# Automatisierte Logik und Programmierung

## Einheit 23

## Korrektheitserhaltende Optimierungen

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

# Optimierung schematisch erzeugter Algorithmen

## Eliminiere überflüssige Berechnungen

- **Simplifikation: Logische Vereinfachung**
  – Äquivalenzumwandlung von Teilausdrücken, ggf. im Kontext
- **Partielle Auswertung**
  – Symbolische Auswertung von Ausdrücken mit Konstanten
- **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

# Simplifikation

## 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

# SIMPLIFIKATION

## 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)`

$$(x.L_1)\circ L_2 = x.(L_1\circ L_2)$$ append / cons

## 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 ∈ (x.[])∘(b.L)` $\mapsto$ `a ∈ x.([]∘(b.L))`

$$(x.L_1)\circ L_2 = x.(L_1\circ L_2)$$ append / cons

# Simplifikation

## 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

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

$\mathtt{[]} \circ L = L$   append / nil

## 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

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

| | |
|---|---|
| $\texttt{[]} \circ \boldsymbol{L} = \boldsymbol{L}$ | append / nil |

## 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

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

| | |
|---|---|
| $\boldsymbol{x \in a.L \;\Leftrightarrow\; x{=}a \vee x \in L}$ | member / cons |

## 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

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

| | |
|---|---|
| $\boldsymbol{x \in a.L \;\Leftrightarrow\; x{=}a \vee x \in L}$ | member / cons |

## 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

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

$\boldsymbol{x \in a.L \;\Leftrightarrow\; x{=}a \vee x \in L}$ member / cons

# Simplifikation

## 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

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

| | |
|---|---|
| $\boldsymbol{x \in a.L \;\Leftrightarrow\; x{=}a \vee x \in L}$ | member / cons |

## 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

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

# Simplifikation

## 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

    $\mathtt{a} \in \mathtt{(x.[])} \circ \mathtt{(b.L)} \;\mapsto\; \mathtt{a=x} \vee \mathtt{a=b} \vee \mathtt{a} \in \mathtt{L}$
- **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 Costas_gs(n,[]) else ∅
```

```
FUNCTION Costas_aux (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)) }
   ∪ ⋃{ Costas_gs(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 Costas_gs(n,[]) else ∅
```

# 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 Costas_gs(n,[]) else ∅
```

**Domänenwissen:** `nodups([])` $\equiv$ `true`

# 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 Costas_gs(n,[]) else ∅
```

---

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

**Domänenwissen:** `nodups([])` ≡ `true`

# 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 Costas_gs(n,[]) else ∅
```

---

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

**Domänenwissen:** `|[]|` $\equiv$ `0`

# 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 Costas_gs(n,[]) else ∅
```

---

```
FUNCTION Costas (n:ℤ) WHERE n≥1
  RETURNS {p:Seq(ℤ) | perm(p,{1..n}) ∧ ∀j<|p|.nodups(dtrow(p,j))}
≡ if true ∧ ∀j<0.nodups(dtrow([],j))
      then Costas_gs(n,[]) else ∅
```

**Domänenwissen:** `|[]|` $\equiv$ `0`

**Domänenwissen:** $\forall x$`<0`$.\,P[x]$ $\equiv$ `true`

# 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 Costas_gs(n,[]) else ∅
```

---

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

---

**Domänenwissen:** $|[]| \quad\equiv\quad 0$

**Domänenwissen:** $\forall x{<}0\,.\,P[x] \quad\equiv\quad \mathtt{true}$

# 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 Costas_gs(n,[]) else ∅
```

---

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

---

**Domänenwissen:** true ∧ true ≡ true

# 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 Costas_gs(n,[]) else ∅
```

---

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

**Domänenwissen:** true ∧ true ≡ true

# 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 Costas_gs(n,[]) else ∅
```

---

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

---

**Domänenwissen:** if true then $a$ else $b$ $\equiv$ $a$

# 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 Costas_gs(n,[]) else ∅
```

---

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

**Domänenwissen:** `if true then` $a$ `else` $b$ $\equiv$ $a$

# 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 Costas_gs(n,[]) else ∅
```

---

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

# CI-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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)) }
   ∪ ⋃{ Costas_gs(n,t) | t∈{ s·i|i∈{1..n}} ∧ nodups(t)
                                            ∧ ∀j<|t|.nodups(dtrow(t,j)) }
```

# CI-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

**Domänenwissen:** $\{z \mid z \in \{x\} \wedge P[z]\} \;\equiv\; \texttt{if } P[x] \texttt{ then } \{x\} \texttt{ else } \emptyset$

# CI-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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)) }
   ∪ ⋃{ Costas_gs(n,t) | t∈{ s·i | i∈{1..n} } ∧ nodups(t)
                                             ∧ ∀j<|t|.nodups(dtrow(t,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,t) | t∈{ s·i | i∈{1..n} } ∧ nodups(t)
                                             ∧ ∀j<|t|.nodups(dtrow(t,j)) }
```

---

**Domänenwissen:** $\{z \mid z \in \{x\} \wedge P[z]\} \equiv \texttt{if } P[x] \texttt{ then } \{x\} \texttt{ else } \emptyset$

# CI-Simplifikationen in der Hilfsfunktion Costas$_{aux}$

