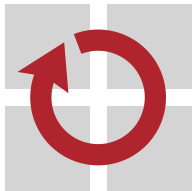
Ein Benchmarkgenerator zur Bewertung von WCET-Analysatoren

16. November 2017

Christian Eichler
eichler@cs.fau.de



Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Motivation

Motivation: WCET-Analysatoren

aiT

PLATIN

RapiTime

XMOS

TAWA

- Einsatz von WCET-Analysatoren um Rechtzeitigkeit von Berechnungen zu garantieren
- **AbsInt aiT** für **Airbus A380**

Motivation: WCET-Analysatoren

aiT

PLATIN

RapiTime

XMOS

TAWA

- Einsatz von WCET-Analysatoren um Rechtzeitigkeit von Berechnungen zu garantieren
- **AbsInt aiT** für **Airbus A380**

☞ **Problem: Genauigkeit & Gültigkeit** sind unbekannt

Motivation: Beispiel

Problemstellung:
Bestimmung der WCET aus existierendem Code

```
int f(int x, int y) {
  if(0 == x) { g(); }
  else { h(); }

  for(int i = 0; i < x; ++i) {
    for(int j = i; j < y; ++j) {
      k();
    }
  }

  if(0 != x) { m(); }
}
```

WCET der Funktion f?

Motivation: Beispiel

Problemstellung:
Bestimmung der WCET aus existierendem Code

```
int f(int x, int y) {
  if(0 == x) { g(); }
  else { h(); }

  for(int i = 0; i < x; ++i) {
    for(int j = i; j < y; ++j) {
      k();
    }
  }

  if(0 != x) { m(); }
}
```

WCET der Funktion f?

- Eingabeabhängige Berechnungen

Motivation: Beispiel

Problemstellung:
Bestimmung der WCET aus existierendem Code

```
int f(int x, int y) {
  if(0 == x) { g(); }
  else { h(); }

  for(int i = 0; i < x; ++i) {
    for(int j = i; j < y; ++j) {
      k();
    }
  }

  if(0 != x) { m(); }
}
```

WCET der Funktion f?

- Eingabeabhängige Berechnungen
- Verschachtelte Schleifen

Motivation: Beispiel

Problemstellung:
Bestimmung der WCET aus existierendem Code

```
int f(int x, int y) {
  if(0 == x) { g(); }
  else { h(); }

  for(int i = 0; i < x; ++i) {
    for(int j = i; j < y; ++j) {
      k();
    }
  }

  if(0 != x) { m(); }
}
```

WCET der Funktion f?

- Eingabeabhängige Berechnungen
- Verschachtelte Schleifen
- Pfade gegenseitigen Ausschlusses

Motivation: Beispiel

Problemstellung:
Bestimmung der WCET aus existierendem Code

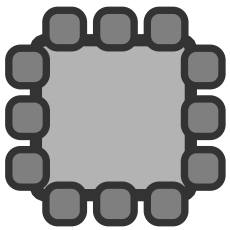
```
int f(int x, int y) {
  if(0 == x) { g(); }
  else { h(); }

  for(int i = 0; i < x; ++i) {
    for(int j = i; j < y; ++j) {
      k();
    }
  }

  if(0 != x) { m(); }
}
```

WCET der Funktion f?

- Eingabeabhängige Berechnungen
- Verschachtelte Schleifen
- Pfade gegenseitigen Ausschlusses
- Hardwaremodellierung



Motivation: Beispiel

Problemstellung:
Bestimmung der WCET aus existierendem Code

```
int f(int x, int y) {
  if(0 == x) { g(); }
  else { h(); }

  for(int i = 0; i < x; ++i) {
    for(int j = i; j < y; ++j) {
      k();
    }
  }

  if(0 != x) { m(); }
}
```

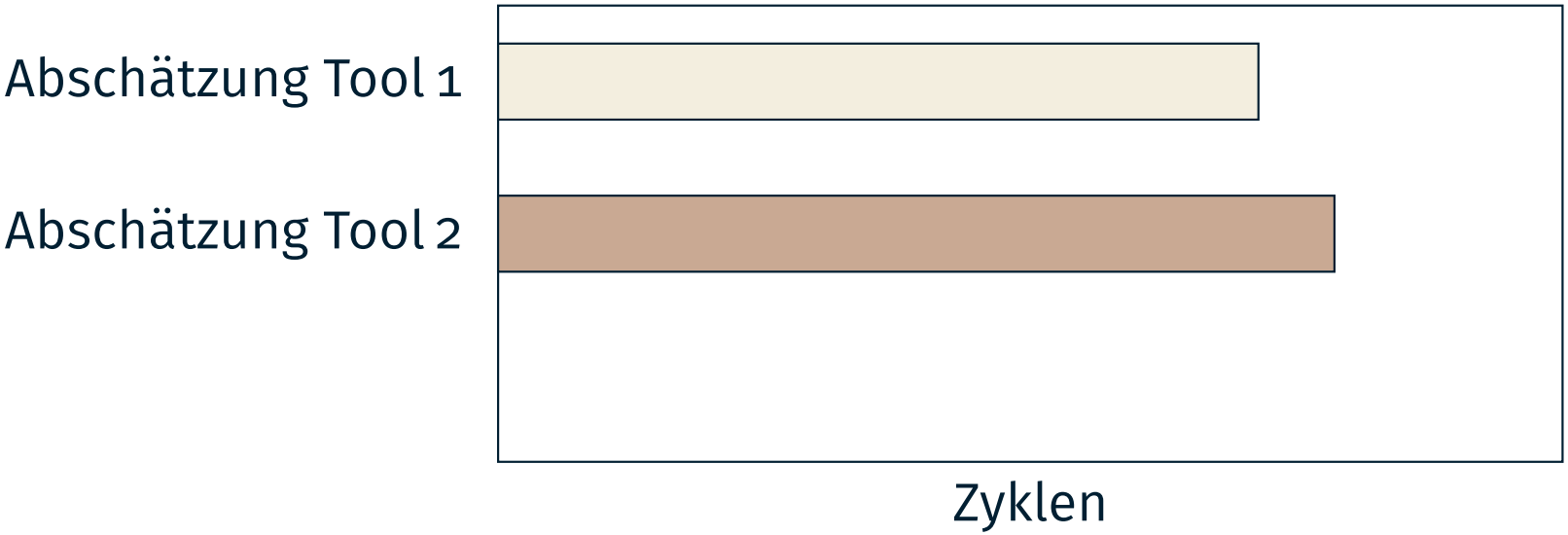
WCET der Funktion f?

- Eingabeabhängige Berechnungen
- Verschachtelte Schleifen
- Pfade gegenseitigen Ausschlusses
- Hardwaremodellierung

☞ **Fehlendes Wissen** über Flussinformation führt zu Überschätzungen der WCET

Motivation: Notwendigkeit von Vergleichswerten

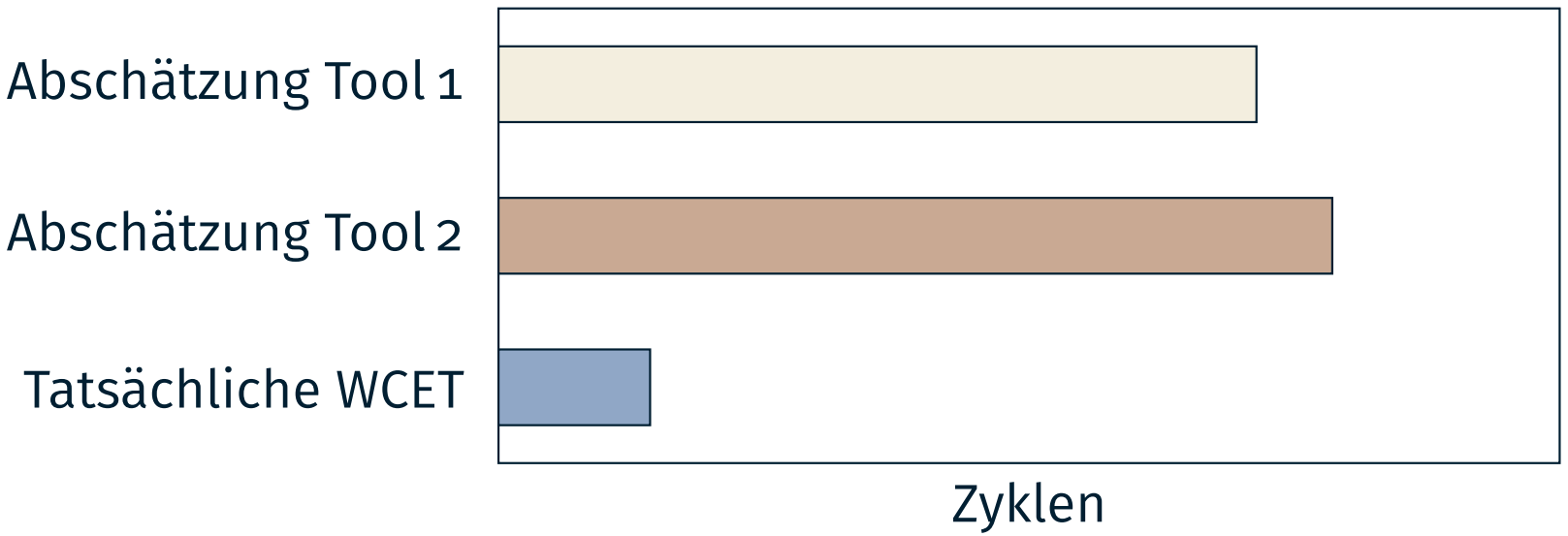
Motivation: Notwendigkeit von Vergleichswerten



Motivation: Notwendigkeit von Vergleichswerten

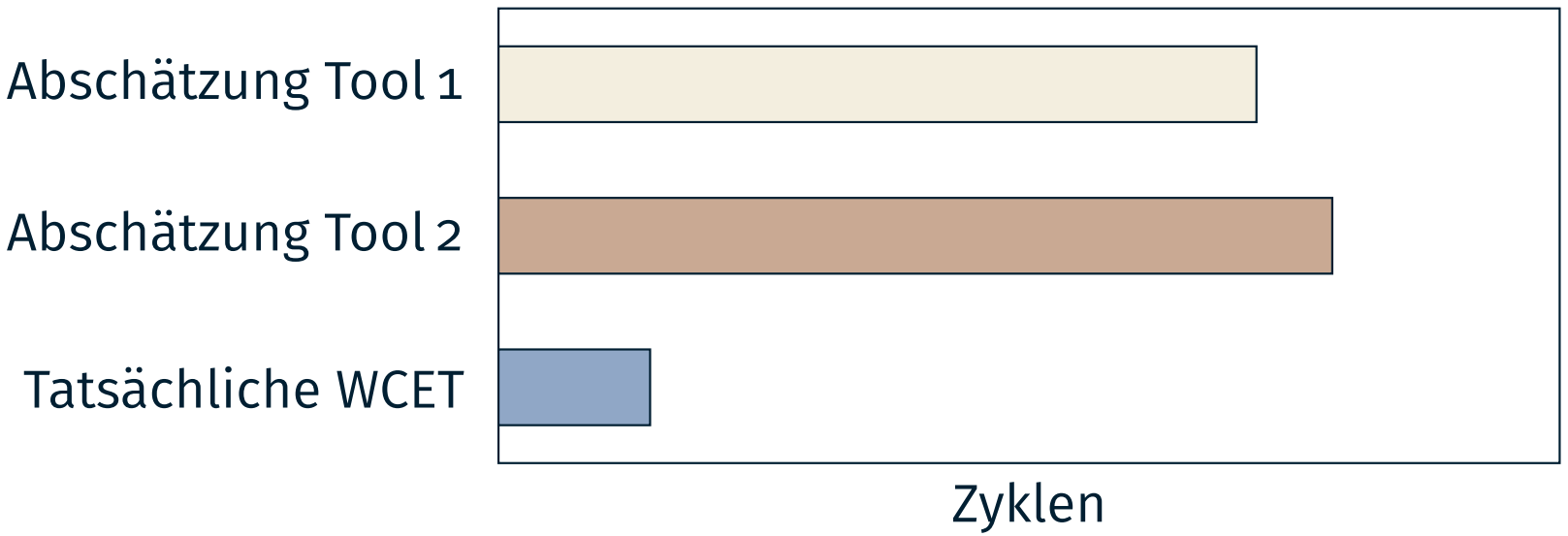
- *Herkömmlicher Ansatz:*
  Relativer Vergleich zwischen WCET-Analysatoren

Motivation: Notwendigkeit von Vergleichswerten



- *Herkömmlicher Ansatz:*
  Relativer Vergleich zwischen WCET-Analysatoren
- *Ziel:* Absoluter Vergleich mit WCET

Motivation: Notwendigkeit von Vergleichswerten



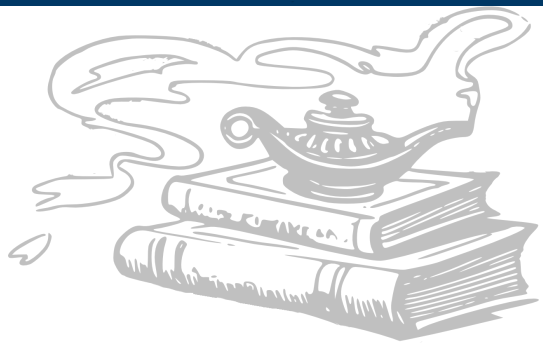
- *Herkömmlicher Ansatz:*
  Relativer Vergleich zwischen WCET-Analysatoren
- *Ziel:* Absoluter Vergleich mit WCET

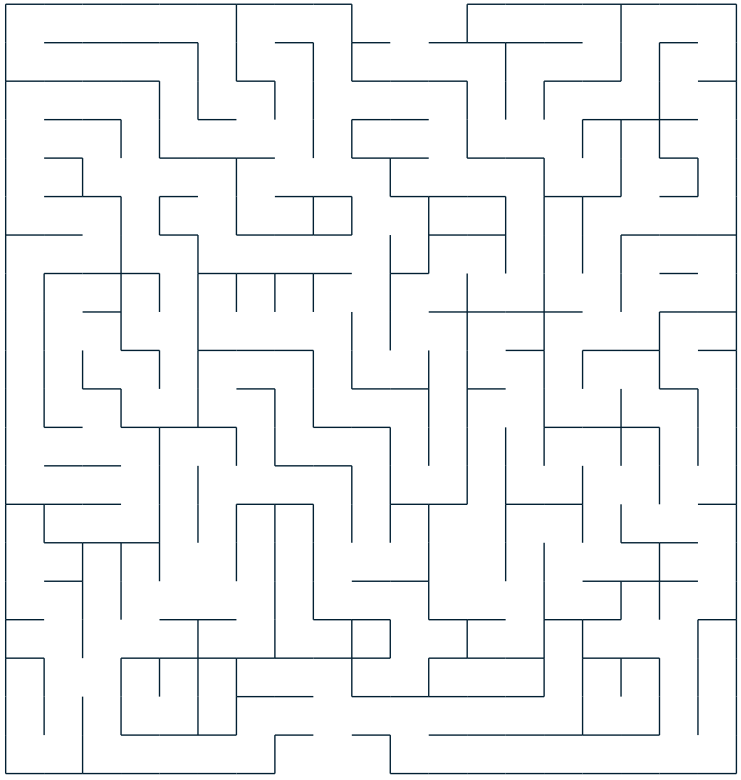
☞ GENE: **Generieren von Benchmarks** mit bekannten Eigenschaften

