

# WeProve

Simon Bischof   Jan Haag   Adrian Herrmann   Lin Jin   Tobias Schlumberger   Matthias Schnetz

Institut für Theoretische Informatik

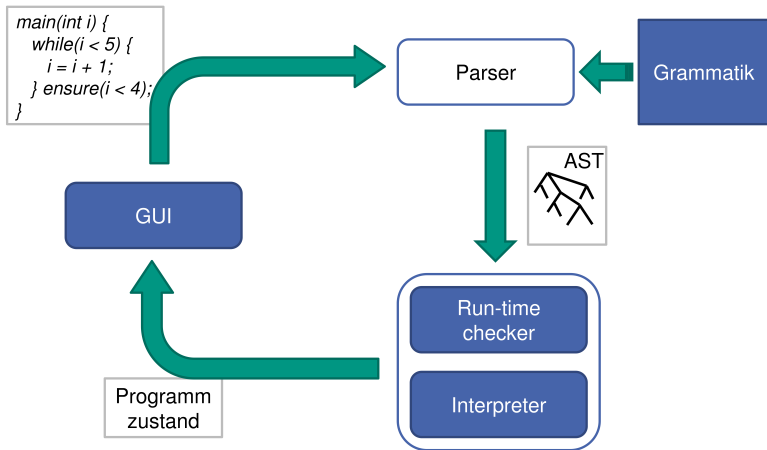


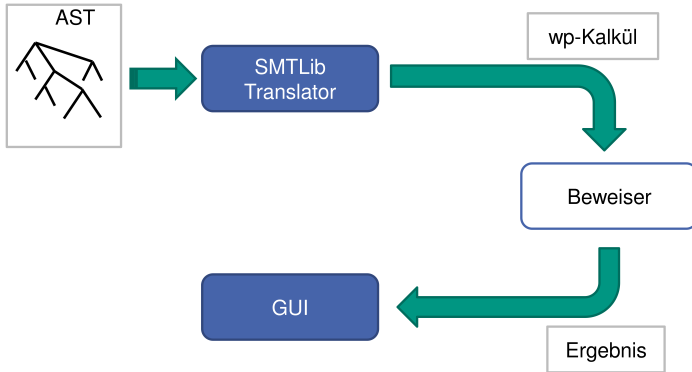
Aufgabenstellung

Kenndaten

Zukunft von W<sub>e</sub>Prove

# Aufgabenstellung: Parser und Interpreter





Zur Verifikation eines Programmes sind zwei Schritte nötig:

1. Ziel des Programms klären

- Was ist der Sinn dieses Programms?
- Welches Endergebnis erwarte Ich?
- Wie soll dieses Ergebnis erreicht werden?

Zur Verifikation eines Programmes sind zwei Schritte nötig:

1. Ziel des Programms klären
  - Was ist der Sinn dieses Programms?
  - Welches Endergebnis erwarte ich?
  - Wie soll dieses Ergebnis erreicht werden?
2. Das Programm um beweisbare Annotationen erweitern
  - Was gilt für die Variablen?
  - Wie lauten geeignete Invarianten für Schleifen?

## 1. Ziel des Programms klären

- Was ist der Sinn dieses Programms?
- Welches Endergebnis erwarte Ich?
- Wie soll dieses Ergebnis erreicht werden?

1. Lorem ipsum dolor sit amet...

## 2. Das Programm um beweisbare Annotationen erweitern

- Was gilt für die Variablen?
- Wie lauten geeignete Invarianten für Schleifen?

2. Lorem ipsum dolor sit amet...



■ 6 Entwickler	9.500 Zeilen
■ 17.000 LOC	Code
■ 100 Klassen	
■ 15 Pakete	3.500 Zeilen
■ Lauffähig unter:	generierter Code
■ Windows XP & Windows 7	
■ Linux	3.500 Zeilen
■ Mac OS X	Code aus der QS
■ Beweisbare Programme:	
■ Summe von 1 bis n	800 Zeilen
■ Russische Multiplikation	Dokumentation

- Einsatzgebiete von formaler Verifikation:
    - Hardware-Bereich: Entwicklung von Prozessoren
    - Software-Bereich: Systeme, deren Zuverlässigkeit wichtig ist
  - Es existieren andere Tools mit größerem Funktionsumfang und Entwicklerteams mit mehr Erfahrung/Mitteln
    - KeY
    - Isabelle/HOL
- ⇒ Tool ist ein guter Einstieg in den Themenbereich der formalen Verifikation (v.a. auch für Studenten)
- ⇒ Lizenzierung unter BSD Lizenz

# Tool Demonstration

TODO: Bild