

Entscheidungsverfahren  
mit Anwendungen in der Softwareverifikation

## **IV: Lineare ganzzahlige Arithmetik**

---

Carsten Sinz  
Institut für Theoretische Informatik

12.11.2019

- **Übungen finden 14-tägig am Mittwoch statt, erster Termin: 13.11.2018**
  - 15:45-17:15 Uhr, Seminarraum 301
- **Übungen bestehen aus theoretischen und praktischen Aufgaben (praktisch: Programmieren)**
  - Lösungen müssen nicht abgegeben werden, aber in der Übung besprochen
- **Insgesamt wird es mindestens 120 Punkte für Übungsaufgaben geben, plus zusätzliche Bonuspunkte**
- **Mindestens 60 Punkte sind zum Bestehen der Übungen erforderlich und Voraussetzung, um die mündliche Prüfung anzutreten**
- **Durch Übungspunkte kann die Note verbessert werden:**
  - $\geq 90$  Punkte: +0,3
  - $\geq 110$  Punkte: +0,7

- **LIA: Linear Integer Arithmetic**

- **Signatur:**

$$\Sigma = \{ =, >, \geq, +, -, \dots, -2, -1, 0, 1, 2, \dots \}$$

- **Beispiele:**

1.  $8x + 6y \leq 0$

2.  $4y \geq 1$

3.  $3x + 3y = 2$

- **Beobachtungen:**

1. kann vereinfacht werden:  $4x + 3y \leq 0$

2. kann verschärft werden:  $y \geq \lceil 1/4 \rceil = 1$

3. Keine ganzzahlige Lösung

- Boolesche Kombination von linearen Gleichungen und Ungleichungen

- Darstellung:

- Lineare Gleichungen: 
$$E = \left\{ \sum_{i=1}^n a_{j,i} x_i = c_j \right\}_{1 \leq j \leq m}$$

- Lineare Ungleichungen: 
$$I = \left\{ \sum_{i=1}^n b_{j,i} x_i \leq d_j \right\}_{1 \leq j \leq m'}$$

- **Entwicklung eines Entscheidungsverfahrens in mehreren Schritten:**

1. Konjunktion von Gleichungen
2. Konjunktion von Gleichungen und Ungleichungen
3. Beliebige Boolesche Kombinationen von Gleichungen und Ungleichungen

- **Eingabe:** Menge von Gleichungen der Form  $a_1x_1 + \dots + a_nx_n = c$
- **Ausgabe:** Gleichungssystem in gelöster Form ( $x_i = \dots$ , „Dreiecksform“) oder Ausgabe „unerfüllbar“
- **Vorverarbeitung (für alle Gleichungen):**
  - **Erfüllbarkeitstest:** Falls  $\text{ggT}(\{a_i\}_{1 \leq i \leq n}) \nmid c$ , gib aus „unerfüllbar“
  - **Normalisieren:** Teile alle  $a_i$  und  $c$  durch  $\text{ggT}(\{a_i\}_{1 \leq i \leq n})$
- **Beispiele:**

$$12x + 15y = 7$$

Erfüllbarkeitstest:  $\text{ggT}(\{a_i\}) = 3$ ,  $3 \nmid 7$ , also Gleichung nicht erfüllbar

$$24x + 12y + 10z = 4$$

Erfüllbarkeitstest:  $\text{ggT}(\{a_i\}) = 2$ ,  $2 \mid 4$ , also Gleichung erfüllbar

Normalisieren:  $12x + 6y + 5z = 2$

- Weiteres Vorgehen:  
Eliminiere Gleichungen schrittweise durch Auflösen nach  $x_i$
- Falls es eine Gleichung gibt mit  $|a_k| = 1$ , löse diese Gleichung nach  $x_k$  auf und ersetze  $x_k$  in allen anderen Gleichungen
- Falls es kein solches  $a_k$  gibt, wähle eine Gleichung  $E$  und ein  $a_k$  mit **kleinstem Betrag** und mache  $a_k$  bei Bedarf positiv (durch Multiplikation der Gleichung mit -1)
- Definiere:
  - $a \bmod' b := a - b \lfloor a/b + 1/2 \rfloor$
  - Setze  $m := a_k + 1$
  - Damit gilt:  $a_k \bmod' m = -1$
- Erzeuge neue Variable  $\sigma$  und füge neue Gleichung hinzu:

