

Graph- and Model-Driven Engineering
Übungsblatt 5

Es gelten die gleichen Regelungen zur Abgabe wie für Blatt 1.

1 Editieren von Modellen (8 Punkte)

Sie entwickeln einen Editor für für Aktivitätsdiagramme; als Definition der Sprache dienen das Metamodell und die zusätzlichen Constraints aus der Vorlesung. Sie möchten Nutzern die Möglichkeit für freieres Editieren zur Verfügung stellen. Unter anderem sollen die folgenden Editieroperationen zur Verfügung gestellt werden: (i) Es soll möglich sein, den Typ einer `SimpleActivity` in `Decision` zu ändern; (ii) es soll möglich sein, die ausgehende Transition einer `Activity` auf eine beliebige andere `Activity`, die nicht die `EndActivity` ist, umzulenken. Für beide Editieroperationen möchten Sie Nutzern außerdem anschließend Vorschläge für Quickfixes machen können.

Aufgaben:

- Entwerfen Sie für jede der beiden Editieroperationen je eine Graphtransaktionsregel, die die entsprechende Editieroperation umsetzt. (2 Punkte)
- Entwerfen Sie für jede der beiden Editieroperationen je einen Quickfix (auch als Graphtransaktionsregel), der nach der Editieroperation angewendet werden kann. Diese sollen so gestaltet sein, dass die Constraints für Aktivitätsdiagramme aus der Vorlesung nach Anwendung des Quickfixes wieder erfüllt sind, sofern das Eingabemodell M vor Anwendung der Editieroperation valide war. Erklären Sie (informell aber präzise), warum Ihre vorgeschlagenen Quickfixes diese Eigenschaft haben. Für den Quickfix bezüglich des Umlenkens von Transitionen dürfen Sie davon ausgehen, dass der Nutzer bei der Anwendung der Editieroperation keinen Zykel von Transitionen gebaut hat. (6 Punkte)

2 Modellierung einer Aufzugsteuerung (14 Punkte)

In dieser Aufgabe sollen Sie die Steuerung eines Aufzugs modellieren. Jedes Stockwerk des Gebäudes besitzt zwei Knöpfe, um den Aufzug zu rufen. Einen, falls der Fahrgast in ein weiter oben liegendes Stockwerk und einen, falls der Fahrgast in ein weiter unten liegendes Stockwerk fahren möchte (*externe Anfrage*). Der Aufzug selbst besitzt ebenfalls

Graph- and Model-Driven Engineering
Übungsblatt 5

einen Knopf pro Stockwerk, mit dem Fahrgäste einen Halt des Aufzugs veranlassen können (*interne Anfrage*). Für jedes Stockwerk kann immer nur eine interne und jeweils eine externe Anfrage für jede Fahrtrichtung vorliegen. Eine Anfrage gilt als abgearbeitet, wenn der Aufzug im entsprechenden Stockwerk anhält. Liegen keine Anfragen vor, fährt der Aufzug in das mittlere Stockwerk und wartet.

Wird eine Anfrage an einen stehenden Aufzug gestellt, bewegt sich dieser, falls nötig, in Richtung dieser Anfrage. Wird eine Anfrage gestellt, die sich auf das Stockwerk bezieht, in dem sich der Aufzug aktuell befindet, wird diese sofort abgearbeitet. Erreicht der Aufzug ein Stockwerk, für das eine interne Anfrage vorliegt, hält der Aufzug dort an. Erreicht der Aufzug hingegen das Stockwerk einer externen Anfrage, wird diese nur dann berücksichtigt, d. h., der Aufzug hält wegen dieser an, wenn der Aufzug in die Fahrtrichtung fährt, auf die sich die Anfrage bezieht oder keine Anfrage in darüberliegenden bzw. darunterliegenden Stockwerken mehr vorhanden ist. Solange in der aktuellen Fahrtrichtung noch Anfragen abzuarbeiten sind (intern oder extern) wechselt der Aufzug die Fahrtrichtung nicht. Sowohl interne als auch externe Anfragen werden erst gelöscht, wenn der Aufzug im entsprechenden Stockwerk anhält.

Aufgaben:

- a) Welche Abhängigkeiten zwischen Aktionen des Aufzugs gehen aus der textuellen Beschreibung hervor? Geben Sie jeweils ein Beispiel für parallel unabhängige und parallel abhängige und sequentiell unabhängige und sequentiell abhängige Aktionen an und beschreiben Sie deren Unabhängigkeit bzw. Abhängigkeit informell. In Analogie zu Abhängigkeiten von Regelanwendungen sei die Bedeutung von *parallel unabhängig* hier, beide Aktionen lassen sich gleichzeitig durchführen. *Sequentiell abhängig* bedeute, die Durchführung einer Aktion ist unter Umständen nur möglich, wenn die andere Aktion zuvor durchgeführt wurde. (4 Punkte)
- b) Modellieren Sie die Aufzugsteuerung in Groove als typisiertes Graphtransformationssystem über dem Typgraphen, der in Abbildung 1 gezeigt ist. (6 Punkte)
- c) Begründen Sie für jeden der Fälle aus Aufgabenteil a), ob die beschriebene (Un-)Abhängigkeit auch der formalen Definition von paralleler bzw. sequentieller (Un-)Abhängigkeit zwischen Anwendungen von Graphtransmutationsregeln entspricht. (4 Punkte)

Graph- and Model-Driven Engineering
Übungsblatt 5

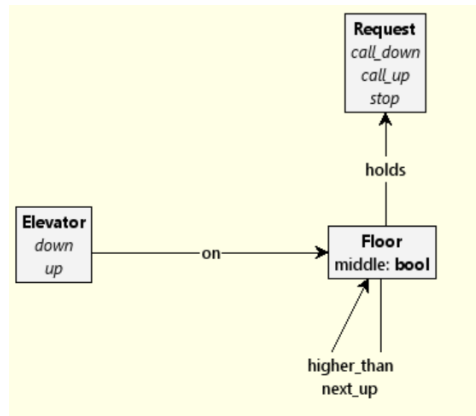


Abbildung 1: Typgraph für die Aufzugsteuerung.