Inhaltsverzeichnis

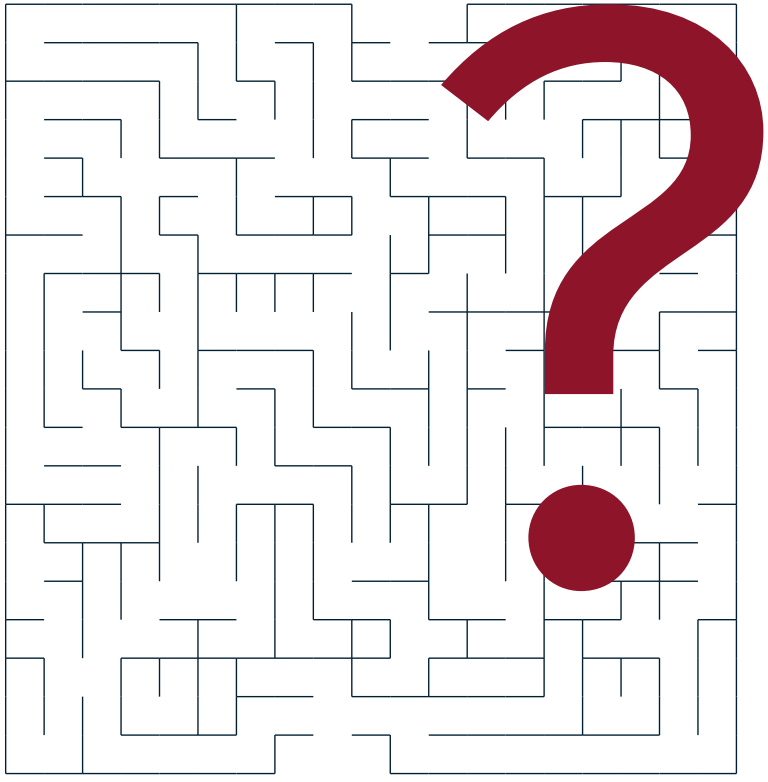
GenE-Ansatz

Grundidee

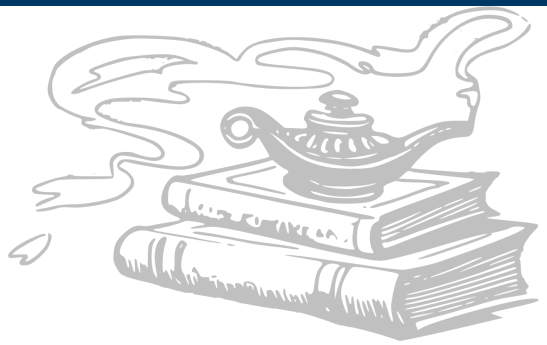


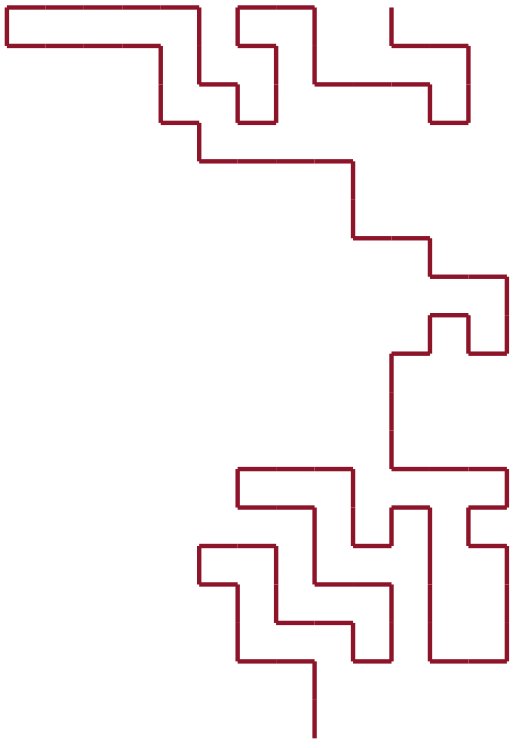


Grundidee

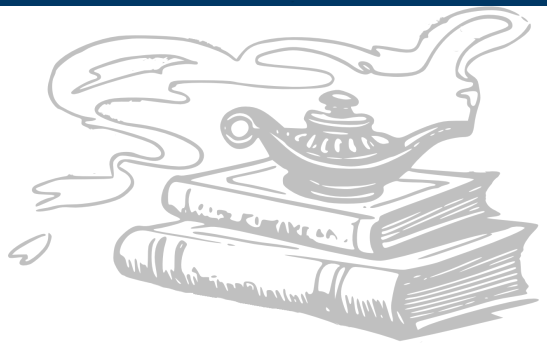


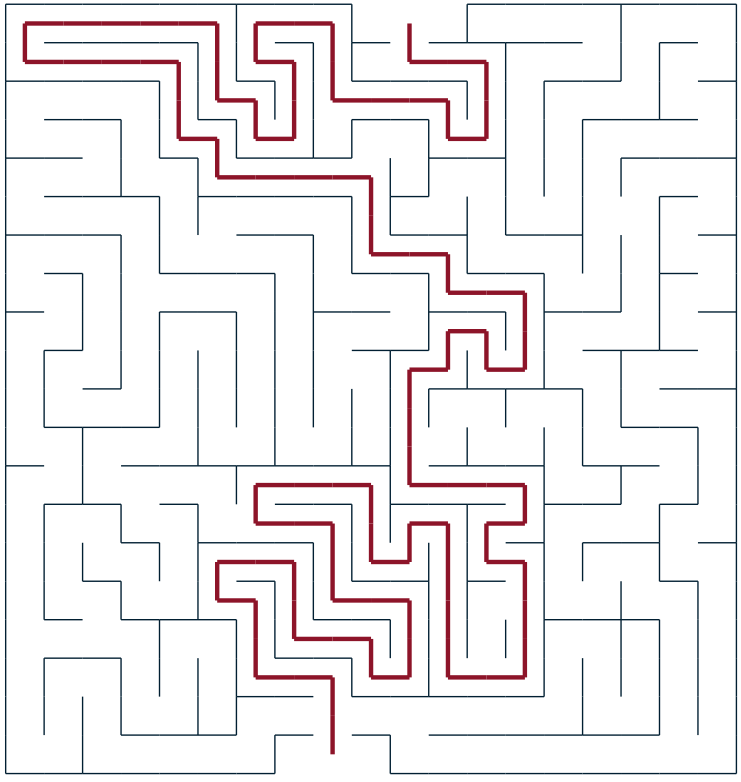
Grundidee



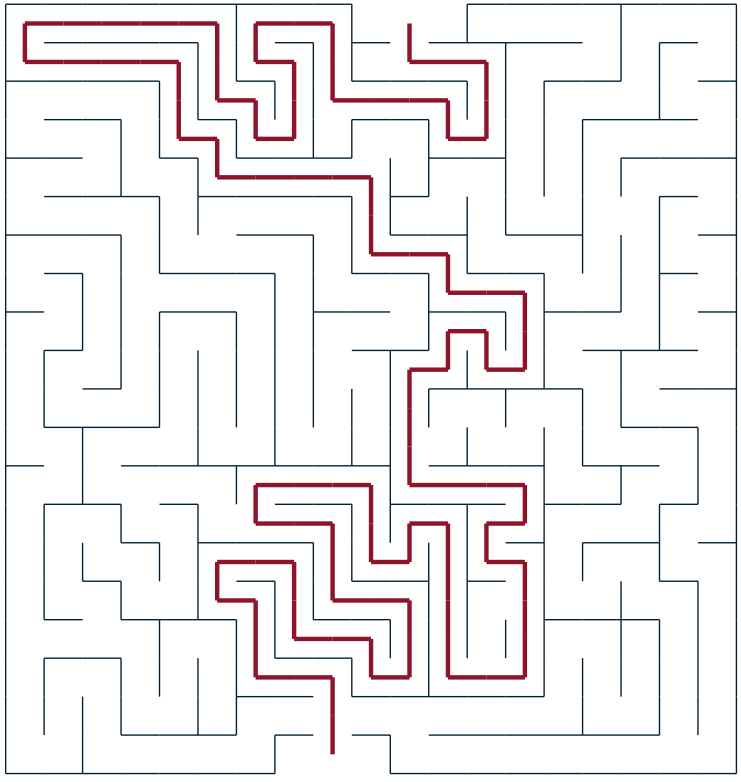


Grundidee



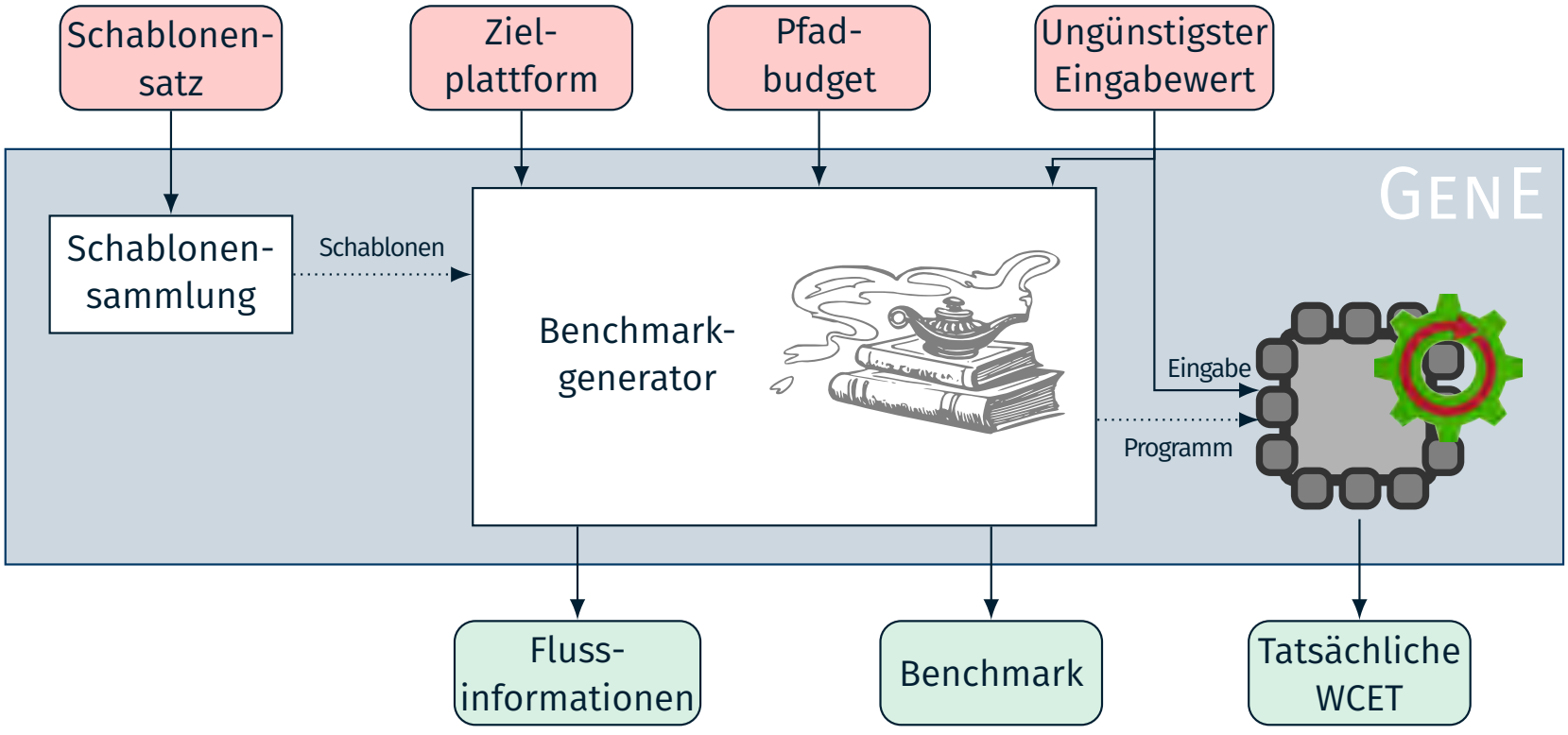


Grundidee

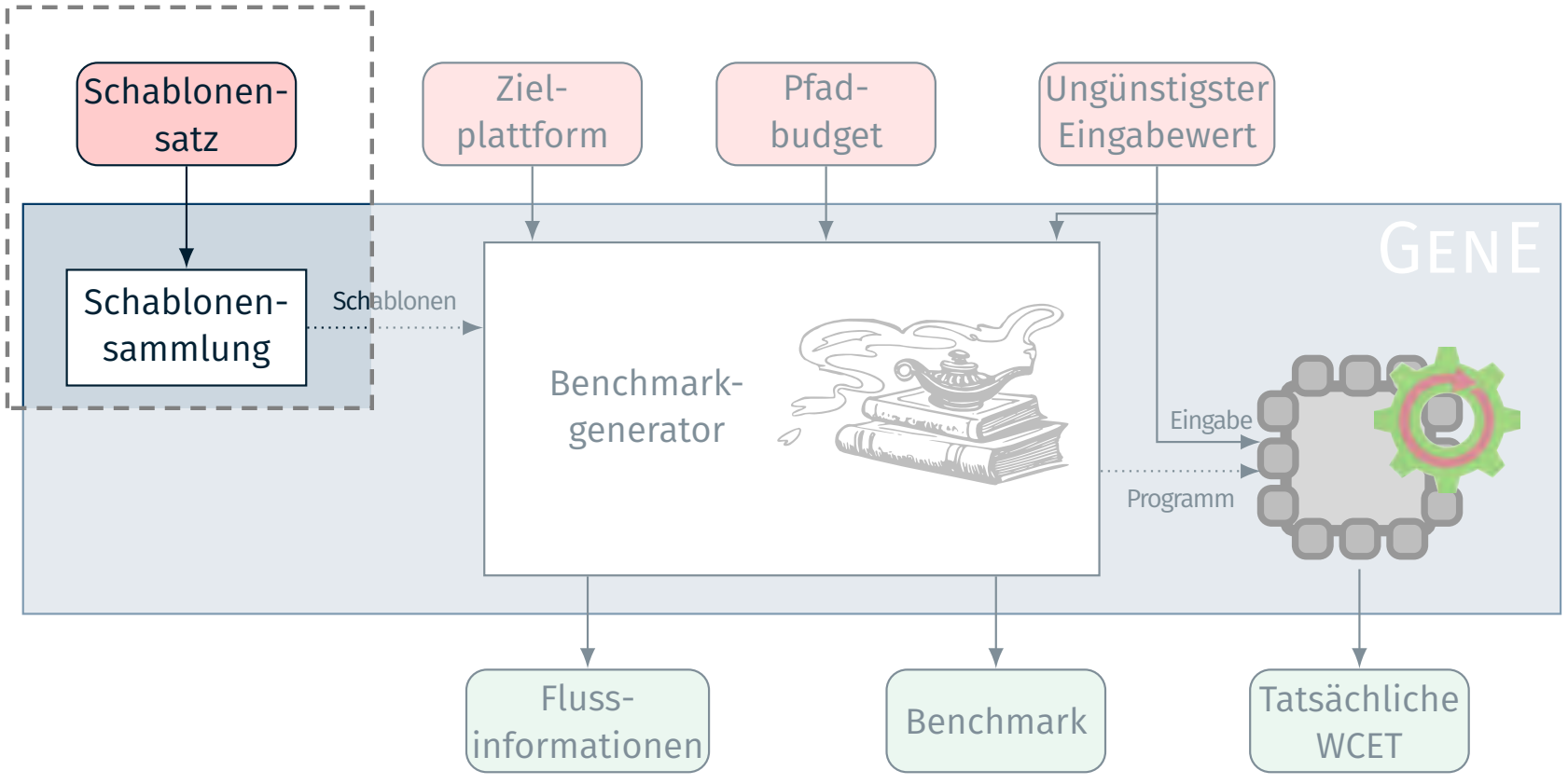


☞ Längster Pfad konstruktionsbedingt bekannt

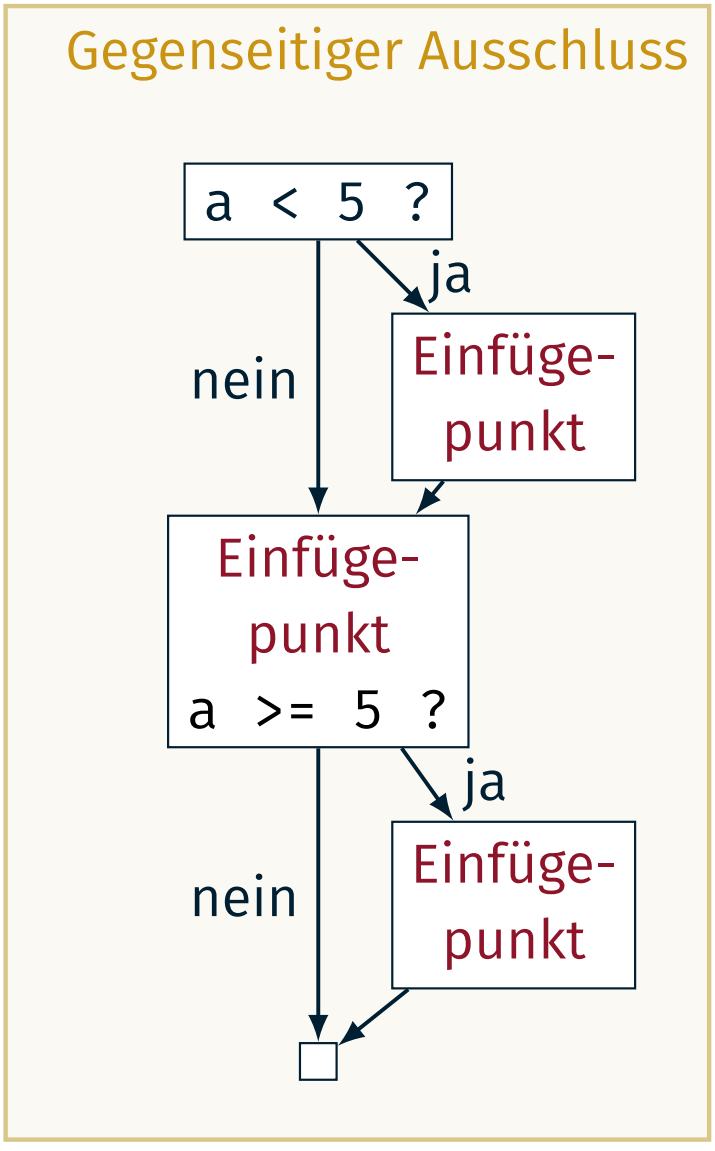
Überblick

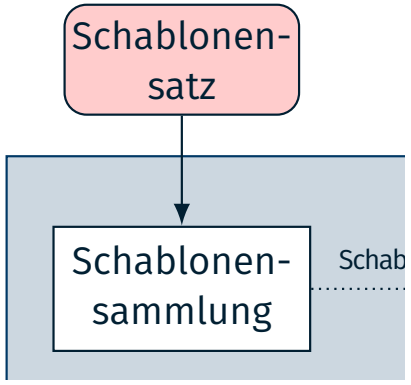

Schablonen-
satz
Ziel-
plattform
Pfad-
budget
Ungünstigster
Eingabewert
GenE
Schablonen-
sammlung
Schablonen
Benchmark-
generator
Eingabe
Programm
Fluss-
informationen
Benchmark
Tatsächliche
WCET

Überblick


Schablonen-
satz
Schablonen-
sammlung
Schablonen
Ziel-
plattform
Pfad-
budget
Ungünstigster
Eingabewert
GenE
Benchmark-
generator
Eingabe
Programm
Fluss-
informationen