$$m\sigma = \sum_i (a_i \bmod' m) \cdot x_i - (c \bmod' m)$$

- Löse die Gleichung  $m\sigma = \sum_i (a_i \bmod' m) \cdot x_i - (c \bmod' m)$  nach  $x_k$  auf (beachte:  $a_k \bmod' m = -1$ ):

$$m\sigma = -x_k + \sum_{i \neq k} (a_i \bmod' m) \cdot x_i - (c \bmod' m)$$

$$x_k = -m\sigma + \sum_{i \neq k} (a_i \bmod' m) \cdot x_i - (c \bmod' m)$$

- Ersetze nun dieses  $x_k$  in allen Gleichungen
- Aus der ursprünglichen Gleichung E:  $a_1x_1 + \dots + a_nx_n = c$  (ang.  $a_k > 0$ ) wird damit:

$$a_kx_k + \sum_{i \neq k} a_ix_i = c$$

$$-a_k m\sigma + a_k \sum_{i \neq k} (a_i \bmod' m) \cdot x_i - a_k(c \bmod' m) + \sum_{i \neq k} a_ix_i = c$$

$$-a_k m\sigma + \sum_{i \neq k} (a_i + a_k(a_i \bmod' m)) \cdot x_i = c + a_k(c \bmod' m)$$

$$-a_k\sigma + \sum_{i \neq k} (\lfloor a_i/m + \frac{1}{2} \rfloor + (a_i \bmod' m)) \cdot x_i = \lfloor c/m + \frac{1}{2} \rfloor + (c \bmod' m)$$

- Was haben wir gewonnen?
  - $x_k$  eliminiert, aber neue Variable  $\sigma$
  - Koeffizienten in neuer Gleichung E' kleiner
- Beispiel: 
$$7x + 12y + 31z = 17$$
$$3x + 5y + 14z = 7$$
- Eliminierung:
  1.  $x$  in Gleichung 1:  $x = -8\alpha - 4y - z - 1$  (neue Variable  $\alpha$ )  
neue Gleichungen:  $\{-7\alpha - 2y + 3z = 3, -24\alpha - 7y + 11z = 10\}$
  2.  $y$  in neuer Gleichung 1:  $y = \alpha + 3\beta$  (neue Variable  $\beta$ )  
neue Gleichungen:  $\{-3\alpha - 2\beta + z = 1, -31\alpha - 21\beta + 11z = 10\}$
  3.  $z$  in neuer Gleichung 1:  $z = 3\alpha + 2\beta + 1$  (keine neue Variable)  
neue Gleichungen:  $\{2\alpha + \beta = -1\}$
  4. Direktes Lösen der letzten Gleichung:  $\{\beta = -2\alpha - 1\}$
- Rücksubstitution liefert Lösung: ( $\alpha$  bel. gewählt, z.B.  $\alpha = 0$ ):
  - $\beta = -1, z = -1, y = -3, x = 12y$



- **Erster Schritt:** Eliminiere Gleichungen (ersetze dabei Variablen auch in Ungleichungen)
  - Reines System von Ungleichungen
- Verwende **Omega-Test** (Variante von Fourier-Motzkin-Elimination für ganze Zahlen)
  - Fourier-Motzkin:
    - entdeckt 1826 durch Fourier, wiederentdeckt 1936 durch Motzkin
    - löst Ungleichungen über rationalen Zahlen
  - Grundlegende Idee: selektiere Variable und eliminiere diese; wiederhole dies, bis nur noch eine Variable verbleibt

- Basis-Idee:
  - Wähle zu eliminierende Variable aus:  $x_k$
  - Schreibe Ungleichungen so um, dass sie obere und untere Schranken für  $x_k$  ausdrücken:

$$\sum_{i=1}^n a_i x_i \leq c \quad \rightarrow \quad a_k x_k \leq c - \sum_{i \neq k} a_i x_i$$

- Falls  $a_k > 0$ : obere Schranke, ansonsten untere Schranke
- Eliminiere  $x_k$
- **Beispiele:** Angenommen, wir wollen  $x$  eliminieren. Handelt es sich um obere oder untere Schranken?

$$x - y \leq 0$$

$$x - z \leq 0$$

$$-x + y + 2z \leq 0$$

$$-z \leq -1$$

- Elimination von  $x_k$ :
  - Wähle sämtliche Gleichungen aus, die obere/untere Schranken für  $x_k$  liefern:

$$\beta_l := c - \sum_{i \neq k}^n a_i x_i \leq a_k x_k \qquad a'_k x_k \leq c' - \sum_{i \neq k}^n a'_i x_i := \beta_u$$

(für  $a_k > 0$ )

