Fehlerbezogene Spezifikation Verteilter Algorithmen

Felix Gärtner

FACHBEREICH INFORMATIK
TECHNISCHE UNIVERSITÄT DARMSTADT

Vortragsgliederung

1. Motivation und Hintergrund
2. Zum Begriff der “fehlerbezogenen Spezifikation”
3. Beispiel: das Übereinstimmungsproblem

Motivation und Hintergrund

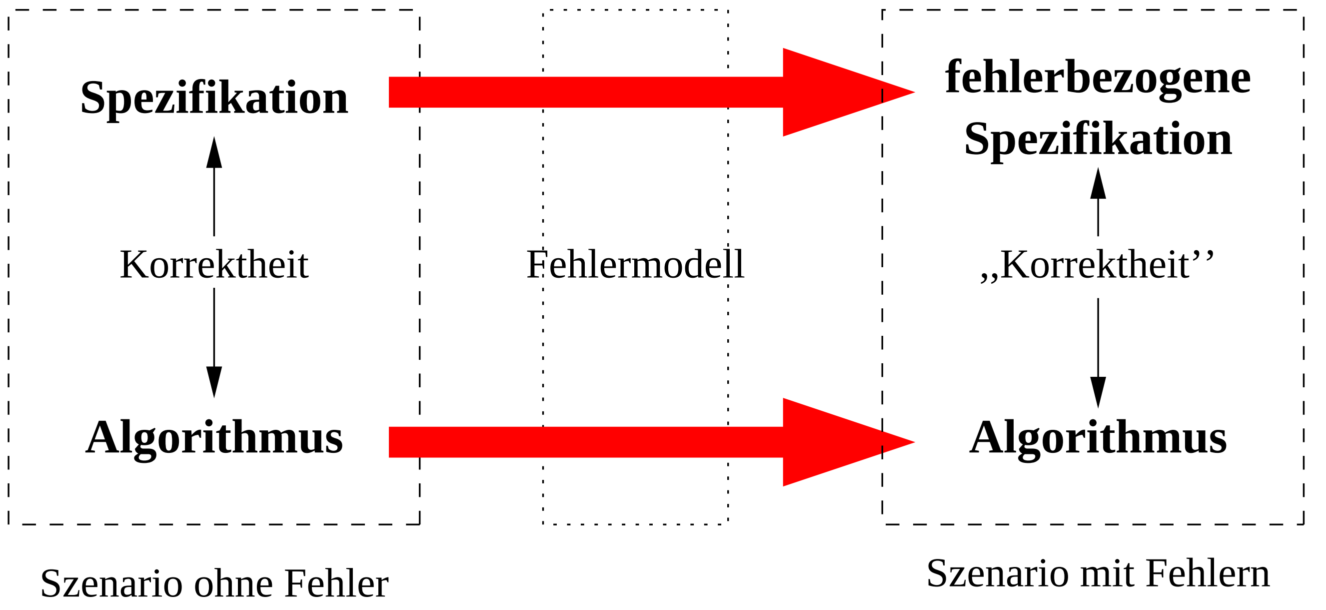
- Zunehmende Abhängigkeit von technischer Infrastruktur.

http://www.cs.virginia.edu/~survive

- Benötigen zuverlässige und sichere Systeme.
- Korrektheit alleine reicht oft nicht aus. Kritische Eigenschaft: Fehlertoleranz.

fehlerbezogene Spezifikation

- Fehlertoleranz: erfüllt Spezifikation trotz interner Fehler.
- Fehlertoleranzeigenschaften müssen verifiziert werden.
- Verifikation nur möglich mit einer Spezifikation.
- Spezifikation muß "fehlerbezogen" sein.



Beispiel: das Übereinstimmungsproblem (*consensus*)

- Netzwerk von Prozessen
- Operationen *propose*(v) und *decide*(v)
- Ein Algorithmus löst das Übereinstimmungsproblem, gdw.
    1. nach endlicher Zeit jeder Prozeß *decide*(v) ausführt,
    2. alle Prozesse denselben Wert entscheiden, und
    3. der entschiedene Wert zuvor von mindestens einem Prozeß mittels *propose*(v) vorgeschlagen wurde.

(NB: Dies ist Spezifikation für den fehlerfreien Fall)

Ein Fehlermodell: *crash*

- In jedem Prozeß gibt es einen (virtuellen) *crash*-Zustand.
- Eine bestimmte Teilmenge von Prozessen wechselt nach endlicher Zeit spontan in diesen Zustand (*crash!*).
- Falls nicht in diesem Zustand, läuft ein Prozeß “normal” weiter.
- Falls in diesem Zustand, tut der Prozeß nichts mehr.

consensus-Spezifikation für *crash*

- Prozesse, die irgendwann abstürzen, heißen "fehlerhaft".
- Nicht-fehlerhafte Prozesse heißen "korrekt".
- Verfeinerte Spezifikation für *consensus*:

  1. Nach endlicher Zeit führt jeder korrekte Prozeß *decide*(v) aus.
  2. Alle korrekten Prozesse entscheiden denselben Wert.
  3. Der entschiedene Wert ist zuvor von mindestens einem korrekten Prozeß mittels *propose*(v) vorgeschlagen worden.

Alternative *consensus*-Spezifikation für *crash*

1. Nach endlicher Zeit führt jeder korrekte Prozeß *decide*(v) aus.
2. Alle Prozesse, die *decide*(v) ausführen, tun dies mit demselben Wert v.
3. Der entschiedene Wert wurde zuvor von mindestens einem Prozeß mittels *propose*(v) vorgeschlagen.

Schlußfolgerungen

- Fehlerbezogene Spezifikationen ...
  - ... sind wichtig, aber nicht-trivial.
  - ... existieren zu einem Problem in einer gewissen Bandbreite.
  - ... hängen vom Fehlermodell ab.
  - ... drücken das Maß an Kooperation aus, welches von fehlerhaften Komponenten zu erwarten ist.

Zusammenfassung

- Fehlertoleranz als kritische Eigenschaft.
- Verifikation erfordert fehlerbezogene Spezifikation.
- Fehlerbezogene Spezifikationen am Beispiel des Übereinstimmungsproblems.
- Ziel: Zusammenhänge zwischen “normalen” Spezifikationen, Fehlermodellen und fehlerbezogenen Spezifikationen untersuchen.
- Fernziel: semiautomatische Verifikation von fehlertoleranten verteilten Algorithmen.