CARL VON OSSIETZKY universität OLDENBURG

Sommersemester 2020

Objektorientierte Modellierung und Programmierung

Dr. Christian Schönberg

CARL VON OSSIETZKY universität OLDENBURG

Regelbasierte Programmierung

CARL VON OSSIETZKY universität OLDENBURG

Lernziele

- Regelsysteme
- JBoss Drools
- Rete-Algorithmus
- IFTTT

CARL VON OSSIETZKY universität OLDENBURG

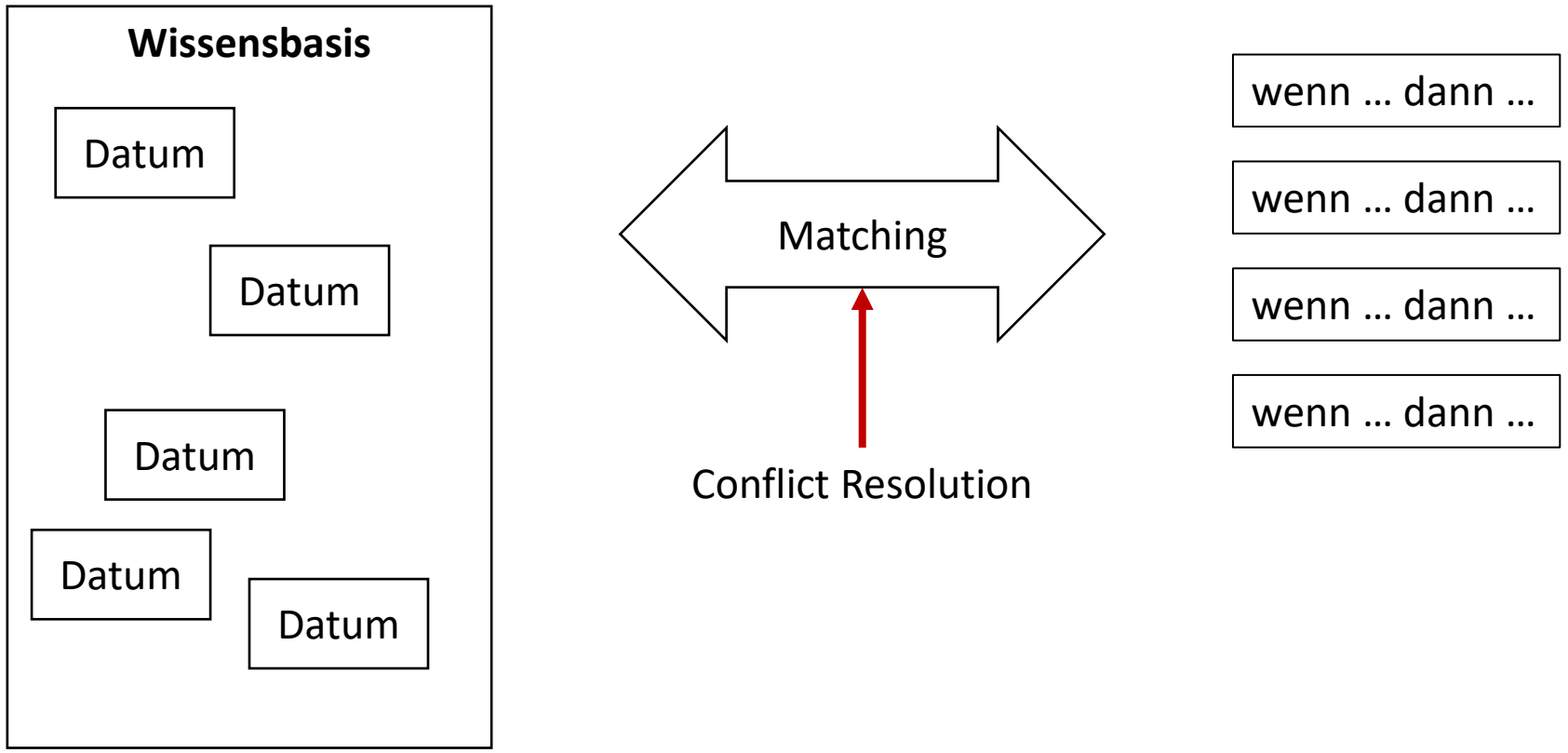
Regelbasierte Programmierung

CARL VON OSSIETZKY universität OLDENBURG

Regelbasierte Programmierung

- Komponenten
  - Wissensbasis
    - enthält Daten, Fakten
  - Regeln
    - bestehen aus Prämisse (Vorbedingung, LHS) und Konklusion (Aktion, RHS)
  - Matching-Algorithmus
    - überwacht die Wissensbasis und die Regeln und stellt fest, wenn die Prämisse einer Regel von einem Datum aus der Wissensbasis erfüllt ist; führt dann die Konklusion aus
  - Conflict Resolution-Algorithmus
    - behandelt Konflikte, d.h. bestimmt die Ausführungsreihenfolge wenn z.B. mehrere Daten eine Regel auslösen, oder wenn mehrere Regeln gleichzeitig ausgelöst werden

Regelbasierte Programmierung



CARL VON OSSIETZKY universität OLDENBURG

JBoss Drools

JBoss Drools

- Business-Rule Management System
- Entwickelt von Red Hat unter der KIE Platform
  - Knowledge Is Everything
- Java-basiert
- Kann zu Stand-Alone Java-Programm kompiliert werden
- Kann externe Ressourcen, Services und Bibliotheken ansprechen
  - externe Wissensdatenbanken
  - Datenverarbeitung
- http://www.drools.org/

CARL VON OSSIETZKY universität OLDENBURG

JBoss Drools (2)

- Wissensbasis
  - Java Objekte
- Regeln
  - eigene Sprache für Regeln, eigene Konstrukte für Prämissen
  - Java-ähnliche Sprache für die Konklusion
- Matching-Algorithmus: Rete bzw. Nachfolger
  - Charles Forgy, 1979
- Conflict Resolution-Algorithmus
  - Priorität
  - Reihenfolge des Einfügens

CARL VON OSSIETZKY universität OLDENBURG

Drools Regeln

```
package de.uni_oldenburg.inf.omp.drools

import de.uni_oldenburg.inf.omp.drools.utils.*;

rule "Rule 1"
when
    Bedingungen
then
    Aktionen
end

rule "Rule 2"
when
    Bedingungen
then
    Aktionen
end
```

CARL VON OSSIETZKY universität OLDENBURG

Drools Regeln (2)

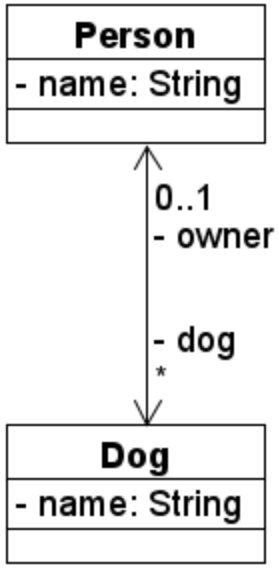
Person
- name: String

```
rule "Hello Hans"
when
    Person(name == "Hans")
then
    System.out.println("Hello there, Hans");
end
```

```
rule "Hello Person"
when
    $person : Person()
then
    System.out.println("Hello " + $person.getName());
end
```

```
rule "Hello Hans"
salience 20
when
    Person(name == "Hans")
then
    System.out.println("Hello there, Hans");
end
```

Drools Regeln (3)




```
rule "Hello Person and Dog"
when
    $person : Person()
    $dog : Dog(owner == $person)
then
    System.out.println("Hello " +
            $person.getName() + " and " +
            $dog.getName());
end
```

CARL VON OSSIETZKY universität OLDENBURG

Verändern der Wissensbasis

- Drools-Regeln können aktiv die Wissensbasis verändern
- **`insert(Object o)`**: Fügt ein neues Objekt in die Wissensbasis ein
- **`delete(Object o)`**: Löscht ein existierendes Objekt aus der Wissensbasis
- **`modify(Object o) { … }`**: Verändert ein existierendes Objekt in der Wissensbasis
- **`insert`** und **`modify`** können dazu führen, dass neue Regeln ausgelöst werden

Rete-Algorithmus

- Matching von Regeln und Daten
- Naives Vorgehen:

```
while (true) {
    for (Rule rule : rules) {
        for (Data data : dataset) {
            if (rule.matches(data)) {
                rule.execute(data);
            }
        }
    }
}
```

- Rete-Algorithmus:
  - reduziere die Menge der Regeln und die Menge der Daten, die betrachtet werden müssen
  - nur Daten, die sich verändert haben, müssen neu betrachtet werden
  - wenn sich (Teil-)Bedingungen in Regel-Prämissen wiederholen, werte sie nur einmal aus