- Für jedes Paar von oberer/unterer Schranke haben wir:

$$\frac{\beta_l}{a_k} \leq x_k \leq \frac{\beta_u}{a'_k}$$

- Dieses Ungleichungspaar besitzt genau dann eine rationale Lösung, falls

$$\frac{\beta_l}{a_k} \leq \frac{\beta_u}{a'_k}$$

- Füge all diese Ungleichung hinzu

# Fourier-Motzkin: Beispiel

- Gegeben:

- (1)  $x - y \leq 0$       obere Schranke für  $x$
- (2)  $x - z \leq 0$       obere Schranke für  $x$
- (3)  $-x + y + 2z \leq 0$       untere Schranke für  $x$
- (4)  $-z \leq -1$

- Eliminiere  $x$ :

- Aus Paar (1)-(3):  $y + 2z \leq x \leq y \quad \rightarrow \quad y + 2z \leq y \quad \rightarrow \quad 2z \leq 0 \quad (5)$
- Aus Paar (2)-(3):  $y + 2z \leq x \leq z \quad \rightarrow \quad y + 2z \leq z \quad \rightarrow \quad y + z \leq 0 \quad (6)$
- (1), (2), (3) können nun gestrichen werden

- Eliminiere  $z$ :

- (4) untere Schranke, (5), (6) obere Schranken
- Aus Paar (4)-(5):  $1 \leq z \leq 0 \quad \rightarrow \quad 1 \leq 0$

- **Widerspruch!**

# Von Fourier-Motzkin zum Omega-Test

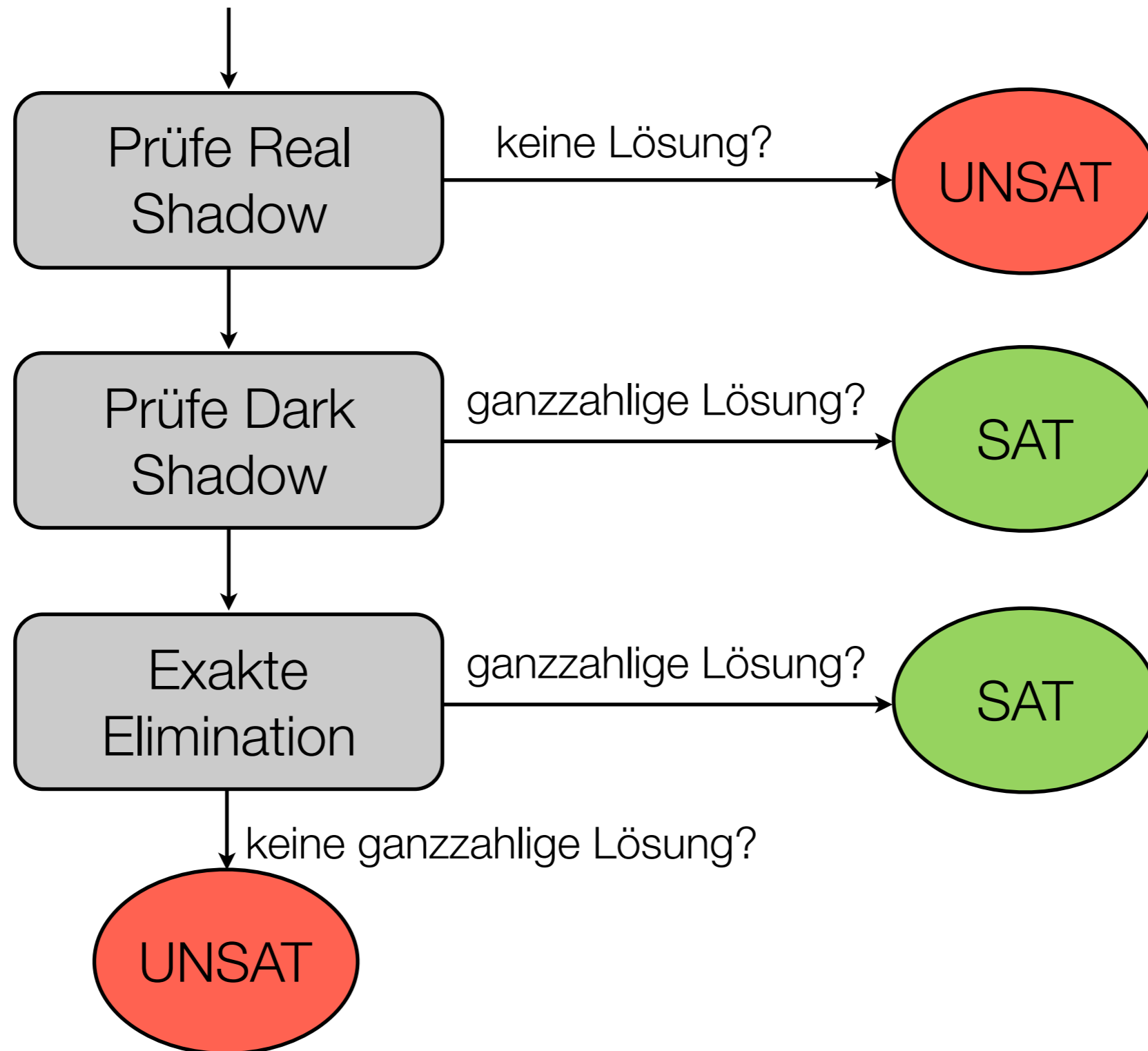
- Fourier-Motzkin: rationale Lösungen
- Omega-Test: ganzzahlige Lösungen
- Idee des Omega-Tests:
  - Berechne zuerst Lösungen für die „rationale / reelle Relaxation“ des ganzzahligen Problems („**real shadow**“):