Benchmark
Tatsächliche
WCET

Schablonen





Schablonen sollten

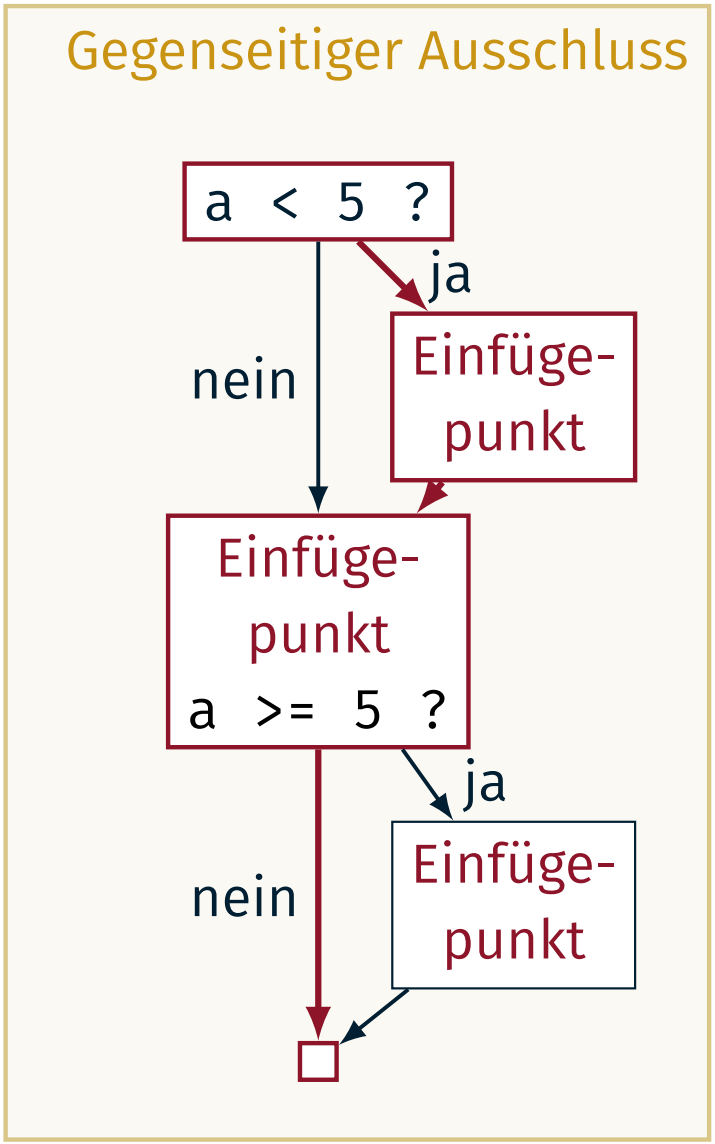
- **Realistisch**
- **Herausfordernd**

sein

Schablonen extrahiert aus:

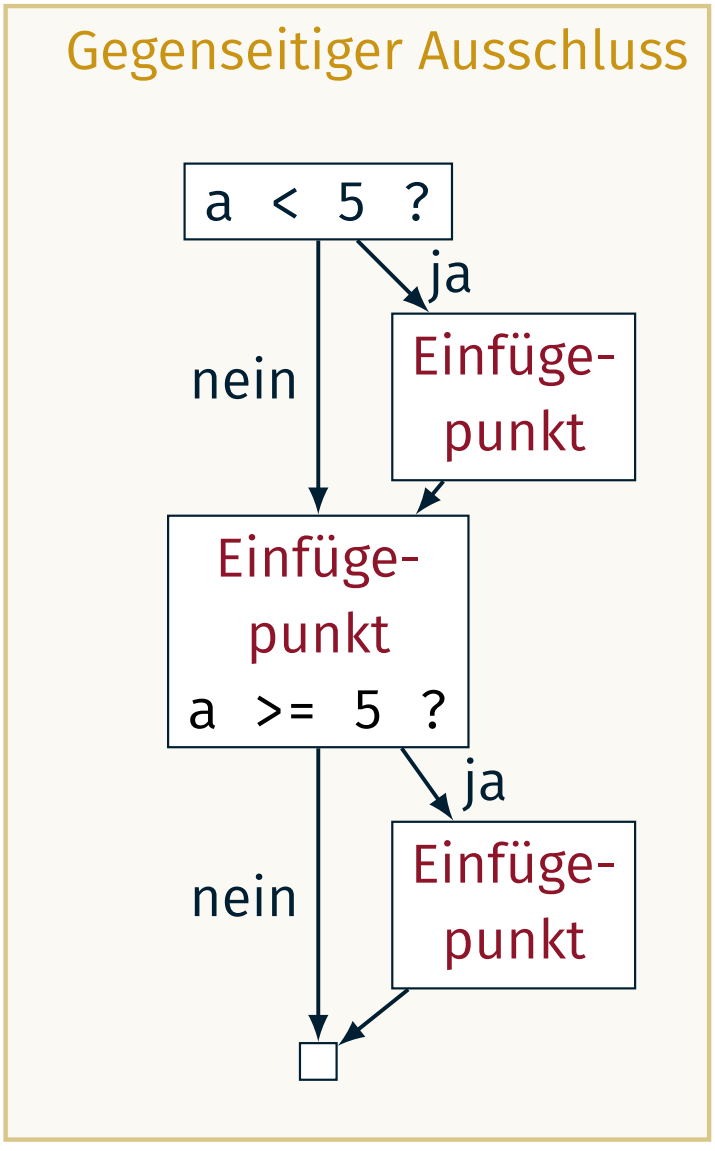
1. **Existierenden Benchmarks**
2. **Industrieanwendungen**
3. Literatur

Schablonen

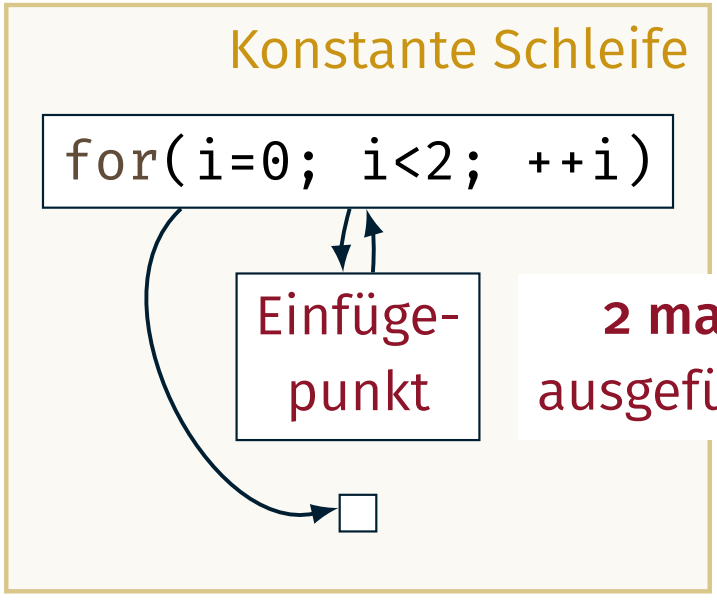


- Schablonen **kennen ihre Flussinformation**

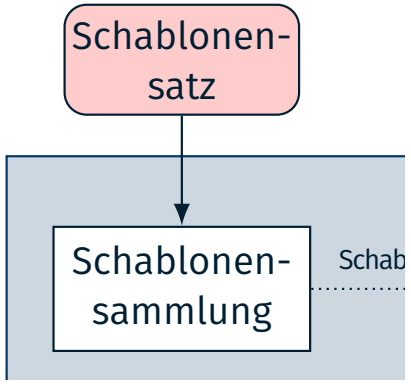
Schablonen



- Schablonen **kennen ihre Flussinformation**
- Schablonen sind **parametrierbar**

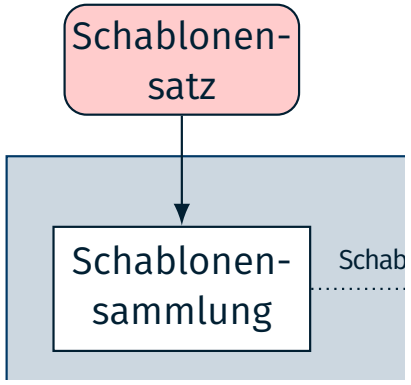


Schablonensatz



- Besteht aus verschiedenen Schablonen, die in Kombination zu Programmen mit bestimmten Eigenschaften führen
- Evaluieren der **Hardwareanalyse**: Einfacher Kontrollfluss
- Evaluieren der **Flussanalyse**: Komplexer Kontrollfluss
- Aber auch: Schablonensätze zur gezielten Überprüfung bestimmter Fähigkeiten
  - ☞ Ein einziger Schleifentyp