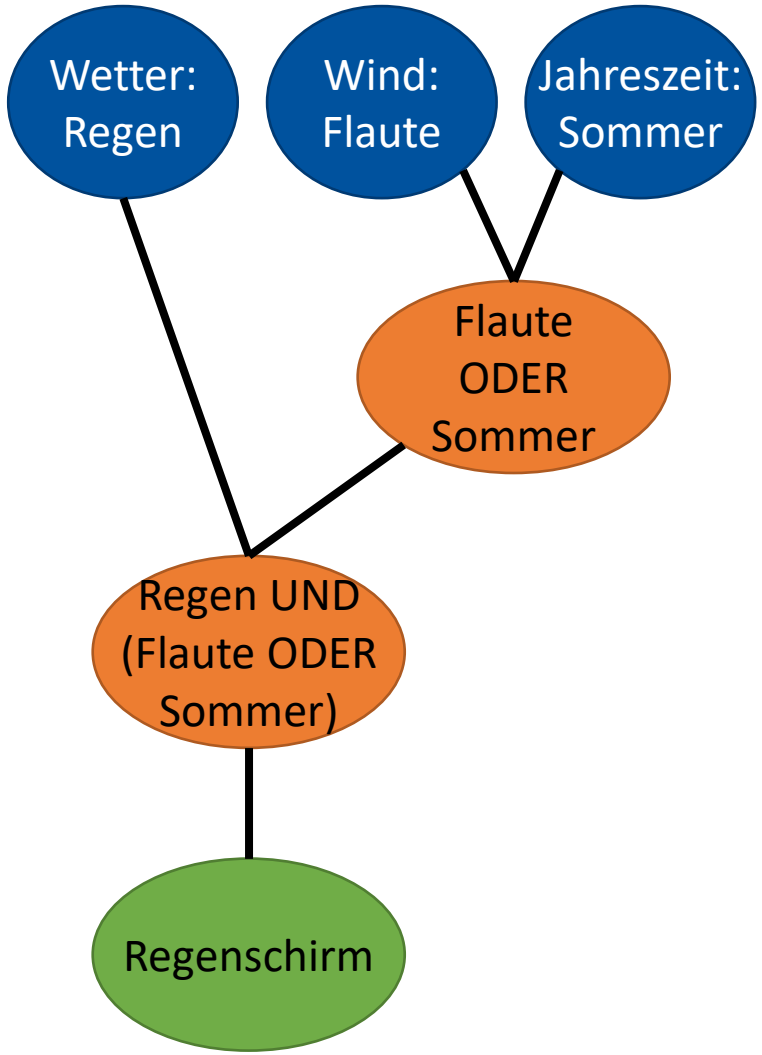
Rete-Algorithmus

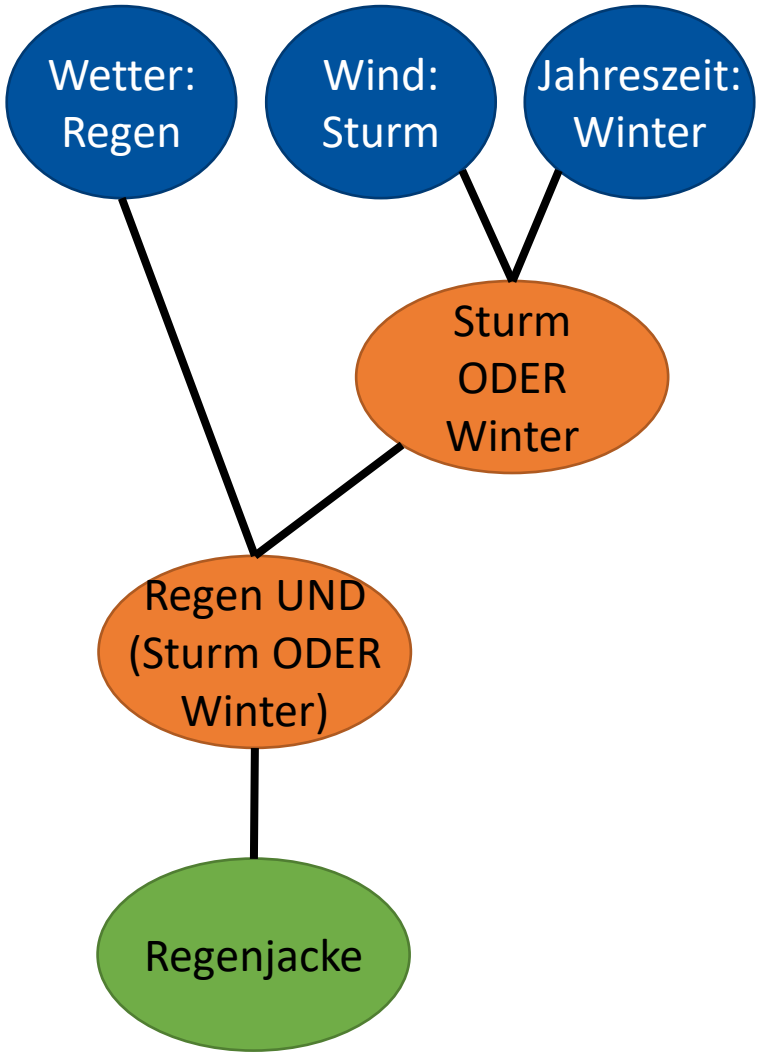
- Betrachte atomare (Teil-)Bedingungen und zusammengesetzte (Teil-) Bedingungen
- Beispiel:
  - Wetter = Regen, Jahreszeit = Sommer, Wind = Flaute
  - (Wetter = Regen UND (Wind = Sturm ODER Jahreszeit = Winter))
- Verknüpfe gleiche Teile von Regel-Prämissen in einem Entscheidungsnetz
- Beispiel:
  - WENN Wetter = Regen UND (Wind = Flaute ODER Jahreszeit = Sommer) DANN empfehle einen Regenschirm
  - WENN Wetter = Regen UND (Wind = Sturm ODER Jahreszeit = Winter) DANN empfehle eine Regenjacke
  - WENN Wetter = Sonne UND (Wind = Flaute ODER Jahreszeit = Sommer) DANN empfehle einen Spaziergang

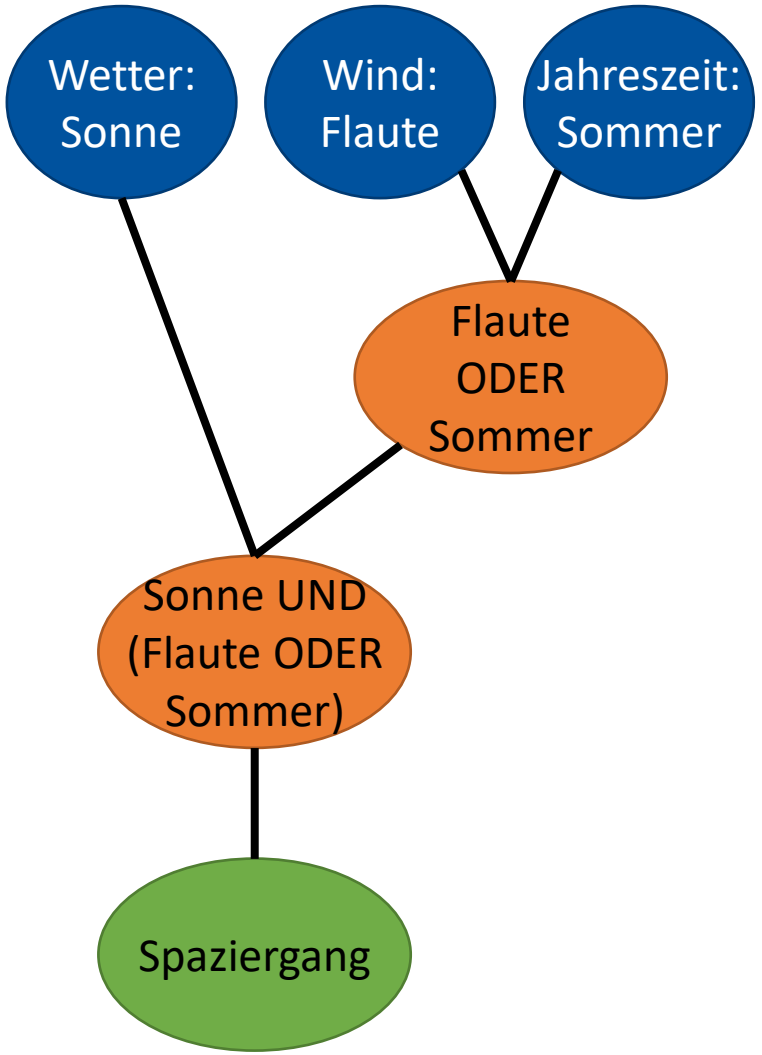
Rete-Algorithmus

- Betrachte atomare (Teil-)Bedingungen und zusammengesetzte (Teil-) Bedingungen
- Beispiel:
  - Wetter = Regen, Jahreszeit = Sommer, Wind = Flaute
  - (Wetter = Regen UND (Wind = Sturm ODER Jahreszeit = Winter))
- Verknüpfe gleiche Teile von Regel-Prämissen in einem Entscheidungsnetz
- Beispiel:
  - WENN Wetter = Regen UND (Wind = Flaute ODER Jahreszeit = Sommer) DANN empfehle einen Regenschirm
  - WENN Wetter = Regen UND (Wind = Sturm ODER Jahreszeit = Winter) DANN empfehle eine Regenjacke
  - WENN Wetter = Sonne UND (Wind = Flaute ODER Jahreszeit = Sommer) DANN empfehle einen Spaziergang

