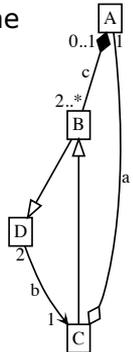
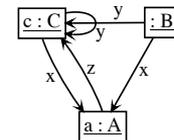


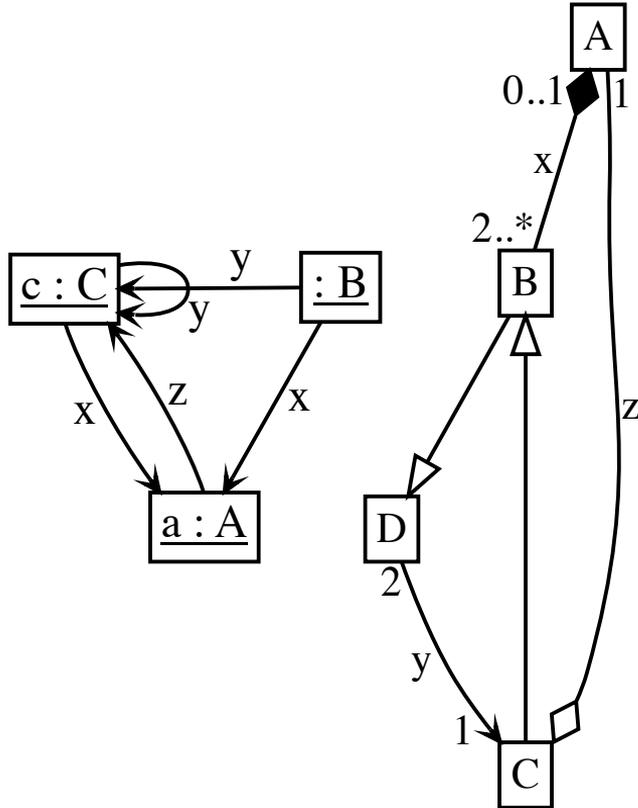
## Generating Diverse Exercise Tasks on UML Class and Object Diagrams, Using Formalisations in Alloy

Marcellus Siegburg und Janis Voigtländer

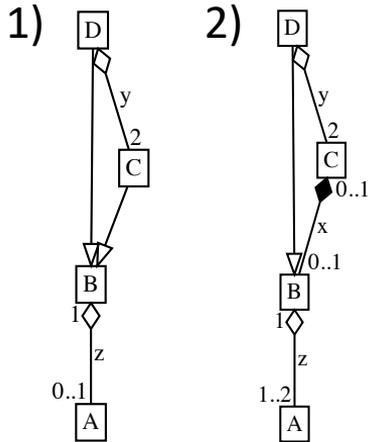
19.02.2020

Which of the links in the object diagram (OD) corresponds to which relationship in the class diagram (CD)?