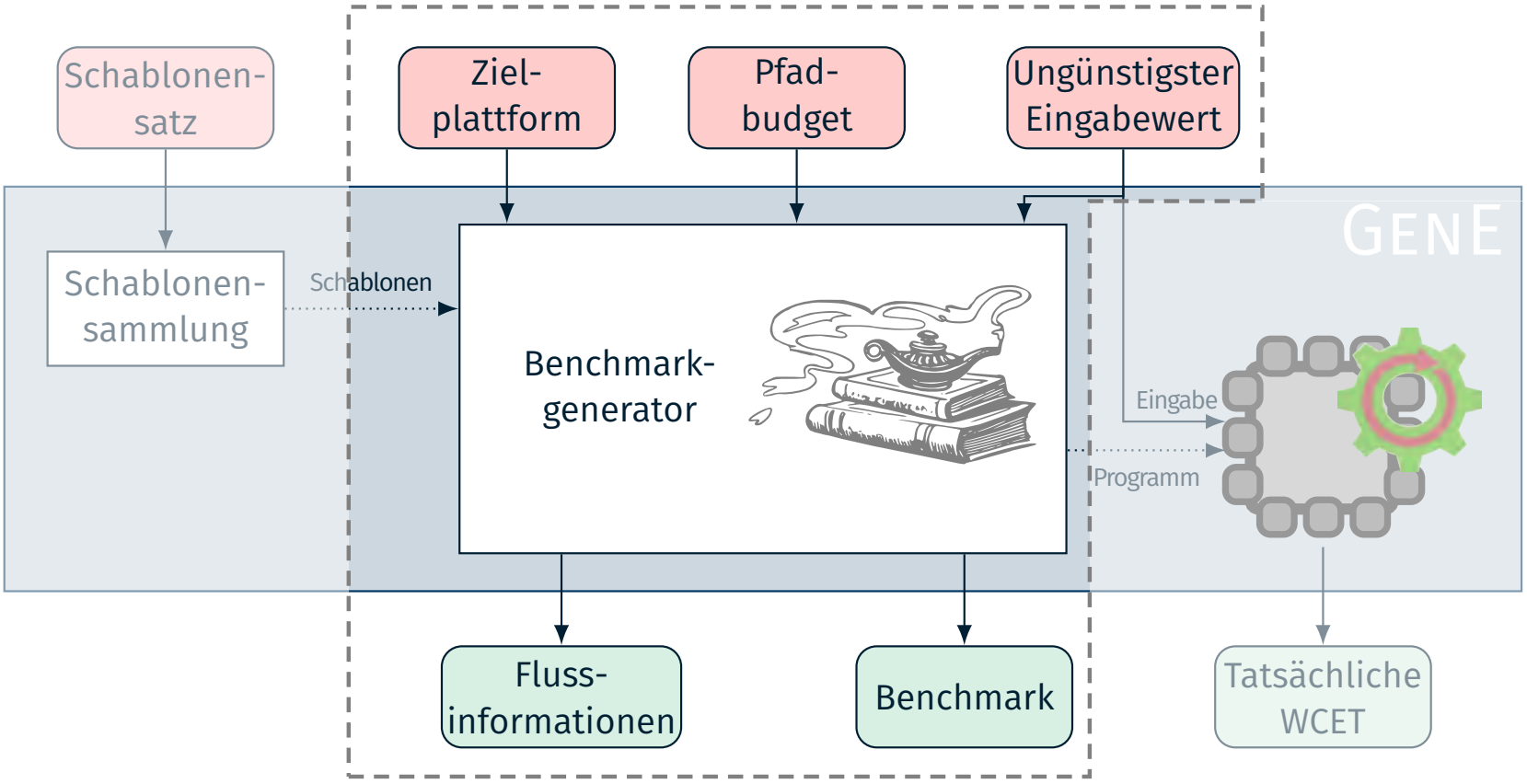
Schablonensatz



- Besteht aus verschiedenen Schablonen, die in Kombination zu Programmen mit bestimmten Eigenschaften führen
- Evaluieren der **Hardwareanalyse**: Einfacher Kontrollfluss
- Evaluieren der **Flussanalyse**: Komplexer Kontrollfluss
- Aber auch: Schablonensätze zur gezielten Überprüfung bestimmter Fähigkeiten
  - ☞ Ein einziger Schleifentyp

✓ Benchmarks **an die Evaluation angepasst**

Überblick


Schablonen-
satz
Ziel-
plattform
Pfad-
budget
Ungünstigster
Eingabewert
GenE
Schablonen-
sammlung
Schablonen
Benchmark-
generator
Eingabe
Programm
Fluss-
informationen
Benchmark
Tatsächliche
WCET

Pfadbudget: Generieren des längsten Pfads

- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads

Pfadbudget: Generieren des längsten Pfads

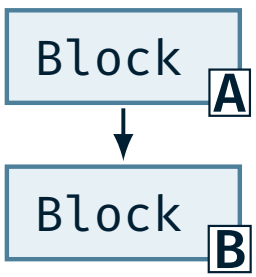
- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt

Pfadbudget: Generieren des längsten Pfads

- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt
- **Instruktionen weisen nicht-konstante Ausführungszeiten auf**

Pfadbudget: Generieren des längsten Pfads

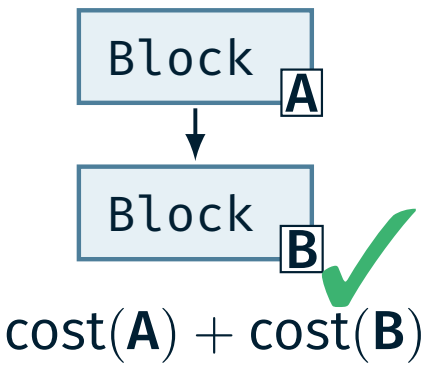
- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt
- **Instruktionen weisen nicht-konstante Ausführungszeiten auf**



$\text{cost}(\mathbf{A}) + \text{cost}(\mathbf{B})$

Pfadbudget: Generieren des längsten Pfads

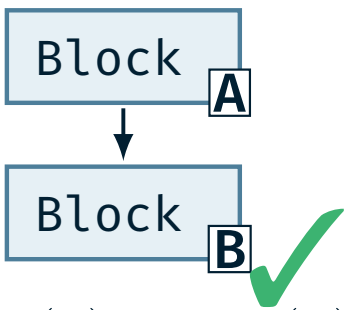
- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt
- **Instruktionen weisen nicht-konstante Ausführungszeiten auf**



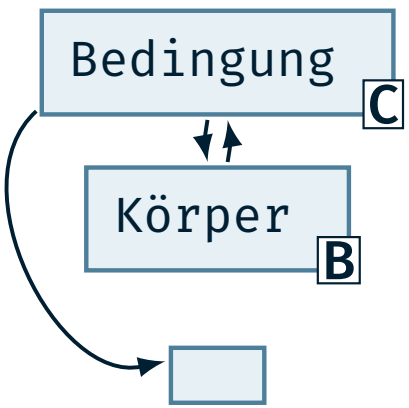
$\text{cost}(\mathbf{A}) + \text{cost}(\mathbf{B})$

Pfadbudget: Generieren des längsten Pfads

- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt
- **Instruktionen weisen nicht-konstante Ausführungszeiten auf**



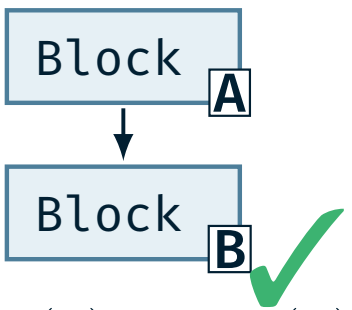
$\text{cost}(\mathbf{A}) + \text{cost}(\mathbf{B})$



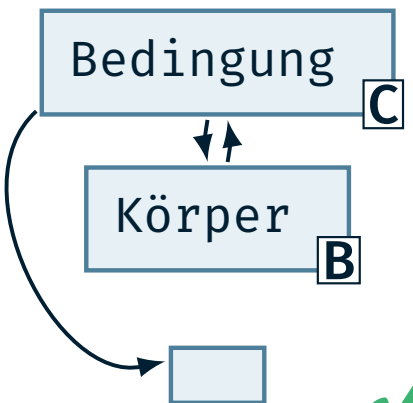
$(N+1) \cdot \text{cost}(\mathbf{C})$
$+N \cdot \text{cost}(\mathbf{B})$

Pfadbudget: Generieren des längsten Pfads

- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt
- **Instruktionen weisen nicht-konstante Ausführungszeiten auf**



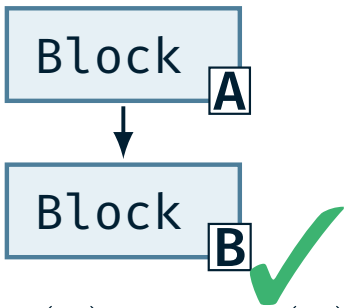
$\text{cost}(\mathbf{A}) + \text{cost}(\mathbf{B})$



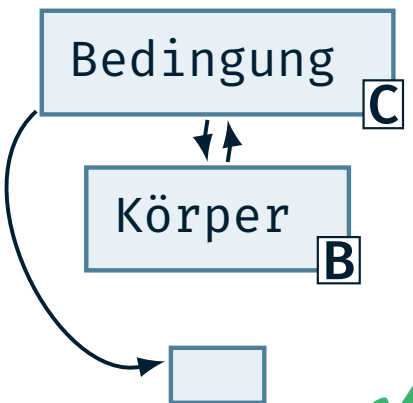
$(N+1) \cdot \text{cost}(\mathbf{C})$
$+N \cdot \text{cost}(\mathbf{B})$

Pfadbudget: Generieren des längsten Pfads

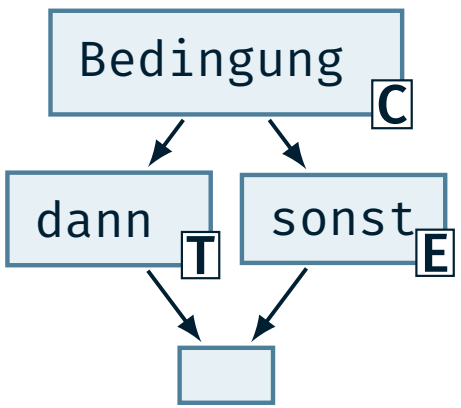
- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt
- **Instruktionen weisen nicht-konstante Ausführungszeiten auf**



$\text{cost}(\mathbf{A}) + \text{cost}(\mathbf{B})$



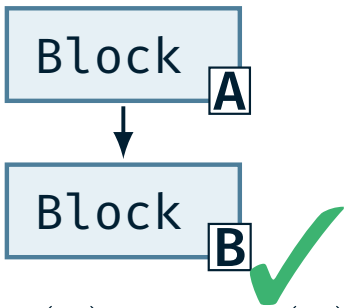
$(N{+}1) \cdot \text{cost}(\mathbf{C})$
$+N \cdot \text{cost}(\mathbf{B})$



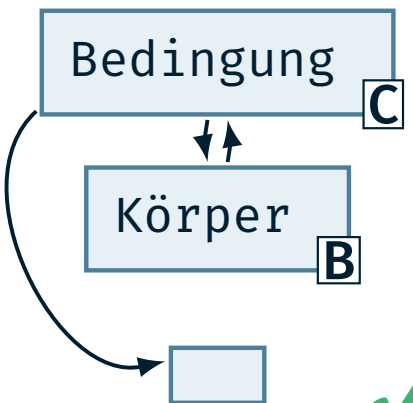
$\text{cost}(\mathbf{C}) +$
$\max(\text{cost}(\mathbf{T}), \text{cost}(\mathbf{E}))$

Pfadbudget: Generieren des längsten Pfads

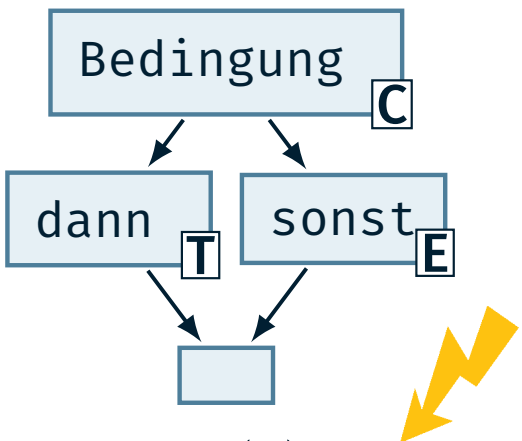
- Beschreibt die **Anzahl der ausgeführten Assembly-Instruktionen** entlang des längsten Pfads
- *Ziel:* Zeitlich längster Pfad konstruktionsbedingt bekannt
- **Instruktionen weisen nicht-konstante Ausführungszeiten auf**



$\text{cost}(\mathbf{A}) + \text{cost}(\mathbf{B})$



$(N{+}1) \cdot \text{cost}(\mathbf{C})$
$+N \cdot \text{cost}(\mathbf{B})$



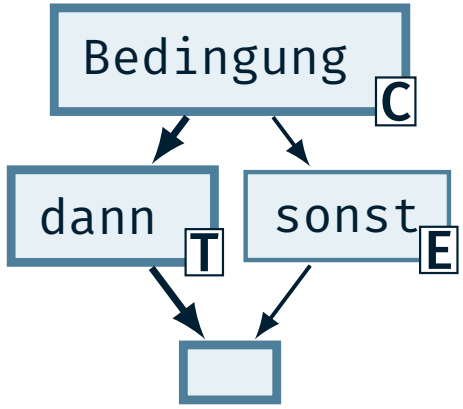
$\text{cost}(\mathbf{C}) +$
$\max(\text{cost}(\mathbf{T}), \text{cost}(\mathbf{E}))$

Pfadbudget: Übergewichtung

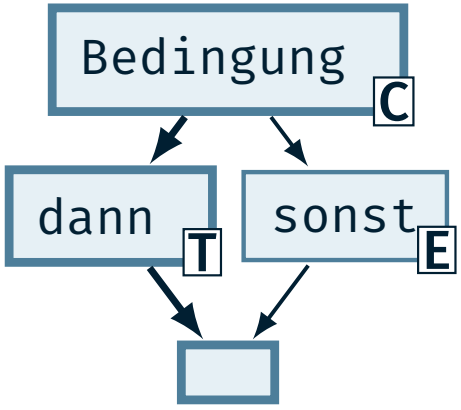
Herausforderung: Moderne Hardwarefeatures

- Caches
- Pipelining

⇒ Ausführdauer einer Instruktion **abhängig von zuvor ausgeführten Instruktionen**



Pfadbudget: Übergewichtung



Herausforderung: Moderne Hardwarefeatures

- Caches
- Pipelining

$\Rightarrow$ Ausführdauer einer Instruktion **abhängig von zuvor ausgeführten Instruktionen**

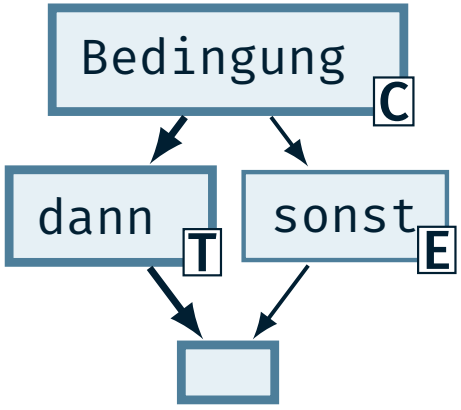
Zielsetzung: $t_{\text{min}}(\text{dann}) \geq t_{\text{max}}(\text{sonst})$

