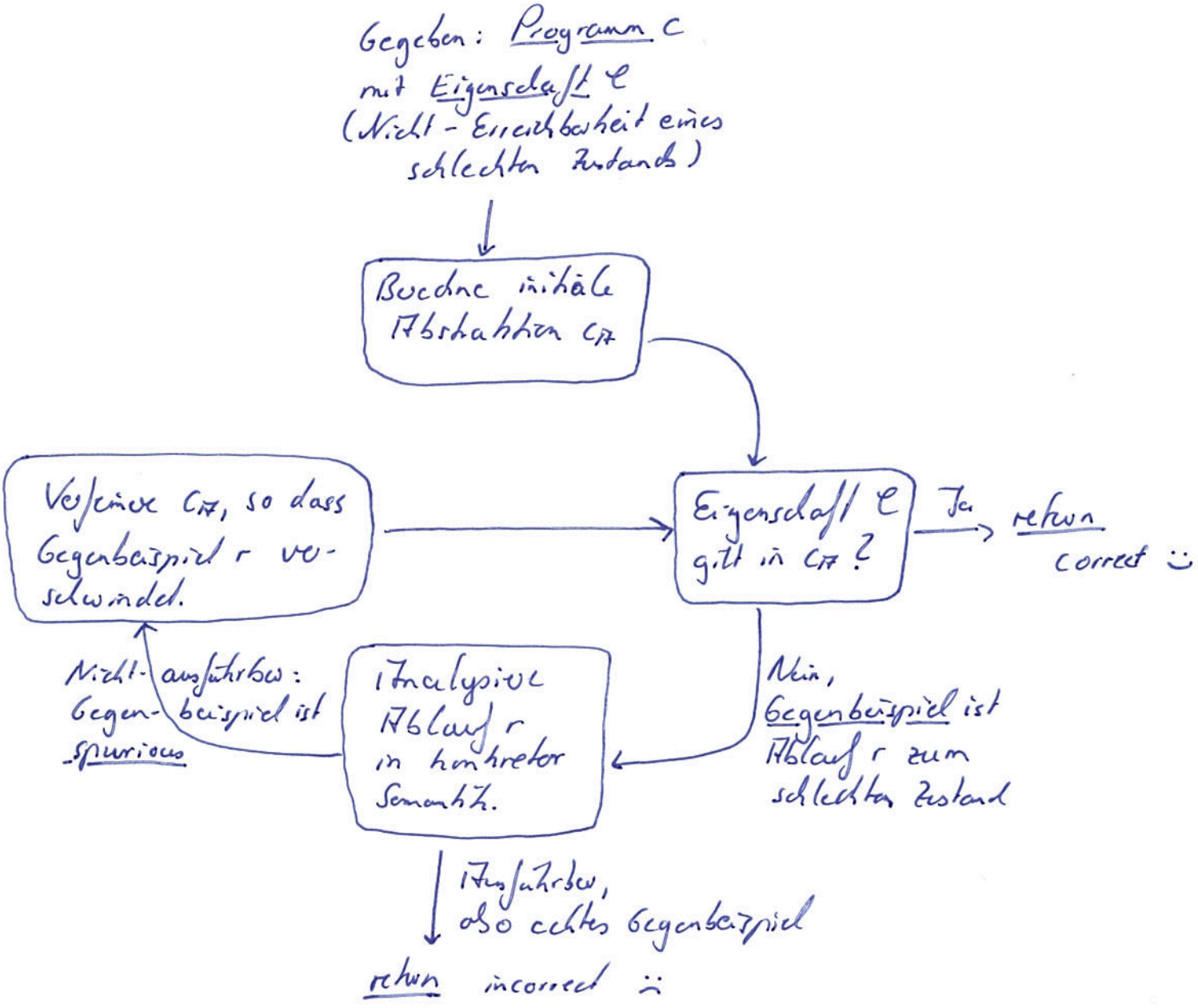
5. Prädikatenabstraktion und Abstraktionsverfeinerung

Problem: Programmeigenschaft lässt sich mit aktueller Abstraktion nicht zeigen, da
- ↳ Programm die Eigenschaft wirklich verletzt oder
- ↳ Abstraktion zu groß ist.

Ziel: Entwicklung eines abstraktionsbasierten Analyseverfahrens, das
- ↳ die Verletzung der Eigenschaft aufzeigt oder
- ↳ die Abstraktion selbstständig verfeinert.

CEGAR (counterexample-guided abstraction refinement)



Problem: Wie verfeinert man die Abstraktion automatisch?

Idee: • Prädikatenabstraktion nutzt Prädikate $p_1, \dots, p_n \in FO$, um Zustände $\sigma : Vars \to \mathbb{Z}$ auf Bitvektoren $(0/1, \dots, 0/1)$ zu abstrahieren (p_1 … p_n) (direktes Produkt)

• Verfeinerung geschieht durch Hinzufügen von Prädikaten

• Wahl der Prädikate: ↳ Initial aus dem Programm abgeleitet
↳ Dann aus den Gegenbeispielen
↳ Keine Interaktion mit dem Nutzer notwendig.

Konzeptuell: • Die Behandlung von Daten ist ein Problem der Logik.
Die Behandlung des Kontrollflusses ist ein Problem der Automatentheorie.
⇒ Prädikatenabstraktion ist eine geeignete Schnittstelle.

• Logik als universelle Sprache:
Die bisherigen Abstraktionen lassen sich als Prädikate über geeigneten Signaturen auffassen.