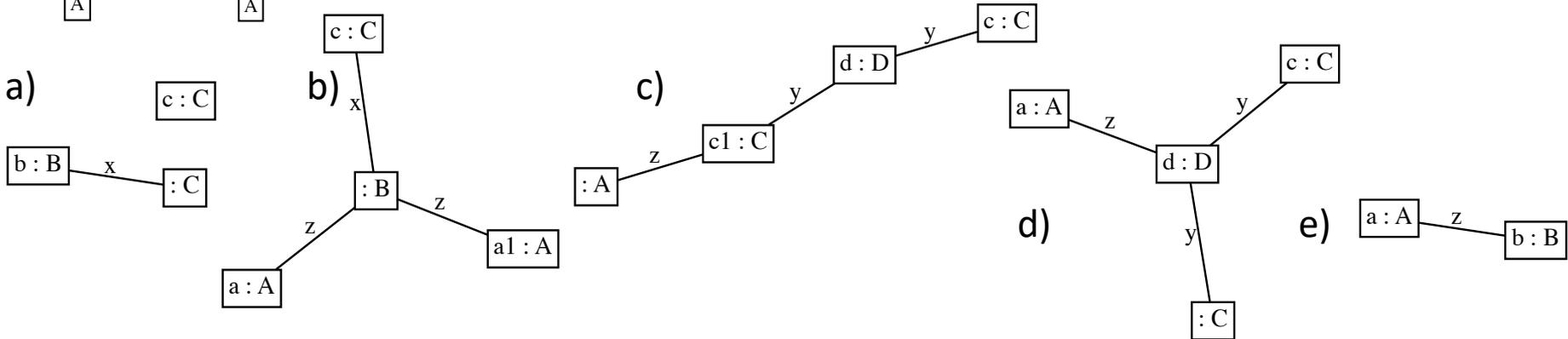


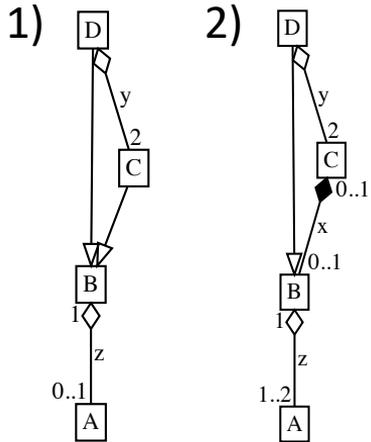
- **Objektdiagramm (OD)**
  - ▶ **Kanten (Links)**
    - » incl. Name, Navigationsrichtung
- **Klassendiagramm (CD)**
  - ▶ **Kanten (Beziehungen):**
    - Assoziation / Aggregation / Komposition / Vererbung**
      - » incl. Name, Multiplizitäten



- Ordnen Sie den CDs ihre validen OD-Instanzen zu!

- Generierung mittels CD2Alloy nach [KSV19]

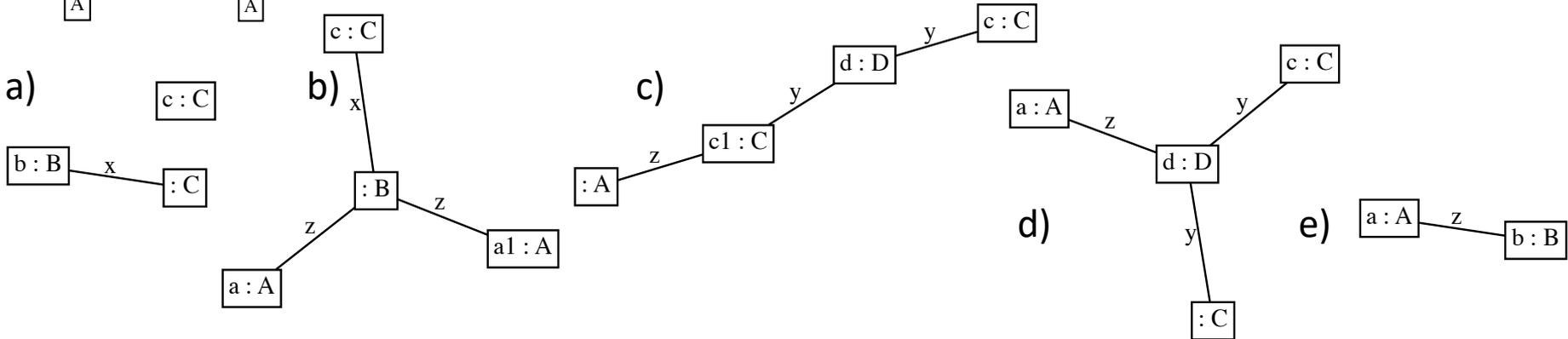




- Ordnen Sie den CDs ihre validen OD-Instanzen zu!

1: c, d, e    2: b, d, e

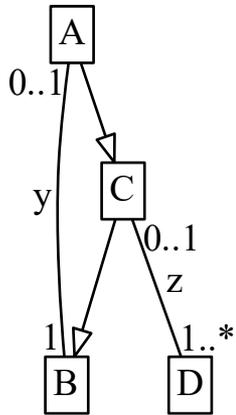
- Generierung mittels CD2Alloy nach [KSV19]



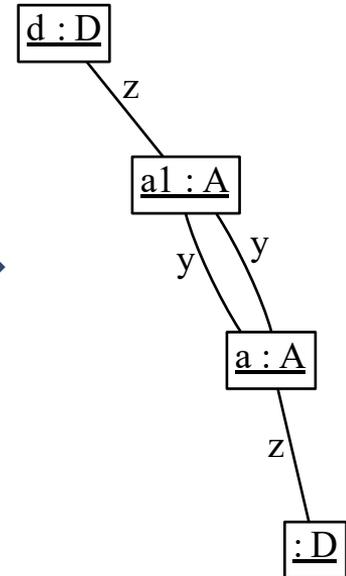
- Alloy: deklarative Sprache, Beschreibung mittels Relationen, verwendet SAT-Solver