Zeitbedarf einer Instruktion: $t_{\texttt{instr}} = t_{\text{icache}} + t_{\text{dcache}} + t_{\text{pipeline}} + t_{\text{calc}}$

```
and r4, r5
```
$t_{\text{min}}(\text{and}) = 1cy$

```
ldr r0, [r0, 42]
```
$t_{\text{max}}(\texttt{ldr}) = 7cy$

Pfadbudget: Übergewichtung



Herausforderung: Moderne Hardwarefeatures

- Caches
- Pipelining

⇒ Ausführdauer einer Instruktion **abhängig von zuvor ausgeführten Instruktionen**

Zielsetzung: $t_{\text{min}}(\text{dann}) \geq t_{\text{max}}(\text{sonst})$

Zeitbedarf einer Instruktion: $t_{\texttt{instr}} = t_{\text{icache}} + t_{\text{dcache}} + t_{\text{pipeline}} + t_{\text{calc}}$

```
and r4, r5
```
$t_{\text{min}}(\texttt{and}) = 1cy$

```
ldr r0, [r0, 42]
```
$t_{\text{max}}(\texttt{ldr}) = 7cy$

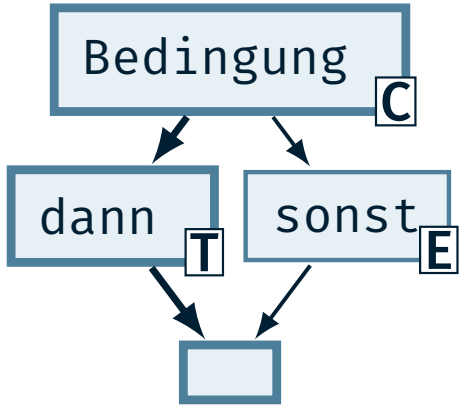
- Bestimmen des Übergewichtungsfaktors aus den Extremwerten der Ausführdauern aller Instruktionen
  - ☞ Übergewichtungsfaktor: $\mathcal{F} = 7$
    - 7 x beliebige Instruktionen kompensieren 1 x Instruktion selbst im ungünstigsten Fall

Pfadbudget: Übergewichtung

Herausforderung: Moderne Hardwarefeatures

- Caches
- Pipelining

⇒ Ausführdauer einer Instruktion **abhängig von zuvor ausgeführten Instruktionen**



Zielsetzung: $t_{\text{min}}(\text{dann}) \geq t_{\text{max}}(\text{sonst})$

Zeitbedarf einer Instruktion: $t_{\texttt{instr}} = t_{\text{icache}} + t_{\text{dcache}} + t_{\text{pipeline}} + t_{\text{calc}}$

```
and r4, r5
```
$t_{\text{min}}(\texttt{and}) = 1cy$

```
ldr r0, [r0, 42]
```
$t_{\text{max}}(\texttt{ldr}) = 7cy$

- Bestimmen des Übergewichtungsfaktors aus den Extremwerten der Ausführdauern aller Instruktionen