```
FUNCTION Costas_aux (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)) }
   ∪ ⋃{ Costas_gs(n,t) | t∈{ s·i | i∈{1..n} } ∧ nodups(t)
                                              ∧ ∀j<|t|.nodups(dtrow(t,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,t) | t∈{ s·i | i∈{1..n} } ∧ nodups(t)
                                              ∧ ∀j<|t|.nodups(dtrow(t,j)) }
```

---

**Domänenwissen:** $\{\, f[x,t] \mid t \in \{g[x,y] \mid y \in S\,\} \wedge h[t]\}$

$$\equiv \quad \{\, f[x,\, g[x,y]\,] \mid y \in S \wedge h[\,g[x,y]\,]\}$$

# CI-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

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

---

**Domänenwissen:** $\{\, f[x,t] \mid t \in \{g[x,y] \mid y \in S\,\} \wedge h[t]\}$

$$\equiv \quad \{\, f[x,\ g[x,y]\,] \mid y \in S \wedge h[\,g[x,y]\,]\}$$

# CI-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

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

# Kontextabhängige Simplifikation

## Vereinfachung unter Berücksichtigung von Kontext

- **Anwendung bedingter Gleichungen**
  - Gleichung $A \equiv B$ mit Vorbedingung $C$
  - Formuliert als Lemma der Form $C \Rightarrow A \equiv B$
- **Anwendbarbarkeit 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 $\text{Costas}_{aux}$

$$\texttt{FUNCTION}\ \texttt{Costas}_{aux}\ (\texttt{n,s}:\mathbb{Z}\times\texttt{Seq}(\mathbb{Z}))$$

$$\quad\texttt{WHERE n}\geq\texttt{1}\ \wedge\ \texttt{range(s)}\subseteq\{\texttt{1..n}\}\ \wedge\ \texttt{nodups(s)}\ \wedge\ \forall\texttt{j<|s|.nodups(dtrow(s,j))}$$

$$\quad\texttt{RETURNS}\ \{\texttt{p:Seq}(\mathbb{Z})\mid\texttt{perm(p,}\{\texttt{1..n}\}\texttt{)}\wedge\ \texttt{s}\sqsubseteq\texttt{p}\wedge\ \forall\texttt{j<|p|.nodups(dtrow(p,j))}\}$$

$$\equiv\quad \texttt{if perm(s,}\{\texttt{1..n}\}\texttt{)}\ \wedge\ \forall\texttt{j<|s|.nodups(dtrow(s,j)) then}\ \{\texttt{s}\}\ \texttt{else}\ \emptyset$$

$$\quad\cup\ \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i)}\mid\texttt{i}\in\{\texttt{1..n}\}\ \wedge\ \texttt{nodups(s}\cdot\texttt{i)}$$

$$\qquad\qquad\qquad\wedge\ \forall\texttt{j<|s}\cdot\texttt{i|.nodups(dtrow(s}\cdot\texttt{i,j))}\}$$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

$\texttt{FUNCTION Costas}_{aux}\ \texttt{(n,s:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{))}$

$\quad\texttt{WHERE n}\geq\texttt{1} \wedge \texttt{range(s)}\subseteq\texttt{\{1..n\}} \wedge \texttt{nodups(s)} \wedge \forall\texttt{j<|s|.nodups(dtrow(s,j))}$

$\quad\texttt{RETURNS \{p:Seq(}\mathbb{Z}\texttt{) | perm(p,\{1..n\})} \wedge \texttt{s}\sqsubseteq\texttt{p} \wedge \forall\texttt{j<|p|.nodups(dtrow(p,j))\}}$

$\equiv\quad\texttt{if perm(s,\{1..n\})} \wedge \forall\texttt{j<|s|.nodups(dtrow(s,j)) then \{s\} else }\emptyset$

$\quad\cup \bigcup\texttt{\{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i) | i}\in\texttt{\{1..n\}} \wedge \texttt{nodups(s}\cdot\texttt{i)}$

$\qquad\qquad\qquad\qquad \wedge \forall\texttt{j<|s}\cdot\texttt{i|.nodups(dtrow(s}\cdot\texttt{i,j))\}}$

**Domänenwissen:** $P \Rightarrow (Q \wedge P \equiv Q)$

**Kontext der Vorbedingung:** $\forall\texttt{j<|s|.nodups(dtrow(s,j))}$

# CD-Simplifikationen in der Hilfsfunktion Costas$_{aux}$

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{Costas_gs(n,s·i)|i∈{1..n} ∧ nodups(s·i)
                                 ∧ ∀j<|s·i|.nodups(dtrow(s·i,j))}
```

---

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) then {s} else ∅
   ∪ ⋃{Costas_gs(n,s·i)|i∈{1..n} ∧ nodups(s·i)
                                 ∧ ∀j<|s·i|.nodups(dtrow(s·i,j))}
```

---

**Domänenwissen:** $P \Rightarrow (Q \wedge P \equiv Q)$

**Kontext der Vorbedingung:** `∀j<|s|.nodups(dtrow(s,j))`

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

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

**Domänenwissen:** $\texttt{perm}(L,M) \equiv \texttt{range}(L)\subseteq M \wedge M\subseteq\texttt{range}(L) \wedge \texttt{nodups}(L)$

**Kontext der Vorbedingung:** $\texttt{range(s)}\subseteq\{\texttt{1..n}\} \wedge \texttt{nodups(s)}$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                    ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡      if {1..n}⊆range(s) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                    ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

**Domänenwissen:** $\texttt{perm}(L,M) \equiv \texttt{range}(L)\subseteq M \wedge M\subseteq\texttt{range}(L) \wedge \texttt{nodups}(L)$

**Kontext der Vorbedingung:** $\texttt{range(s)}\subseteq\{1..n\} \wedge \texttt{nodups(s)}$

**Domänenwissen:** $M\subseteq M' \equiv M\setminus M'=\emptyset$

# CD-Simplifikationen in der Hilfsfunktion Costas$_{aux}$

```
FUNCTION Costas_aux (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))}
≡     if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                     ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡     if {1..n}\range(s)=∅ then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                     ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

**Domänenwissen:** $\mathtt{perm}(L,M) \equiv \mathtt{range}(L)\subseteq M \wedge M\subseteq\mathtt{range}(L) \wedge \mathtt{nodups}(L)$