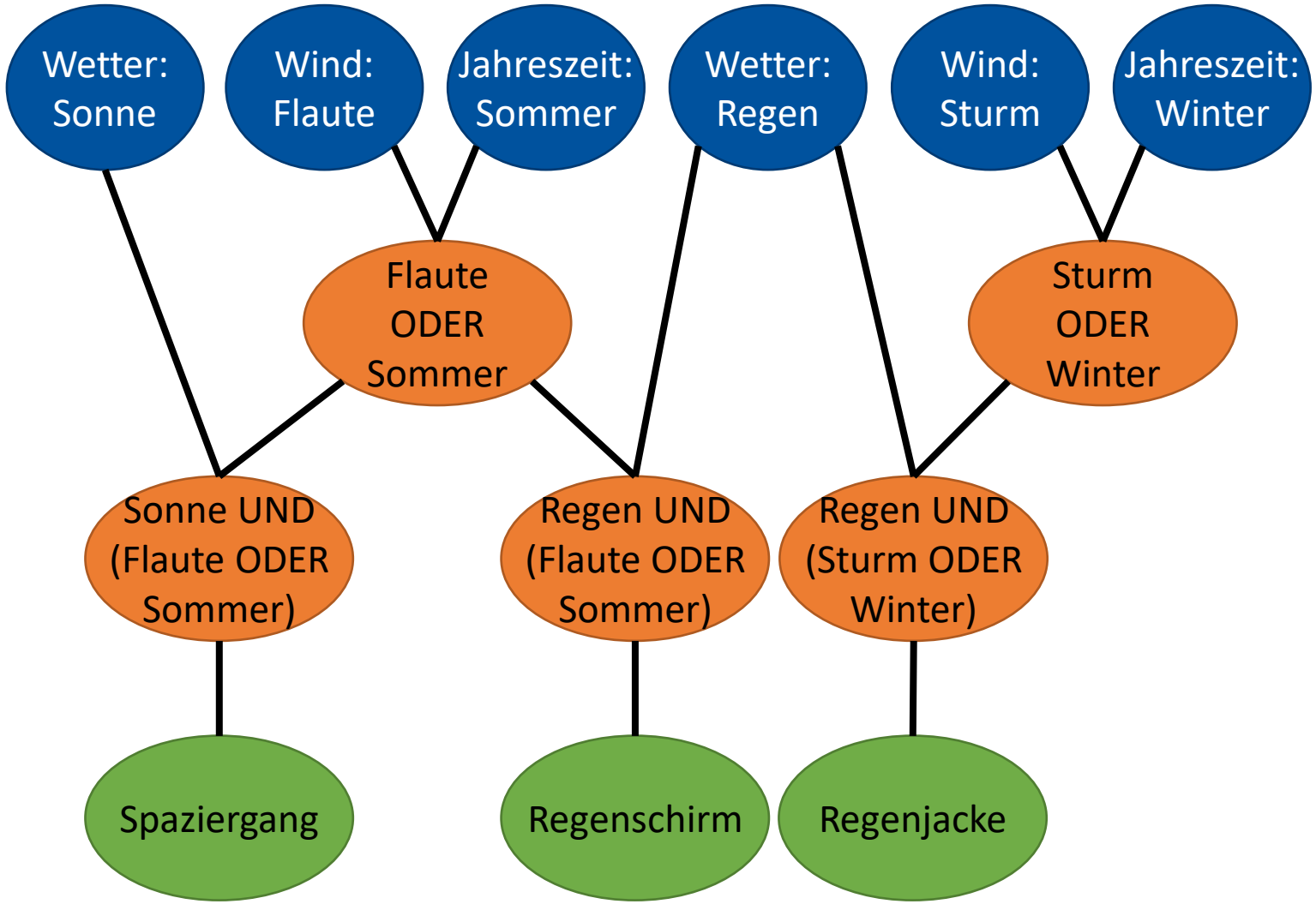
Beispiel: Rete-Algorithmus







Beispiel: Rete-Algorithmus

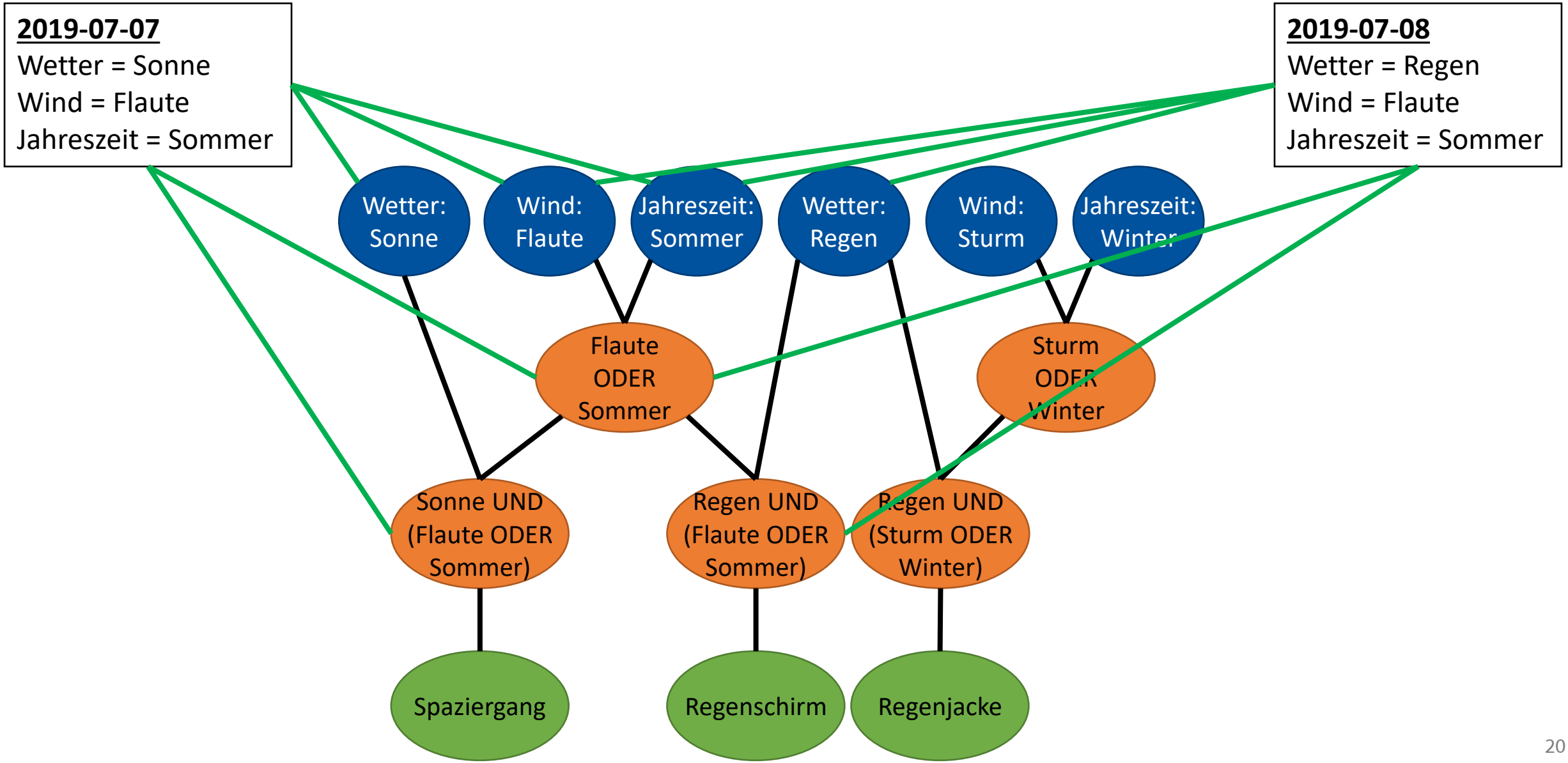


CARL VON OSSIETZKY universität OLDENBURG

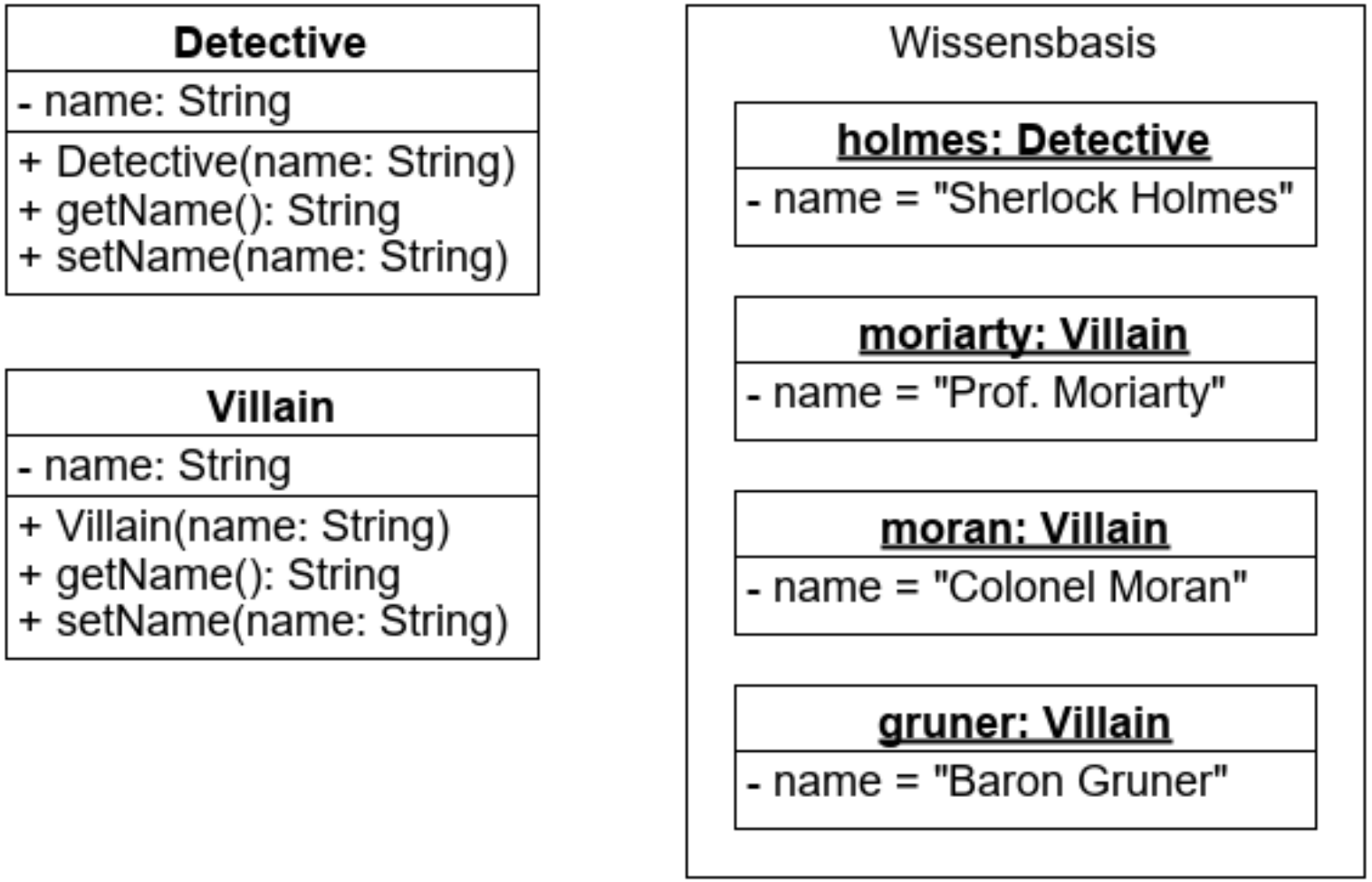
Rete-Algorithmus

- An den Knoten des Entscheidungsnetzes werden die Daten gespeichert, für die der Knoten erfüllt ist
- Diese Mengen werden aktualisiert, wenn sich Daten ändern, wegfallen oder hinzukommen (vgl. **`insert`**, **`delete`**, **`modify`**)
- Das Entscheidungsnetz wird aktualisiert, wenn sich Regeln ändern, wegfallen oder hinzukommen

Beispiel: Rete-Algorithmus



Beispiel: Sherlock Holmes



CARL VON OSSIETZKY universität OLDENBURG

Beispiel: Sherlock Holmes

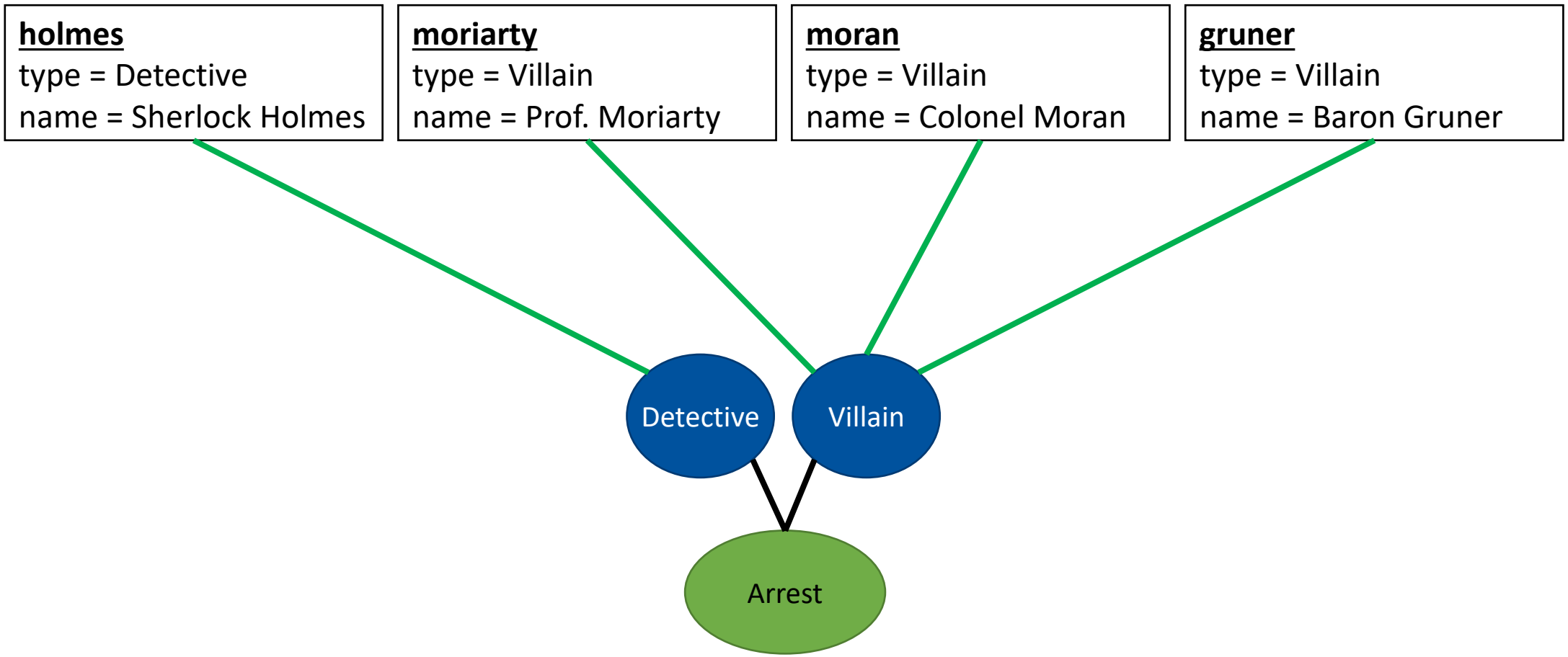
```
package de.uni_oldenburg.inf.omp.drools.detective

rule "Arrest"
when
    $detective : Detective()
    $villain : Villain()
then
    System.out.println($detective.getName() + " arrests " +