  ☞ Übergew

  ☞ Ausreichend groß wählen: $\mathcal{F} = 25$

  - 7 x bel[illegible] selbst im ungünstigsten Fall

Pfadbudget: Übergewichtung

Lösung
Kompensieren von Unsicherheiten durch **Übergewichtung**

Ungünstigster Eingabewert: $\boxed{\cdot\,\cdot}\boxed{0}\boxed{1}$

Übergewichtungsfaktor: $\mathcal{F} = 25$

Pfadbudget: Übergewichtung

Lösung
Kompensieren von Unsicherheiten durch **Übergewichtung**

Ungünstigster Eingabewert: $\boxed{\cdot\,\cdot}\boxed{0}\boxed{1}$

Übergewichtungsfaktor: $\mathcal{F} = 25$

Budget: 1010

Pfadbudget: Übergewichtung

Lösung
Kompensieren von Unsicherheiten durch **Übergewichtung**

Ungünstigster Eingabewert: | · · | 0 | 1 |

Übergewichtungsfaktor: $\mathcal{F} = 25$

Budget: 1010

```
if(input[0])
```

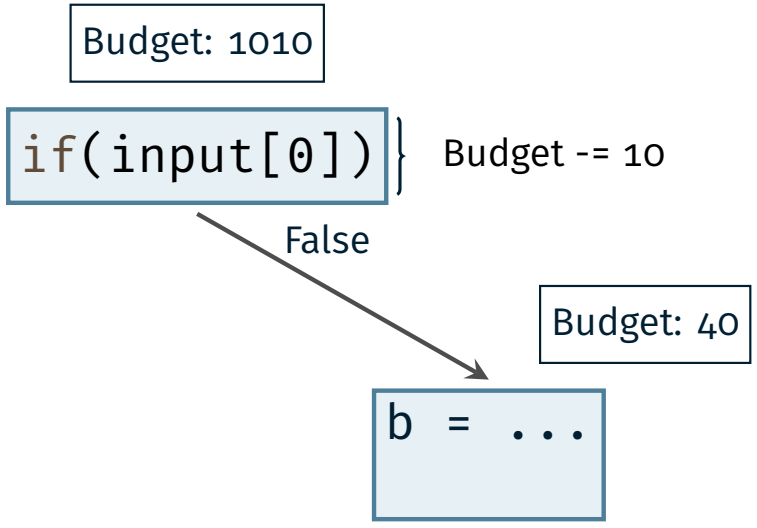
Budget -= 10

Pfadbudget: Übergewichtung

Lösung
Kompensieren von Unsicherheiten durch **Übergewichtung**

Ungünstigster Eingabewert: | · · | 0 | 1 |

Übergewichtungsfaktor: $\mathcal{F} = 25$

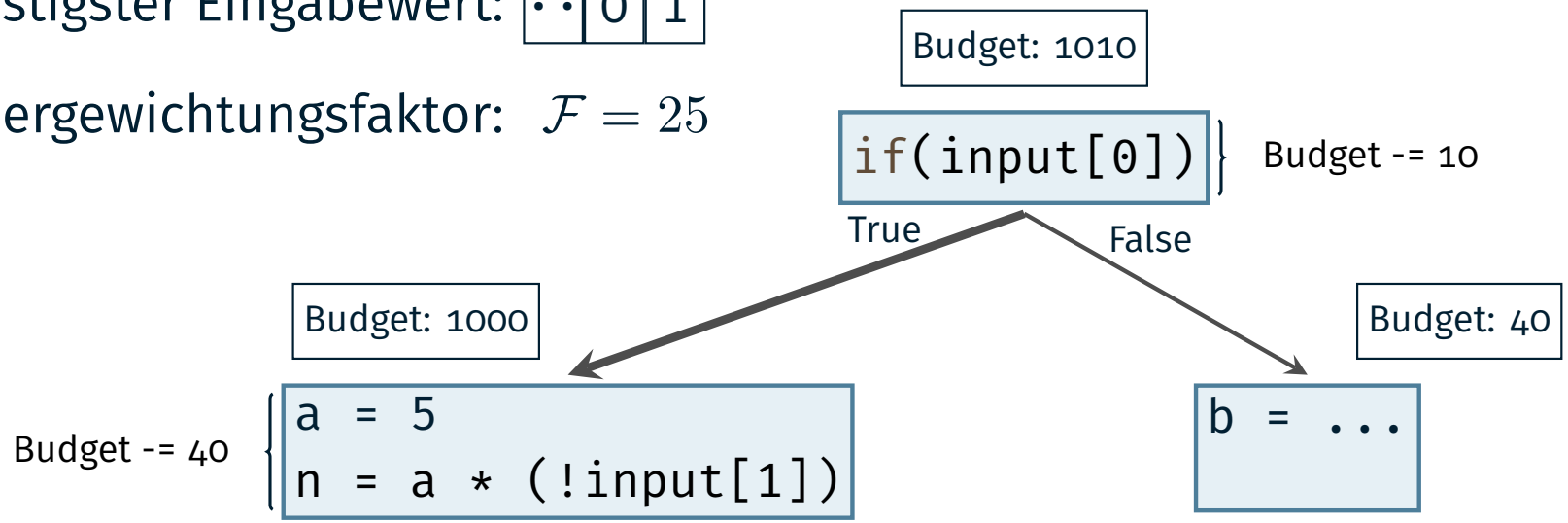


Pfadbudget: Übergewichtung

Lösung
Kompensieren von Unsicherheiten durch **Übergewichtung**

Ungünstigster Eingabewert: | · · | 0 | 1 |

Übergewichtungsfaktor: $\mathcal{F} = 25$

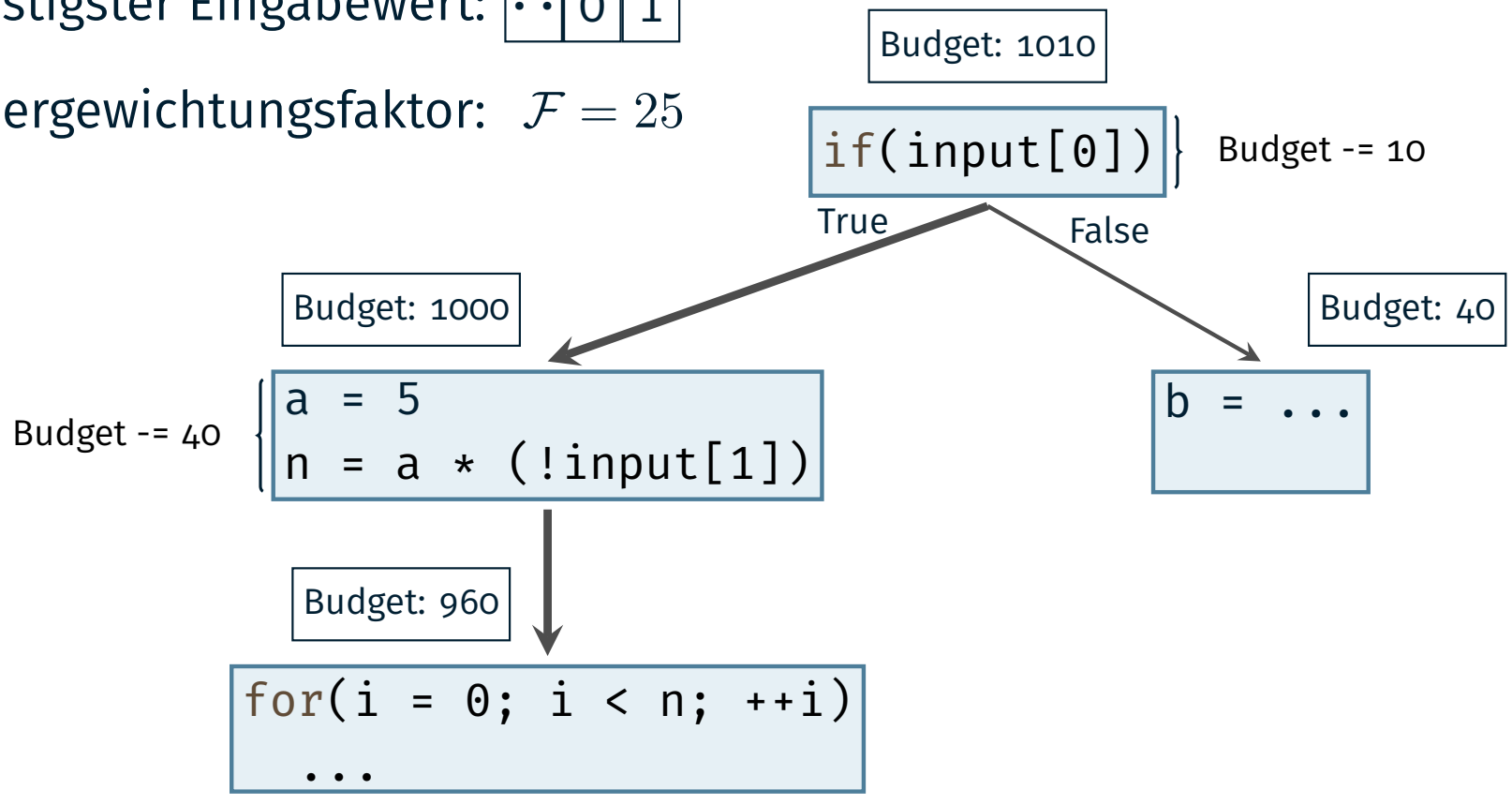


Pfadbudget: Übergewichtung

Lösung
Kompensieren von Unsicherheiten durch **Übergewichtung**

Ungünstigster Eingabewert: | ·· | 0 | 1 |

Übergewichtungsfaktor: $\mathcal{F} = 25$

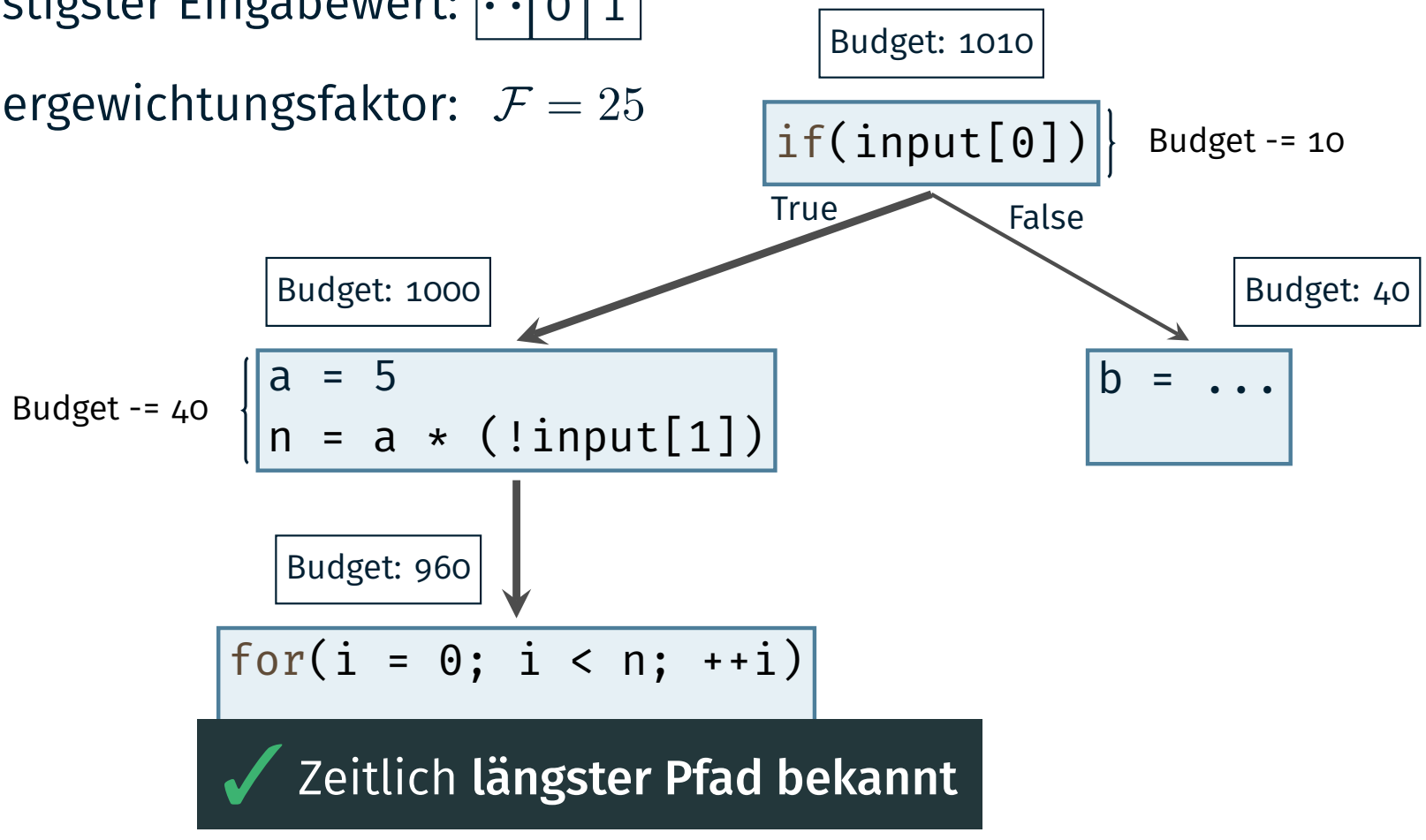


Pfadbudget: Übergewichtung

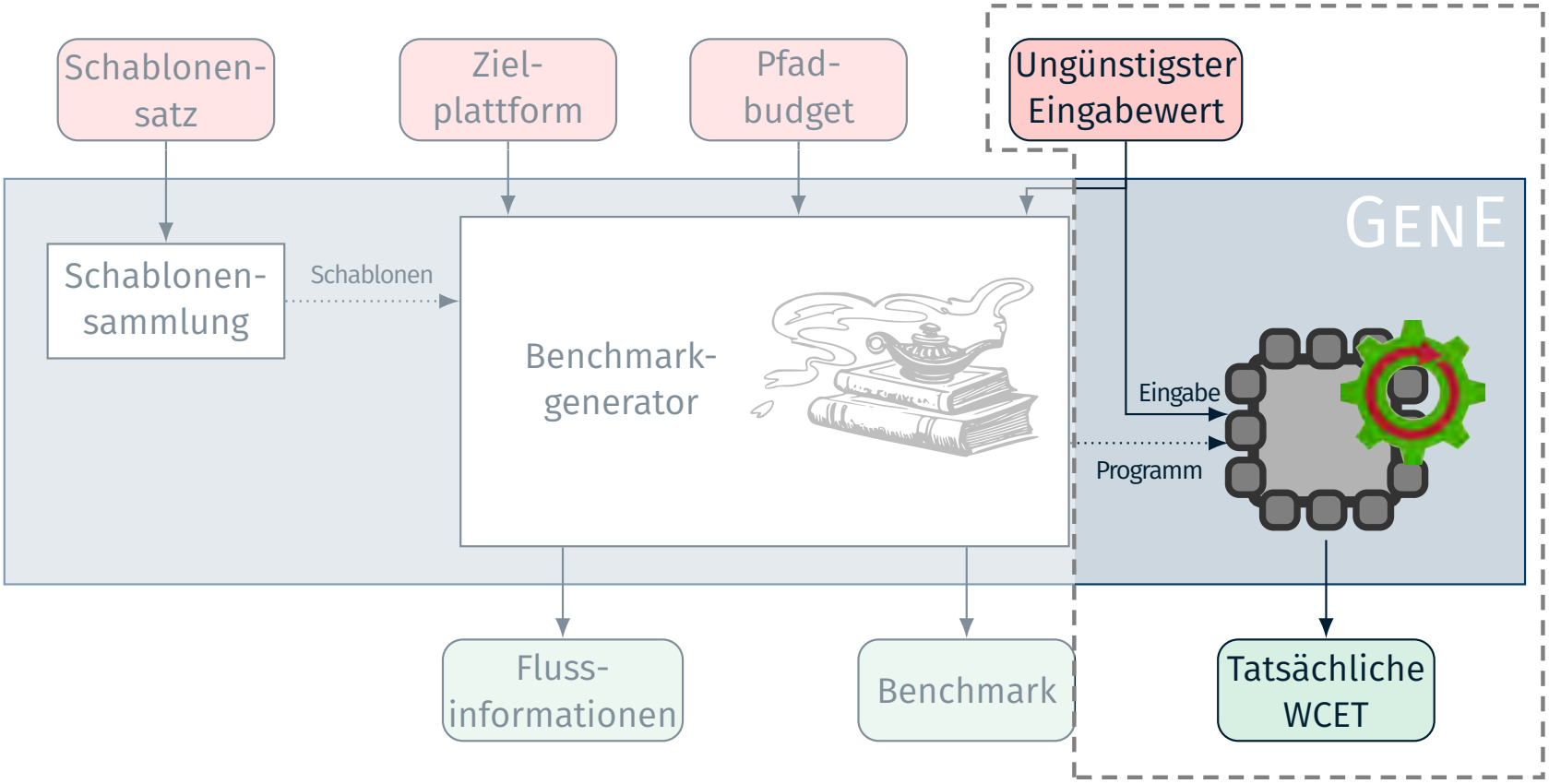
Lösung
Kompensieren von Unsicherheiten durch **Übergewichtung**

Ungünstigster Eingabewert: | · · | 0 | 1 |

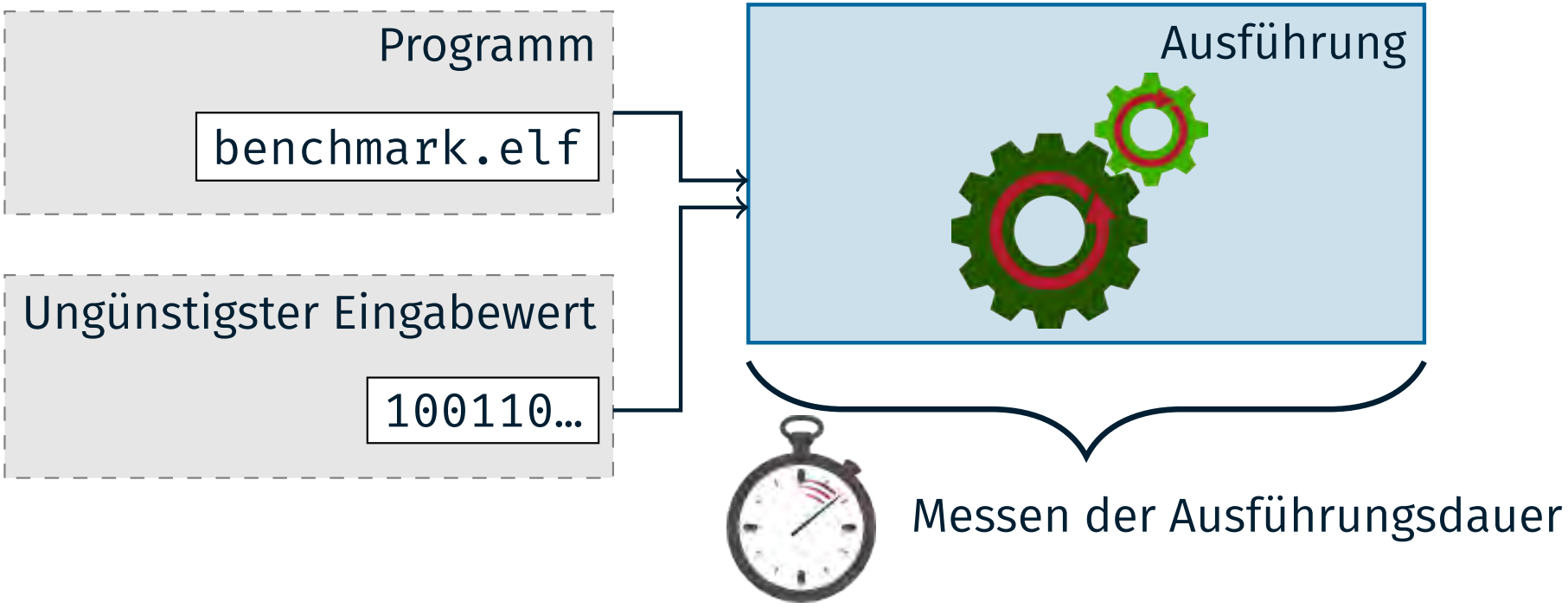
Übergewichtungsfaktor: $\mathcal{F} = 25$



Überblick: Bestimmen der WCET



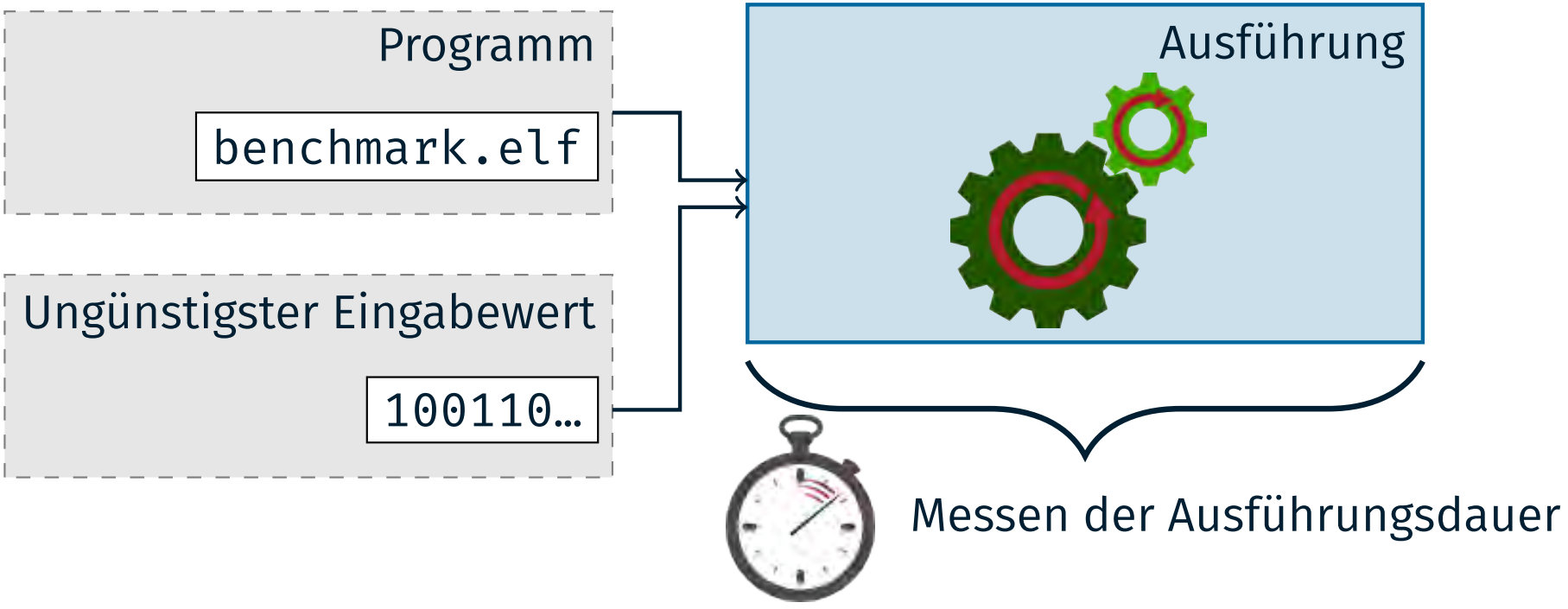
Bestimmen der WCET



Bestimmen der **tatsächlichen WCET** via Ausführung:

1. Zyklengenauer Simulator
2. Zielhardware

Bestimmen der WCET



Bestimmen der **tatsächlichen WCET** via Ausführung:

1. Zyklengenauer Simulator
2. Zielhardware

✓ Tatsächliche **WCET bekannt**

Implementierung: GenE & Aladdin

GenE

- GenE als Tool für die **LLVM Compiler Infrastructure**
- Generieren von LLVM Zwischensprache
  - + Systemnahe Darstellung
  - + Unabhängig von der Zielplattform

Implementierung: GenE & Aladdin

GenE

- GenE als Tool für die **LLVM Compiler Infrastructure**
- Generieren von LLVM Zwischensprache
  - + Systemnahe Darstellung
  - + Unabhängig von der Zielplattform

Aladdin

Automatisierte Messung und Evaluation

- + Umfangreiche Evaluationen
- + Reproduzierbarkeit
- + Zusammenfassen der Ergebnisse verschiedener Evaluationen

