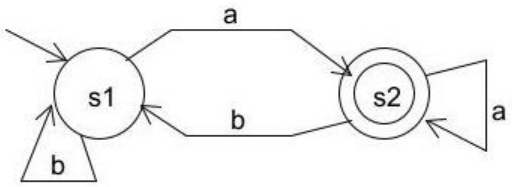
Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Logik

Sommersemester 2022
12. Aufgabenblatt

(Selbststudium, keine Abnahme, bzw. auf Nachfrage)

Aufgabe 20 (0 Punkte)



Ein endlicher deterministischer Automat dient zur Berechnung, ob eine Folge von Zeichen (hier ein String) akzeptiert wird oder nicht. Zur Berechnung hat der Automat einen Startzustand (mit eingehendem Pfeil gekennzeichnet) und eine Überführungsfunktion mit der festgelegt wird, wie aus einem aktuellen Zustand mit einem eingelesenen Zeichen in einen Folgezustand übergegangen werden kann. Die Überführungsfunktion ist graphisch als gerichtete Kante zwischen aktuellem und Folgezustand, markiert mit dem gelesenen Zeichen. Eine Teilmenge der endlichen Menge von Zuständen sind Endzustände, mit einem Doppelkreis markiert. Ist die Eingabe vollständig abgearbeitet und der Automat befindet sich in einem der Endzustände, wird die Eingabe akzeptiert, sonst nicht. Der Automat auf der rechten Seite akzeptiert z. B. ababaa, aber nicht abb.
Überlegen Sie sich, wie generell ein konkreter gegebener Automat mit Drools umgesetzt werden kann, dabei ist die Überführungsfunktion mit Drools-Regeln umzusetzen.
Zur Realisierung steht das Projekt logikAufgabeDroolsAutomat zur Verfügung, das bereits ein vorkonfiguriertes Programm enthält. Die Regeln sollen in der Datei regeln.drl stehen, die notwendigen Schritte zur Übergabe von Informationen zur Drools-Engine und zurück müssen Sie sich selbst überlegen. Bringen Sie die Klasse main.Main zunächst zum Laufen. Sollten Sie kleukersSEU direkt unter C: installiert haben, sollte dies direkt möglich sein. Sonst müssen aus dem Modulepath alle Jar-Dateien bis auf slf4j-simple-1.7.32.jar gelöscht und dann alle Jar-Dateien aus drools/binaries in kleukersSEU hinzugefügt werden.
Setzen Sie dann den Beispielautomaten mit Drools um, am Ende soll folgender Programmablauf möglich sein, Eingaben sind grün.

```
Text (Ende mit x): ababaa
true
Text (Ende mit x): abb
false
Text (Ende mit x): x
```

Hinweise: Beachten Sie, dass die Werte der Klassen String sowie Boolean und des Typs boolean immutable sind, d. h. der Wert kann nicht geändert werden. Sollen solche Werte änderbar sein, sind sie in Klassen zu kapseln, wie es im Projekt mit den Klassen Text und Ergebnis erfolgt.
Sollten Sie ein Objekt verändern sollen, muss es im then-Teil in einem modify-Block stehen, wie es z. B. in Folie 270 angedeutet ist.
Für eine neue Nutzung der Drools-Enginge müssen alte Objekte entfernt werden. Dazu gibt die Methode insert() ein FactHandle-Objekt zurück, dass nach der Berechnung zum Löschen mit delete() genutzt werden kann. Alternativ könnte eine neue Session erstellt werden.

```
FactHandle f1 = kSession.insert(start);
...
kSession.delete(f1);
```

Aufgabe 21 (0 Punkte)

Laden Sie von der Veranstaltungsseite das Projekt logikDroolsRechteAufgabe, mit dem Objekte einer stark vereinfachten Klasse Recht verwaltet werden sollen, dabei beschreibt jedes Objekt wer, aus was (also welches Artefakt, wie ein user-Verzeichnis), wie (schreibend, lesend) zugreifen kann. Lesen Sie sich die Klasse Systemtest, insbesondere die Tests genau

durch und schreiben Sie Drools-Regeln in den geforderten Regeldateien, die dazu führen, dass alle Tests erfüllt werden. Es werden jeweils String-Listen berechnet, die folgende Fragen beantworten.

Welche nutzenden Personen (Ergebnis: wer) dürfen das Artefakt „user“ lesen oder schreiben?

Welche nutzenden Personen (Ergebnis: wer) haben bei zumindest einem Artefakt gleichzeitig Lese- und Schreibrechte?

Welche nutzenden Personen (Ergebnis: wer) dürfen auf Artefakte ausschließlich nur lesend (oder gar nicht) zugreifen?

Für welche bekannten Artefakte (Ergebnis: worauf) wurden keine Schreibrechte vergeben?

Welche nutzenden Personen dürfen (Ergebnis: wer) auf alle bekannten Artefakte zugreifen?

Sorgen Sie dafür, dass alle Tests laufen.