5.1 Prädikatenabstraktion

Idee: • Seien Prädikate $p_1, \dots, p_n$ über Vars gegeben.
Abstrakte Zustände $\sigma : Vars \to \mathbb{Z}$
auf Boolesche Kombinationen von $p_1, \dots, p_n$.

• Beliebige Boolesche Kombinationen scharfen nicht.

• Nutze kartesische Prädikatenabstraktion auf Konjunktionen.

Definition:

• Ein Prädikat ist ein Boolescher Ausdruck $b \in BExp$
// Siehe Syntax von Programmen: b, a, c.

• Ein Zustand $\sigma : Vars \to \mathbb{Z}$ erfüllt b, $\sigma \models b$,
falls $[\![b]\!](\sigma) = true$.

• Prädikat p ist schwächer als b, $b \models p$,
falls
$$\forall \sigma \in State : \sigma \models b \text{ impliziert } \sigma \models p.$$
Man sagt auch b ist stärker als p.

- Prädikate b und ρ heißen <u>äquivalent</u>, $b \dashv\vdash \rho$, falls $b \vdash \rho$ und $\rho \vdash b$.
- Sei $P = \{p_1, \dots, p_n\} \subseteq BExp$ eine endliche Menge von Prädikaten und $\neg P := \{\neg p_1, \dots, \neg p_n\}$.
  Der <u>Prädikatenabstraktionsverband</u> ist
  $$Abs(P) := (\{\bigwedge Q \mid Q \subseteq P \cup \neg P\}, \vdash).$$
  Schreibe $true := \bigwedge \emptyset$ und $false := \bigwedge \{p_i, \neg p_i, \dots\}$.
  Elemente in $Abs(P)$ heißen <u>Cubes</u> q.

<u>Lemma</u>:
$Abs(P)$ ist ein vollständiger Verband.

<u>Beweis</u>:
- $\bot = false$, $\top = true$
- $q_1 \sqcap q_2 = q_1 \wedge q_2$
- $q_1 \sqcup q_2 = \overline{q_1 \vee q_2}$.
  Dabei ist $\overline{b}$ die <u>stärkste Formel in $Abs(P)$</u>, die aus b folgt.
  Sie ist tatsächlich eindeutig bestimmt als
  $$\overline{b} \dashv\vdash \bigwedge \{q \in Abs(P) \mid b \vdash q\}$$
  $$\dashv\vdash \bigwedge \{l \in P \cup \neg P \mid b \vdash l\}.$$
  Die zweite Äquivalenz dient als Berechnungsmethode für $\overline{\cdot}$.

<u>Beispiel</u>:
Sei $P = \{p_1, p_2, p_3\}$.
a) Für $q_1 := p_1 \wedge \neg p_2$ sowie $q_2 := \neg p_2 \wedge p_3$ gilt
$$q_1 \sqcap q_2 = p_1 \wedge \neg p_2 \wedge \neg p_2 \wedge p_3 \dashv\vdash p_1 \wedge \neg p_2 \wedge p_3$$
$$q_1 \sqcup q_2 = \overline{(p_1 \wedge \neg p_2) \vee (\neg p_2 \wedge p_3)}$$
$$\dashv\vdash \overline{\neg p_2 \wedge (p_1 \vee p_3)}$$
$$\dashv\vdash \neg p_2 \wedge \dots, \text{ wobei } \dots \text{ abhängig von den Prädikaten,}$$
zum Beispiel $p_4 = x > 4 \vdash p_1 \vee p_3 = x > 6$

(2) Für $q_1 := p_1 \wedge p_2$ sowie $q_2 := p_1 \wedge \neg p_2$ gilt

$$q_1 \sqcap q_2 = p_1 \wedge p_2 \wedge p_1 \wedge \neg p_2 \models \text{false}$$

$$q_1 \sqcup q_2 = \overline{(p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)}$$

$$\models \overline{p_1 \wedge (p_2 \vee \neg p_2)} \models \overline{p_1} \models p_1.$$

Definition:
Die Galoisverbindung zw Prädikatenabstraktion ist definiert durch

$$\alpha : \mathbb{P}(State) \to Abs(P) \quad \text{und} \quad \gamma : Abs(P) \to \mathbb{P}(State)$$

mit

$$\gamma(q) := \{\sigma \in State \mid \sigma \models q\}$$

Was ist die Abstraktionsfunktion?
Gegeben $\sigma \in State$, definiere

$$q_\sigma := \bigwedge \{l \in P \cup \neg P \mid \sigma \models l\}$$

Dann gilt für $State' \subseteq State$:

$$\alpha(State') := \bigsqcup \{q_\sigma \mid \sigma \in State'\}.$$

Beispiel:
Betrachte $P = \{p_1, p_2, p_3\}$ mit $p_1 := x \leq y$, $p_2 := x = y$, $p_3 := x > y$.
Sei $State' = \{\sigma_1, \sigma_2\}$ mit $\sigma_1 = (x=1, y=2)$, $\sigma_2 = (x=2, y=2)$.

(1) $\alpha(State') = q_{\sigma_1} \sqcup q_{\sigma_2}$

$$= (p_1 \wedge \neg p_2 \wedge \neg p_3) \sqcup (p_1 \wedge p_2 \wedge \neg p_3)$$

$$= \overline{(p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (p_1 \wedge p_2 \wedge \neg p_3)}$$

$$= p_1 \wedge \neg p_3.$$

(2) Für $q = p_1 \wedge \neg p_2$ gilt

$$\gamma(q) = \{\sigma \in State \mid \sigma(x) < \sigma(y)\}.$$