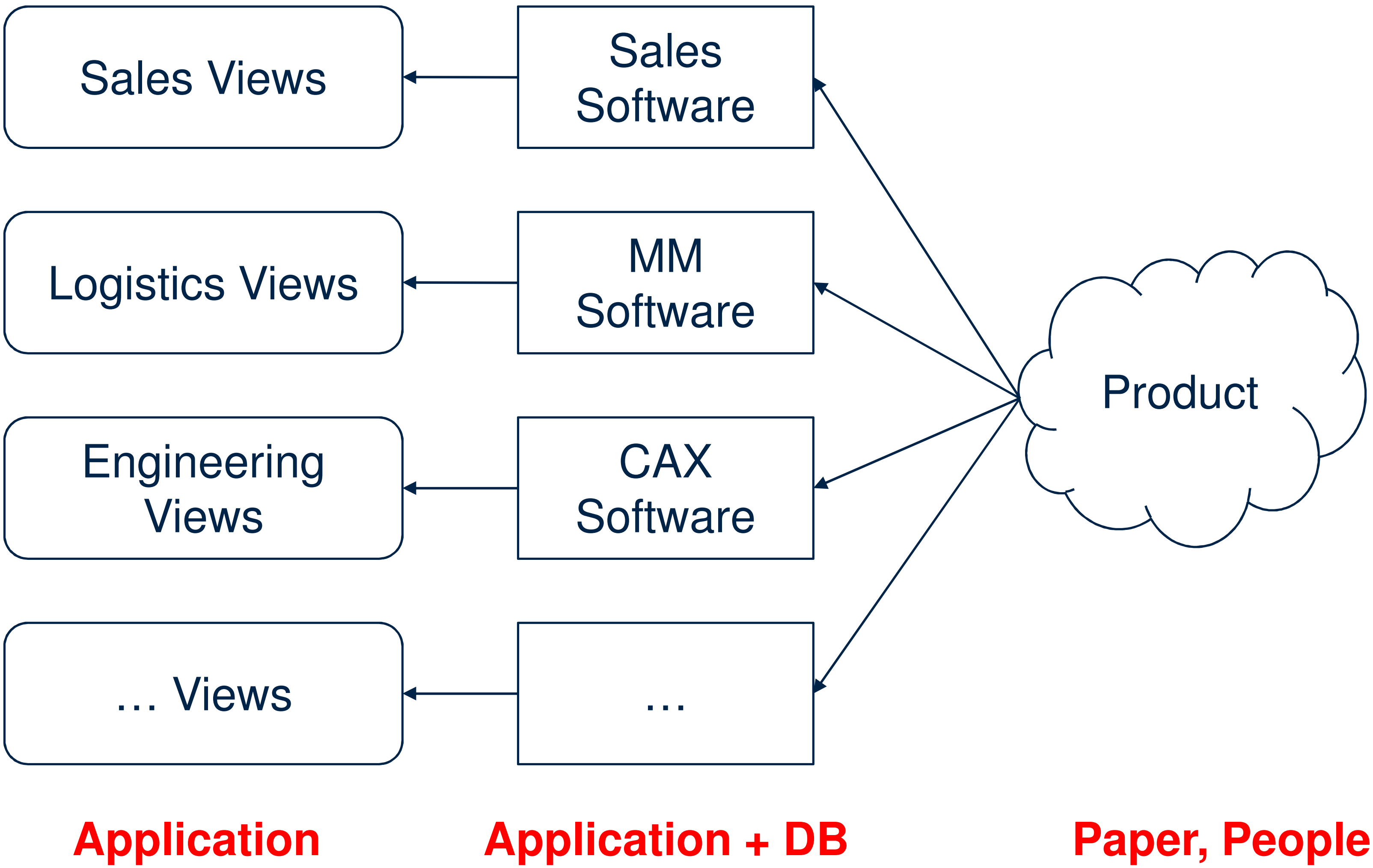
develop “semantic” GOOGLE

