

Automatisierte Logik und Programmierung

Einheit 22



Korrektheitserhaltende Optimierungen



1. Logische Vereinfachungen
2. Partielle Auswertung
3. Endliche Differenzierung
4. Fallanalyse
5. Datentyp-Verfeinerung

Eliminiere überflüssige Berechnungen

- **Simplifikation: Logische Vereinfachung**
 - Äquivalenzumwandlung von Teilausdrücken, ggf. im Kontext
- **Partielle Auswertung**
 - Symbolische Auswertung von Ausdrücken mit konstanten Komponenten
- **Endliche Differenzierung**
 - Inkrementelle Berechnung von Teilausdrücken in Schleifen
- **Fallanalyse**
 - Analyse und Vereinfachung von Teilausdrücken
- **Datentyp-Verfeinerung**
 - Bestimmung konkreter Implementierungen für abstrakte Datentypen
- **Sprachabhängige Optimierung & Compilierung**
 - Ausnutzen der Besonderheiten einer konkreten Zielsprache

Logische Vereinfachung von Teilausdrücken

- **Transformation in äquivalente Ausdrücke**

- Term-Rewriting mit gerichteten Gleichungen
- Gleichungen formuliert als Lemmata der Wissensbank
- Identifikation geeigneter Lemmata über Operatoren im Ausdruck

$$a \in (x.[\])^{\circ}(b.L) \mapsto a=x \vee a=b \vee a \in L$$

- **Kontextunabhängige CI-Simplifikation**

- Anwendung einfacher Gleichungen ohne Rahmenbedingungen
- Nur der aktuelle Teilausdruck muß betrachtet werden
- Automatische Ausführung solange anwendbare Lemmata vorhanden

- **Kontextabhängige CD-Simplifikation**

- Anwendung von Gleichungen mit Rahmenbedingungen
- Rahmenbedingungen müssen durch Kontext erfüllt werden

- **Benutzerinteraktion**

- Auswahl von Teilausdruck und Art der Vereinfachung (CI/CD)
- Optional bei CD: Begrenzung der Vor- und Rückwärtsinferenzen

OPTIMIERUNG DES COSTAS-ARRAYS ALGORITHMUS

Ausgangspunkt: schematischer Globalsuchalgorithmus

```
FUNCTION Costas (n:ℤ) WHERE n ≥ 1
  RETURNS {p:Seq(ℤ) | perm(p, {1..n}) ∧ ∀j < |p|.nodups(dtrow(p, j))}
≡ if nodups([]) ∧ ∀j < |[]|.nodups(dtrow([], j))
   then Costasgs(n, []) else ∅

FUNCTION Costasaux (n, s:ℤ × Seq(ℤ))
  WHERE n ≥ 1 ∧ range(s) ⊆ {1..n} ∧ nodups(s) ∧ ∀j < |s|.nodups(dtrow(s, j))
  RETURNS {p:Seq(ℤ) | perm(p, {1..n}) ∧ s ⊆ p ∧ ∀j < |p|.nodups(dtrow(p, j))}
≡ {p | p ∈ {s} ∧ perm(p, {1..n}) ∧ ∀j < |p|.nodups(dtrow(p, j)) }
  ∪ ⋃ {Costasgs(n, t) | t ∈ {s.i | i ∈ {1..n}} ∧ nodups(t)
                                             ∧ ∀j < |t|.nodups(dtrow(t, j))}
```

CI-SIMPLIFIKATIONEN IM HAUPTALGORITHMUS **COSTAS**

```
FUNCTION Costas (n:ℤ) WHERE n ≥ 1  
  RETURNS {p:Seq(ℤ) | perm(p, {1..n}) ∧ ∀j < |p|.nodups(dtrow(p, j))}  
≡ if nodups([]) ∧ ∀j < |[]|.nodups(dtrow([], j))  
   then Costasgs(n, []) else ∅
```

```
FUNCTION Costas (n:ℤ) WHERE n ≥ 1  
  RETURNS {p:Seq(ℤ) | perm(p, {1..n}) ∧ ∀j < |p|.nodups(dtrow(p, j))}  
≡ Costasgs(n, [])
```