            $villain.getName() + ".");
    delete($villain);
end
```

CARL VON OSSIETZKY universität OLDENBURG

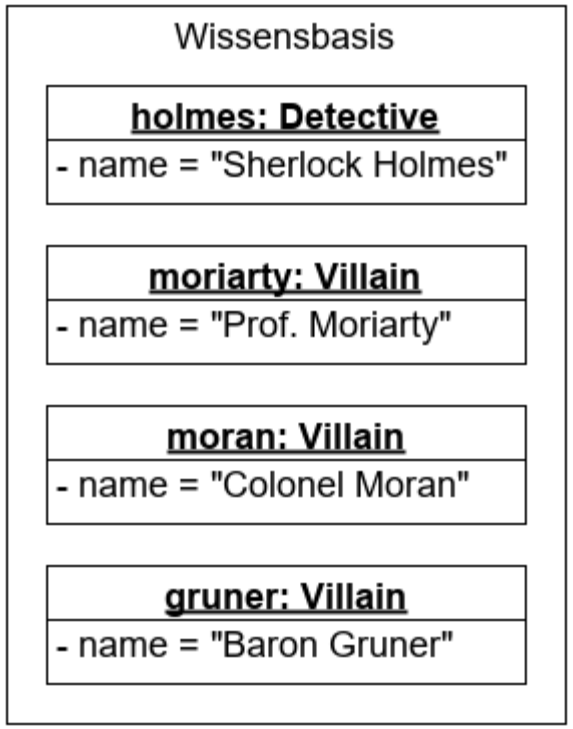
Beispiel: Sherlock Holmes

```
package de.uni_oldenburg.inf.omp.drools.detective;

import org.kie.api.KieServices;
import org.kie.api.runtime.KieContainer;
import org.kie.api.runtime.KieSession;

public class DetectiveRunner {

    public static void main(String[] args) {
        KieServices kieServices = KieServices.Factory.get();
        KieContainer kContainer = kieServices.getKieClasspathContainer();
        KieSession kSession = kContainer.newKieSession("ksession-rules");

        kSession.insert(new Detective("Sherlock Holmes"));
        kSession.insert(new Villain("Prof. Moriarty"));
        kSession.insert(new Villain("Colonel Moran"));
        kSession.insert(new Villain("Baron Gruner"));
        kSession.fireAllRules();
        kSession.dispose();
    }
}
```

