

# Auswertung: Implementierung gewichteter Parser für CFG

Thomas Ruprecht  
thomas.ruprecht@tu-dresden.de

Lehrstuhl Grundlagen der Programmierung  
Institut für theoretische Informatik  
TU Dresden

28. Mai 2019

## Ausführungszeit

Programmiersprache	Alg.	Grammatik aus ...		
		20 Bäumen	200 Bäumen	2000 Bäumen
C++	CKY			
	CKY			
	CKY			
Rust	CKY			
	CKY			
Python	CKY			
Rust	CKY			

## Ausführungszeit

Programmiersprache	Alg.	Grammatik aus ...		
		20 Bäumen	200 Bäumen	2000 Bäumen
C++	CKY			
	CKY			
Rust	CKY			
	CKY			
Python	CKY	19.6s	$\geq 40s$	$\geq 200s$
Rust	CKY			

## Ausführungszeit

Programmiersprache	Alg.	Grammatik aus ...		
		20 Bäumen	200 Bäumen	2000 Bäumen
C++	CKY			
	CKY	480ms	$\geq 40s$	$\geq 200s$
Rust	CKY			
	CKY			
Python	CKY	19.6s	$\geq 40s$	$\geq 200s$
Rust	CKY			

## Ausführungszeit

Programmiersprache	Alg.	Grammatik aus ...		
		20 Bäumen	200 Bäumen	2000 Bäumen
C++	CKY			
	CKY	480ms	$\geq 40s$	$\geq 200s$
Rust	CKY			
	CKY	253ms	31s	$\geq 200s$
Python	CKY	19.6s	$\geq 40s$	$\geq 200s$
Rust	CKY			

## Ausführungszeit

Programmiersprache	Alg.	Grammatik aus ...		
		20 Bäumen	200 Bäumen	2000 Bäumen
C++	CKY			
	CKY	480ms	$\geq 40s$	$\geq 200s$
Rust	CKY	43ms	1.6s	48.4s
	CKY	253ms	31s	$\geq 200s$
Python	CKY	19.6s	$\geq 40s$	$\geq 200s$
Rust	CKY			

## Ausführungszeit

Programmiersprache	Alg.	Grammatik aus ...		
		20 Bäumen	200 Bäumen	2000 Bäumen
C++	CKY	35ms	511ms	6s
	CKY	480ms	$\geq 40s$	$\geq 200s$
Rust	CKY	43ms	1.6s	48.4s
	CKY	253ms	31s	$\geq 200s$
Python	CKY	19.6s	$\geq 40s$	$\geq 200s$
Rust	CKY			

## Ausführungszeit

Programmiersprache	Alg.	Grammatik aus ...		
		20 Bäumen	200 Bäumen	2000 Bäumen
C++	CKY	35ms	511ms	6s
	CKY	480ms	$\geq 40s$	$\geq 200s$
Rust	CKY	43ms	1.6s	48.4s
	CKY	253ms	31s	$\geq 200s$
Python	CKY	19.6s	$\geq 40s$	$\geq 200s$
Rust	CKY	41ms	244ms	4.15s

# Auffälligkeiten

Behandlung der unären Regeln:

- ▶ aus einer Datenstruktur sollte ein Eintrag mit geringstem Gewicht genommen werden: dafür eignet sich ein *Heap*
- ▶ wer keine Dijkstra-Suche benutzt muss so lange iterieren, wie Gewichte geändert werden

# Auffälligkeiten

Behandlung der unären Regeln:

- ▶ aus einer Datenstruktur sollte ein Eintrag mit geringstem Gewicht genommen werden: dafür eignet sich ein *Heap*
- ▶ wer keine Dijkstra-Suche benutzt muss so lange iterieren, wie Gewichte geändert werden

Datenstruktur für Regeln:

- ▶ `Map<Rule, Weight>` eignet sich nicht
- ▶ alle Regeln (nullstellige, unäre und binäre) auf einem Haufen
- ▶ getrennte Arrays für Regelsorten, evtl. indiziert durch Nichtterminal auf rechter Seite

# Auffälligkeiten

Behandlung der unären Regeln:

- ▶ aus einer Datenstruktur sollte ein Eintrag mit geringstem Gewicht genommen werden: dafür eignet sich ein *Heap*
- ▶ wer keine Dijkstra-Suche benutzt muss so lange iterieren, wie Gewichte geändert werden

Datenstruktur für Regeln:

- ▶ `Map<Rule, Weight>` eignet sich nicht
- ▶ alle Regeln (nullstellige, unäre und binäre) auf einem Haufen
- ▶ getrennte Arrays für Regelsorten, evtl. indiziert durch Nichtterminal auf rechter Seite

Datenstruktur für Parsecharts:

- ▶ es wurden häufig ineinander verschachtelte Datenstrukturen (Array/Liste/Hash-Map) verwendet
- ▶ Array von  $(\sum_{i \in \{1, \dots, n\}} n) \cdot |N| = \frac{n \cdot (n+1) \cdot |N|}{2}$  Gewichten

# Auffälligkeiten

Behandlung der unären Regeln:

- ▶ aus einer Datenstruktur sollte ein Eintrag mit geringstem Gewicht genommen werden: dafür eignet sich ein *Heap*
- ▶ wer keine Dijkstra-Suche benutzt muss so lange iterieren, wie Gewichte geändert werden

Datenstruktur für Regeln:

- ▶ `Map<Rule, Weight>` eignet sich nicht
- ▶ alle Regeln (nullstellige, unäre und binäre) auf einem Haufen
- ▶ getrennte Arrays für Regelsorten, evtl. indiziert durch Nichtterminal auf rechter Seite

Datenstruktur für Parsecharts:

- ▶ es wurden häufig ineinander verschachtelte Datenstrukturen (Array/Liste/Hash-Map) verwendet
- ▶ Array von  $(\sum_{i \in \{1, \dots, n\}} n) \cdot |N| = \frac{n \cdot (n+1) \cdot |N|}{2}$  Gewichten

Suche in Listen/Iteratoren, häufiges kopieren, häufige Vergleiche von Strings, ...