3 Unabhängigkeit (14 Punkte)

Gegeben sind zwei Regelpaare und dazu je ein Graph, der als Grundlage für die Ausführung dieser Regeln dient.

- Zeigen Sie für das in Abbildung 2 dargestellte Regelpaar, dass dieses ausgehend vom gegebenen Graphen *parallel abhängige Regelanwendungen* besitzt. Argumentieren sie hierbei mit der auf Morphismen basierenden Definition paralleler Unabhängigkeit. Zeichnen Sie hierzu alle benötigten Graphen auf, visualisieren Sie die in der Definition betrachteten Mengen und zeigen Sie, dass die Bedingungen für parallele Unabhängigkeit *nicht* erfüllt sind. (2 Punkte)
- Angenommen, das gegebene Regelsystem besteht nur aus den zwei Regeln *pop* und *push*. Ist das oben gefundene parallel abhängige Paar von Regelanwendungen bezüglich dieses Regelsystems lokal konfluent? Falls ja, geben Sie die entsprechenden Sequenzen von Regelanwendungen und das resultierende Ergebnis an (es reicht zu sagen, welche Regel wie oft angewendet werden muss). Falls nein, begründen Sie (informell) wieso nicht. (2 Punkte)
- Geben Sie *alle* kritischen Paare für das Regelpaar (*pop, push*) an. Machen Sie dabei jeweils sowohl den Ausgangsgraphen als auch die beiden Ansätze der Regelanwendungen klar. (5 Punkte)

Graph- and Model-Driven Engineering
Übungsblatt 5

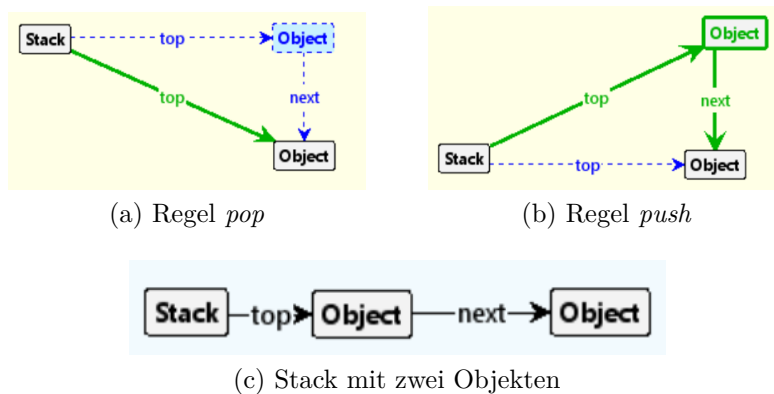


Abbildung 2: Transformationsregeln *pop* und *push* und Zustandsgraph (c) eines Stacks.

- d) Zeigen Sie für das in Abbildung 3 dargestellte Regelpaar, dass dieses bezüglich des gegebenen Graphen *sequentiell unabhängige Regelanwendungen* besitzt. Argumentieren Sie hierbei mit der mengentheoretischen Definition sequentieller Unabhängigkeit. Zeichnen Sie hierzu alle benötigten Graphen auf, visualisieren Sie die in der Definition betrachteten Mengen und zeigen Sie, dass die Bedingungen für sequentielle Unabhängigkeit erfüllt sind. (2 Punkte)
- e) Das Parallelismustheorem besagt, dass statt der sequentiell unabhängigen Sequenz aus Aufgabenteil d) äquivalent die parallele Regel von *put* und *get* an einem passenden Ansatz angewendet werden kann und das gleiche Ergebnis berechnet. Geben Sie diesen Ansatz auf dem Graphen aus Abbildung 3c an. (3 Punkte)

Graph- and Model-Driven Engineering
Übungsblatt 5

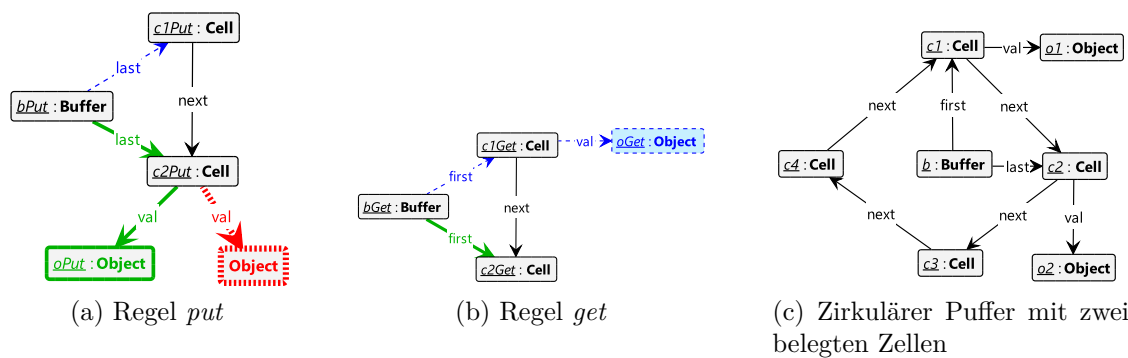


Abbildung 3: Die Regeln *put* und *get* (mit benannten Elementen) und eine Instanz eines zirkulären Puffers mit zwei belegten Zellen.