Selbststabilisierende Replikation

Felix Gärtner

Übersicht

1. Motivation und Begriffe
2. Was ist Selbststabilisierung?
3. Wie kann man selbststabilisierende Algorithmen bauen? (Methodologie)
4. Anwendung auf Replikation
5. offene Probleme

Fehlertoleranz

- Herkömmliche Fehlertoleranzuntersuchungen: Korrektheit in Bezug auf Fehlermodell (*crash*, *failstop*, etc.)
- permanente vs. vorübergehende Fehler
- maskierende vs. nichtmaskierende Fehlertoleranz

Replikation (von Diensten)

Bereitstellung mehrerer identischer Kopien eines Datenelementes an verschiedenen Stellen in einem verteilten System.

+ Steigerung der Leistung (*performance*)
+ Steigerung der Verfügbarkeit (*availability*)
– erhöhter Änderungsaufwand
– Gefahr von Inkonsistenzen

starke/strenge Konsistenz $\leftrightarrow$ schwache Konsistenz

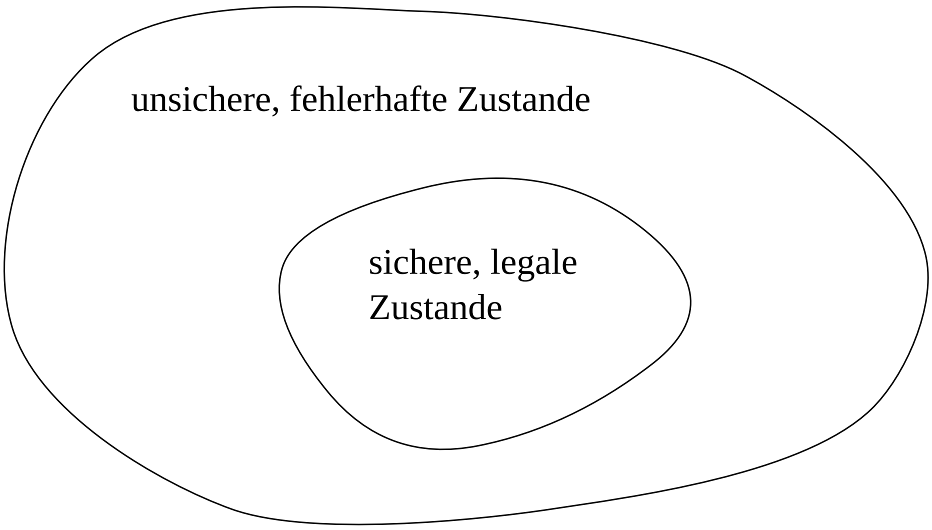
Selbststabilisierung

- Menge Z von (fehlerhaften) Systemzuständen
- Prädikat $P \subseteq Z$ beschreibt "legale" Zustände

Definition: Selbststabilisierung

Ein System stabilisiert sich selbst zu P, gdw.

1. Gilt P, so wird P bei normaler Ausführung nicht verlassen.
2. *Jede* Ausführung enthält einen Zustand aus P.



\+ toleriert beliebige vorübergehende Fehler

– nicht-maskierende Fehlertoleranz

Methodologie I: Komposition selbststabilisierender Algorithmen

herkömmliche Komposition von Algorithmen A und B:

$$\alpha \xrightarrow{A} \beta \quad \overset{\text{Terminierung}}{\|} \quad \beta \xrightarrow{B} \gamma$$

Komposition jetzt:

$$\alpha \xrightarrow{A,B} \beta \xrightarrow{B} \gamma$$

Voraussetzung: B "beeinflußt" A nicht.

Methodologie II: Automatisierte Selbststabilisierung

1. selbststab. Schnappschußalgorithmus

2. selbststab. Rücksetzmechanismus

Voraussetzung:

- entscheidbares Prädikat P

- $\neg P$ muß durch Schnappschuß erkannt werden können

Methodologie III: lokale Überprüfbarkeit

Definition: lokales Prädikat

enthält nur Bedingungen, die sich auf einen einzigen Prozeß beziehen.

