

Wir wollen die in der Vorlesung theoretisch erarbeiteten Fragestellungen praktisch ausprobieren. Dafür verwenden wir das Text-basierte algebraische System BOBJ, eine in Java programmierte Variante der OBJ-Familie mit leicht anderer Syntax.

## Algebraische Spezifikationen mit BOBJ (erste Schritte)

### Vorbereitung

#### 1. BOBJ vorbereiten

##### Github Codespace verwenden:

Im Repository <https://github.com/fh-wedel/FSV-2024> ist ein GitHub Codespace konfiguriert, den Sie, wie in der Vorlesung vorgestellt, verwenden können, um die Übung zu bearbeiten. Forken Sie dazu das Repository und starten Sie in Ihrem Fork den Codespace.

##### Alternativ können Sie BOBJ auch manuell installieren:

BOBJ ist vollständig in Java implementiert und läuft daher unter Linux und Windows.

- (a) Laden Sie bitte die BOBJ 0.9-Version `bobj.zip` vom Moodle-Server.  
Die BOBJ-Webseite finden Sie unter  
`http://cseweb.ucsd.edu/groups/tatami/bobj.`
- (b) Extrahieren Sie bitte das Archiv `bobj.zip` an beliebigem Ort und setzen Sie bitte die Environment-Variable `BOBJ_HOME` auf diesen Pfad (Linux: `export BOBJ_HOME=.../bobj0.9`).
- (c) Starten sie BOBJ mit Hilfe des Shell-Scripts `bobj0.9/bobj.sh` (bzw. Windows: `bobj.bat`)<sup>1</sup>

#### 2. Counter-Spezifikation

Wir wollen zunächst die Counter-Signatur in OBJ umsetzen:

- (a) Laden Sie die Spezifikation `counter.bob` vom Moodle-Server. Laden Sie sie mit `in counter` in das laufende BOBJ-System.
- (b) Mit `parse <term>` . kann ein Term auf syntaktische Gültigkeit bzgl. der Signatur überprüft werden. Prüfen sie die Terme `reset`, `inc(reset)`, `inc(dec(reset))`, `dec`, `reset(inc)`.
- (c) Ergänzen Sie die Counter-Spezifikation bitte um Gleichungen, die ganzzahlige Counter spezifizieren.
- (d) Mit `reduce` (kurz `red`) kann ein Term gemäß der durch die Gleichungen bestimmten Ableitungsregeln reduziert werden. Reduzieren Sie bitte die Terme `reset`, `dec(inc(reset))`, `dec(dec(inc(reset)))`, `inc(dec(reset))`.
- (e) Wie muss man die Spezifikation ergänzen, damit sie der Intuition entspricht, dass sich `dec` und `inc` vollständig aufheben.
- (f) Wie würde eine Counter-Spezifikation aussehen, die die in der Vorlesung auch diskutierten Ringzähler spezifiziert? Legen Sie eine eigene Spezifikation an und überprüfen sie diese (`red`-Aufrufe dürfen auch in den Quellfiles stehen).

---

<sup>1</sup>Das GNU-Programm `rlwrap` stellt Kommandozeilen-Historie und Editiermöglichkeiten für BOBJ zur Verfügung: `rlwrap ./bobj.sh`

### 3. Stack-Spezifikation

- (a) Bitte spezifizieren Sie den in der Vorlesung diskutierten Stack natürlicher Zahlen mit Fehlerwerten `topless` und `underflow` beim Zugriff auf leere Stacks.
- (b) Welchen Wert hat `top(push(3, pop(pop(push(2, empty)))))`, welchen `top(push(3, empty))`?
- (c) Reduzieren Sie bitte weitere Ausdrücke Ihrer Wahl.

### 4. Spezifikationen eines Lagers

Es sollen die Grundzüge eines IT-Systems zur Lagerverwaltung algebraisch spezifiziert werden.

*Ladungsträger* (etwa Euro-Paletten oder Gitterboxen) mit einer eindeutigen Kennung und einer Beschreibung der auf ihnen verfügbaren Güter sollen in einem Lager auf *Fächern* (Regalplätzen) eingelagert werden.

Fächer sind im Lager eindeutig durch ihren Ort bestimmt, der durch Gangnummer, Regalnummer innerhalb des Ganges und Ebene innerhalb des Regals festgelegt wird. So bezeichnet beispielsweise der Ort 17/03/02 das Fach in Gang 17, Regal 3 und Ebene 2.

Bei dem eingesetzten Regalsystem handelt es sich um ein *Einplatzregal*, das heißt, jedes Fach wird mit höchstens einem Ladungsträger belegt. Dadurch ist jeder Ladungsträger direkt verfügbar und das Lager kann in FiFo-Art (First in First out) betrieben werden.



Zur Illustration: Einplatzregal (Quelle: Jungheinrich)

Im zu spezifizierenden Teil des Systems soll es möglich sein, das Ein- und Auslagern der Ladungsträger zu verwalten<sup>2</sup>. Ein Fach ist entweder von einem Ladungsträger belegt oder es ist leer. Ein verwalteter Ladungsträger befindet sich immer in einem Fach.

- (a) Spezifizieren Sie bitte in BOBJ geeignete Sorten, die die oben angesprochenen Aspekte des Systems darstellen können.
- (b) Spezifizieren Sie bitte in BOBJ die Signatur einer Operationen (Konstante) `leeresLager`, die ein Lager repräsentiert, in dem keine Ladungsträger eingelagert sind, also alle Fächer leer sind.
- (c) Spezifizieren Sie bitte in BOBJ die Signaturen der Operationen `einlagern` und `auslagern`, bei der ein Ladungsträger mit gegebener Kennung und Beschreibung in einem gegebenen Fach (Gang/Regal/Ebene) eingelagert bzw. mit gegebener Kennung ausgelagert wird.
- (d) Bitte beschreiben Sie in BOBJ die Aussage, dass das Einlagern eines Ladungsträgers mit der Kennung  $k$  unmittelbar gefolgt vom Auslagern des selben Ladungsträgers  $k$  das Lager unverändert lässt.

---

<sup>2</sup>Zusätzlich muss das System auch die Topologie des Lagers verwalten können (Stammdatenverwaltung), was aber hier nicht Gegenstand der Betrachtung sein soll.