⇒ Übersetzung von CDs in Alloy-Spezifikationen

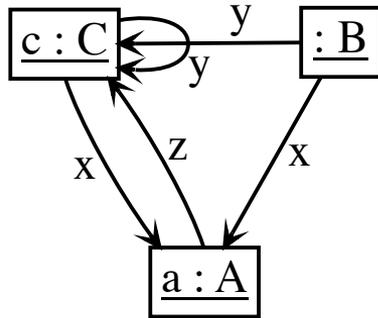
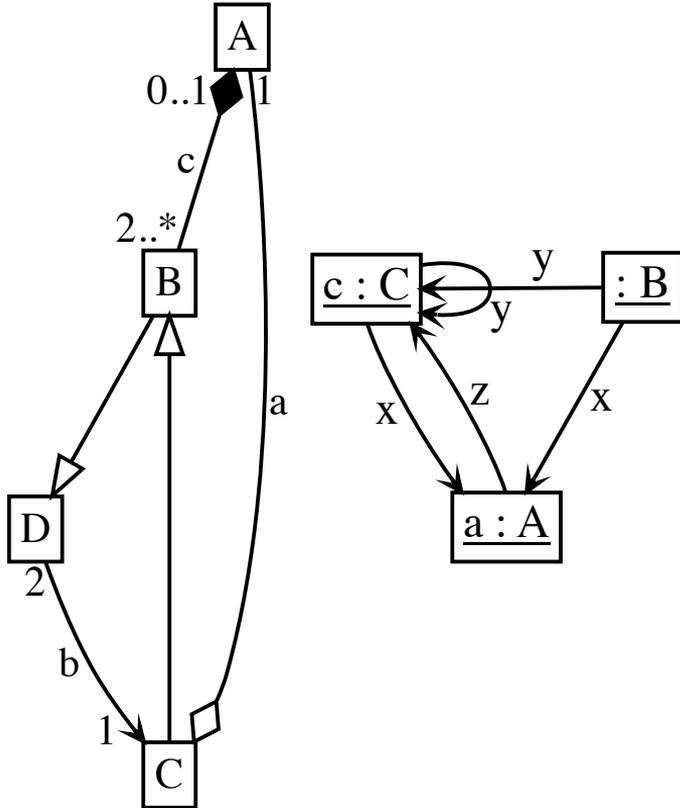
➔ Interpretation der gefundenen OD-Instanzen



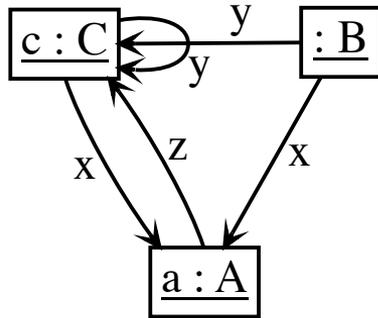
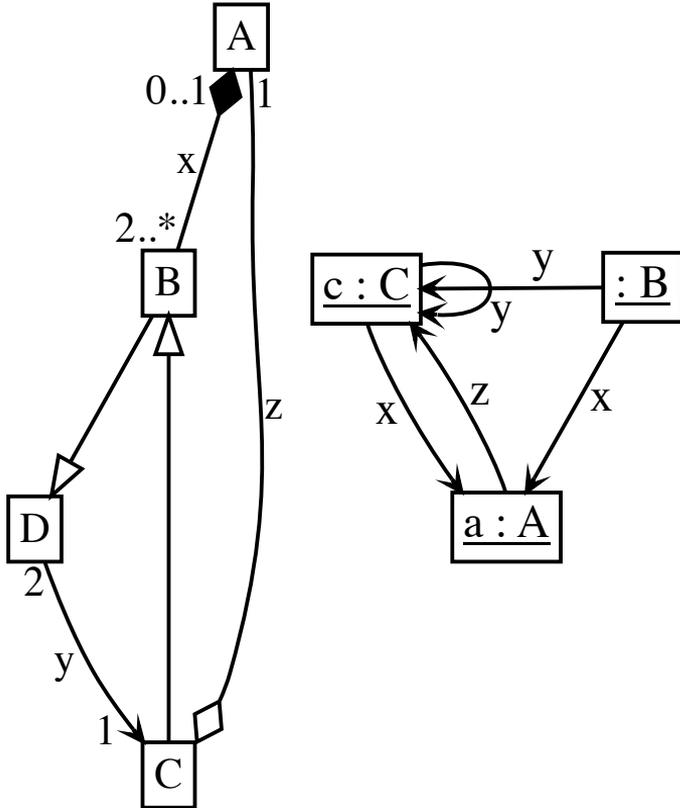
```
...  
one sig y extends FName {}  
...  
fun ASubsCD : set Obj { A }  
fun BSubsCD : set Obj { B + CSubsCD }  
fun CSubsCD : set Obj { C + ASubsCD }  
...  
ObjLUAttrib[ASubsCD, y, BSubsCD, 1, 1]  
ObjLU[BSubsCD, y, ASubsCD, 0, 1]  
...
```