Beispiel: Sherlock Holmes

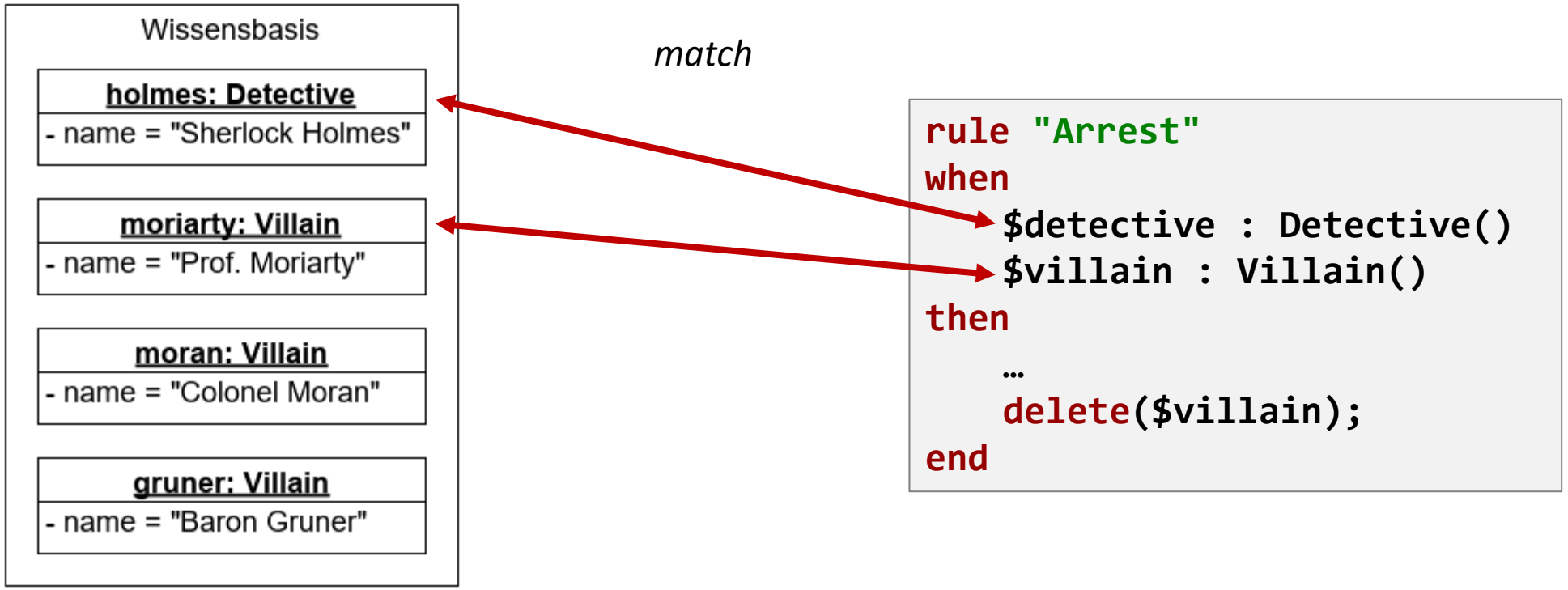


Beispiel: Sherlock Holmes

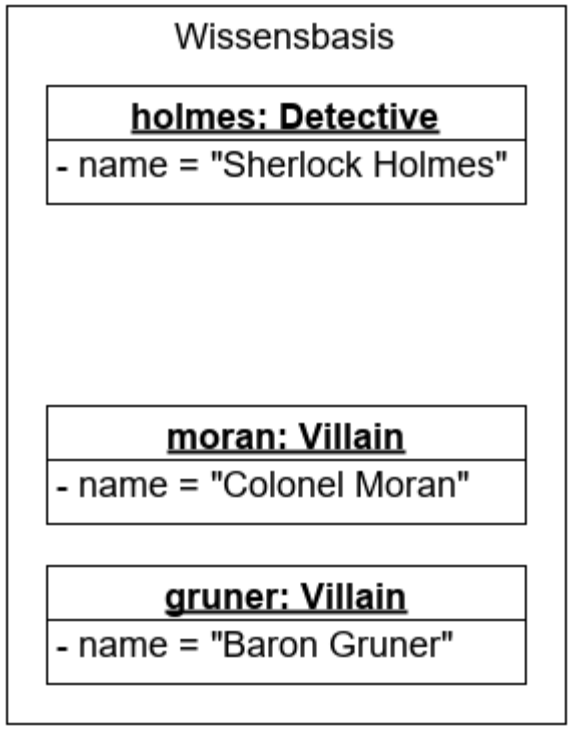



```
rule "Arrest"
when
    $detective : Detective()
    $villain : Villain()
then
    …
    delete($villain);
end
```

Beispiel: Sherlock Holmes

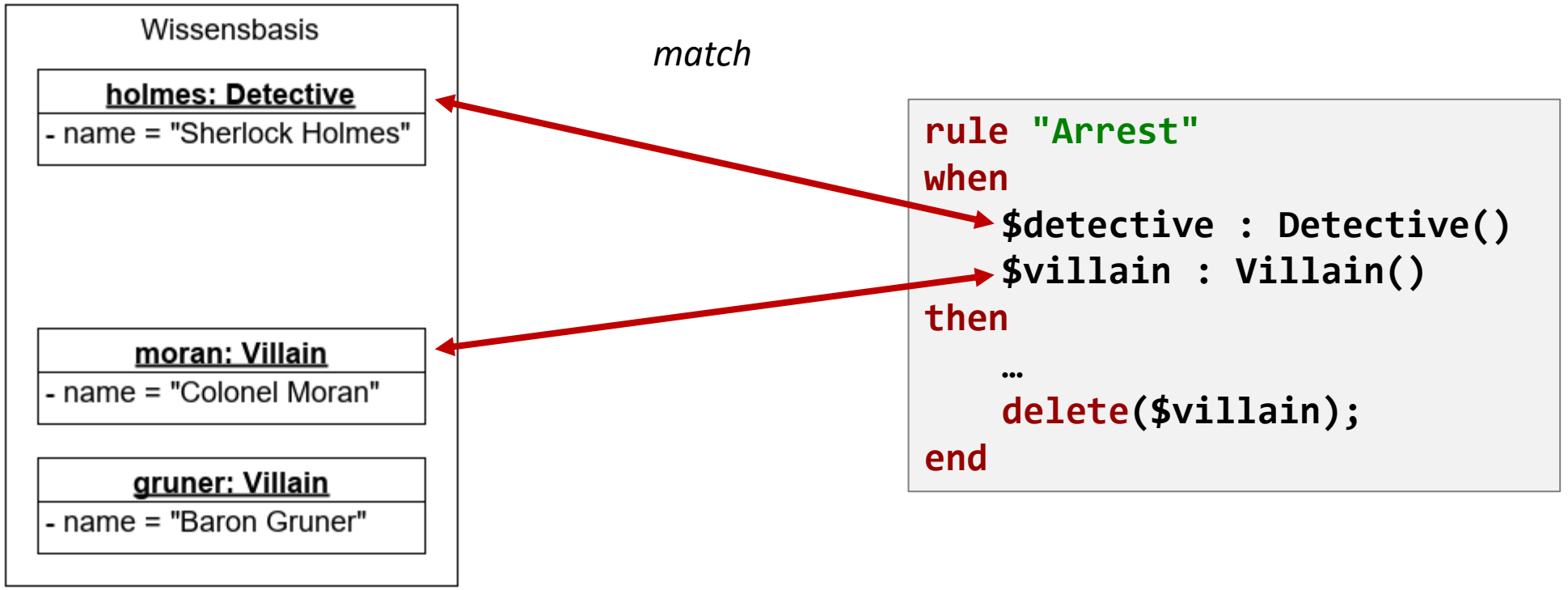


Beispiel: Sherlock Holmes

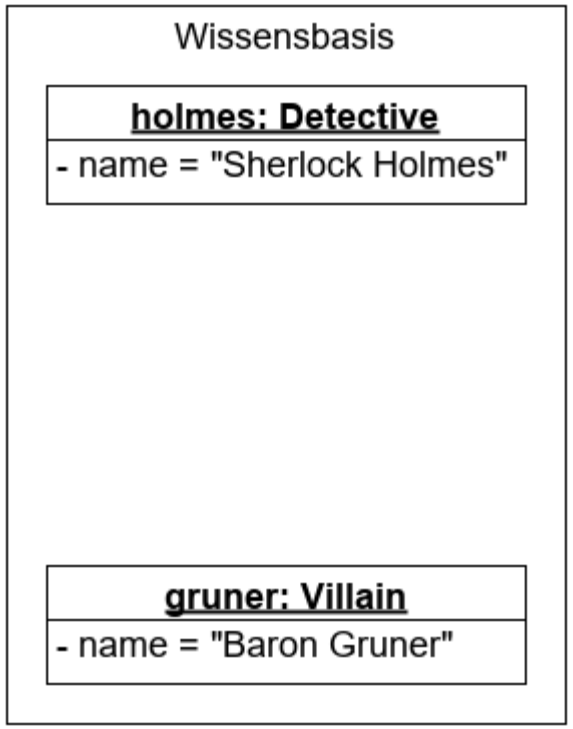



```
rule "Arrest"
when
    $detective : Detective()
    $villain : Villain()
then
    …
    delete($villain);
end
```