CI-SIMPLIFIKATIONEN IN DER HILFSFUNKTION Costas_{aux}

```

FUNCTION  $\text{Costas}_{aux}$  (n, s:  $\mathbb{Z} \times \text{Seq}(\mathbb{Z})$ )
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
  RETURNS {p:  $\text{Seq}(\mathbb{Z})$  |  $\text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))$ }
 $\equiv$ 
  {p |  $p \in \{s\} \wedge \text{perm}(p, \{1..n\}) \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))$ }
   $\cup \bigcup \{\text{Costas}_{gs}(n, t) \mid t \in \{s \cdot i \mid i \in \{1..n\}\} \wedge \text{nodups}(t) \wedge \forall j < |t|. \text{nodups}(\text{dtrow}(t, j))\}$ 

```

```

FUNCTION  $\text{Costas}_{aux}$  (n, s:  $\mathbb{Z} \times \text{Seq}(\mathbb{Z})$ )
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
  RETURNS {p:  $\text{Seq}(\mathbb{Z})$  |  $\text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))$ }
 $\equiv$ 
  if  $\text{perm}(s, \{1..n\}) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$  then {s} else  $\emptyset$ 
   $\cup \bigcup \{\text{Costas}_{gs}(n, s \cdot i) \mid i \in \{1..n\} \wedge \text{nodups}(s \cdot i) \wedge \forall j < |s \cdot i|. \text{nodups}(\text{dtrow}(s \cdot i, j))\}$ 

```

KONTEXTABHÄNGIGE SIMPLIFIKATION

Vereinfachung unter Berücksichtigung von Kontext

- **Anwendung bedingter Gleichungen**
 - Gleichung $A \equiv B$ mit Vorbedingung C
 - Formuliert als Lemma der Form $C \Rightarrow A \equiv B$
- **Anwendbarkeit abhängig vom Kontext**
 - Ersetze Teilausdruck A durch B , wenn C aus dem Kontext folgt
 - Kontext ergibt sich aus Syntaxbaum des Gesamtausdrucks
 - Bei Programmen: benachbarter Programmcode und Vorbedingung
 - Hauptanwendung: Elimination redundanter Teilausdrücke
- **Komplizierter als CI-Simplifikation**
 - Vorbedingung kann auch Teil einer Gleichung (z.B. $C \wedge A \equiv B$) sein
 - Gleichung kann wird zuweilen auch rückwärts angewandt
 - Automatische Anwendung muß tiefenbeschränkt werden

CD-SIMPLIFIKATIONEN IN DER HILFSFUNKTION Costas_{aux}

```

FUNCTION  $\text{Costas}_{aux}$  (n, s:  $\mathbb{Z} \times \text{Seq}(\mathbb{Z})$ )
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
  RETURNS {p:  $\text{Seq}(\mathbb{Z})$  |  $\text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))$ }
 $\equiv$     if  $\text{perm}(s, \{1..n\}) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$  then {s} else  $\emptyset$ 
       $\cup \bigcup \{ \text{Costas}_{gs}(n, s \cdot i) \mid i \in \{1..n\} \wedge \text{nodups}(s \cdot i)$ 
                                      $\wedge \forall j < |s \cdot i|. \text{nodups}(\text{dtrow}(s \cdot i, j)) \}$ 

```

```

FUNCTION  $\text{Costas}_{aux}$  (n, s:  $\mathbb{Z} \times \text{Seq}(\mathbb{Z})$ )
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
  RETURNS {p:  $\text{Seq}(\mathbb{Z})$  |  $\text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))$ }
 $\equiv$     if  $\{1..n\} \setminus \text{range}(s) = \emptyset$  then {s} else  $\emptyset$ 
       $\cup \bigcup \{ \text{Costas}_{gs}(n, s \cdot i) \mid i \in \{1..n\} \setminus \text{range}(s) \wedge \forall j < |s|. s_{|s \cdot i| - j} - i \notin \text{dtrow}(s, j) \}$ 