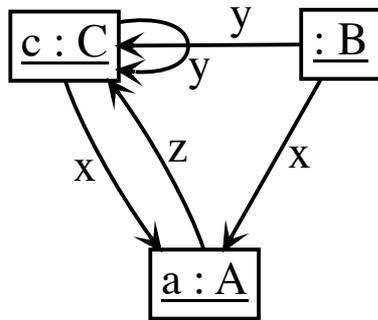
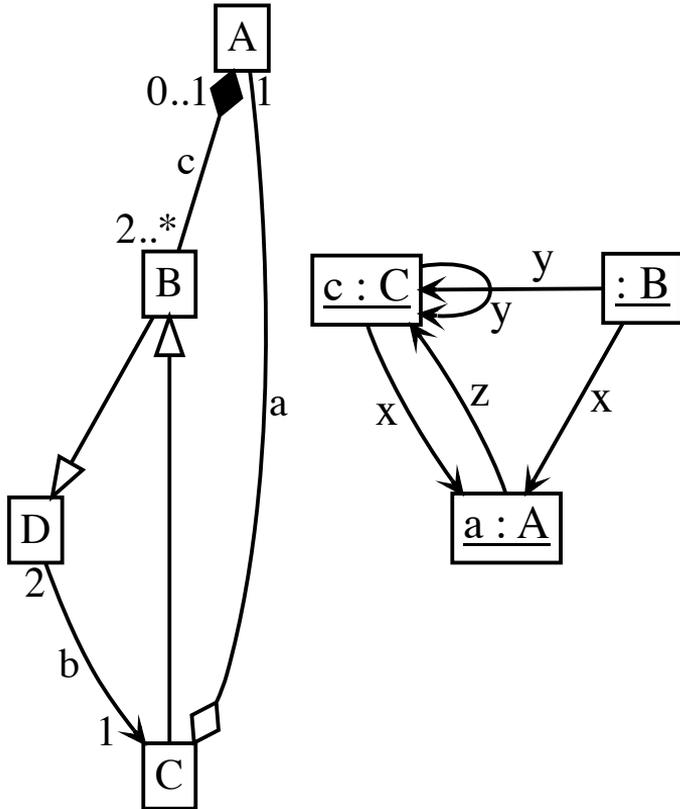
- **Aufgaben mit unterschiedlichen Schwierigkeitsgraden**
  - **verschiedene Aufgaben (idealerweise pro Studierender) mit ähnlichem Schwierigkeitsgrad**
  - **Korrektheit durch Synthese von Aufgaben**
- ↳ **Automatische Überprüfung von Einreichungen und automatisches Feedback zu Einreichungen**
- ⇒ **Abrufen und Überprüfen unterschiedlicher Lernkompetenzen durch verschiedene Aufgabentypen**