Beispiel: Sherlock Holmes

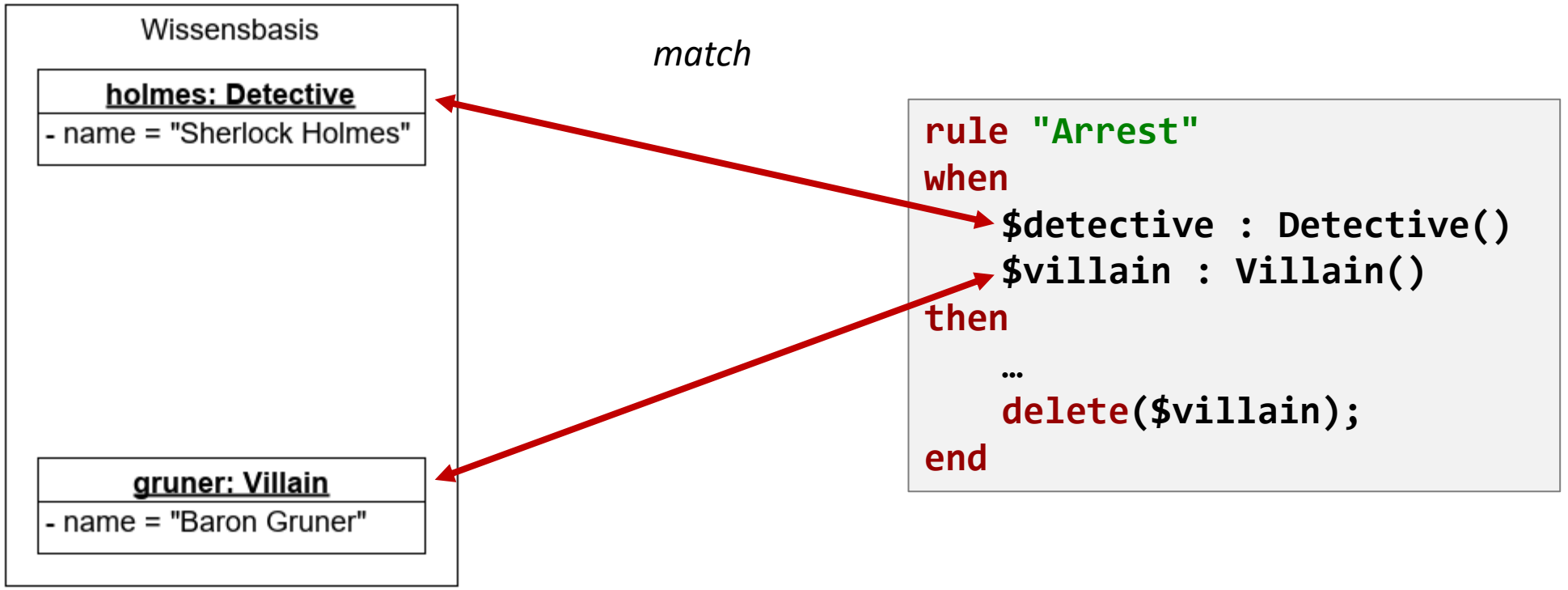


Beispiel: Sherlock Holmes

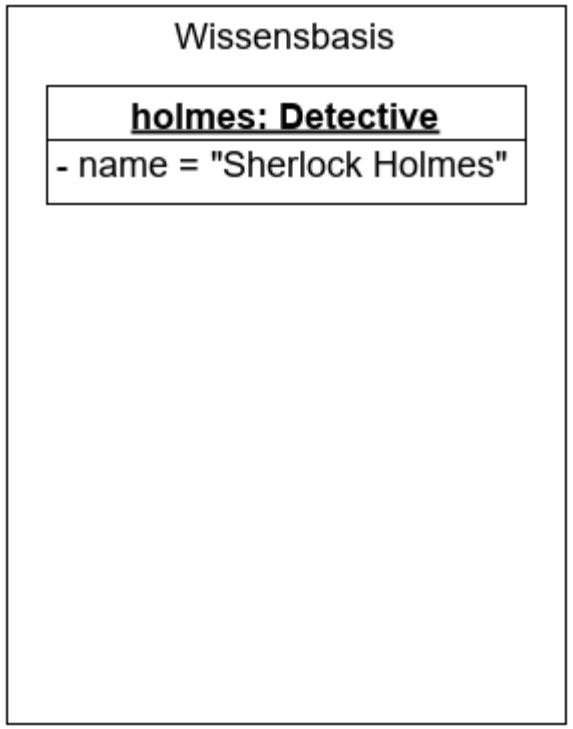



```
rule "Arrest"
when
    $detective : Detective()
    $villain : Villain()
then
    …
    delete($villain);
end
```

Beispiel: Sherlock Holmes

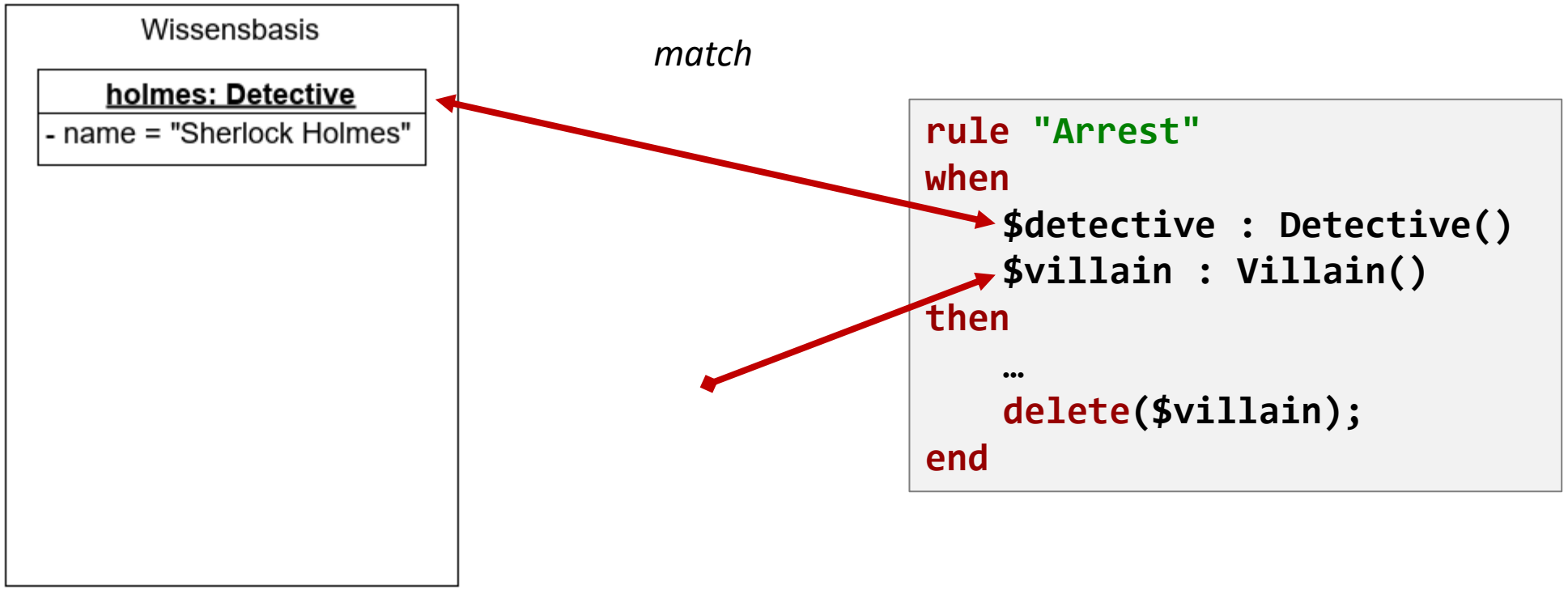


Beispiel: Sherlock Holmes

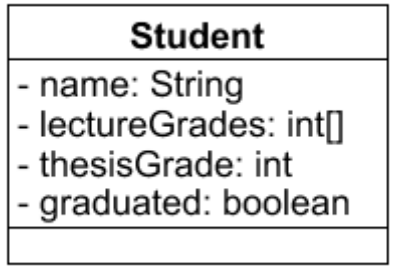



```
rule "Arrest"
when
    $detective : Detective()
    $villain : Villain()
then
    …
    delete($villain);
end
```

Beispiel: Sherlock Holmes



Beispiel: Studenten



- mindestens 4 benotete Vorlesungen
- keine Vorlesung mit Note 5
- Abschlussarbeit mit Note besser als 5

➔ Abschluss

- Abschluss

➔ Ausgabe des Namens

CARL VON OSSIETZKY universität OLDENBURG

Beispiel: Studenten

```
package de.uni_oldenburg.inf.omp.drools.student

rule "Graduate"
when
    $student : Student(thesisGrade > 0, thesisGrade < 5,
                       lectureGrades.size() >= 4,
                       lectureGrades not contains 5,
                       !graduated)
then
    modify($student) {
        setGraduated(true);
    }
    System.out.println($student.getName() + " is about to graduate.");
end

rule "Graduation"
when
    $student : Student(graduated)
then
    System.out.println("Graduating " + $student.getName() + ".");
end
```

CARL VON OSSIETZKY universität OLDENBURG

Beispiel: Studenten

```
package de.uni_oldenburg.inf.omp.drools.student;

…

KieServices kieServices = KieServices.Factory.get();
KieContainer kContainer = kieServices.getKieClasspathContainer();
KieSession kSession = kContainer.newKieSession("ksession-rules");

kSession.insert(
    new Student("Harry Potter", new int[] { 2, 1, 2, 4, 3 }, 2));
kSession.insert(
    new Student("Ronald Weasley", new int[] { 3, 2, 5, 4 }, 3));
kSession.insert(
    new Student("Hermione Granger", new int[] { 1, 1, 1, 1, 1 }, 1));
kSession.insert(
    new Student("Charles Weasley", new int[] { 1, 2, 4, 2, 2, 3 }, 2, true));
kSession.fireAllRules();
kSession.dispose();
```

```
Hermione Granger is about to graduate.
Harry Potter is about to graduate.
Graduating Harry Potter
Graduating Hermione Granger
Graduating Charles Weasley
```