Evaluation

Evaluation: Aufbau

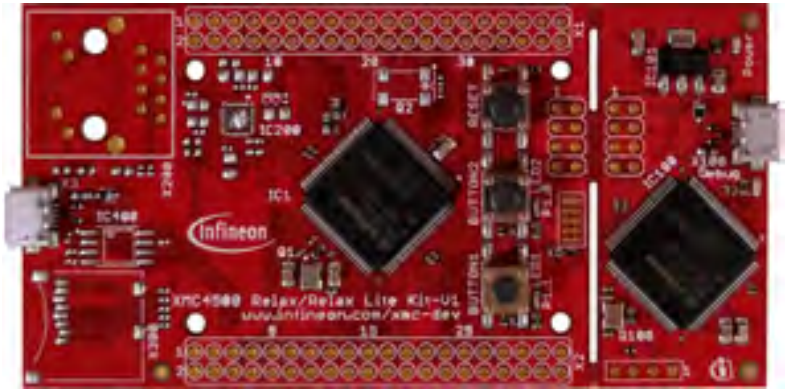
aiT

PLATIN

RapiTime

XMOS

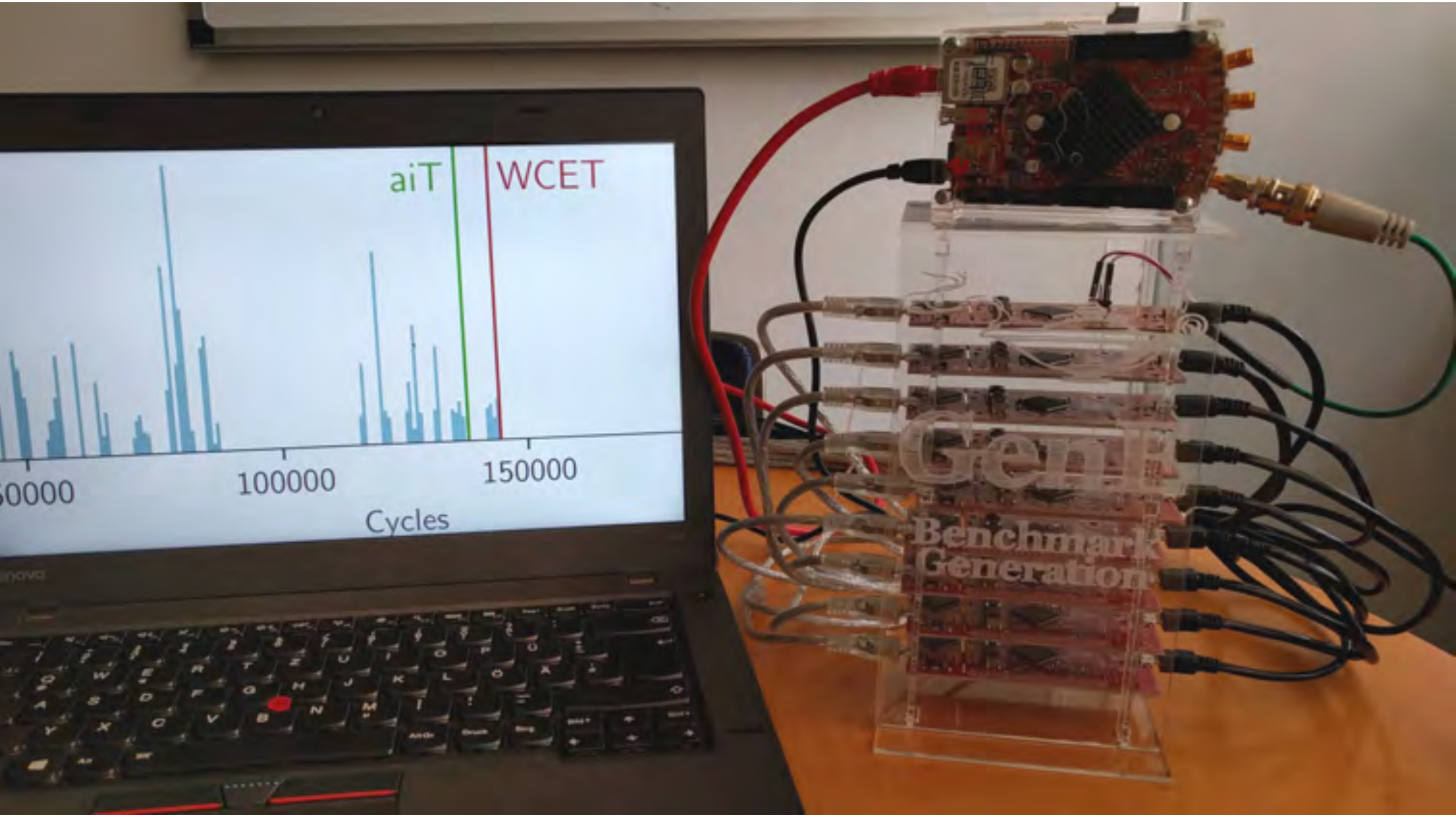
OTAWA



ARM Cortex-M4

- 3-stufige Pipeline
- Taktfrequenz: 120 MHz
- 4 KB Instruktionscache

Evaluation: Aufbau



Evaluation: Stärken und Schwächen

Methode

Erzeugen eines Benchmarks aus nur einer herausfordernden Schablone:

- **Genau eine** Schleife
- Arithmetische Operationen & Zuweisungen

	Konstante Schleifengrenze	Eingabeabh. Schleifengrenze	Nicht geschlossen darstellbare Schleifengrenze	Dreieck-schleife
aiT				
PLATIN				

Evaluation: Stärken und Schwächen

Methode

Erzeugen eines Benchmarks aus nur einer herausfordernden Schablone:

- **Genau eine** Schleife
- Arithmetische Operationen & Zuweisungen

	Konstante Schleifengrenze	Eingabeabh. Schleifengrenze	Nicht geschlossen darstellbare Schleifengrenze	Dreieck-schleife
aiT	✓			
PLATIN	✓			

Evaluation: Stärken und Schwächen

Methode

Erzeugen eines Benchmarks aus nur einer herausfordernden Schablone:

- **Genau eine** Schleife
- Arithmetische Operationen & Zuweisungen

	Konstante Schleifengrenze	Eingabeabh. Schleifengrenze	Nicht geschlossen darstellbare Schleifengrenze	Dreieck-schleife
aiT	✓	✓		
PLATIN	✓	✓		

Evaluation: Stärken und Schwächen

Methode

Erzeugen eines Benchmarks aus nur einer herausfordernden Schablone:

- **Genau eine** Schleife
- Arithmetische Operationen & Zuweisungen

	Konstante Schleifengrenze	Eingabeabh. Schleifengrenze	Nicht geschlossen darstellbare Schleifengrenze	Dreieck-schleife
aiT	✓	✓	✓	
PLATIN	✓	✓	✗	

Evaluation: Stärken und Schwächen

Methode

Erzeugen eines Benchmarks aus nur einer herausfordernden Schablone:

- **Genau eine** Schleife
- Arithmetische Operationen & Zuweisungen

	Konstante Schleifengrenze	Eingabeabh. Schleifengrenze	Nicht geschlossen darstellbare Schleifengrenze	Dreieck-schleife
aiT	✓	✓	✓	✗
PLATIN	✓	✓	✗	✓

Evaluation: Stärken und Schwächen

Methode

Erzeugen eines Benchmarks aus nur einer herausfordernden Schablone:

- **Genau eine** Schleife
- Arithmetische Operationen & Zuweisungen

	Konstante Schleifengrenze	Eingabeabh. Schleifengrenze	Nicht geschlossen darstellbare Schleifengrenze	Dreieck-schleife
aiT	✓	✓	✓	✗
PLATIN	✓	✓	✗	✓

✓ GENE deckt Stärken und Schwächen auf

Evaluation: Überschätzung von PLATIN/aiT

Methode

Berechnen der Überschätzung für 10 000 verschiedene Benchmarks
$\rightarrow$ Optimum: 0% Überschätzung

Evaluation: Überschätzung von PLATIN/aiT

Methode

Berechnen der Überschätzung für 10 000 verschiedene Benchmarks
$\rightarrow$ Optimum: 0% Überschätzung

	Instruktionscache	
	deaktiviert	aktiviert
PLATIN		
aiT		

Evaluation: Überschätzung von PLATIN/aiT

Methode

Berechnen der Überschätzung für 10 000 verschiedene Benchmarks
→ Optimum: 0% Überschätzung

	Instruktionscache	
	deaktiviert	aktiviert
PLATIN	96%	
aiT	23%	

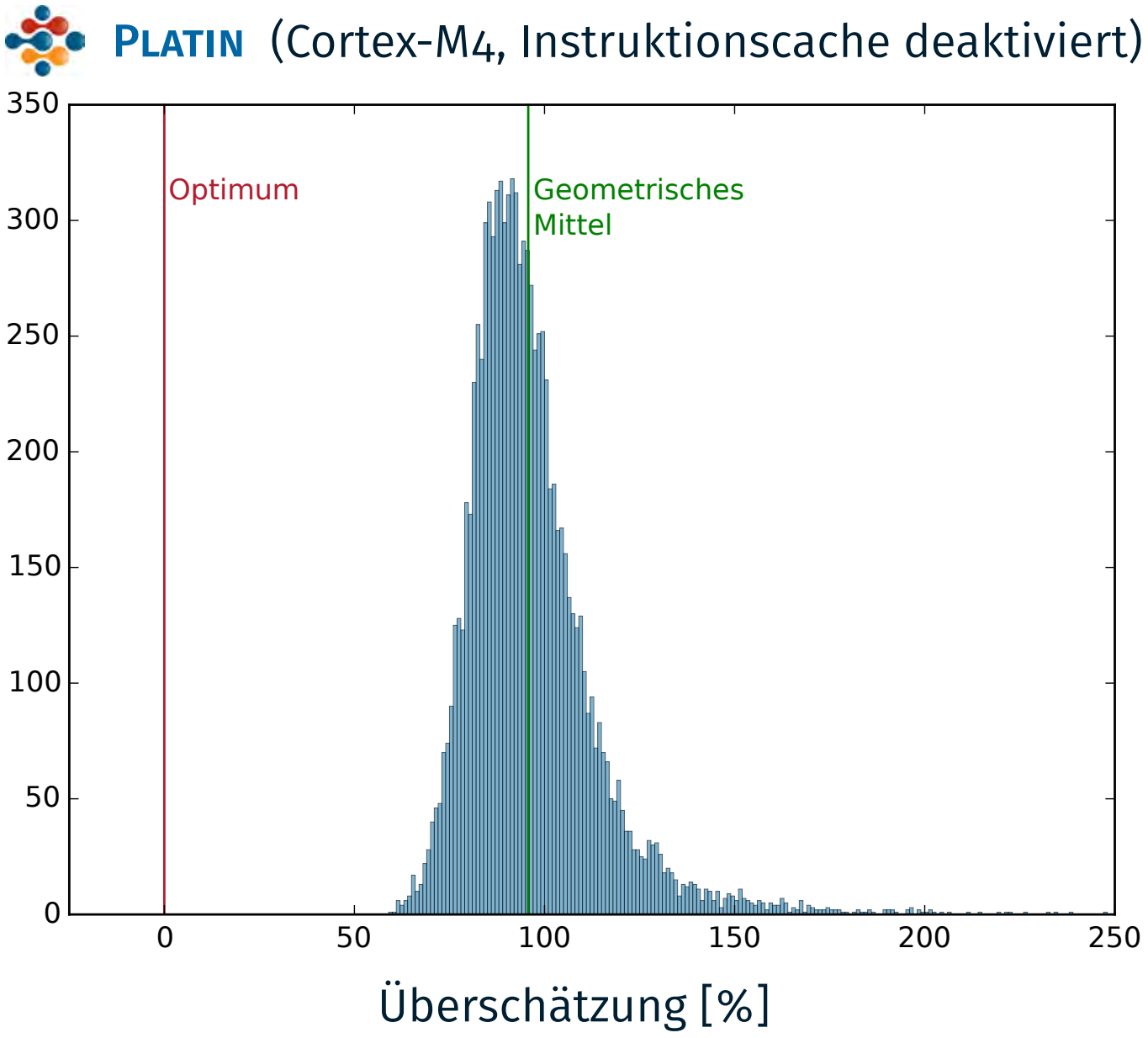
Evaluation: Überschätzung von PLATIN/aiT

Methode

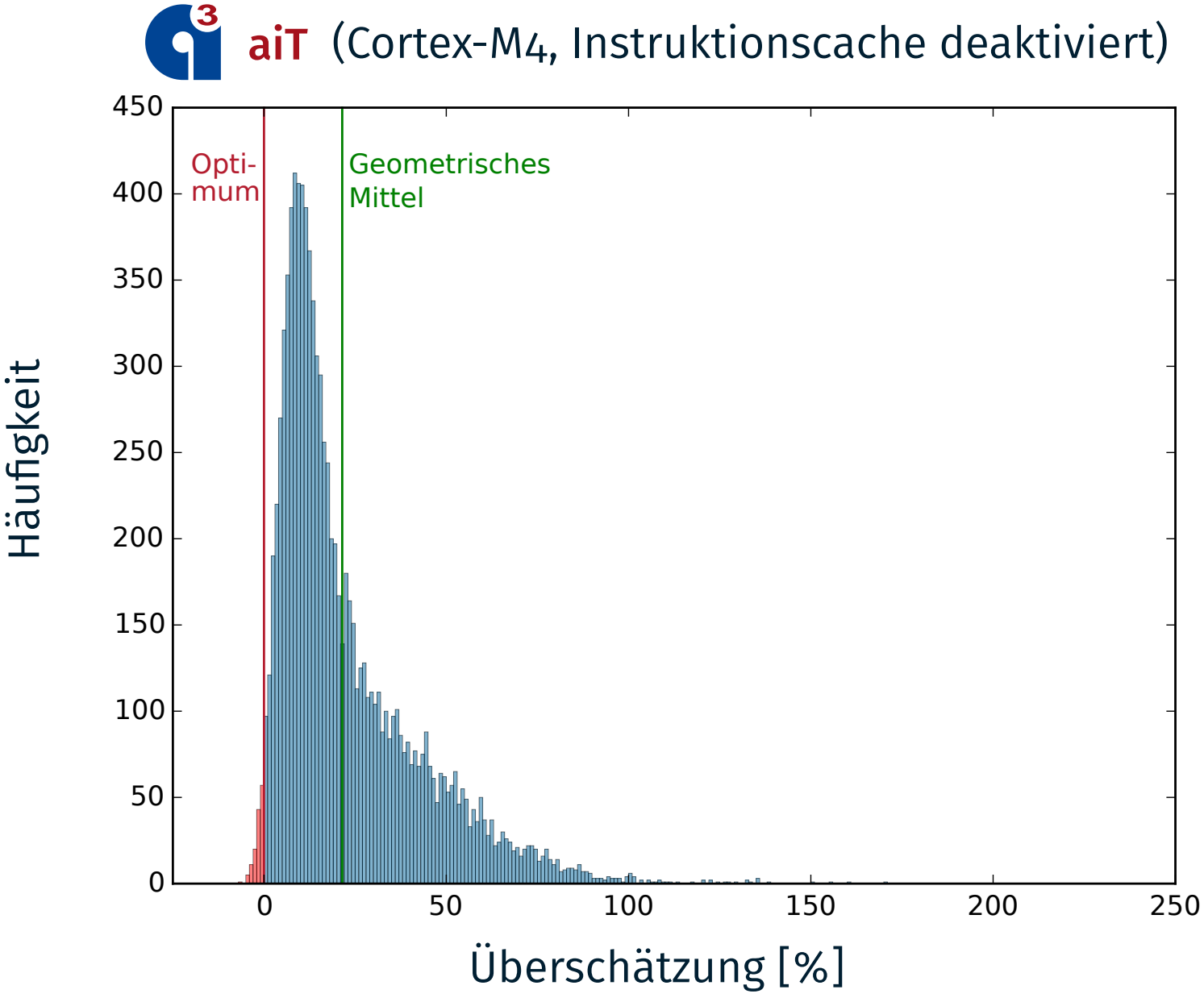
Berechnen der Überschätzung für 10 000 verschiedene Benchmarks
→ Optimum: 0% Überschätzung