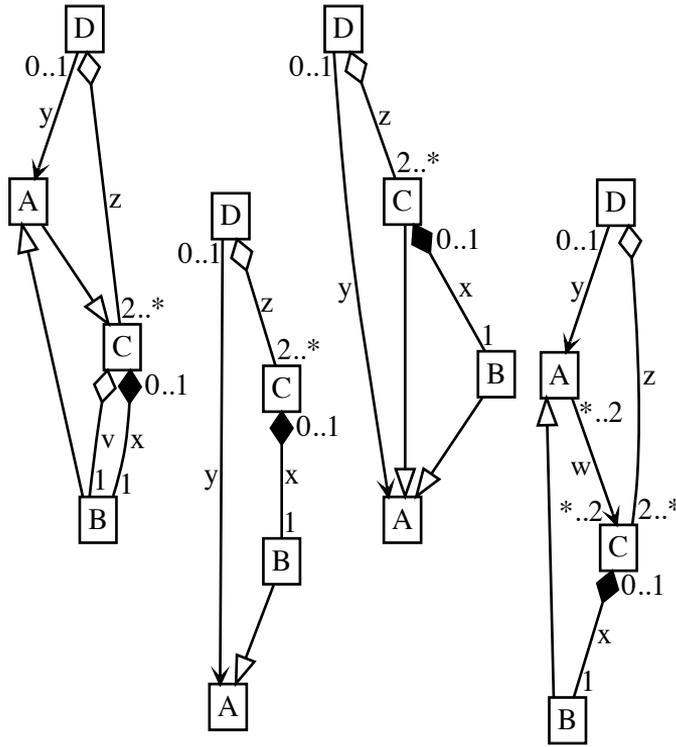
- **Aufgabe:**
  - ▶ Ordnen Sie allen Links im OD, Beziehungen im CD zu!
- **Generierung:**
  - ▶ CD erstellen, wie bei altem Aufgabentyp
  - ▶ OD erstellen, wie bei altem Aufgabentyp
  - ▶ CD Beziehungen neu benennen



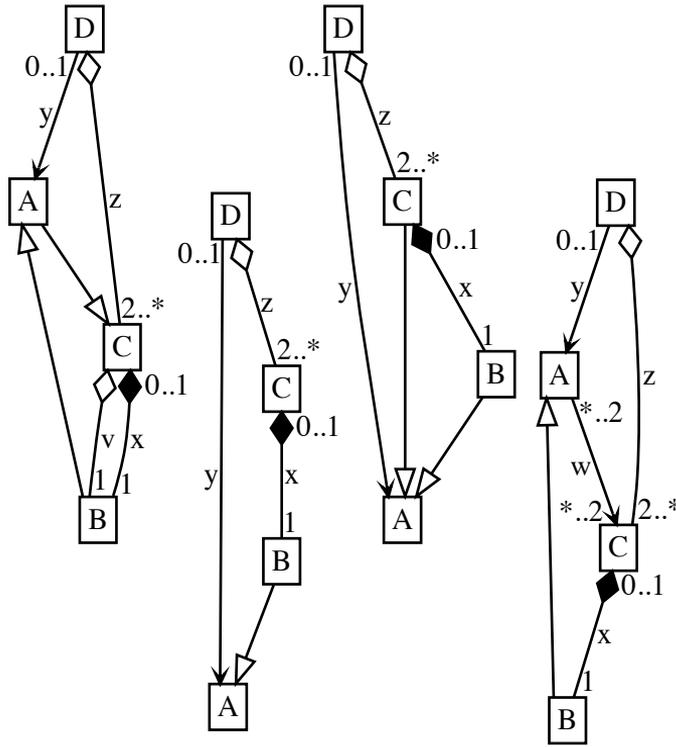
- **Aufgabe:**
  - ▶ Ordnen Sie allen Links im OD, Beziehungen im CD zu!
- **Generierung:**
  - ▶ CD erstellen, wie bei altem Aufgabentyp
  - ▶ OD erstellen, wie bei altem Aufgabentyp
  - ▶ CD Beziehungen neu benennen



- **Aufgabe:**
  - ▶ Ordnen Sie allen Links im OD, Beziehungen im CD zu!
- **Generierung:**
  - ▶ CD erstellen, wie bei altem Aufgabentyp
  - ▶ OD erstellen, wie bei altem Aufgabentyp
  - ▶ CD Beziehungen neu benennen