$$\beta_l \leq a_k x_k \quad a'_k x_k \leq \beta_u \quad \rightarrow \quad \beta_l a'_k \leq a_k a'_k x_k \leq \beta_u a_k \quad \rightarrow \quad \beta_l a'_k \leq \beta_u a_k$$

- Wenn  $\beta_l a'_k \leq \beta_u a_k$  keine Lösung besitzt, wissen wir, dass das Problem unerfüllbar ist
- Wenn  $\beta_l a'_k \leq \beta_u a_k$  eine ganzzahlige Lösung besitzt, so kann es trotzdem sein, dass das Ursprungsproblem keine Lösung hat. (Warum?)

- Es kann sein, dass es eine ganzzahlige Lösung für  $\beta_l a'_k \leq \beta_u a_k$  gibt, nicht aber für  $\beta_l a'_k \leq a_k a'_k x_k \leq \beta_u a_k$ .
  - Kein Vielfaches von  $a_k a'_k$  ist zwischen  $\beta_l a_k$  und  $\beta_u a_k$ .
- Prüfe nun die folgende Ungleichung („dark shadow“):
$$\beta_u a_k - \beta_l a'_k \geq (a_k - 1)(a'_k - 1)$$
- Falls diese eine Lösung besitzt, so wissen wir, dass auch das Ursprungsproblem eine Lösung hat
- Falls nicht, muss eine exakte Elimination durchgeführt werden
  - Wähle den größten Koeffizienten  $a'_k$  für  $x_k$  in einer oberen Schranke
  - Prüfe für alle  $0 \leq i \leq (a_k a'_k - a_k - a'_k)/a'_k$ , ob eine Lösung existiert mit der zusätzlichen Bedingung  $a_k x_k = \beta_l + i$ .

# Übersicht Omega-Test



- Betrachte das folgende System  $P$  von Ungleichungen in LIA:

$$\begin{aligned} 3 &\leq 11x + 13y \leq 21 \\ -8 &\leq 7x - 9y \leq 6 \end{aligned}$$

- Keine Gleichungen, daher Start der Fourier-Motzkin-Elimination.
- Wir wollen zuerst  $x$  eliminieren:

$$\begin{aligned} 3 - 13y &\leq 11x \leq 21 - 13y && \textcircled{1} \\ -8 + 9y &\leq 7x \leq 6 + 9y && \textcircled{2} \end{aligned}$$

- „**Real shadow**“:  $\beta_l a'_k \leq \beta_u a_k$  für Ungleichungspaar  $\beta_l a'_k \leq a_k a'_k x_k \leq \beta_u a_k$ 
  - Paar 1 ( $\textcircled{1}$ - $\textcircled{1}$ ):  $\beta_l = 3-13y$ ,  $\beta_u = 21-13y$ ,  $a_k = a'_k = 11$ :
    - $(3-13y) \cdot 11 \leq (21-13y) \cdot 11$
    - $33 \leq 231$  ✓
  - Paar 2: ( $\textcircled{1}$ - $\textcircled{2}$ ):  $\beta_l = 3-13y$ ,  $\beta_u = 6+9y$ ,  $a_k = 11$ ,  $a'_k = 7$ :
    - $(3-13y) \cdot 7 \leq (6+9y) \cdot 11$
    - $21-91y \leq 66+99y \rightarrow 0 \leq 45+190y$