**Kontext der Vorbedingung:** $\mathtt{range(s)}\subseteq\{1..n\} \wedge \mathtt{nodups(s)}$

**Domänenwissen:** $M\subseteq M' \equiv M\setminus M'=\emptyset$

# CD-Simplifikationen in der Hilfsfunktion Costas$_{aux}$

```
FUNCTION Costas_aux (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))}
≡     if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
    ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                      ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡     if {1..n}\range(s)=∅ then {s} else ∅
    ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                      ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

**Domänenwissen:** nodups($L\cdot x$) $\equiv$ nodups($L$) $\wedge$ $x \notin L$

**Kontext der Vorbedingung:** nodups(s)

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

$$
\begin{array}{l}
\texttt{FUNCTION Costas}_{aux}\ \texttt{(n,s:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{))}\\
\quad \texttt{WHERE n}\geq\texttt{1} \ \wedge\ \texttt{range(s)}\subseteq\texttt{\{1..n\}} \ \wedge\ \texttt{nodups(s)} \ \wedge\ \forall\texttt{j<|s|.nodups(dtrow(s,j))}\\
\quad \texttt{RETURNS \{p:Seq(}\mathbb{Z}\texttt{) | perm(p,\{1..n\})} \wedge\ \texttt{s}\sqsubseteq\texttt{p} \wedge\ \forall\texttt{j<|p|.nodups(dtrow(p,j))\}}\\
\equiv \qquad \texttt{if \{1..n\}}\backslash\texttt{range(s)=}\emptyset\texttt{ then \{s\} else } \emptyset\\
\quad \cup \bigcup\texttt{\{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i) | i}\in\texttt{\{1..n\}} \ \wedge\ \texttt{i}\notin\texttt{s}\\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\ \forall\texttt{j<|s}\cdot\texttt{i|.nodups(dtrow(s}\cdot\texttt{i,j))\}}
\end{array}
$$

**Domänenwissen:** $\texttt{nodups}(L\cdot x) \ \equiv\ \texttt{nodups}(L) \wedge x\notin L$

**Kontext der Vorbedingung:** $\texttt{nodups(s)}$

**Domänenwissen:** $x\in M \ \wedge\ x\notin L \ \equiv\ x\in M\backslash\texttt{range}(L)$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{Costas_gs(n,s·i)|i∈{1..n} ∧ nodups(s·i)
                                 ∧ ∀j<|s·i|.nodups(dtrow(s·i,j))}
```

---

```
FUNCTION Costas_aux (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))}
≡      if {1..n}\range(s)=∅ then {s} else ∅
   ∪ ⋃{Costas_gs(n,s·i)|i∈{1..n}\range(s)
                        ∧ ∀j<|s·i|.nodups(dtrow(s·i,j))}
```

**Domänenwissen:** $\texttt{nodups}(L\cdot x) \equiv \texttt{nodups}(L) \wedge x \notin L$

**Kontext der Vorbedingung:** nodups(s)

**Domänenwissen:** $x \in M \wedge x \notin L \equiv x \in M\backslash\texttt{range}(L)$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                    ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡      if {1..n}\range(s)=∅ then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n}\range(s)
                          ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

**Domänenwissen:** $\texttt{dtrow}(L\cdot x, i) \;\equiv\; \texttt{dtrow}(L, i)\cdot(L_{|L\cdot x|-i} - x)$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

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

**Domänenwissen:** $\texttt{dtrow}(L\cdot x,i) \equiv \texttt{dtrow}(L,i)\cdot(L_{|L\cdot x|-i} - x)$

$\texttt{nodups}(L\cdot x) \equiv \texttt{nodups}(L) \wedge x \notin L$

# CD-Simplifikationen in der Hilfsfunktion Costas$_{aux}$

$$
\begin{array}{ll}
\multicolumn{2}{l}{\mathtt{FUNCTION\ Costas}_{aux}\ \mathtt{(n,s:}\mathbb{Z}\times\mathtt{Seq(}\mathbb{Z}\mathtt{))}}\\
\multicolumn{2}{l}{\quad \mathtt{WHERE\ n}\geq 1\ \wedge\ \mathtt{range(s)}\subseteq\{\mathtt{1..n}\}\ \wedge\ \mathtt{nodups(s)}\ \wedge\ \forall \mathtt{j}<|\mathtt{s}|.\mathtt{nodups(dtrow(s,j))}}\\
\multicolumn{2}{l}{\quad \mathtt{RETURNS}\ \{\mathtt{p:Seq(}\mathbb{Z}\mathtt{)}\mid \mathtt{perm(p,}\{\mathtt{1..n}\}\mathtt{)}\wedge \mathtt{s}\sqsubseteq\mathtt{p}\wedge \forall \mathtt{j}<|\mathtt{p}|.\mathtt{nodups(dtrow(p,j))}\}}\\
\equiv & \quad \mathtt{if\ perm(s,}\{\mathtt{1..n}\}\mathtt{)}\ \wedge\ \forall \mathtt{j}<|\mathtt{s}|.\mathtt{nodups(dtrow(s,j))\ then}\ \{\mathtt{s}\}\ \mathtt{else}\ \emptyset\\
 & \cup \bigcup\{\mathtt{Costas}_{gs}\mathtt{(n,s\cdot i)}\mid \mathtt{i}\in\{\mathtt{1..n}\}\ \wedge\ \mathtt{nodups(s\cdot i)}\\
 & \qquad\qquad\qquad\qquad \wedge\ \forall \mathtt{j}<|\mathtt{s\cdot i}|.\mathtt{nodups(dtrow(s\cdot i,j))}\ \}
\end{array}
$$

---

$$
\begin{array}{ll}
\multicolumn{2}{l}{\mathtt{FUNCTION\ Costas}_{aux}\ \mathtt{(n,s:}\mathbb{Z}\times\mathtt{Seq(}\mathbb{Z}\mathtt{))}}\\
\multicolumn{2}{l}{\quad \mathtt{WHERE\ n}\geq 1\ \wedge\ \mathtt{range(s)}\subseteq\{\mathtt{1..n}\}\ \wedge\ \mathtt{nodups(s)}\ \wedge\ \forall \mathtt{j}<|\mathtt{s}|.\mathtt{nodups(dtrow(s,j))}}\\
\multicolumn{2}{l}{\quad \mathtt{RETURNS}\ \{\mathtt{p:Seq(}\mathbb{Z}\mathtt{)}\mid \mathtt{perm(p,}\{\mathtt{1..n}\}\mathtt{)}\wedge \mathtt{s}\sqsubseteq\mathtt{p}\wedge \forall \mathtt{j}<|\mathtt{p}|.\mathtt{nodups(dtrow(p,j))}\}}\\
\equiv & \quad \mathtt{if}\ \{\mathtt{1..n}\}\backslash\mathtt{range(s)=}\emptyset\ \mathtt{then}\ \{\mathtt{s}\}\ \mathtt{else}\ \emptyset\\
 & \cup \bigcup\{\mathtt{Costas}_{gs}\mathtt{(n,s\cdot i)}\mid \mathtt{i}\in\{\mathtt{1..n}\}\backslash\mathtt{range(s)}\\
 & \qquad\qquad \wedge\ \forall \mathtt{j}<|\mathtt{s\cdot i}|.\mathtt{nodups(dtrow(s,j))}\ \wedge\ \mathtt{s}_{|s\cdot i|-j}\,\mathtt{-i}\notin\mathtt{dtrow(s,j)}\ \}
\end{array}
$$

