

# Compiler

## Blatt 3

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik  
Hochschule München

Letzte Änderung: 27.04.2017 09:44

Das Repository für dieses Blatt bekommen Sie unter dem Link <https://classroom.github.com/assignment-invitations/52571060047ef0ab78e4b649fe33b312>.

Der Code ist mit

```
stack exec ea
```

ausführbar, akzeptiert aber noch nichts. Mit

```
stack test
```

können Sie die Spezifikation ausführen.

### Aufgabe 1 — Endlicher Automat für Integer

Entwickeln Sie einen endlichen Automat der `Integer` akzeptiert. Die (ausführbare) Spezifikation finden Sie im Modul `EAIntegerSpec`.

- Geben Sie das Zustandsübergangsdiagramm an.
- Geben Sie den Endlichen Automat als mathematisches Objekt an.
- Implementieren Sie den endlichen Automat in Haskell im Modul `EAInteger`.

## Aufgabe 2 — Endlicher Automat für Float

Entwickeln Sie einen endlichen Automat der Floats akzeptiert. Die (ausführbare) Spezifikation finden Sie im Modul `EFloatSpec`. Die Floats dürfen beliebig lang und beliebig genau sein.

- a) Geben Sie das Zustandsübergangsdiagramm an.
- b) Geben Sie den Endlichen Automat als mathematisches Objekt an.
- c) Implementieren Sie den endlichen Automat in Haskell im Modul `EFloat`.

Zum Ausprobieren müssen Sie statt der `accept`-Funktion aus dem Modul `EAInteger` die aus `EFloat` in der `main`-Funktion nutzen.

## Aufgabe 3 — Scannen inkl. Klassifizierung

Ändern Sie die `main`-Funktion so ab, dass bei einem akzeptierten Integer bzw. Float jeweils das Tupel mit der Klassifizierung und dem Wert ausgegeben wird:

```
Bitte Wort eingeben: 123
===> (Integer,123)
Bitte Wort eingeben: hallo
===> rejected
Bitte Wort eingeben: 1.3245e10
===> (Float,1.3245e10)
```