



# Funktionale Programmierung und Verifikation Anleitung Beweise

Funktionale Programmierung (Technische Universität München)

Weakest Precondition um Z zu beweisen:

1) Basic Prinzip:

- a)  $WP[[ \text{statement} ]]$  (vorherige Bedingung)  $\equiv$  \*statement in vorheriger Bedingung einsetzen und bestmöglich vereinfachen\*
- b) siehe: einfaches Beispiel rechts
- c) wichtige Sonderfälle auf der letzten Seite dieses Dokuments

2) Mit Loop Invariante:

- a) Weakest Precondition wie beim Basic Prinzip bis man zur Schleife kommt
- b) dann: Loop Invariant I überlegen
  - i) Mit weakest Preconditions durch die Schleife arbeiten
- c) Lokale Konsistenz der Loop Invariante beweisen:
  - i) wie?  $(\neg b \wedge B_{\text{no}}) \vee (b \wedge B_{\text{yes}})$  mit dem Vergleichsprädikat b formulieren und vereinfachen
    - (1) ist I das Ergebnis gilt lokale Konsistenz
    - (2) sonst: reingeschissen
- d) Ist lokale Konsistenz bewiesen: weiter nach oben arbeiten mit WP und hoffentlich am Ende true erhalten

- 1) Führe eine Variable  $r$  ein, die die Schleifendurchläufe runterzählt
  - a) Anfang:  $r > 0$
  - b) zu beweisen:
    - i) Am Anfang des No-Branches A gilt immer  $r > 0$
    - ii) Am Fastende des No-Branches B gilt immer  $r$  ist größer als der Wert, auf den es danach gesetzt wird
  - c) wie?
    - i) Schleifeninvariante bilden
    - ii) lokale Konsistenz beweisen
    - iii) schauen, ob die WP an den Punkten A und B unsere zu beweisenden Bedingungen implizieren
      - (1) wenn ja: beweis fertig
      - (2) wenn nein, reingeschissen oder Terminierung nicht möglich