---

**Domänenwissen:**

$$\mathtt{nodups}(L\cdot x)\ \equiv\ \mathtt{nodups}(L)\ \wedge\ x\notin L$$

$$\forall i<|L\cdot x|.P[i]\ \equiv\ \forall i<|L|.P[i]\ \wedge\ P[|L|]$$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                     ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡      if {1..n}\range(s)=∅ then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n}\range(s)
                    ∧ ∀j<|s|.nodups(dtrow(s,j)) ∧ s_{|s·i|-j}-i ∉ dtrow(s,j)
                          ∧ nodups(dtrow(s,|s|)) ∧ s_{|s·i|-j}-i ∉ dtrow(s,|s|) }
```

**Domänenwissen:** $\forall i{<}|L{\cdot}x|.\,P[i] \;\equiv\; \forall i{<}|L|.\,P[i] \wedge P[|L|]$

$\mathtt{dtrow}(L,|L|) \;\equiv\; []$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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))}
≡     if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n}  ∧ nodups(s·i)
                                     ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡     if {1..n}\range(s)=∅ then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n}\range(s)
                         ∧ ∀j<|s|.nodups(dtrow(s,j)) ∧ s_{|s·i|-j}-i ∉ dtrow(s,j)
                               ∧ nodups([]) ∧ s_{|s·i|-j}-i ∉ [] }
```

---

**Domänenwissen:**

$$\texttt{dtrow}(L,|L|) \equiv \texttt{[]}$$

$$\texttt{nodups([])} \equiv \texttt{true}$$

$$x \notin \texttt{[]} \equiv \texttt{true}$$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

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

---

**Domänenwissen:**

$$
\begin{array}{rcl}
\texttt{nodups([])} & \equiv & \texttt{true} \\
x \notin \texttt{[]} & \equiv & \texttt{true} \\
\forall x.\, P[x] \wedge Q[x] & \equiv & \forall x.\, P[x] \ \wedge\ \forall x.\, Q[x]
\end{array}
$$

# CD-Simplifikationen in der Hilfsfunktion Costas$_{aux}$

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

---

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

---

**Domänenwissen:** $\forall x.\,P[x]\wedge Q[x]\ \equiv\ \forall x.\,P[x]\ \wedge\ \forall x.\,Q[x]$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

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

---

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

**Kontext der Vorbedingung:** $\forall\texttt{j<|s|.nodups(dtrow(s,j))}$

# CD-Simplifikationen in der Hilfsfunktion $\text{Costas}_{aux}$

```
FUNCTION Costas_aux (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))}
≡     if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n} ∧ nodups(s·i)
                                     ∧ ∀j<|s·i|.nodups(dtrow(s·i,j)) }
```

---

```
FUNCTION Costas_aux (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))}
≡     if {1..n}\range(s)=∅ then {s} else ∅
   ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n}\range(s) ∧ ∀j<|s|. s_{|s·i|-j} -i ∉ dtrow(s,j) }
```

**Kontext der Vorbedingung:** `∀j<|s|.nodups(dtrow(s,j))`

# CD-Simplifikationen in der Hilfsfunktion Costas$_{aux}$

```
FUNCTION Costas_aux (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))}
≡      if perm(s,{1..n}) ∧ ∀j<|s|.nodups(dtrow(s,j)) then {s} else ∅
   ∪ ⋃{Costas_gs(n,s·i)|i∈{1..n} ∧ nodups(s·i)
                                ∧ ∀j<|s·i|.nodups(dtrow(s·i,j))}
```

---

```
FUNCTION Costas_aux (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))}
≡      if {1..n}\range(s)=∅ then {s} else ∅
   ∪ ⋃{Costas_gs(n,s·i)|i∈{1..n}\range(s) ∧ ∀j<|s|. s_{|s·i|-j} -i ∉ dtrow(s,j)}
```

# Partielle Auswertung

## 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

# Partielle Auswertung

## 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

# Partielle Auswertung

## 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:**

| | | |
|---|---|---|
| | $\mid [x;5] \circ L \mid$ | unfold append |
| $\mapsto$ | $\mid x.([5] \circ L) \mid$ | unfold append |

# PARTIELLE AUSWERTUNG

## 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:**

| | | |
|---|---|---|
| | $\texttt{|}\,\texttt{[}x\texttt{;5]}\circ L\,\texttt{|}$ | unfold append |
| $\mapsto$ | $\texttt{|}x\texttt{.}\,\texttt{([5]}\circ L\texttt{)}\,\texttt{|}$ | unfold append |
| $\mapsto$ | $\texttt{|}x\texttt{.}\,\texttt{(5.}\,\texttt{([]}\circ L\texttt{))}\,\texttt{|}$ | unfold append |

# Partielle Auswertung

## 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:**

$$
\begin{array}{lll}
 & |\,[x;5] \circ L\,| & \text{unfold append} \\
\mapsto & |\,x.\,([5] \circ L)\,| & \text{unfold append} \\
\mapsto & |\,x.\,(5.\,([] \circ L))\,| & \text{unfold append} \\
\mapsto & |\,x.\,(5.\,L)\,| & \text{unfold length}
\end{array}
$$

## 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:**

| | | |
|---|---|---|
| | $\texttt{|[}x\texttt{;5]} \circ L\texttt{|}$ | unfold append |
| $\mapsto$ | $\texttt{|}x\texttt{.([5]} \circ L\texttt{)|}$ | unfold append |
| $\mapsto$ | $\texttt{|}x\texttt{.(5.([]} \circ L\texttt{))|}$ | unfold append |
| $\mapsto$ | $\texttt{|}x\texttt{.(5.}L\texttt{)|}$ | unfold length |
| $\mapsto$ | $\texttt{1 + |5.}L\texttt{|}$ | unfold length |

# Partielle Auswertung

## 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:**

| | |
|---|---|
| $\quad$ \|[$x$;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 |

# Partielle Auswertung

## 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 |
| $\mapsto$ | $2 + \|L\|$ | |

## 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

- **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**

$\texttt{FUNCTION Costas}_{aux}\texttt{(n,s:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{))}$ WHERE ... RETURNS ...

