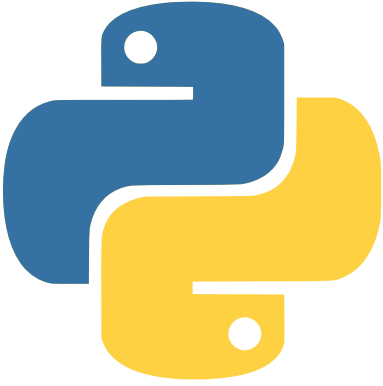
Universität Stuttgart
Projekt digit@L – BOOST. SKILLS. SUPPORT.



Programmierkurs Python

SymPy: Arbeiten mit Ausdrücken

Dominik Göddeke

IANS SimTech

Stiftung Innovation in der Hochschullehre

SymPy:
Arbeiten mit Ausdrücken:
Faktorisieren und Ausmultiplizieren

Faktorisieren und Ausmultiplizieren

- `factor()` und `expand()`: Faktorisierung und Erweiterung arithmetischer Ausdrücke

Faktorisieren und Ausmultiplizieren

- `factor()` und `expand()`: Faktorisierung und Erweiterung arithmetischer Ausdrücke
- Argument: symbolischer Ausdruck

Faktorisieren und Ausmultiplizieren

- `factor()` und `expand()`: Faktorisierung und Erweiterung arithmetischer Ausdrücke
- Argument: symbolischer Ausdruck
- `display` nicht vergessen

Faktorisieren und Ausmultiplizieren

- `factor()` und `expand()`: Faktorisierung und Erweiterung arithmetischer Ausdrücke
- Argument: symbolischer Ausdruck
- `display` nicht vergessen
- Demonstration: **binomische Formeln**

Codebeispiel

Substitution von Werten

Substitution von Werten

- `subs()`: Einsetzen von numerischen Werten in symbolische Ausdrücke

Substitution von Werten

- `subs()`: Einsetzen von numerischen Werten in symbolische Ausdrücke
- Wichtig: alle SymPy-Objekte sind **immutable**

Substitution von Werten

- `subs()`: Einsetzen von numerischen Werten in symbolische Ausdrücke
- Wichtig: alle SymPy-Objekte sind **immutable**
- Deshalb: Ergebnis von `subs()` muss neuer symbolischer Gleichung zugewiesen werden

Substitution von Werten

- `subs()`: Einsetzen von numerischen Werten in symbolische Ausdrücke
- Wichtig: alle SymPy-Objekte sind **immutable**
- Deshalb: Ergebnis von `subs()` muss neuer symbolischer Gleichung zugewiesen werden
- Symbolische Gleichung ist „ganz normale Python-Variable“

Codebeispiel

Vergleich von Gleichungen

Vergleich von Gleichungen

- Standardoperator == in SymPy „unzuverlässig“

Vergleich von Gleichungen

- Standardoperator == in SymPy „unzuverlässig“
- Keine mathematische Gleichheit, keine beliebig komplexe automatische Vereinfachung

Vergleich von Gleichungen

- Standardoperator == in SymPy „unzuverlässig“
- Keine mathematische Gleichheit, keine beliebig komplexe automatische Vereinfachung
- Rein **struktureller Vergleich**, mit Berücksichtigung elementarer Rechengesetze wie Kommutativität

Vergleich von Gleichungen

- Standardoperator `==` in SymPy „unzuverlässig“
- Keine mathematische Gleichheit, keine beliebig komplexe automatische Vereinfachung
- Rein **struktureller Vergleich**, mit Berücksichtigung elementarer Rechengesetze wie Kommutativität
- Idee für Ausweg: `expand()` und `factor()`

Vergleich von Gleichungen

- Standardoperator == in SymPy „unzuverlässig“
- Keine mathematische Gleichheit, keine beliebig komplexe automatische Vereinfachung
- Rein **struktureller Vergleich**, mit Berücksichtigung elementarer Rechengesetze wie Kommutativität
- Idee für Ausweg: `expand()` und `factor()`
- Aber: **schnell Sackgasse**

Codebeispiel

Der Satz von Richardson

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**
  - Paraphrasiert: jede hinreichend komplizierte Gleichung der Form $f = 0$, also insbesondere die Gleichung $g - h = 0$ für zwei zu vergleichende Ausdrücke g und h, ist **unentscheidbar** in der Computeralgebra