- **Aufgabe:**
  - ▶ Entscheiden Sie für die gegebenen CD, ob sie gültig sind!
- **Generierung:**
  - ▶ Erstellung eines ungültigen CD
  - ▶ Änderung des CD
    - » 4 (un)gültige CDs



- **Aufgabe:**

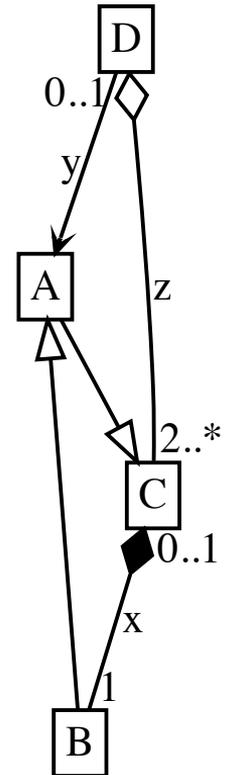
- ▶ **Entscheiden Sie für die gegebenen CD, ob sie gültig sind!**

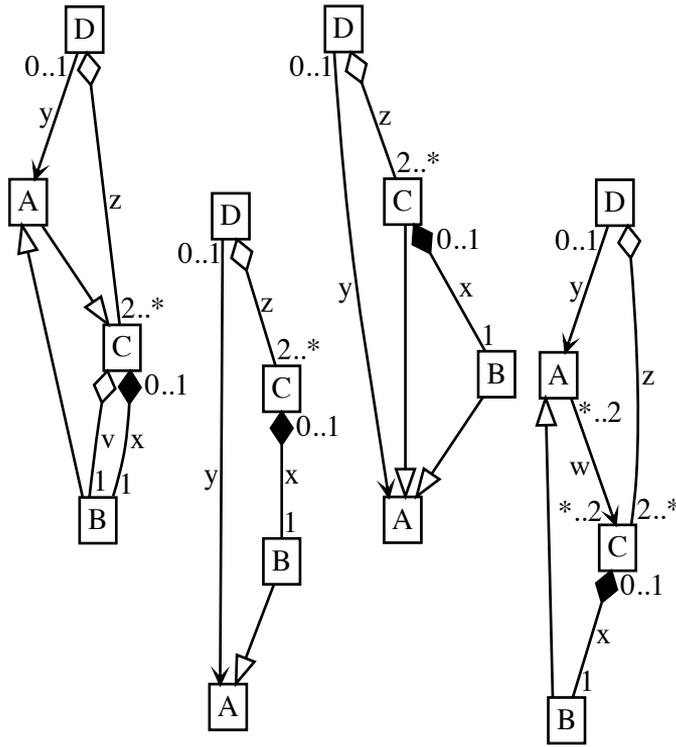
- **Generierung:**

- ▶ **Erstellung eines ungültigen CD**

- ▶ **Änderung des CD**

- » **4 (un)gültige CDs**





- **Aufgabe:**

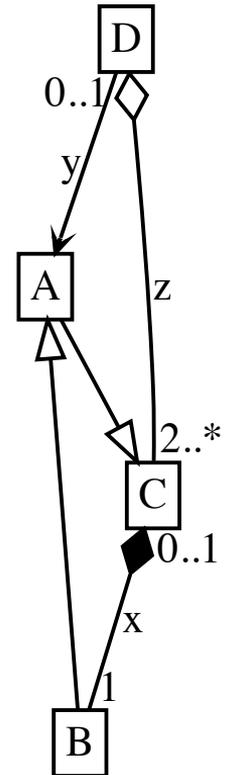
- ▶ **Entscheiden Sie für die gegebenen CD, ob sie gültig sind!**

- **Generierung:**

- ▶ **Erstellung eines ungültigen CD**

- ▶ **Änderung des CD**

- » **4 (un)gültige CDs**



```
abstract sig Class {}

abstract sig Relationship {
  from : one Class,
  to : one Class
}
...
sig Inheritance extends ↵
↵ Relationship {}
...
pred noInheritanceCycles [is : ↵
↵ set Inheritance] {
  no c : Class | c in ↵
↵ c.^((~ from :> is).(is <: to))
}
...
```