$\equiv \quad \texttt{if } \{1..n\}\backslash\texttt{range(s)}=\emptyset \texttt{ then } \{s\} \texttt{ else } \emptyset$

$\quad \cup \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i)} \mid \texttt{i} \in \{1..n\}\backslash\texttt{range(s)} \wedge \forall \texttt{j<|s|. s}_{|s\cdot i|-j}\texttt{-i} \notin \texttt{dtrow(s,j)}\}$

# 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**

$$\texttt{FUNCTION Costas}_{aux}\texttt{(n,s:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{))  WHERE ...  RETURNS ...}$$

$$\equiv \quad \texttt{if } \{\texttt{1..n}\}\backslash\texttt{range(s)=}\emptyset \texttt{ then } \{\texttt{s}\} \texttt{ else } \emptyset$$

$$\cup \bigcup \{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i)} \mid \texttt{i} \in \{\texttt{1..n}\}\backslash\texttt{range(s)} \wedge \forall \texttt{j<|s|. s}_{|s\cdot i|-j}\texttt{-i} \notin \texttt{dtrow(s,j)}\}$$

---

- Benutzer identifiziert $\{\texttt{1..n}\}\backslash\texttt{range(s)}$ als wiederkehrende Berechnung

# 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**

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

---

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

---

- Benutzer identifiziert $\{\texttt{1..n}\}\backslash\texttt{range(s)}$ als wiederkehrende Berechnung
- System ändert $\texttt{Costas}_{aux}\texttt{(n,s)}$ in $\texttt{Costas}_{aux}\texttt{(n,s,pool)}$
  und ändert Aufrufe von $\texttt{Costas}_{aux}$ entsprechend

# 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**

$\texttt{FUNCTION Costas}_{aux}\texttt{(n,s:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{))}\ \ \texttt{WHERE}\ \ldots\ \ \texttt{RETURNS}\ \ldots$

$\equiv\quad \texttt{if \{1..n\}\textbackslash range(s)=}\emptyset\texttt{ then \{s\} else }\emptyset$

$\quad\cup \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i)}\mid \texttt{i}\in\texttt{\{1..n\}\textbackslash range(s)} \wedge \forall\texttt{j<|s|. s}_{|s\cdot i|-j}\texttt{-i}\notin\texttt{dtrow(s,j)}\}$

---

$\texttt{FUNCTION Costas}_{aux}\texttt{(n,s,pool }\ldots\texttt{)}\ \ \texttt{WHERE}\ \ldots\ \texttt{pool = \{1..n\}\textbackslash range(s)}$

$\equiv\quad \texttt{if \{1..n\}\textbackslash range(s)=}\emptyset\texttt{ then \{s\} else }\emptyset$

$\quad\cup \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i,\{1..n\}\textbackslash range(s}\cdot\texttt{i))}\mid \texttt{i}\in\texttt{\{1..n\}\textbackslash range(s)}$

$\qquad\qquad \wedge \forall\texttt{j<|s|. s}_{|s\cdot i|-j}\texttt{-i}\notin\texttt{dtrow(s,j)}\}$

---

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

# 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 Costas_aux(n,s:ℤ×Seq(ℤ))  WHERE …  RETURNS …
≡     if {1..n}\range(s)=∅ then {s} else ∅
  ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n}\range(s) ∧ ∀j<|s|. s_{|s·i|-j} -i ∉ dtrow(s,j)}
```

---

```
FUNCTION Costas_aux(n,s,pool …)  WHERE … pool = {1..n}\range(s)
≡     if {1..n}\range(s)=∅ then {s} else ∅
  ∪ ⋃{ Costas_gs(n,s·i,{1..n}\range(s·i)) | i∈{1..n}\range(s)
                                 ∧ ∀j<|s|. s_{|s·i|-j} -i ∉ dtrow(s,j)}
```

---

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

# 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**

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

---

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

---

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

# Endliche Differenzierung – Formalia

- **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 $O[x,y]$
  $\equiv$ ....$E[x]$....$f'(\,t[x],\ E[t[x]]\,)$..

# Endliche Differenzierung – Formalia

- **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 $c$

# Endliche Differenzierung – Formalia

- **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 $\texttt{Costas}_{aux}$

$\texttt{FUNCTION Costas}_{aux}\ \texttt{(n,s:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{))}$

$\quad\texttt{WHERE n}\geq\texttt{1}\ \wedge\ \texttt{range(s)}\subseteq\{\texttt{1..n}\}\ \wedge\ \texttt{nodups(s)}\ \wedge\ \forall\texttt{j<|s|.nodups(dtrow(s,j))}$

$\quad\texttt{RETURNS}\ \{\texttt{p:Seq(}\mathbb{Z}\texttt{)}\mid\texttt{perm(p,}\{\texttt{1..n}\}\texttt{)}\wedge\ \texttt{s}\sqsubseteq\texttt{p}\wedge\ \forall\texttt{j<|p|.nodups(dtrow(p,j))}\}$

$$\equiv\qquad \texttt{if}\ \{\texttt{1..n}\}\backslash\texttt{range(s)=}\emptyset\ \texttt{then}\ \{\texttt{s}\}\ \texttt{else}\ \emptyset$$

$$\cup\ \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i)}\mid\texttt{i}\in\{\texttt{1..n}\}\backslash\texttt{range(s)}\ \wedge\ \forall\texttt{j<|s|.}\ \texttt{s}_{|s\cdot i|-j}\texttt{-i}\notin\texttt{dtrow(s,j)}\}$$

# Endliche Differenzierung von $\texttt{Costas}_{aux}$

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

Benutzer identifiziert $\texttt{\{1..n\}}\backslash\texttt{range(s)}$ als wiederkehrende Berechnung

# Endliche Differenzierung von $\texttt{Costas}_{aux}$

```
FUNCTION Costas_aux (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))}
≡       if {1..n}\range(s)=∅ then {s} else ∅
  ∪ ⋃{ Costas_gs(n,s·i) | i∈{1..n}\range(s) ∧ ∀j<|s|. s_{|s·i|-j} -i ∉ dtrow(s,j)}
```

---