```

Auswertung von Ausdrücken mit Konstanten

- **Symbolische Auswertung zur Entwurfszeit**

- **Reduktion** von Ausdrücken, deren “Wert” bestimmt werden kann
- Anwendbar, solange ein konstanter Teilausdruck vorhanden
- Formale Technik: **Auffalten von Definitionen** + **Simplifikation**
Je nach Inhalt der Wissensbank auch reine Simplifikation

Beispiel:

$ [x; 5] \circ L $	unfold append
$\mapsto x . ([5] \circ L) $	unfold append
$\mapsto x . (5 . ([] \circ L)) $	unfold append
$\mapsto x . (5 . L) $	unfold length
$\mapsto 1 + 5 . L $	unfold length
$\mapsto 1 + 1 + L $	simplify
$\mapsto 2 + L $	

- **Benutzerinteraktion:**

- Auswahl von auszuwertendem **Ausdruck** und **Optimierungstechnik**

ENDLICHE DIFFERENZIERUNG

- **Inkrementelle Berechnung statt ständiger Neuberechnung**

- Ersetze Teilausdruck $E[x]$ in $f(x)$ durch neue Variable c
- Initialisiere c mit Ausdruck für Basiswerte
- Bestimme differentielle Veränderungen bei rekursivem Aufruf

- **Endliche Differenzierung am Beispiel**

FUNCTION $\text{Costas}_{aux}(n, s: \mathbb{Z} \times \text{Seq}(\mathbb{Z}))$ WHERE ... RETURNS ...
 $\equiv$ if $\{1..n\} \setminus \text{range}(s) = \emptyset$ then $\{s\}$ else $\emptyset$
 $\cup \bigcup \{ \text{Costas}_{gs}(n, s \cdot i) \mid i \in \{1..n\} \setminus \text{range}(s) \wedge \forall j < |s| . s_{|s| \cdot i - j} - i \notin \text{dtrow}(s, j) \}$

FUNCTION $\text{Costas}_{aux}(n, s, \text{pool} \dots)$ WHERE ... $\text{pool} = \{1..n\} \setminus \text{range}(s)$
 $\equiv$ if $\text{pool} = \emptyset$ then $\{s\}$ else $\emptyset$
 $\cup \bigcup \{ \text{Costas}_{gs}(n, s \cdot i, \text{pool} \setminus \{i\}) \mid i \in \text{pool} \wedge \forall j < |s| . s_{|s| \cdot i - j} - i \notin \text{dtrow}(s, j) \}$

- Benutzer identifiziert $\{1..n\} \setminus \text{range}(s)$ als wiederkehrende Berechnung
- System ändert $\text{Costas}_{aux}(n, s)$ in $\text{Costas}_{aux}(n, s, \text{pool})$ und ändert Aufrufe von Costas_{aux} entsprechend
- System ergänzt $\text{pool} = \{1..n\} \setminus \text{range}(s)$ zur Eingabebedingung
- System vereinfacht Vorkommen von $\{1..n\} \setminus \text{range}(s)$ zu pool

• Abstraktion über Ausdruck $E[x]$ in Funktion $f(x)$

FUNCTION *main*... WHERE ... RETURNS ... SUCH THAT... $\equiv$ $f(x_0)$
 FUNCTION $f(x:D):R$ WHERE $I[x]$ RETURNS y SUCH THAT $O[x,y]$
 $\equiv$ $E[x]$ $f(t[x])$..

- Erweitere f zu neuem f' , so daß $f(x) = f'(x, E[x])$
 - Ersetze alle Aufrufe der Art $f(t[x])$ durch $f'(t[x], E[t[x]])$
 - Ergänze Gleichung $c=E[x]$ zur Eingabebedingung von f'
-

FUNCTION *main*... WHERE ... RETURNS ... SUCH THAT... $\equiv$ $f'(x_0, E[x_0])$
 FUNCTION $f(x, c:D \times D'):R$ WHERE $I[x] \wedge c=E[x]$ RETURNS y SUCH THAT $O[x,y]$
 $\equiv$ $E[x]$ $f'(t[x], E[t[x]])$..