	Instruktionscache	
	deaktiviert	aktiviert
PLATIN	96%	301%
aiT	23%	36%

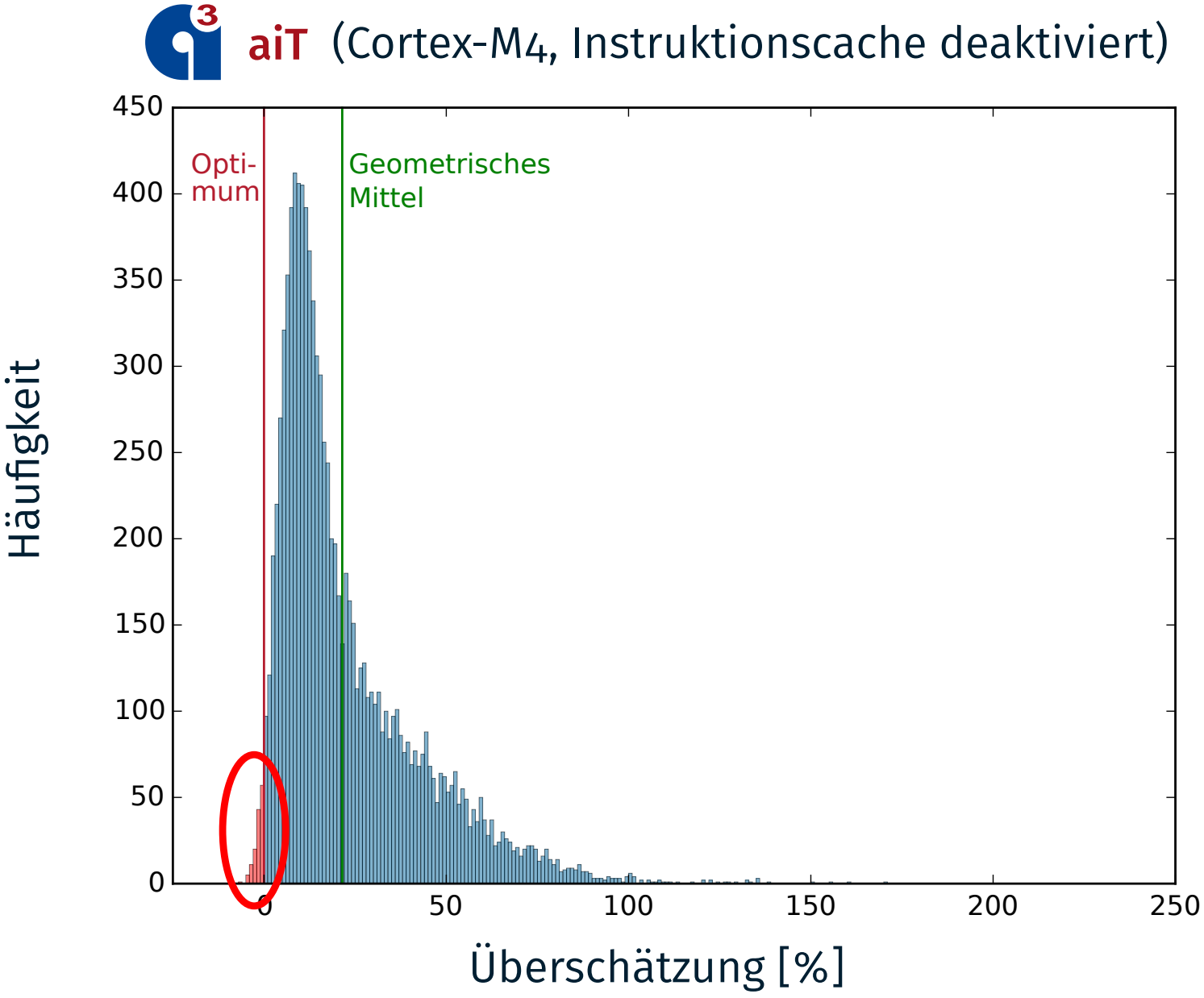
Evaluation: Überschätzung von PLATIN



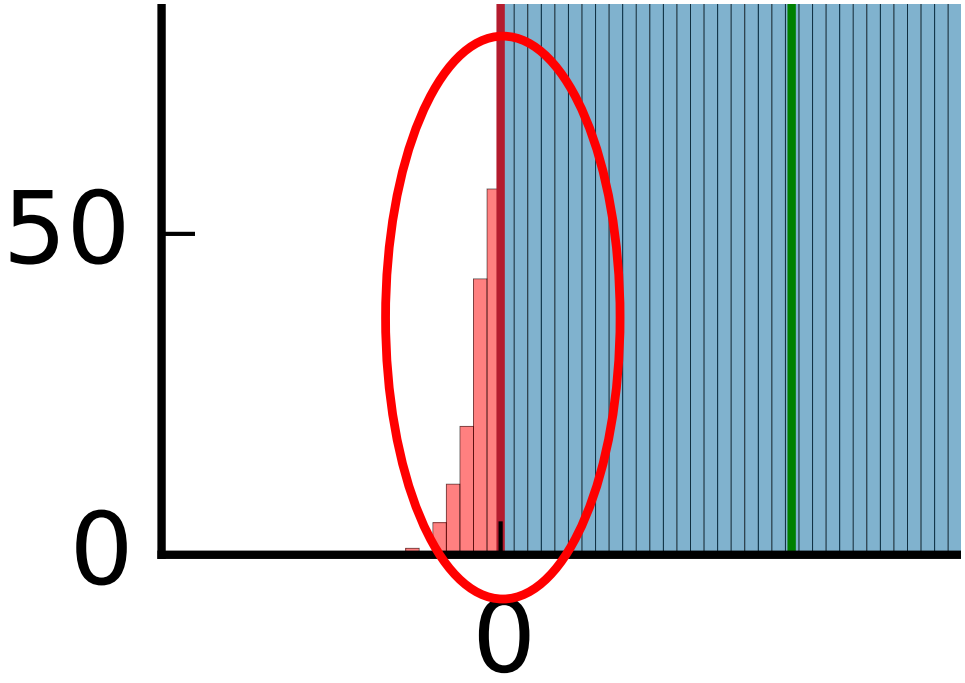
Evaluation: Überschätzung von aiT



Evaluation: Überschätzung von aiT

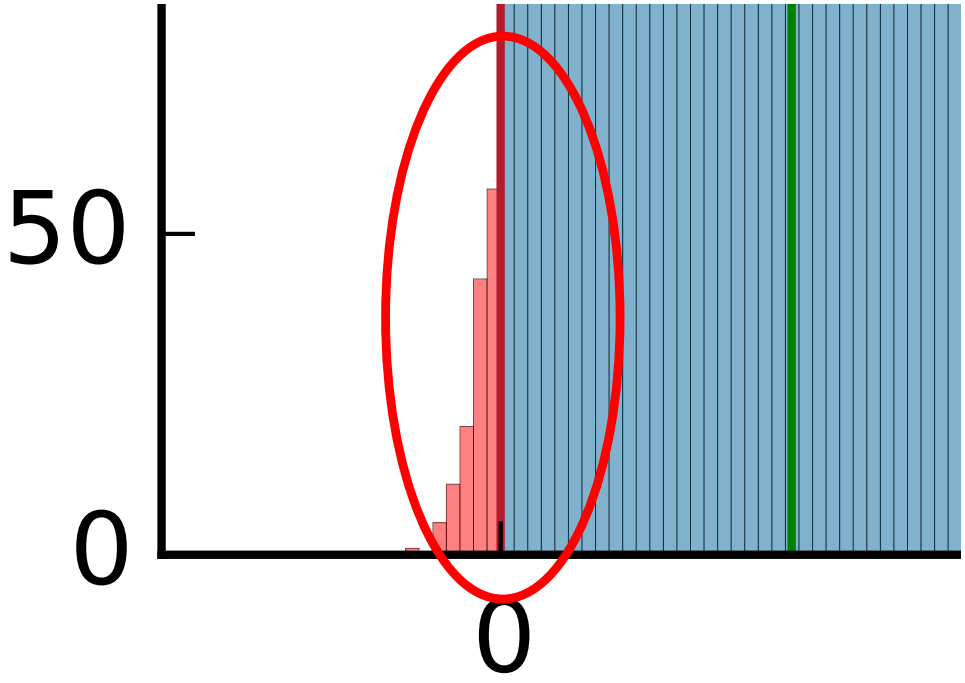


Evaluation: Erkennen von Fehlern in WCET-Analysatoren



- *Beobachtung*: Unterschätzungen auf bis zu 94% der tatsächlichen WCET
- *AbsInt*: Mithilfe von GENE-Benchmarks wurden **Fehler im Cortex-M4 Hardwaremodell** von aiT gefunden:
  - Pipelinemodellierung
  - Speichertimings

Evaluation: Erkennen von Fehlern in WCET-Analysatoren



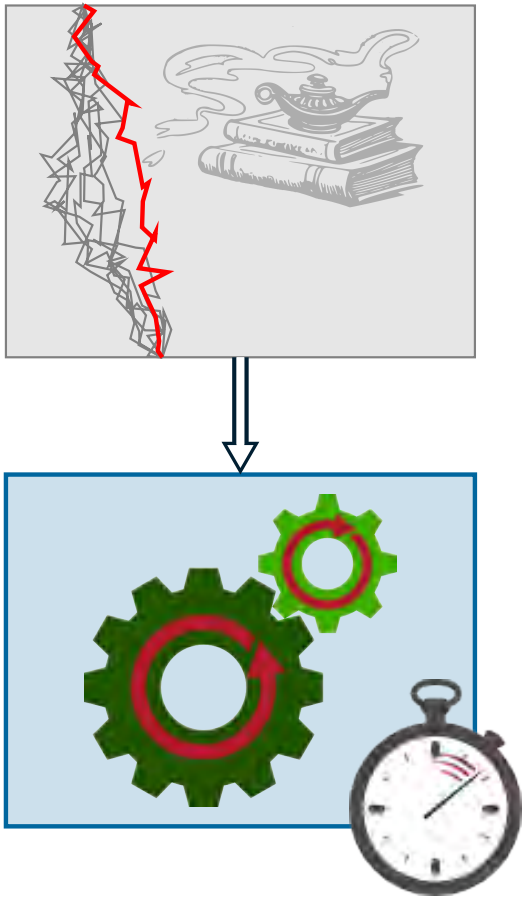
- *Beobachtung*: Unterschätzungen auf bis zu 94% der tatsächlichen WCET
- *AbsInt*: Mithilfe von GENE-Benchmarks wurden **Fehler im Cortex-M4 Hardwaremodell** von aiT gefunden:
  - Pipelinemodellierung

✓ Abschätzen der Überschätzung & **Erkennen von Unterschätzungen**

Fazit

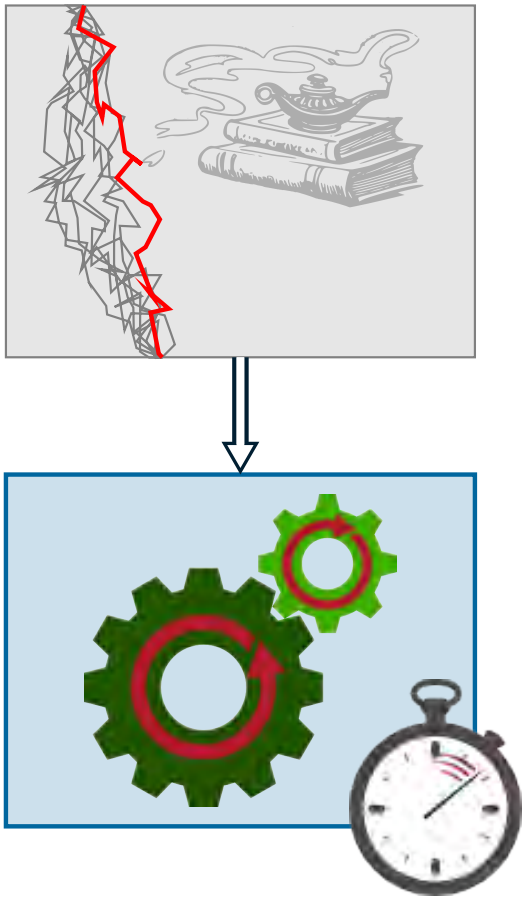
Fazit

- *Motivation:* **Fehlende Vergleichswerte** bei existierendem Code



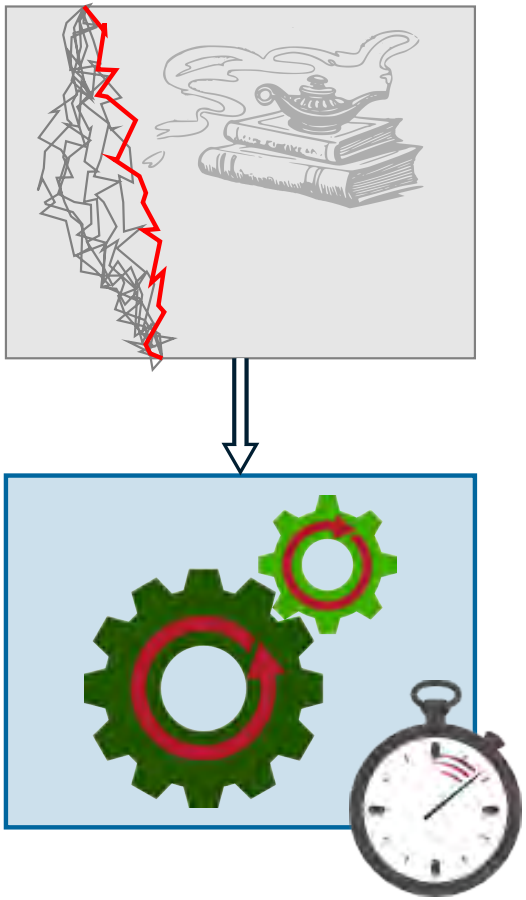
Fazit

- *Motivation:* **Fehlende Vergleichswerte** bei existierendem Code
- Erzeugen von Code mit bekannter Eingabe für den längsten Ausführungspfad
- Bestimmen der WCET durch **Ausführung** auf der Zielhardware



Fazit

- *Motivation:* **Fehlende Vergleichswerte** bei existierendem Code
- Erzeugen von Code mit bekannter Eingabe für den längsten Ausführungspfad
- Bestimmen der WCET durch **Ausführung** auf der Zielhardware



GenE

✓ Bekannte WCET

✓ Komplexe Benchmarks

✓ Kann Fehler in WCET-Analysatoren finden

Fragen?

Der Quellcode ist verfügbar unter:
`https://gitlab.cs.fau.de/gene`

- Wägemann, Distler, Eichler, Schröder-Preikschat.
Benchmark Generation for Timing Analysis.
Proc. of the 23rd Real-Time and Embedded Technology and Applications Symposium (**RTAS '17**)

- Eichler, Wägemann, Distler, Schröder-Preikschat.
Tooling Support for Benchmarking Timing Analysis..
Proc. of the 23rd Real-Time and Embedded Technology and Applications Symposium,
Demo Session (**RTAS Demo '17**)