```
FUNCTION Costas_aux(n,s,pool:ℤ×Seq(ℤ)×Set(ℤ))
  WHERE n≥1 ∧ range(s)⊆{1..n} ∧ nodups(s) ∧ ∀j<|s|.nodups(dtrow(s,j))
            ∧ pool = {1..n}\range(s)
  RETURNS {p:Seq(ℤ) | perm(p,{1..n}) ∧ s⊑p ∧ ∀j<|p|.nodups(dtrow(p,j))}
≡      if pool=∅ then {s} else ∅
  ∪ ⋃{ Costas_gs(n,s·i,pool\{i}) | i∈pool ∧ ∀j<|s|. s_{|s·i|-j} -i ∉ dtrow(s,j)}
```

---

Benutzer identifiziert $\{\texttt{1..n}\}\backslash\texttt{range(s)}$ als wiederkehrende Berechnung

System ändert $\texttt{Costas}_{aux}\texttt{(n,s)}$ in $\texttt{Costas}_{aux}\texttt{(n,s,pool)}$ und ändert Aufrufe

System ergänzt $\texttt{pool} = \{\texttt{1..n}\}\backslash\texttt{range(s)}$ zur Eingabebedingung

und vereinfacht Vorkommen von $\{\texttt{1..n}\}\backslash\texttt{range(s)}$ zu $\texttt{pool}$

# Endliche Differenzierung von $\mathtt{Costas}_{aux}$

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

---

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

Benutzer identifiziert $\mathtt{|s|}$ als wiederkehrende Berechnung

# Endliche Differenzierung von $\texttt{Costas}_{aux}$

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

---

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

---

Benutzer identifiziert `|s|` als wiederkehrende Berechnung

System ändert $\texttt{Costas}_{aux}\texttt{(n,s,pool)}$ in $\texttt{Costas}_{aux}\texttt{(n,s,pool,ssize)}$

System ergänzt `ssize = |s|` zur Eingabebedingung

und vereinfacht Vorkommen von `|s|` zu `ssize`

# Endliche Differenzierung von $\texttt{Costas}_{aux}$

FUNCTION $\texttt{Costas}_{aux}$ $(\texttt{n},\texttt{s}:\mathbb{Z}\times\texttt{Seq}(\mathbb{Z}))$

WHERE $\texttt{n}\geq 1 \;\wedge\; \texttt{range(s)}\subseteq\{1..\texttt{n}\} \;\wedge\; \texttt{nodups(s)} \;\wedge\; \forall \texttt{j}<|\texttt{s}|.\texttt{nodups(dtrow(s,j))}$

RETURNS $\{\texttt{p}:\texttt{Seq}(\mathbb{Z})\mid \texttt{perm}(\texttt{p},\{1..\texttt{n}\}) \wedge \texttt{s}\sqsubseteq\texttt{p} \wedge \forall \texttt{j}<|\texttt{p}|.\texttt{nodups(dtrow(p,j))}\}$

$\equiv$ if $\{1..\texttt{n}\}\setminus\texttt{range(s)}=\emptyset$ then $\{\texttt{s}\}$ else $\emptyset$

$\cup \bigcup\{\texttt{Costas}_{gs}(\texttt{n},\texttt{s}\cdot\texttt{i}) \mid \texttt{i}\in\{1..\texttt{n}\}\setminus\texttt{range(s)} \;\wedge\; \forall \texttt{j}<|\texttt{s}|.\, \texttt{s}_{|s\cdot i|-j} - \texttt{i} \notin \texttt{dtrow(s,j)}\}$

---

FUNCTION $\texttt{Costas}_{aux}(\texttt{n},\texttt{s},\texttt{pool},\texttt{ssize}:\mathbb{Z}\times\texttt{Seq}(\mathbb{Z})\times\texttt{Set}(\mathbb{Z})\times\mathbb{Z})$

WHERE $\texttt{n}\geq 1 \;\wedge\; \texttt{range(s)}\subseteq\{1..\texttt{n}\} \;\wedge\; \texttt{nodups(s)} \;\wedge\; \forall \texttt{j}<|\texttt{s}|.\texttt{nodups(dtrow(s,j))}$

$\wedge\; \texttt{pool} = \{1..\texttt{n}\}\setminus\texttt{range(s)} \;\wedge\; \texttt{ssize}=|\texttt{s}|$

RETURNS $\{\texttt{p}:\texttt{Seq}(\mathbb{Z})\mid \texttt{perm}(\texttt{p},\{1..\texttt{n}\}) \wedge \texttt{s}\sqsubseteq\texttt{p} \wedge \forall \texttt{j}<|\texttt{p}|.\texttt{nodups(dtrow(p,j))}\}$

$\equiv$ if $\texttt{pool}=\emptyset$ then $\{\texttt{s}\}$ else $\emptyset$

$\cup \bigcup\{\texttt{Costas}_{gs}(\texttt{n},\texttt{s}\cdot\texttt{i},\texttt{pool}\setminus\{\texttt{i}\},\texttt{ssize+1}) \mid \texttt{i}\in\texttt{pool}$

$\wedge\; \forall \texttt{j}<\texttt{ssize}.\, \texttt{s}_{\texttt{ssize}+1-\texttt{j}} - \texttt{i} \notin \texttt{dtrow(s,j)}\}$

## Modifizierter Aufruf aus Hauptfunktion

FUNCTION $\texttt{Costas}$ $(\texttt{n}:\mathbb{Z})$ WHERE $\texttt{n}\geq 1$

RETURNS $\{\texttt{p}:\texttt{Seq}(\mathbb{Z})\mid \texttt{perm}(\texttt{p},\{1..\texttt{n}\}) \wedge \forall \texttt{j}<|\texttt{p}|.\texttt{nodups(dtrow(p,j))}\}$

$\equiv \texttt{Costas}_{gs}(\texttt{n},[\,],\{1..\texttt{n}\},0)$

# Fallanalyse

- **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 $\neg P[x]$
  - Distributiere `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 $\mathtt{Costas}_{aux}$

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

# Fallanalyse in $\texttt{Costas}_{aux}$

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

Markierung von Ausdruck und Testprädikat

# Fallanalyse in $\mathtt{Costas}_{aux}$

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

---

$$\begin{array}{l}
\mathtt{FUNCTION}\ \mathtt{Costas}_{aux}(\mathtt{n},\mathtt{s},\mathtt{pool},\mathtt{ssize}:\mathbb{Z}\times\mathtt{Seq}(\mathbb{Z})\times\mathtt{Set}(\mathbb{Z})\times\mathbb{Z})\\
\quad\mathtt{WHERE}\ \ldots\ \mathtt{RETURNS}\ \ldots\\
\equiv\quad \mathtt{if}\ \mathtt{pool}=\emptyset\ \mathtt{then}\ \{\mathtt{s}\}\ \mathtt{else}\ \emptyset\\
\quad\cup\ \mathtt{if}\ \mathtt{pool}=\emptyset\ \mathtt{then}\ \bigcup\{\mathtt{Costas}_{gs}(\mathtt{n},\mathtt{s}\cdot\mathtt{i},\mathtt{pool}\backslash\{\mathtt{i}\},\mathtt{ssize}+1)\ |\ \mathtt{i}\in\mathtt{pool}\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge\ \forall \mathtt{j}<\mathtt{ssize}.\ \mathtt{s}_{\mathtt{ssize}+1-\mathtt{j}}-\mathtt{i}\notin\mathtt{dtrow}(\mathtt{s},\mathtt{j})\}\\