Configuration Today

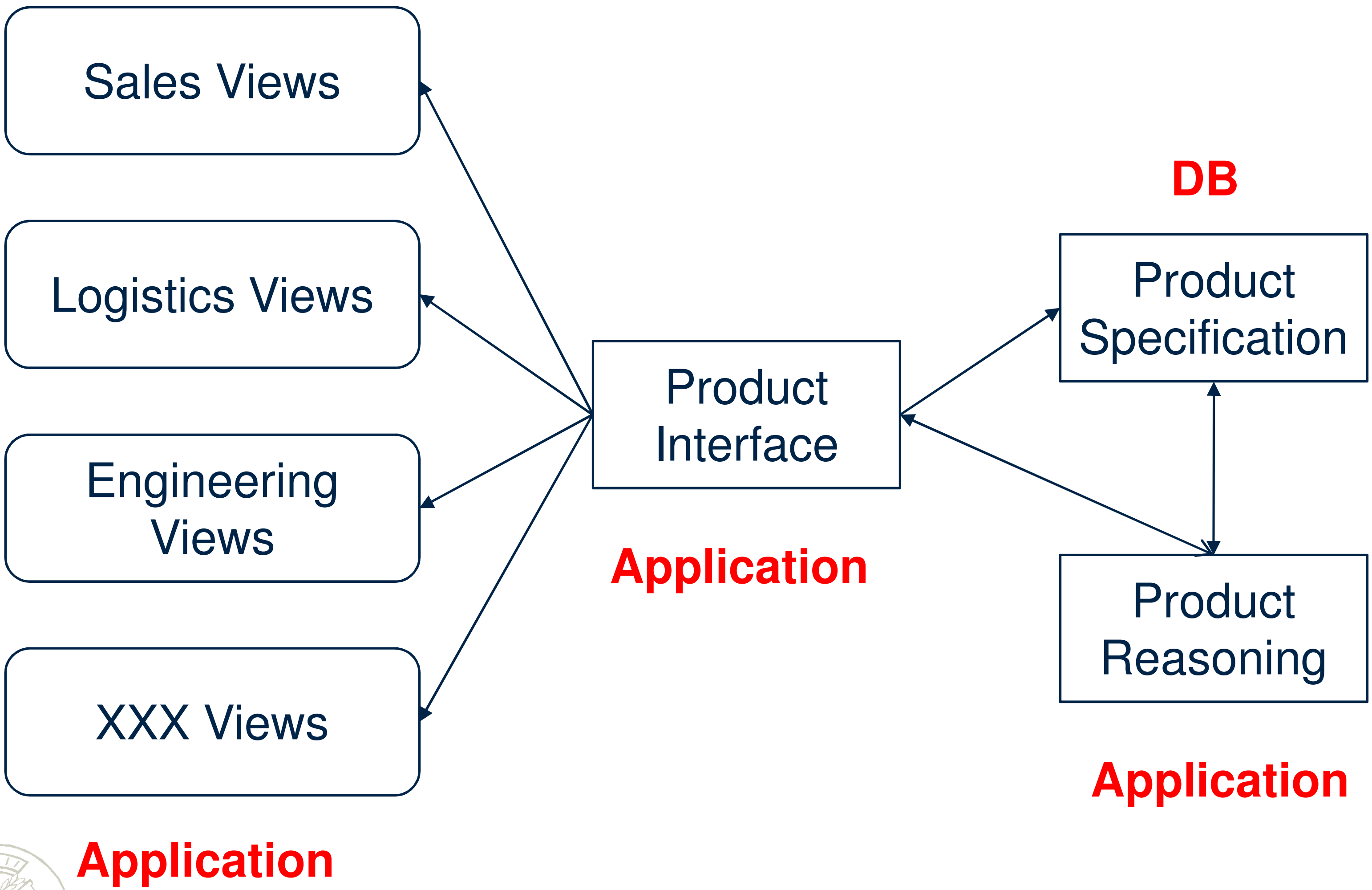
The car industry:

