Universität Rostock Traditio et Innovatio

Imperative Programmierung

Termumformungen

Prof. Dr.-Ing. habil. Gero Mühl

Architektur von Anwendungssystemen (AVA)
Fakultät für Informatik und Elektrotechnik (IEF)
Universität Rostock

Boolean

- nicht(und(Wahr, Falsch))
  = nicht(Falsch)
  = Wahr

- und(oder(nicht(Falsch), Falsch), Wahr)
  = und(oder(Wahr, Falsch), Wahr)
  = und(Wahr, Wahr)
  = Wahr

- oder(nicht(Wahr), nicht(und(Wahr, Falsch)))
  = oder(Falsch, nicht(Falsch))
  = oder(Falsch, Wahr)
  = Wahr

Natürliche Zahlen

- add(Suc(Zero), Suc(Zero))
  = Suc(add(Suc(Zero), Zero))
  = Suc(Suc(Zero))

- mult(Suc(Suc(Zero)), Suc(Suc(Zero)))
  = add(mult(Suc(Suc(Zero)), Suc(Zero)), Suc(Suc(Zero)))
  = add(Suc(Suc(Zero)), Suc(Suc(Zero)))
  = ...
  = Suc(Suc(Suc(Suc(Zero))))

Natürliche Zahlen und Boolean

- gleich(Suc(Suc(Zero)), Suc(Zero))
  = gleich(Suc(Zero), Zero)
  = Falsch

- ungleich(Suc(Zero), Suc(Zero))
  = nicht(gleich(Suc(Zero), Suc(Zero)))
  = nicht(gleich(Zero, Zero))
  = nicht(Wahr)
  = Falsch

Natürliche Zahlen und Boolean

> gleich(add(Suc(Zero), Suc(Zero)),
add(Suc(Suc(Zero)), Zero))

= gleich(Suc(add(Suc(Zero), Zero)),
add(Suc(Suc(Zero)), Zero))

= gleich(Suc(Suc(Zero)), add(Suc(Suc(Zero)), Zero))

= gleich(Suc(Suc(Zero)), Suc(Suc(Zero)))

= gleich(Suc(Zero), Suc(Zero))

= gleich(Zero, Zero)

= Wahr

Stack

- depth(push(1, push(5, init)))
  = 1 + depth(push(5, init))
  = 1 + 1 + depth(init)
  = 1 + 1 + 0
  = 2

- push(top(push(1, push(5, init))), init)
  = push(1, init)

Queue

- remove(insert(1, insert(2, init)))
  = insert(1, remove(insert(2, init)))
  = insert(1, init)

- front(insert(1, insert(2, init)))
  = front(insert(2, init))
  = 2

Priority Queue

Höhere Priorität

- pfront(pinsert(1, 2, pinsert(2, 3, init)))
  = pfront(pinsert(1, 2, insert(2, 3, init)))
  = pfront(insert(2, 3, insert(1, 2, init)))
  = 2

- maxprio(pinsert(1, 2, pinsert(2, 3, init)))
  = maxprio(insert(2, 3, insert(1, 2, init)))
  = 3

- premove(pinsert(2, 3, pinsert(1, 2, init)))
  = premove(insert(2, 3, insert(1, 2, init)))
  = insert(1, 2, init)

Priority Queue

> pinsert(11, 1, pinsert(9, 3, pinsert(15, 2, init)))
= pinsert(11, 1, pinsert(9, 3, insert(15, 2, init)))
= pinsert(11, 1, insert(9, 3, insert(15, 2, init)))
= insert(9, 3, pinsert(11, 1, insert(15, 2, init)))
= insert(9, 3, insert(15, 2, pinsert(11, 1, init)))
= insert(9, 3, insert(15, 2, insert(11, 1, init)))

Hohe Priorität
Mittlere Priorität
Niedrige Priorität

Tabelle

- read(6, insert(5, "Giraffe", insert(6, "Zebra", init)))
  = read(6, insert(6, "Zebra", init))
  = "Zebra"

- read(4, insert(5, "Giraffe", insert(6, "Zebra", init)))
  = read(4, insert(6, "Zebra", init))
  = read(4, init)
  = "ERROR"

Tabelle

> delete(6, insert(5, "Giraffe", insert(6, "Zebra", init)))
= insert(5, "Giraffe", delete(6, insert(6, "Zebra", init)))
= insert(5, "Giraffe", init)

> delete(4, insert(5, "Giraffe", insert(6, "Zebra", init)))
= insert(5, "Giraffe", delete(4, insert(6, "Zebra", init)))
= insert(5, "Giraffe", insert(6, "Zebra", delete(4, init)))
= insert(5, "Giraffe", insert(6, "Zebra", init))

Tabelle

- update(6, "Kuh",
  insert(5, "Giraffe", insert(6, "Zebra", init)))
  = insert(5, "Giraffe",
  update(6, "Kuh", insert(6, "Zebra", init)))
  = insert(5, "Giraffe", insert(6, "Kuh", init))

- length(insert(5, "Giraffe", insert(6, "Zebra", init)))
  = 1 + length(insert(6, "Zebra", init))
  = 1 + 1 + length(init)
  = 1 + 1 + 0
  = 2

Vielen Dank für Ihre Aufmerksamkeit!

Univ.-Prof. Dr.-Ing. Gero Mühl

gero.muehl@uni-rostock.de
https://www.ava.uni-rostock.de