\qquad\qquad\quad \mathtt{else}\ \bigcup\{\mathtt{Costas}_{gs}(\mathtt{n},\mathtt{s}\cdot\mathtt{i},\mathtt{pool}\backslash\{\mathtt{i}\},\mathtt{ssize}+1)\ |\ \mathtt{i}\in\mathtt{pool}\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge\ \forall \mathtt{j}<\mathtt{ssize}.\ \mathtt{s}_{\mathtt{ssize}+1-\mathtt{j}}-\mathtt{i}\notin\mathtt{dtrow}(\mathtt{s},\mathtt{j})\}
\end{array}$$

---

Aufspalten des Ausdrucks auf beide Fälle

# FALLANALYSE IN $\texttt{Costas}_{aux}$

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

---

$$
\begin{array}{l}
\texttt{FUNCTION Costas}_{aux}\texttt{(n,s,pool,ssize:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{)}\times\texttt{Set(}\mathbb{Z}\texttt{)}\times\mathbb{Z}\texttt{)}\\
\quad\texttt{WHERE ... RETURNS ...}\\
\equiv \quad \texttt{if pool=}\emptyset\texttt{ then }\{\texttt{s}\}\texttt{ else }\emptyset\\
\quad\cup\ \texttt{if pool=}\emptyset\texttt{ then } \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i,pool}\backslash\{\texttt{i}\}\texttt{,ssize+1)} \mid \texttt{i}\in\texttt{pool}\\
\qquad\qquad\qquad\qquad \wedge\ \forall\texttt{j<ssize. s}_{\texttt{ssize+1-j}}\texttt{-i}\notin\texttt{dtrow(s,j)}\}\\
\qquad\qquad\texttt{else } \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i,pool}\backslash\{\texttt{i}\}\texttt{,ssize+1)} \mid \texttt{i}\in\texttt{pool}\\
\qquad\qquad\qquad\qquad \wedge\ \forall\texttt{j<ssize. s}_{\texttt{ssize+1-j}}\texttt{-i}\notin\texttt{dtrow(s,j)}\}
\end{array}
$$

Domänenwissen: $\bigcup\{\, f\texttt{(i)} \mid \texttt{i}\in\emptyset\} = \emptyset$

Kontext der Fallanalyse: $\texttt{pool=}\emptyset$

# FALLANALYSE IN $\texttt{Costas}_{aux}$

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

---

$$
\begin{array}{l}
\texttt{FUNCTION Costas}_{aux}\texttt{(n,s,pool,ssize:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{)}\times\texttt{Set(}\mathbb{Z}\texttt{)}\times\mathbb{Z}\texttt{)} \\
\quad \texttt{WHERE ... RETURNS ...} \\
\equiv \quad \texttt{if pool=}\emptyset\texttt{ then }\{\texttt{s}\}\texttt{ else }\emptyset \\
\quad \cup\ \texttt{if pool=}\emptyset\texttt{ then }\emptyset \\
\qquad\qquad \texttt{else } \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i,pool}\backslash\{\texttt{i}\}\texttt{,ssize+1)} \mid \texttt{i}\in\texttt{pool} \\
\qquad\qquad\qquad \wedge \forall\texttt{j<ssize. s}_{\texttt{ssize}+1-\texttt{j}}\texttt{-i} \notin \texttt{dtrow(s,j)}\}
\end{array}
$$

---

Domänenwissen: $\bigcup\{\, f\texttt{(i)} \mid \texttt{i}\in\emptyset\} = \emptyset$

Kontext der Fallanalyse: $\texttt{pool=}\emptyset$

# Fallanalyse in $\mathtt{Costas}_{aux}$

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

---

$$
\begin{array}{l}
\mathtt{FUNCTION}\ \mathtt{Costas}_{aux}(\mathtt{n,s,pool,ssize}:\mathbb{Z}\times\mathtt{Seq}(\mathbb{Z})\times\mathtt{Set}(\mathbb{Z})\times\mathbb{Z})\\
\quad\mathtt{WHERE}\ \ldots\ \mathtt{RETURNS}\ \ldots\\
\equiv\quad\mathtt{if\ pool}=\emptyset\ \mathtt{then}\ \{\mathtt{s}\}\ \mathtt{else}\ \emptyset\\
\quad\cup\ \mathtt{if\ pool}=\emptyset\ \mathtt{then}\ \emptyset\\
\qquad\qquad\mathtt{else}\ \bigcup\{\mathtt{Costas}_{gs}(\mathtt{n,s\cdot i,pool}\backslash\{\mathtt{i}\}\mathtt{,ssize+1})\mid\mathtt{i}\in\mathtt{pool}\\
\qquad\qquad\qquad\qquad\wedge\ \forall\mathtt{j<ssize.}\ \mathtt{s}_{\mathtt{ssize+1-j}}\mathtt{-i}\notin\mathtt{dtrow(s,j)}\}
\end{array}
$$

---

Distribution von $\mathtt{if\ pool}=\emptyset\ \mathtt{then}\ \ldots\ \mathtt{else}\ ..$ über $\cup$

# Fallanalyse in $\texttt{Costas}_{aux}$

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

---

$$
\begin{array}{l}
\texttt{FUNCTION Costas}_{aux}\texttt{(n,s,pool,ssize:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{)}\times\texttt{Set(}\mathbb{Z}\texttt{)}\times\mathbb{Z}\texttt{)} \\
\quad \texttt{WHERE ... RETURNS ...} \\
\equiv \quad \texttt{if pool=}\emptyset\texttt{ then }\{\texttt{s}\}\cup\emptyset \\
\qquad\qquad\qquad \texttt{else } \emptyset \cup \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i,pool}\backslash\{\texttt{i}\}\texttt{,ssize+1)} \mid \texttt{i}\in\texttt{pool} \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\; \forall\texttt{j<ssize.}\,\texttt{s}_{\texttt{ssize+1-j}}\texttt{-i} \notin \texttt{dtrow(s,j)}\}
\end{array}
$$

---

Distribution von `if pool=∅ then ... else ..` über ∪

# Fallanalyse in $\texttt{Costas}_{aux}$

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

---

$$
\begin{array}{l}
\texttt{FUNCTION Costas}_{aux}\texttt{(n,s,pool,ssize:}\mathbb{Z}\times\texttt{Seq(}\mathbb{Z}\texttt{)}\times\texttt{Set(}\mathbb{Z}\texttt{)}\times\mathbb{Z}\texttt{)}\\
\quad\texttt{WHERE ... RETURNS ...}\\
\equiv \quad \texttt{if pool=}\emptyset\texttt{ then }\{\texttt{s}\}\cup\emptyset\\
\qquad\qquad\qquad \texttt{else } \emptyset \cup \bigcup\{\texttt{Costas}_{gs}\texttt{(n,s}\cdot\texttt{i,pool}\backslash\{\texttt{i}\}\texttt{,ssize+1)} \mid \texttt{i}\in\texttt{pool}\\
\qquad\qquad\qquad\qquad\qquad \wedge\ \forall\texttt{j<ssize.}\ \texttt{s}_{\texttt{ssize+1-j}}\texttt{-i} \notin \texttt{dtrow(s,j)}\}
\end{array}
$$

Domänenwissen: $S\cup\emptyset = \emptyset = \emptyset\cup S$

# Fallanalyse in $\texttt{Costas}_{aux}$