- **CD besteht aus Klassen und Beziehungen**
- **Eigenschaften des CD werden durch Prädikate ausgedrückt**
  - ▶ z.B. keine Vererbungszyklen
  - ▶ Prädikate können gezielt (de)aktiviert werden
- **Generator erzeugt alle CDs, die zur entsprechenden Beschreibung passen (bis zu festgelegter Größe)**

```
classDiagram[
  . . . ,
  0,
  0,
  0,
  False,
  False,
  False,
  False,
  False]
```

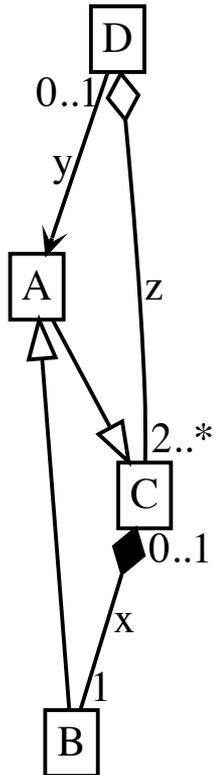
- die Anzahl der Kompositionen mit falschen Multiplizitäten am Rautenende,
- die Anzahl der Relationen mit anderen falschen Multiplizitäten
- die Anzahl der Selbst-Beziehungen
- ob es Doppelbeziehungen gibt
- ob es gegenläufige Beziehungen gibt
- ob es Mehrfachvererbungen gibt
- ob es Vererbungszyklen gibt
- ob es Kompositionszyklen gibt

...

```
abstract sig Change {  
  add : lone ↪  
  ↪ Relationship,  
  remove : lone ↪  
  ↪ Relationship  
}
```

...

- **Definition von Veränderungen eines CD als Hinzufügen / Entfernen maximal einer Beziehung**
- **Definition von Eigenschaften eines veränderten CD mit Hilfe des Prädikats `classDiagram`**



- **Aufgabe:**

- ▶ Wählen Sie alle gültigen Reparaturen für das gegebene CD!

- 1 Füge eine Aggregation für C aus Bs hinzu, bei der C 0..\* mal und B genau einmal partizipiert

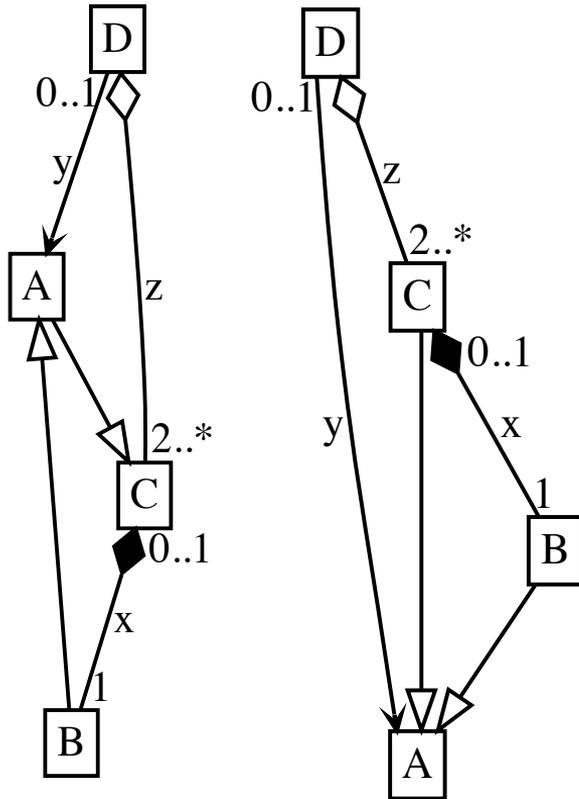
- 2 Entferne eine Vererbung, bei der A von C erbt

- 3 Ersetze eine Vererbung bei der A von C erbt durch eine Vererbung bei der C von A erbt

- 4 Ersetze eine Vererbung bei der A von C erbt durch eine Assoziation von A nach C, bei der A \*..2 mal und C \*..2 mal partizipiert