• Simplifikation mit Gleichung $c = E[x]$

- Transformiere Ausdrücke der Form $E[g[x]]$ in die Form $g'[E[x]]$
- Ersetze alle Vorkommen von $E[x]$ durch c

• Benutzerinteraktion:

- Auswahl des zu ersetzenden Teilausdrucks $E[x]$ in $f(x)$

ENDLICHE DIFFERENZIERUNG VON Costas_{aux}

```

FUNCTION  $\text{Costas}_{aux}$  (n, s:  $\mathbb{Z} \times \text{Seq}(\mathbb{Z})$ )
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
  RETURNS  $\{p: \text{Seq}(\mathbb{Z}) \mid \text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))\}$ 
 $\equiv$     if  $\{1..n\} \setminus \text{range}(s) = \emptyset$  then  $\{s\}$  else  $\emptyset$ 
       $\cup \bigcup \{\text{Costas}_{gs}(n, s \cdot i) \mid i \in \{1..n\} \setminus \text{range}(s) \wedge \forall j < |s|. s_{|s|-j} - i \notin \text{dtrow}(s, j)\}$ 

```

```

FUNCTION  $\text{Costas}_{aux}$ (n, s, pool, ssize:  $\mathbb{Z} \times \text{Seq}(\mathbb{Z}) \times \text{Set}(\mathbb{Z}) \times \mathbb{Z}$ )
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
       $\wedge \text{pool} = \{1..n\} \setminus \text{range}(s) \wedge \text{ssize} = |s|$ 
  RETURNS  $\{p: \text{Seq}(\mathbb{Z}) \mid \text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))\}$ 
 $\equiv$     if pool =  $\emptyset$  then  $\{s\}$  else  $\emptyset$ 
       $\cup \bigcup \{\text{Costas}_{gs}(n, s \cdot i, \text{pool} \setminus \{i\}, \text{ssize} + 1) \mid i \in \text{pool}$ 
           $\wedge \forall j < \text{ssize}. s_{\text{ssize}+1-j} - i \notin \text{dtrow}(s, j)\}$ 

```

Modifizierter Aufruf aus Hauptfunktion

```

FUNCTION  $\text{Costas}$  (n:  $\mathbb{Z}$ ) WHERE  $n \geq 1$ 
  RETURNS  $\{p: \text{Seq}(\mathbb{Z}) \mid \text{perm}(p, \{1..n\}) \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))\}$ 
 $\equiv \text{Costas}_{gs}(n, [], \{1..n\}, 0)$ 

```

- **Separate Analyse von Einzelfällen**

- Auswertung von Tests aus anderen Programmteilen
- Zweck: globale Vereinfachung durch lokale Einzelanalyse
- Formal: Erzeugung eines Kontexts + CD-Simplifikation

- **Kontexterzeugung mit Prädikat P**

- Ersetze Ausdruck $E[x]$ durch **if** $P[x]$ **then** $E[x]$ **else** $E[x]$
- Vereinfache $E[x]$ in den entsprechenden Kontexten $P[x]$ und $\neg P[x]$
- Distribuiere **if** $P[x]$ **then...else...**
über Ausdrücke außerhalb von $E[x]$

- **Benutzerinteraktion:**

- Auswahl des zu ersetzenden Teilausdrucks $E[x]$
- Auswahl des Prädikats $P[x]$
- Auslösung der nachfolgenden Simplifikationen

FALLANALYSE IN Costas_{aux}

```

FUNCTION  $\text{Costas}_{aux}(n, s, \text{pool}, \text{ssize} : \mathbb{Z} \times \text{Seq}(\mathbb{Z}) \times \text{Set}(\mathbb{Z}) \times \mathbb{Z})$ 
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
         $\wedge \text{pool} = \{1..n\} \setminus \text{range}(s) \wedge \text{ssize} = |s|$ 
  RETURNS  $\{p : \text{Seq}(\mathbb{Z}) \mid \text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))\}$ 
 $\equiv$  if  $\text{pool} = \emptyset$  then  $\{s\}$  else  $\emptyset$ 
     $\cup \bigcup \{ \text{Costas}_{gs}(n, s \cdot i, \text{pool} \setminus \{i\}, \text{ssize} + 1) \mid i \in \text{pool}$ 
                                      $\wedge \forall j < \text{ssize}. s_{\text{ssize}+1-j} - i \notin \text{dtrow}(s, j) \}$ 

