

Lehrstuhl für Informatik 8 (Theoretische Informatik) Lehrveranstaltungen WiSe 2021/22

Advanced Mechanized Reasoning in Coq

- In SemProg, we formalized educationally idealized languages.
- Now it's time to enter the coqpit. There is so much to learn:
- Coinduction, typeclasses, termination metrics, binders (de Bruijn, HOAS or otherwise), separation logic, logical relations, step indexing and partial evaluation ...
- Adjustments to interests and needs of the audience are possible
- For each of you, it should be the beginning of a longer journey and an own project



Algebra des Programmierens



Solide und flexible mathematische Grundlagen für effektive Programmierung und Systemsemantik. Kategorien, Funktoren, Algebren und Koalgebren für

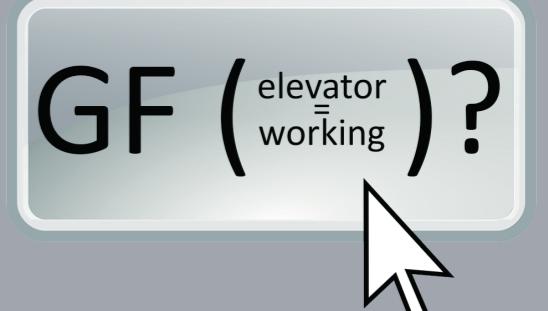
- induktive Datentypen: strukturelle Induktion und Rekursion per initiale Semantik
- Automaten & Transitionssysteme: Bisimulation und Systemverhalten per finale Semantik

Wichtig: Dualität von Prozessen und Daten!

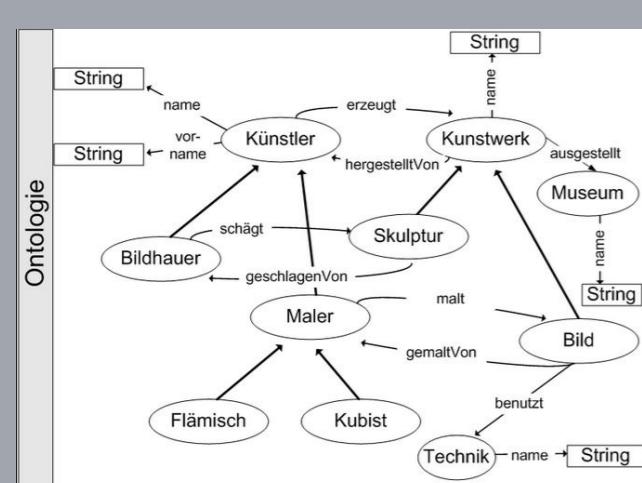
Formale Methoden der Softwareentwicklung

- Tired of manually checking your code for memory leaks?
- Looking for hands-on experience with automated verification tools?
- Want to know why our elevators keep breaking down?

} Let formal logic
be your guide!



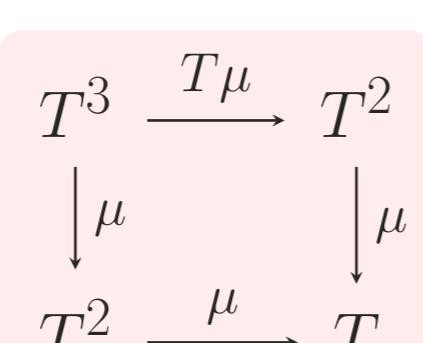
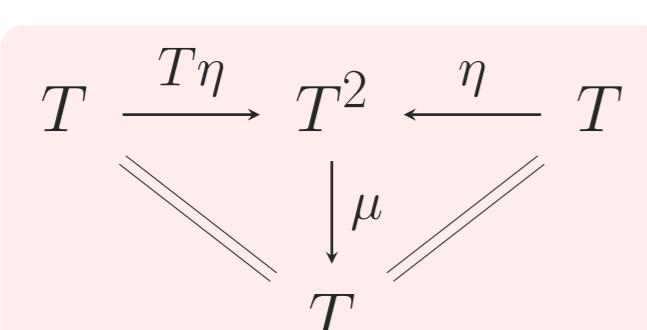
Ontologien im Semantic Web



Ontologien sind Repositorien formalisierten Wissens; sie werden z. B. zur Vernetzung von Daten und zur Entscheidungsunterstützung etwa in Medizin und Ingenieurtechnik verwendet. Der Kurs gibt einen Einblick in semantische und algorithmische Aspekte wichtiger Ontologiesprachen. Besonderes Augenmerk liegt dabei auf den sogenannten Beschreibungslogiken, einer Familie von Modallogiken, die vor allem als Grundlage der *Web Ontology Language* (OWL) des W3C bekannt sind.

Monad-Based Programming

```
class Functor m => Monad m where
  return :: a -> m a           -- η
  bind   :: m a -> (a -> m b) -> m b -- μ
  -- associative law: (fmap bind) = bind . bind
  -- unit laws: (fmap unit) = bind . unit = id
```



Computational monads are a centerpiece of modern semantics and functional programming, prominently manifested in HASKELL and influencing SCALA, ML and F#. We provide

- a systematic theoretical and practical monad-oriented account of various computational aspects of program semantics
- a uniform monad-based presentation of such diverse computational effects as store, non-determinism, exceptions, and continuations
- an introduction to advanced theoretical and practical features of HASKELL programming