Opel Corsa

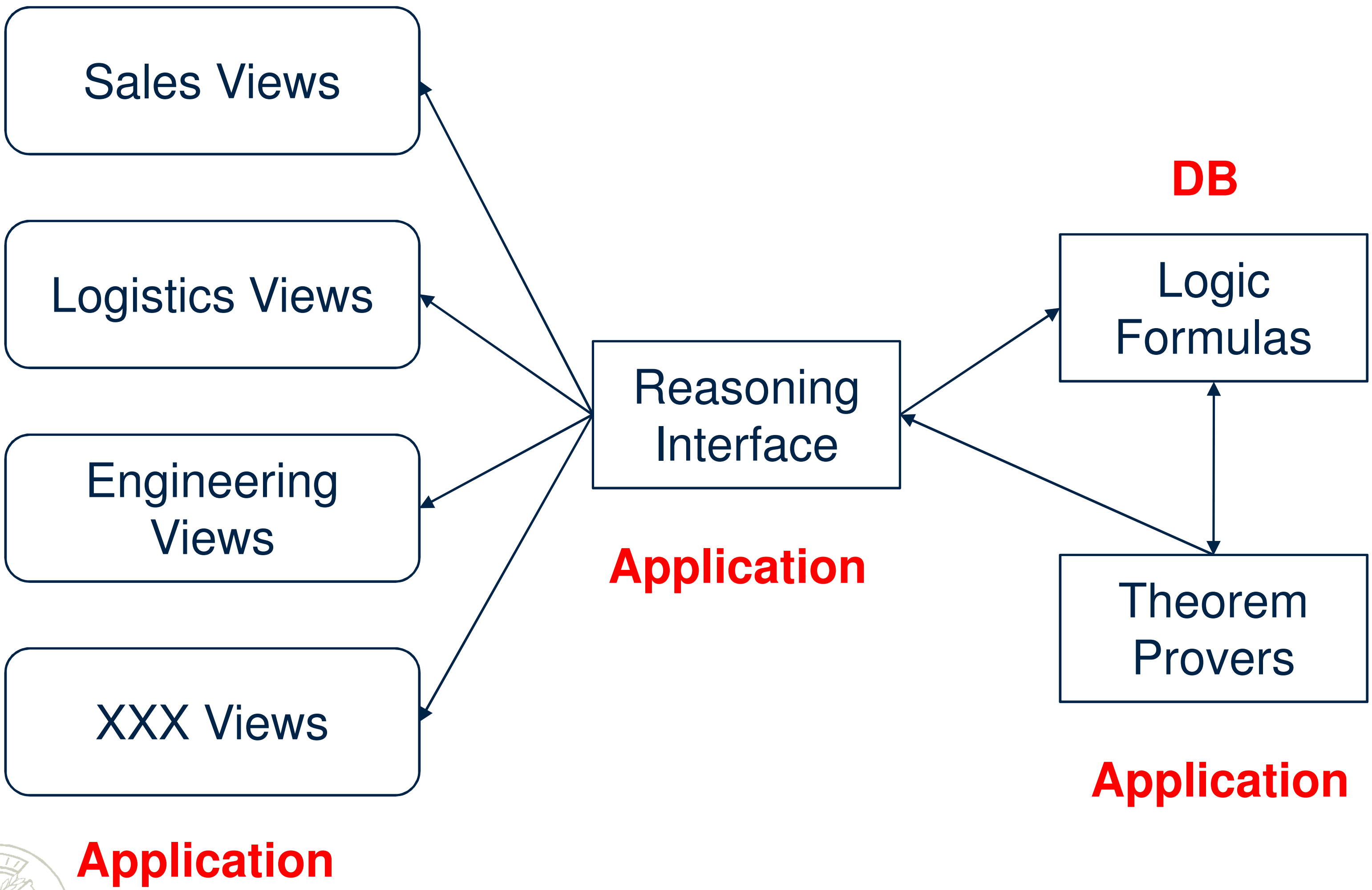
Today's Architecture



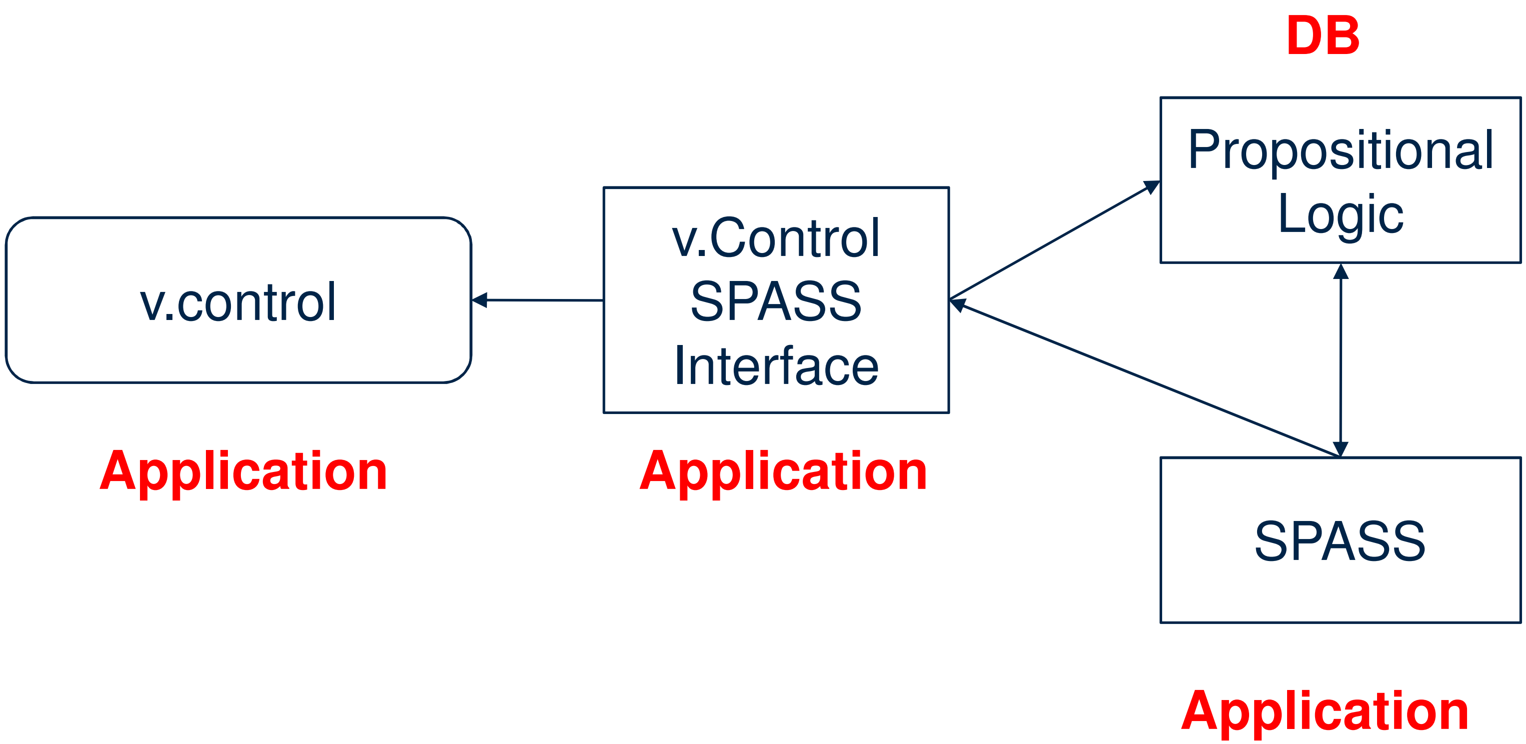
Future's Architecture



Scientific View



Concrete Example



max planck institut informatik

Opel Corsa

PROSTEP IMP

in cooperation with

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

Daniel Doenigus, PROSTEP IMP

Propositional Logic

- 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

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

max planck institut informatik

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, $M \cup \{\neg P\}$));

$\neg A \vee \neg B \vee E$
$\neg A \vee B$
$\neg E$
$A \vee D$

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

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

$$\begin{aligned}
&\text{4-Holes} \Rightarrow \text{Wheels} \\
&\text{5-Holes} \Rightarrow \text{Wheels} \\
&\text{4-Holes} \Rightarrow \neg\text{5-Holes} \\
&\text{5-Holes} \Rightarrow \neg\text{4-Holes}
\end{aligned}$$

$$\begin{aligned}
&\text{Diesel} \Rightarrow \text{Engines} \\
&\text{Gasoline} \Rightarrow \text{Engines} \\
&\text{Diesel} \Rightarrow \neg\text{Gasoline} \\
&\text{Gasoline} \Rightarrow \neg\text{Diesel}
\end{aligned}$$

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

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 2009: v.control + SPASS approx. 3000 nodes
- in x years: for a reasonable product approx. 60000 nodes

- SAT Seminar:

  http://www.mpi-inf.mpg.de/departments/rg1/teaching/sat-ws10/

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

Thank you for your attention