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**
  - Paraphrasiert: jede hinreichend komplizierte Gleichung der Form $f = 0$, also insbesondere die Gleichung $g - h = 0$ für zwei zu vergleichende Ausdrücke g und h, ist **unentscheidbar** in der Computeralgebra
- Aber: zahlreiche „Ausweichmöglichkeiten“, in der Praxis oft kein Problem

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**
  - Paraphrasiert: jede hinreichend komplizierte Gleichung der Form $f = 0$, also insbesondere die Gleichung $g - h = 0$ für zwei zu vergleichende Ausdrücke g und h, ist **unentscheidbar** in der Computeralgebra
- Aber: zahlreiche „Ausweichmöglichkeiten“, in der Praxis oft kein Problem
- **Strategie für Gleichheitstest**

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**
  - Paraphrasiert: jede hinreichend komplizierte Gleichung der Form $f = 0$, also insbesondere die Gleichung $g - h = 0$ für zwei zu vergleichende Ausdrücke g und h, ist **unentscheidbar** in der Computeralgebra
- Aber: zahlreiche „Ausweichmöglichkeiten“, in der Praxis oft kein Problem
- **Strategie für Gleichheitstest**
  - Bilde Differenz der zu vergleichenden Ausdrücke

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**
  - Paraphrasiert: jede hinreichend komplizierte Gleichung der Form $f = 0$, also insbesondere die Gleichung $g - h = 0$ für zwei zu vergleichende Ausdrücke g und h, ist **unentscheidbar** in der Computeralgebra
- Aber: zahlreiche „Ausweichmöglichkeiten“, in der Praxis oft kein Problem
- **Strategie für Gleichheitstest**
  - Bilde Differenz der zu vergleichenden Ausdrücke
  - Vereinfache resultierenden Ausdruck so weit wie möglich

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**
  - Paraphrasiert: jede hinreichend komplizierte Gleichung der Form $f = 0$, also insbesondere die Gleichung $g - h = 0$ für zwei zu vergleichende Ausdrücke g und h, ist **unentscheidbar** in der Computeralgebra
- Aber: zahlreiche „Ausweichmöglichkeiten“, in der Praxis oft kein Problem
- **Strategie für Gleichheitstest**
  - Bilde Differenz der zu vergleichenden Ausdrücke
  - Vereinfache resultierenden Ausdruck so weit wie möglich
  - Teste Ergebnis auf null

Der Satz von Richardson

- Pessimismus aus der reinen Mathematik: **Satz von Richardson**
  - Paraphrasiert: jede hinreichend komplizierte Gleichung der Form $f = 0$, also insbesondere die Gleichung $g - h = 0$ für zwei zu vergleichende Ausdrücke g und h, ist **unentscheidbar** in der Computeralgebra
- Aber: zahlreiche „Ausweichmöglichkeiten“, in der Praxis oft kein Problem
- **Strategie für Gleichheitstest**
  - Bilde Differenz der zu vergleichenden Ausdrücke
  - Vereinfache resultierenden Ausdruck so weit wie möglich
  - Teste Ergebnis auf null
- `simplify()` tut genau das, was der Name verheißt

Codebeispiel

Ausweg bei komplizierten Ausdrücken

Ausweg bei komplizierten Ausdrücken

- Falls Satz von Richardson zuschlägt: Problem unendscheidbar

Ausweg bei komplizierten Ausdrücken

- Falls Satz von Richardson zuschlägt: Problem unendscheidbar
- Ausweg: Numerik

Ausweg bei komplizierten Ausdrücken

- Falls Satz von Richardson zuschlägt: Problem unendscheidbar
- Ausweg: Numerik
- `equals()`: randomisierte Auswertungspunkte, Vergleich der resultierenden Funktionswerte

Ausweg bei komplizierten Ausdrücken

- Falls Satz von Richardson zuschlägt: Problem unendscheidbar
- Ausweg: Numerik
- `equals()`: randomisierte Auswertungspunkte, Vergleich der resultierenden Funktionswerte
- Ergebnis: **heuristische Aussage über Gleichheit**

Codebeispiel

Impressum, Danksagung und Quellen

Stiftung Innovation in der Hochschullehre

Gefördert durch die Stiftung Innovation in der Hochschullehre im Rahmen des Projekts digit@L, https://stiftung-hochschullehre.de

SimTech

Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie

---

Autor: Dominik Göddeke, IANS, Universität Stuttgart

ZOERR

Veröffentlicht auf dem Zentralen OER Repositorium Baden-Württemberg, https://www.zoerr.de