- „**Real shadow**“:  $\beta_l a'_k \leq \beta_u a_k$  für Ungleichungspaar  $\beta_l a'_k \leq a_k a'_k x_k \leq \beta_u a_k$ 
  - Paar 3:  $98 \geq 0$  ✓
  - Paar 4:  $235 \geq 190y$
- Insgesamt: Neues System nach Eliminierung von x:  
 $-45 \leq 190y \leq 235$
- Besitzt dieses ganzzahlige Lösungen: **ja!** ( $y = 0$  oder  $y = 1$ )
  - Prüfung des **real shadow** liefert also kein verwertbares Ergebnis!
  - Jetzt Prüfung des **dark shadow**!
- „**Dark shadow**“ wird geprüft anhand der Gleichung  $\beta_u a_k - \beta_l a'_k \geq (a_k - 1)(a'_k - 1)$ 
  - Paar 1 (①-①):  $\beta_l = 3-13y$ ,  $\beta_u = 21-13y$ ,  $a_k = a'_k = 11$ :
    - $(21-13y) \cdot 11 - (3-13y) \cdot 11 \geq 100$
    - $231-143y-33+143y \geq 100$  →  $198 \geq 100$  ✓

- Generierung der 3 weiteren Paare liefert für den **dark shadow**:

$$15 \leq 190y \leq 175$$

- Dieses System besitzt **keine** ganzzahlige Lösung!
- **Also**: Exakte Elimination erforderlich!

- Wir müssen für jede untere Schranke prüfen, ob  $P \cup \{ a_k x_k = \beta_l + i \}$  eine ganzzahlige Lösung besitzt für  $0 \leq i \leq (a_k a'_k - a_k - a'_k) / a'_k$  mit  $a'_k = 11$ .

1.  $\beta_l = 3 - 13y$ ,  $a_k = 11$ : zusätzliche Gleichung:  $P_i = \{ 11x = 3 - 13y + i \}$  mit  $0 \leq i \leq (11 \cdot 11 - 22) / 11 = 9$

Für kein  $i$  mit  $0 \leq i \leq 9$  besitzt  $P \cup P_i$  eine ganzzahlige Lösung.

2.  $\beta_l = -8 + 9y$ ,  $a_k = 7$ : zusätzliche Gleichung:  $P_i = \{ 7x = -8 + 9y + i \}$  mit  $0 \leq i \leq \lfloor (7 \cdot 11 - 7 - 11) / 11 \rfloor = 5$

Für kein  $i$  mit  $0 \leq i \leq 5$  besitzt  $P \cup P_i$  eine ganzzahlige Lösung.

→ Das Ungleichungssystem  $P$  besitzt keine ganzzahlige Lösung!

- Parallelisieren von Schleifen
  - Gibt es Datenabhängigkeiten zwischen verschiedenen Iterationen?

• Z.B.:

```
1. for (i=11; i <= 20; i++)  
    a[i] = a[i] + 3;
```

**parallel**

```
2. for (i=11; i <= 20; i++)  
    a[i] = a[i-1] + 3;
```

**nicht parallel**

```
3. for (i=11; i <= 20; i++)  
    a[i] = a[i-10] + 3;
```

**parallel?**

- Eine Datenabhängigkeit besteht, wenn es verschiedene Iterationen  $i_1$ ,  $i_2$  und einen Array-Index  $j$  gibt mit:
  - in  $i_1$  findet ein schreibender Zugriff auf  $a[j]$  statt, in  $i_2$  ein lesender oder
  - in  $i_1$  und  $i_2$  findet ein schreibender Zugriff auf  $a[j]$  statt

- Beispiel:

```
1. for (i=11; i <= 20; i++)
```

```
    a[i] = a[i] + 3;
```

```
2. for (i=11; i <= 20; i++)
```

```
    a[i] = a[i-1] + 3;
```

```
3. for (i=11; i <= 20; i++)
```

```
    a[i] = a[i-10] + 3;
```

- Beispiel 1:  $11 \leq i_1, i_2 \leq 20, i_1 \neq i_2, j = i_1 = i_2$  **unerfüllbar, d.h. parallel**
- Beispiel 2:  $11 \leq i_1, i_2 \leq 20, i_1 \neq i_2, j = i_1, j=i_2-1$  **erfüllbar, d.h. nicht parallel**
- Beispiel 3:  $11 \leq i_1, i_2 \leq 20, i_1 \neq i_2, j = i_1, j=i_2-10$  **unerfüllbar, d.h. parallel**