- **Generierung:**

- ▶ exakt wie bei vorherigem Aufgabentyp

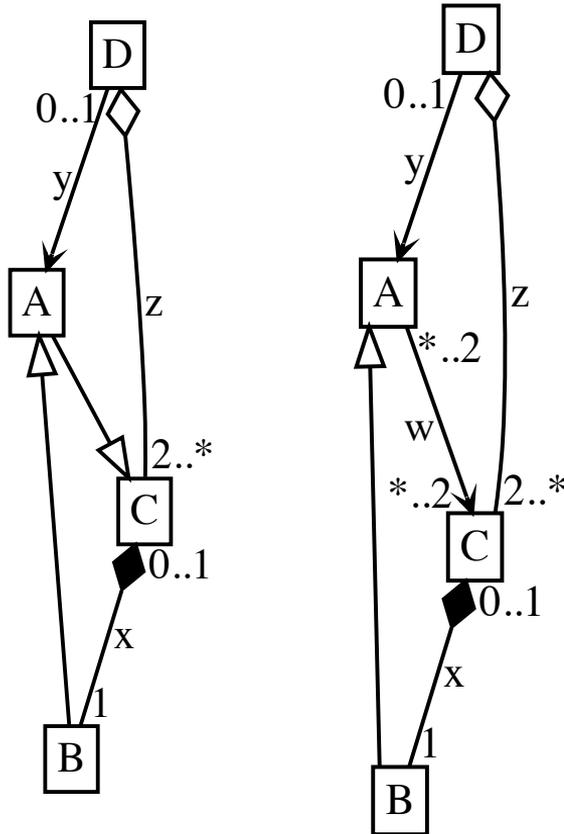


## • Aufgabe:

- ▶ Wählen Sie alle gültigen Reparaturen für das gegebene CD!
- 1 Füge eine Aggregation für C aus Bs hinzu, bei der C 0..\* mal und B genau einmal partizipiert
- 2 Entferne eine Vererbung, bei der A von C erbt
- 3 Ersetze eine Vererbung bei der A von C erbt durch eine Vererbung bei der C von A erbt
- 4 Ersetze eine Vererbung bei der A von C erbt durch eine Assoziation von A nach C, bei der A \*..2 mal und C \*..2 mal partizipiert

## • Generierung:

- ▶ exakt wie bei vorherigem Aufgabentyp



## • Aufgabe:

▶ Wählen Sie alle gültigen Reparaturen für das gegebene CD!

1 Füge eine Aggregation für C aus Bs hinzu, bei der C 0..\* mal und B genau einmal partizipiert

2 Entferne eine Vererbung, bei der A von C erbt

3 Ersetze eine Vererbung bei der A von C erbt durch eine Vererbung bei der C von A erbt

4 Ersetze eine Vererbung bei der A von C erbt durch eine Assoziation von A nach C, bei der A \*..2 mal und C \*..2 mal partizipiert

## • Generierung:

▶ exakt wie bei vorherigem Aufgabentyp

- **Weitere Konfiguration der Aufgabentypen ist möglich:**
  - ▶ **Komplexität anpassbar durch Änderung der Anzahl der Elemente**
  - ▶ **Wahl der angezeigten Komponenten in CD und OD (Navigationsrichtung, Name)**
- **Aufgabenvariationen möglich**
- **Automatische Bewertung der erstellten Aufgaben möglich**
- **Verwendung der Generatoren**
  - ▶ **E-Learning-Kurs mit Aufgaben aller vorgestellter Typen**
  - ▶ **Großflächiger Einsatz im Wintersemester 2020/21 geplant**

Browser address bar: autotool.fmi.iw.uni-due.de/mohol20/select-valid/1

Navigation: Autotool School Exercise

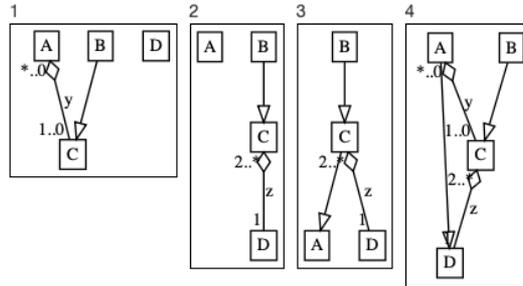
Log in

## Try exercise SelectValidCd-Quiz

<https://autotool.fmi.iw.uni-due.de/mohol20>

### Task

Consider the following class diagrams.



Which of these class diagrams are valid?  
Please state your answer by giving a list of numbers, stating all valid class diagrams.

Please note: hovering over or clicking on edges or their labels highlights both parts.