Z-Spezifikationen mit Jaza animieren

Jaza[1] ist ein Z-Animator, der es erlaubt, Z-Ausdrücke interaktiv auszuwerten und auch, sich schrittweise durch einen Z-Zustandsraum zu bewegen. Jaza ist ein in Haskell programmiertes, kommandozeilen-orientiertes Programmm.

Jaza unter Windows oder Linux

Auf Moodle stehen Jaza–Binaries für Windows und für 64-Bit-Linux zur Verfügung. Bitte das entsprechende Archiv entpacken. Jaza wird über das darin befindliche Executable von der Kommando-Zeile aus gestartet: `C:>jaza` bzw. `$./jaza`.

Jaza unter Mac OS

Um Jaza unter Mac OS laufen zu lassen, muss auf den Haskell-98-Interpretierer *Hugs* zurückgegriffen werden. Dieser lässt sich wie unter

`https://georgegarside.com/blog/macos/installing-hugs-98-haskell`

beschrieben unter Mac OS installieren (einfach via `homebrew`).

Bitte dann den Jaza-Quellcode `jaza_1_1_source.tgz` vom Moodle laden und Jaza mit `runhugs TextUI` starten.

Mit Jaza arbeiten

Nach dem Start meldet sich Jaza mit dem `JAZA>`-Prompt. Nun können Kommandos eingegeben werden (`help` gibt eine Übersicht über die möglichen Kommandos).

Z-Spezifikationen werden in Jaza in der LaTeX-Notation eingegeben/eingelesen. Die Spezifikation:

FactX

$$x, top : \mathbb{N}$$
$$factors : \mathbb{P}\,\mathbb{N}$$

$$factors = \{f : 0 \mathinner{..} x \mid (\exists\, g : 0 \mathinner{..} x \bullet f * g = x)\}$$
$$x < top$$

schreibt sich in dieser Notation so:

```
\begin{schema}{FactX}
  x, top : \nat\\
  factors : \power \nat\\
\where
  factors=\{f: 0 \upto x | (\exists g: 0 \upto x @ f*g = x) \}\\
  x<top
\end{schema}
```

[1] `https://web.archive.org/web/20170411233059/http://www.cs.waikato.ac.nz/~marku/jaza/`

1. Bitte laden Sie Jaza vom Moodle–Server. In den Archiven befindet sich auch das Jaza-Handbuch `userman/master.pdf`. Ziehen sie es ggf. als Referenz zu Rate. Starten Sie bitte Jaza und berechnen Sie mit `eval 3+4` Ihren ersten Z-Ausdruck.

2. Die obige Spezifikation findet sich auch auf dem Moodle-Server als `factx.zed`. Laden Sie bitte diese Spezifikation in Jaza mit Hilfe des `load factx.zed`-Kommandos.

3. Nun kann nach Belegungen gesucht werden, in denen das Schema wahr wird:

   ```
   do [ FactX | x=12 ]
   ```

   Allerdings muss der Suchraum, den Jaza durchsucht, endlich (und klein) sein. (*top* ist aber eine beliebige natürliche, bisher unbeschränkte Zahl).

   Testen Sie bitte `FactX` weiter mit den folgenden Ausdrücken:

`do [ FactX | x=12 \land top=1000 ]`	feste Werte für `x` und `top`.
`do [ FactX | x<10 \land top=10 ]`	Einschränkung für `x` und fester Wert für `top`.
`do FactX[ top:=10 ]`	Variable `top` textuell durch Wert ersetzt.

   `=` bezeichnet die Forderung nach Gleichheit, `:=` bezeichnet die Substitution einer Variablen durch einen Wert.

   Gibt es mehrer Lösungen, so kann man sich mit `next` und `prev` durch den Suchraum bewegen. `curr` zeigt die aktuelle Lösung an. Probieren Sie bitte auch die Kommandos `show` (z.B. `show FactX`) und `why`.

`PhoneBook` animieren

In Jaza ist es möglich, Spezifikationen schrittweise auszuführen:

```
do init' ; op1 ; op2 ; op3 ...
```

Auf dem Moodle-Server findet sich eine leicht an Jaza angepasste Version des einfachen Telefonbuchs aus der Vorlesung und der letzten Übung.

4. Laden Sie diese Version (`load phonebook.zed`).
   Mit `do InitPhoneBook'` wird der Anfangszustand (als aktuelle Lösung) etabliert.

5. Wenden Sie nun die Operation `addName` mit `; addName` mehrfach an. Namen geben Sie bitte als in " (Anführungszeichen) eingefasste Strings ein. Zahlen (Telefonnummern) werden direkt, ohne Anführungszeichen, eingegeben. Nehmen Sie bitte "Tick", "Trick" und "Track"mit den Telefonnummern 11, 22 bzw. 33 auf.

6. Wir sehen uns zunächst den guten Fall der Suche an: Suchen Sie nun bitte mit `; lookup` nach "Tick" dann nach dem nichtvorhandenen "Donald". Wie ist der aktuelle Zustand? Mit `undo` lässt er sich zurücknehmen.

7. Suchen Sie nun bitte mit `; DoFindOp` — der robusten Such-Operation — nach "Tick" dann nach dem nichtvorhandenen "Donald". Welchen Wert bekommt jeweils die Variable `rep!`? Welchen Wert hat `phone!`? Insbesondere wenn nach "Donald" gesucht wurde. (Blättern sie wiederum mit `next` und `prev` durch die möglichen Nachfolgezustände)