```
FUNCTION Costas_aux(n,s,pool,ssize:ℤ×Seq(ℤ)×Set(ℤ)×ℤ)
  WHERE n≥1 ∧ range(s)⊆{1..n} ∧ nodups(s) ∧ ∀j<|s|.nodups(dtrow(s,j))
             ∧ pool = {1..n}\range(s) ∧ ssize=|s|
  RETURNS {p:Seq(ℤ) | perm(p,{1..n}) ∧ s⊑p ∧ ∀j<|p|.nodups(dtrow(p,j))}
≡     if pool=∅ then {s} else ∅
  ∪ ⋃{ Costas_gs(n,s·i,pool\{i},ssize+1) | i∈pool
                                       ∧ ∀j<ssize. s_(ssize+1−j) −i ∉ dtrow(s,j)}
```

---

```
FUNCTION Costas_aux(n,s,pool,ssize:ℤ×Seq(ℤ)×Set(ℤ)×ℤ)
  WHERE ... RETURNS ...
≡   if pool=∅ then {s}
               else ⋃{ Costas_gs(n,s·i,pool\{i},ssize+1) | i∈pool
                                       ∧ ∀j<ssize. s_(ssize+1−j) −i ∉ dtrow(s,j)}
```

Domänenwissen: $S\cup\emptyset = \emptyset = \emptyset\cup S$

# FALLANALYSE IN $\texttt{Costas}_{aux}$

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

---

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

# Datentypverfeinerung

**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 einzeln für jede Variable
  – System ersetzt abstrakte Notation durch konkrete Implementierung
  und fügt ggf. Konversionen ein

# Datentypverfeinerung für $\texttt{Costas}_{aux}$

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

# Datentypverfeinerung für $\texttt{Costas}_{aux}$

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

---

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

# Datentypverfeinerung für $\texttt{Costas}_{aux}$

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

---

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

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

# Datentypverfeinerung für $\mathtt{Costas}_{aux}$

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

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

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

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

# Datentypverfeinerung für $\texttt{Costas}_{aux}$

```
FUNCTION Costas_aux(n,s,pool,ssize:ℤ×Seq(ℤ)×Set(ℤ)×ℤ)
  WHERE n≥1 ∧ range(s)⊆{1..n} ∧ nodups(s) ∧ ∀j<|s|.nodups(dtrow(s,j))
            ∧ pool = {1..n}\range(s) ∧ ssize=|s|
  RETURNS {p:Seq(ℤ) | perm(p,{1..n}) ∧ s⊑p ∧ ∀j<|p|.nodups(dtrow(p,j))}
≡   if pool=∅ then {s}
               else ⋃{ Costas_gs(n,s·i,pool\{i},ssize+1) | i∈pool
                                              ∧ ∀j<ssize. s_(ssize+1−j) −i ∉ dtrow(s,j)}
```

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

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

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

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

# Formale Optimierungen im Rückblick

- **Schematische Algorithmen müssen optimiert werden**
  - Synthese erzeugt Algorithmus durch Einsetzen von Parametern
    Algorithmus ist garantiert korrekt, aber oft voller Redundanzen
  - Benutzer erkennen Optimierungsmöglichkeiten oft unmittelbar
- **Korrektheit von Optimierungen muß gesichert sein**
  - Optimierung "von Hand" stellt Korrektheitsgarantie in Frage
  - Optimierung muß genauso formal sein wie ursprüngliche Synthese
- **Es gibt viele Arten von Optimierungsmöglichkeiten**
  - Logische Vereinfachung mit und ohne Kontext, Fallanalyse
  - Partielle Auswertung in Anwesenheit von Konstanten
  - Endliche Differenzierung ersetzt Schleifen durch neue Parameter
  - Datentyp-Verfeinerung bestimmt Implementierungsstrukturen

  Alle Optimierungsarten basieren auf formal-logischen Inferenzen
- **Ausführung analog zum interaktiven Beweisern**
  - Benutzer wählt Art und Stelle der Optimierungsschritte
  - System garantiert Korrektheit aller ausgeführten Optimierungen