Beispiel: Studenten

Reihenfolge:

- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Agenda:

Harry Potter

Ronald Weasley

Hermione Granger

Charles Weasley (graduated)

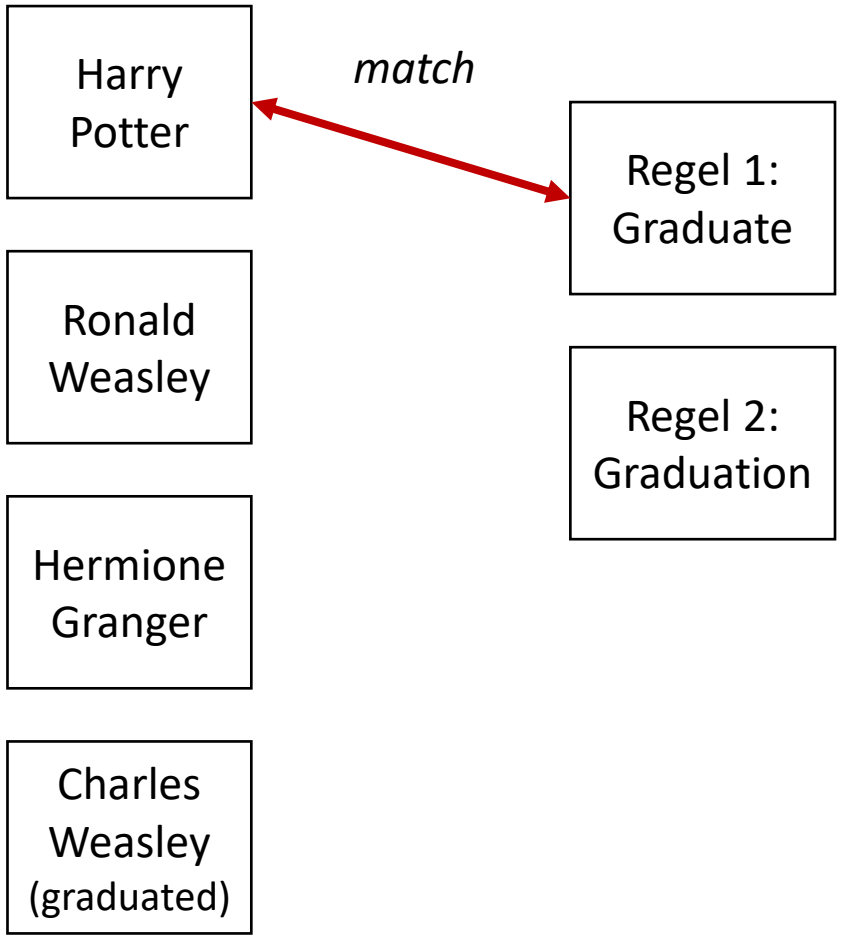
Regel 1: Graduate

Regel 2: Graduation

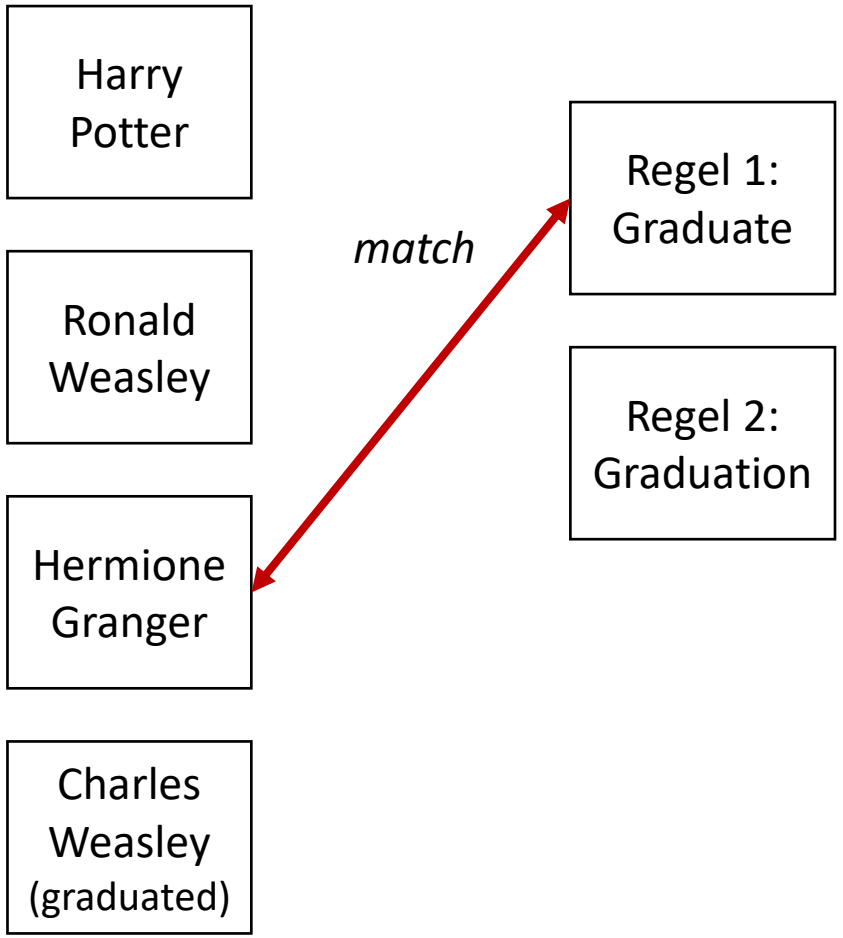
Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Agenda:
- **Regel 1 (Harry Potter)**



Beispiel: Studenten



Reihenfolge:

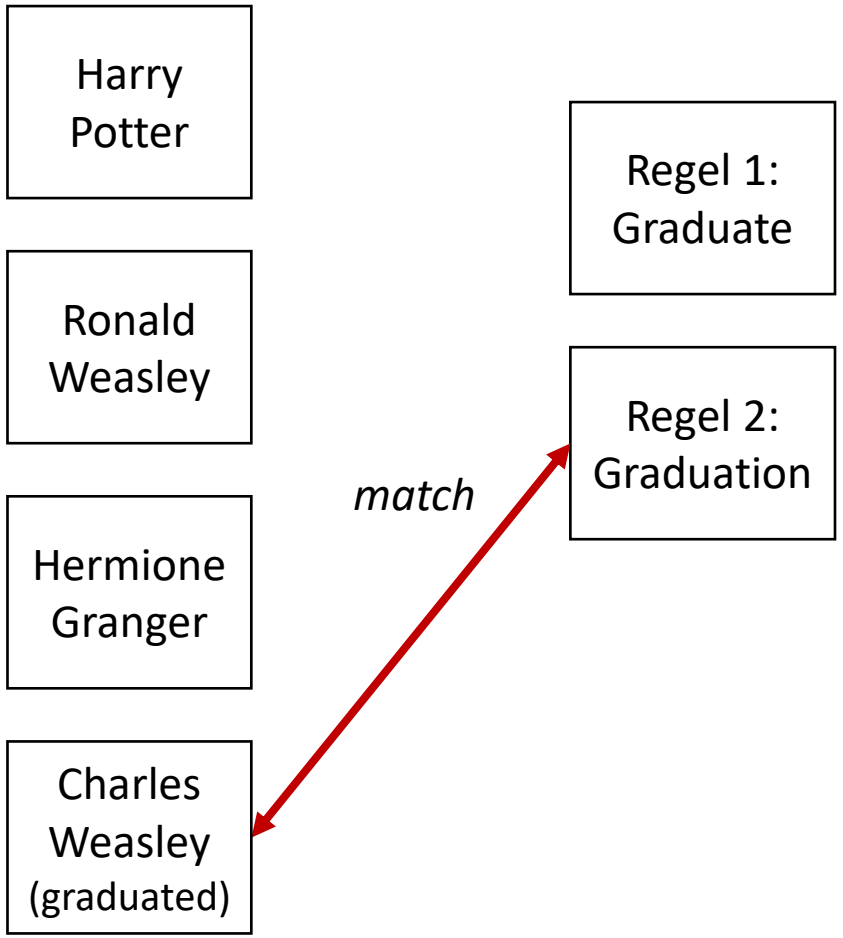
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Agenda:

- **Regel 1 (Hermione Granger)**
- Regel 1 (Harry Potter)

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO



Agenda:
- Regel 1 (Hermione Granger)
- Regel 1 (Harry Potter)
- **Regel 2 (Charles Weasley)**

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Agenda:
- ~~Regel 1 (Hermione Granger)~~
- Regel 1 (Harry Potter)
- Regel 2 (Charles Weasley)