Für

$$P \equiv P_1 \wedge P_2 \wedge \cdots \wedge P_n$$

mit lokalen Prädikaten P_i kann $\neg P$ einfach erkannt werden.

Definition: lokale Überprüfbarkeit

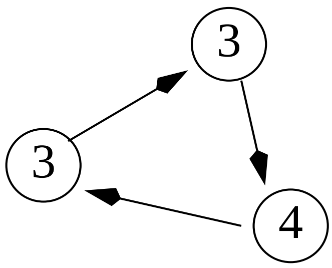
P lokal überprüfbar, gdw. es als Konjunktion von lokalen Prädikaten geschrieben werden kann.

Methodologie IV: lokale Korrigierbarkeit

Idee: korrigiere die einzelnen P_i lokal.

$\stackrel{?}{\Rightarrow}$ P gilt wieder

Gegenbeispiel



‘‘alle sollen den gleichen Wert haben’’

update in Pfeilrichtung

hinreichende Kriterien für Stabilisierung:

(1) Alle P_i sind stabil.

oder allgemeiner:

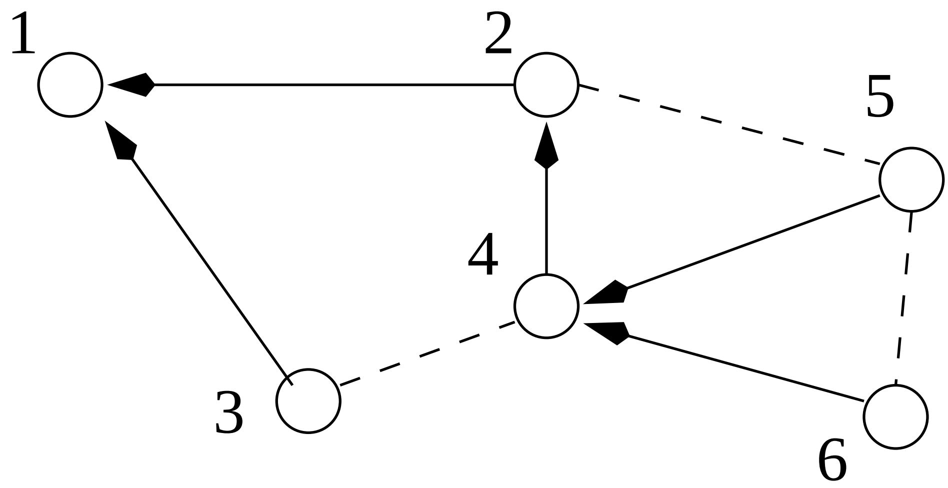
(2) Der durch die P_i definierte Abhängigkeitsgraph enthält keine Zyklen.

oder noch allgemeiner:

(3) ???

selbststabilisierende Replikationsalgorithmen

Anwendung von lokalem Prüfen und lokaler Korrektur. Replikat auf Knoten i: r_i. (Lese-) Zugriffsmöglichkeit auf Werte der Nachbarn durch gemeinsame Variable.



Legalitätsprädikat: $P \equiv \forall i, j : r_i = r_j$

- (selbststabilisierender) aufspannender Baum mit Wurzel
- Knoten überprüfen periodisch Replikatwert des Elternknotens (Aktualisierung falls ungleich)

$\Rightarrow$ stabilisiert sich zu Konsistenz

Ergebnisse

- Selbststab. Replikationsalgorithmus.

+ verträgt alle vorübergehenden Fehler

+ verträgt permanente Fehler solange verbunden

→ selbststabilisierender *broadcast*

→ Basis für weitere Verfahren

– ist nicht-maskierend (Anwendung?)

offene Fragen

- Hierarchie der Verfahren?
- Komplexität der Verfahren?
- Anwendung von Selbststabilisierung auf andere Probleme?
- Maskierende Selbststabilisierung?

Zusammenfassung

1. Begriffe: Fehlertoleranz und Replikation

2. Methodik für den Entwurf selbststabilisierender Algorithmen

3. Anwendung der Entwurfsmethodik auf Replikations
   $\Rightarrow$ selbststabilisierende Replikationsalgorithmen
   $\Rightarrow$ selbststabilisierender *broadcast*

4. Ergebnisse und offene Fragen