```

```

FUNCTION  $\text{Costas}_{aux}(n, s, \text{pool}, \text{ssize} : \mathbb{Z} \times \text{Seq}(\mathbb{Z}) \times \text{Set}(\mathbb{Z}) \times \mathbb{Z})$ 
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$ 
         $\wedge \text{pool} = \{1..n\} \setminus \text{range}(s) \wedge \text{ssize} = |s|$ 
  RETURNS  $\{p : \text{Seq}(\mathbb{Z}) \mid \text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))\}$ 
 $\equiv$  if  $\text{pool} = \emptyset$  then  $\{s\}$ 
    else  $\bigcup \{ \text{Costas}_{gs}(n, s \cdot i, \text{pool} \setminus \{i\}, \text{ssize} + 1) \mid i \in \text{pool}$ 
                                      $\wedge \forall j < \text{ssize}. s_{\text{ssize}+1-j} - i \notin \text{dtrow}(s, j) \}$ 

```

Ersetze abstrakte Datentypen durch effiziente konkrete Implementierung

- **Endliche Mengen**

- **Listen**: Standardimplementierung
- **Bitvektor**: Mengen über endlichem Domain
- **Charakteristische Funktion**: effiziente Elementrelation/Einfügen/Löschen

- **Folgen**

- **Verkettete Liste**: Standardimplementierung
- **Umgekehrt verkettete Liste**: gut für append-Operation

- **Benutzerinteraktion:**

- System stellt **Auswahl von Implementierungen** bereit
- Benutzer wählt Nichtstandard-Implementierung **einzel**n für jede Variable
- System ersetzt abstrakte Notation durch konkrete Implementierung und fügt ggf. Konversionen ein

DATENTYPVERFEINERUNG FÜR Costas_{aux}

```
FUNCTION  $\text{Costas}_{aux}(n, s, \text{pool}, \text{ssize} : \mathbb{Z} \times \text{Seq}(\mathbb{Z}) \times \text{Set}(\mathbb{Z}) \times \mathbb{Z})$   
  WHERE  $n \geq 1 \wedge \text{range}(s) \subseteq \{1..n\} \wedge \text{nodups}(s) \wedge \forall j < |s|. \text{nodups}(\text{dtrow}(s, j))$   
         $\wedge \text{pool} = \{1..n\} \setminus \text{range}(s) \wedge \text{ssize} = |s|$   
  RETURNS  $\{p : \text{Seq}(\mathbb{Z}) \mid \text{perm}(p, \{1..n\}) \wedge s \sqsubseteq p \wedge \forall j < |p|. \text{nodups}(\text{dtrow}(p, j))\}$   
 $\equiv$  if  $\text{pool} = \emptyset$  then  $\{s\}$   
    else  $\bigcup \{ \text{Costas}_{gs}(n, s \cdot i, \text{pool} \setminus \{i\}, \text{ssize} + 1) \mid i \in \text{pool}$   
               $\wedge \forall j < \text{ssize}. s_{\text{ssize}+1-j} - i \notin \text{dtrow}(s, j) \}$ 
```

$n : \mathbb{Z}$ $\mapsto$ Standardimplementierung positiver ganzen Zahlen

$s : \text{Seq}(\mathbb{Z})$, Elemente werden hinten angehängt $\mapsto$ umgekehrt verkettete Liste

$\text{pool} : \text{Set}(\mathbb{Z})$: Elemente werden aus fester Menge entnommen $\mapsto$ Bitvektor

$\text{ssize} : \mathbb{Z}$ $\mapsto$ Standardimplementierung positiver ganzen Zahlen