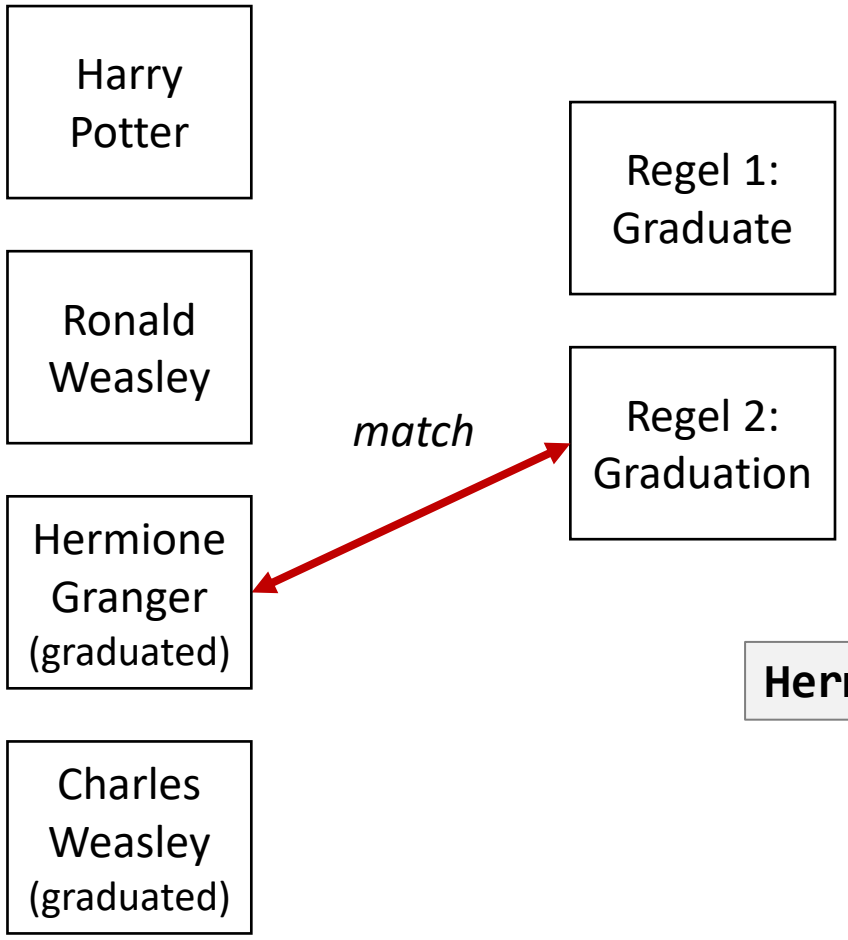
Harry Potter
Ronald Weasley
Hermione Granger (graduated)
Charles Weasley (graduated)

Regel 1: Graduate
Regel 2: Graduation

```
Hermione Granger is about to graduate.
```

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO



Agenda:
- ~~Regel 1 (Hermione Granger)~~
- Regel 1 (Harry Potter)
- **Regel 2 (Hermione Granger)**
- Regel 2 (Charles Weasley)

```
Hermione Granger is about to graduate.
```

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Harry Potter (graduated)

Ronald Weasley

Hermione Granger (graduated)

Charles Weasley (graduated)

Regel 1: Graduate

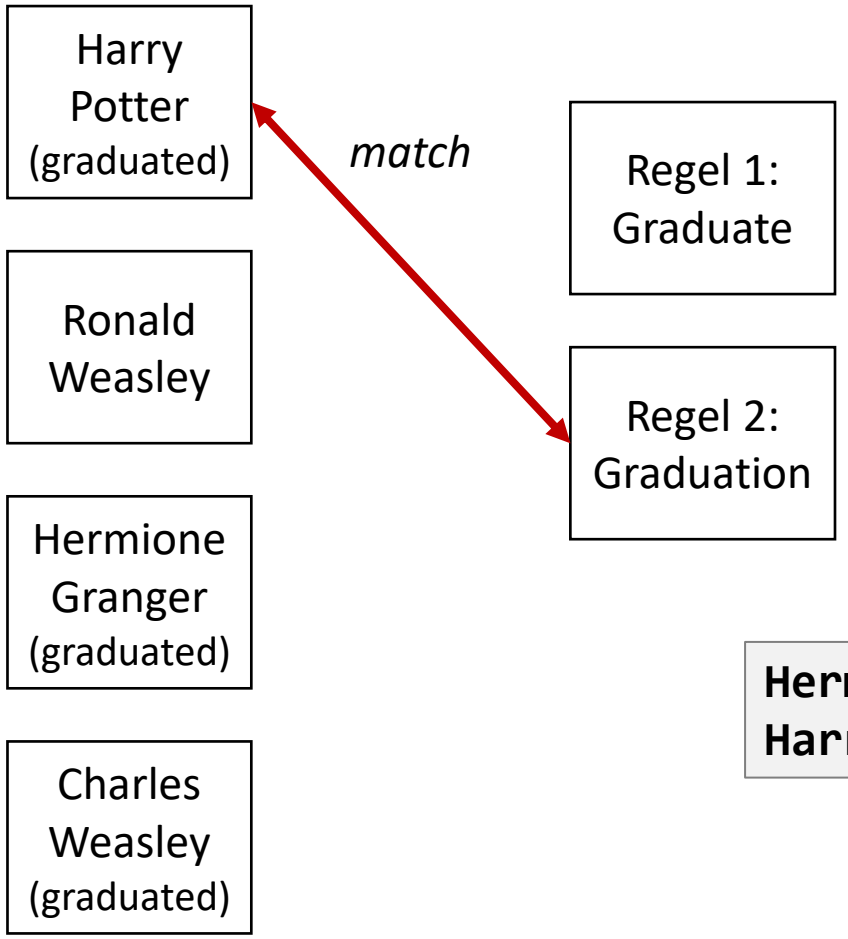
Regel 2: Graduation

Agenda:
- ~~Regel 1 (Hermione Granger)~~
- ~~Regel 1 (Harry Potter)~~
- Regel 2 (Hermione Granger)
- Regel 2 (Charles Weasley)

```
Hermione Granger is about to graduate.
Harry Potter is about to graduate.
```

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO



Agenda:
- ~~Regel 1 (Hermione Granger)~~
- ~~Regel 1 (Harry Potter)~~
- **Regel 2 (Harry Potter)**
- Regel 2 (Hermione Granger)
- Regel 2 (Charles Weasley)

```
Hermione Granger is about to graduate.
Harry Potter is about to graduate.
```

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Harry Potter (graduated)

Ronald Weasley

Hermione Granger (graduated)

Charles Weasley (graduated)

Regel 1: Graduate

Regel 2: Graduation

Agenda:
- ~~Regel 1 (Hermione Granger)~~
- ~~Regel 1 (Harry Potter)~~
- ~~Regel 2 (Harry Potter)~~
- Regel 2 (Hermione Granger)
- Regel 2 (Charles Weasley)

```
Hermione Granger is about to graduate.
Harry Potter is about to graduate.
Graduating Harry Potter.
```

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Harry Potter (graduated)

Ronald Weasley

Hermione Granger (graduated)

Charles Weasley (graduated)

Regel 1: Graduate

Regel 2: Graduation

Agenda:
- ~~Regel 1 (Hermione Granger)~~
- ~~Regel 1 (Harry Potter)~~
- ~~Regel 2 (Harry Potter)~~
- ~~Regel 2 (Hermione Granger)~~
- Regel 2 (Charles Weasley)

```
Hermione Granger is about to graduate.
Harry Potter is about to graduate.
Graduating Harry Potter.
Graduating Hermione Granger.
```

Beispiel: Studenten

Reihenfolge:
- Regeln Matching FIFO
- Daten Matching FIFO
- veränderte Daten Matching LIFO
- Regel Ausführung LIFO/FIFO

Harry Potter (graduated)

Ronald Weasley

Hermione Granger (graduated)

Charles Weasley (graduated)

Regel 1: Graduate

Regel 2: Graduation

Agenda:
- ~~Regel 1 (Hermione Granger)~~
- ~~Regel 1 (Harry Potter)~~
- ~~Regel 2 (Harry Potter)~~
- ~~Regel 2 (Hermione Granger)~~
- ~~Regel 2 (Charles Weasley)~~

```
Hermione Granger is about to graduate.
Harry Potter is about to graduate.
Graduating Harry Potter.
Graduating Hermione Granger.
Graduating Charles Weasley.
```

CARL VON OSSIETZKY universität OLDENBURG

IFTTT

- IF This Then That
- Kostenloser Web-Service, mit dem Ketten von bedingten Anweisungen erstellt werden können
- Oft genutzt für Kommunikation und Soziale Netzwerke
  - Gmail, Telegram, Facebook, Twitter, Instagram, …
- Oft genutzt für Smart Homes
- https://ifttt.com/

Beispiele: IFTTT (von der Webseite)

- Tweete automatisch meine Instagram-Bilder
- Schalte das Licht in der Einfahrt an, wenn der Pizza-Bote kommt
- Erinnere mich an meinen Regenschirm bevor ich losgehe während es regnet
- Wenn mir ein Youtube-Video gefällt, füge Lieder daraus zu meiner Spotify-Playlist hinzu
- Schalte mein Telefon auf stumm, wenn ich in die Bibliothek gehe
- Schalte Bluetooth und WLAN aus, wenn ich raus gehe
- Kopiere neue iOS-Kontakte in die Google-Kontakte

CARL VON OSSIETZKY universität OLDENBURG

Lernziele

- Regelsysteme
- JBoss Drools
- Rete-Algorithmus
- IFTTT