

2019

## Praktikum Betriebssicherheit

### Aufgabenblatt 2

#### Ereignisbaumanalyse

##### Anforderungen:

- Die Aufgabe wird in Python programmiert.
- Die Aufgabe wird von jedem Teilnehmer einzeln erstellt!
- Der Teilnehmer kommt zur Abnahme auf den Dozenten zu. Die Abnahme erfolgt für jeden Teilnehmer einzeln. Die Kenntnis des Quellcodes wird erwartet.
- Programmcode wird auf Ilias hochgeladen. Die Lokation wird im Praktikum bekanntgegeben. Das File hat folgendes Format:
  - <Name>\_<Vorname>\_<Matrikelnummer>\_Aufgabe\_2\_Programmcode.py
- **Es gibt eine Frist für die Abnahme, Abgabe des Berichts und Hochladen der Files. Diese wird im Praktikum bekanntgegeben.**

##### Einleitung

Die Ereignisbaumanalyse gibt Rückschlüsse auf Folgeereignisse wenn Anfangsereignisse eintreten. Der Baum kann aus einem Typ von Knoten erstellt werden, und zwar aus einem Zustandsknoten. Das initiiierende Ereignis expandiert auf die Folgeereignisse welche Zustandsänderungen hervorrufen. So kann die Wahrscheinlichkeit der Zustände ermittelt werden. Weitere Knoten stellen die XOR-Knoten dar, welche mehrere Folgeereignisse zusammenfassen können.

##### Aufgabe

Es soll mit Hilfe von dem Element Zustandsknoten ein Ereignisbaum implementiert werden. Dafür soll das Element als Klasse STATENODE definiert werden. Die Klasse STATENODE soll wiederum Listen als Attribute enthalten, die Objekte von weiteren STATENODEs aufnehmen können. Weiteres Element des Ereignisbaumes ist der XOR-Knoten.

Der folgende Ereignisbaum (Bild 1) soll mit einem Programm innerhalb des Praktikums modelliert werden:

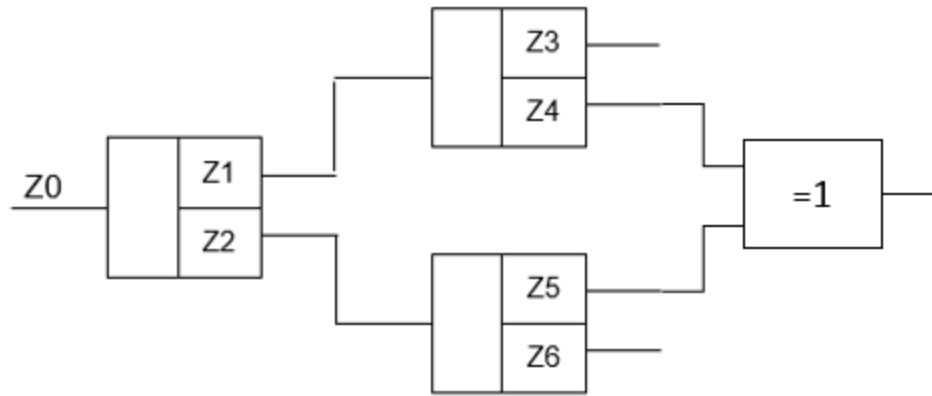


Bild 1: Ereignisbaum

Die Wahrscheinlichkeit für den Eintritt von Z0 ist 0.55. Die Eintrittswahrscheinlichkeiten der weiteren Zustände sind wie folgt:

$\Pr(Z1|Z0)=0,3$   
 $\Pr(Z2|Z0)=0,7$   
 $\Pr(Z3|Z1)=0,2$   
 $\Pr(Z4|Z1)=0,8$   
 $\Pr(Z5|Z2)=0,15$   
 $\Pr(Z6|Z2)=0,85$

Z0 ist das initiiierende Ereignis. Dieses kann auch aus dem STATENODE Klasse instanziiert werden. Z0 enthält die Zustände Z1 und Z2 etc. Die STATENODE Klasse soll folgende Struktur haben:

```

class STATENODE:
    def __init__(self,name):
        self.name = name #Name
        self.nodes=[] #Liste der Kinder
        self.parentattr=None #Vaterknoten
        self.prop=0 #Probability
    def add(self,node,prop):
        #.....
        self.nodes.append(node)
        return
    #.....
    def getprop(self):
        #.....
        return
  
```

Bild 2: Klasse STATENODE

- a.) Bauen Sie aus den instanziierten Objekten der definierten Verknüpfungselemente den Ereignisbaum aus Bild 1 auf. Verwenden Sie dabei die Methoden add(...) um darunterliegende Objekte an die Verknüpfungselemente anzuhängen. Beispiel:

Z0 = STATENODE('Z0')  
 Z1 = STATENODE('Z1')

```
Z2 = STATENODE(,Z2')
```

```
Z0.add(Z1,0.3)
```

```
Z0.add(Z2,0.7)
```

....

Die Methode `add(...)` soll das Kindelement und die Eintrittswahrscheinlichkeit als Parameter erhalten. Stellen Sie sicher, dass die Elemente doppelt verknüpft sind. Das bedeutet, dass von einem Kindelement (Z3, Z4 etc.) zum Vatorelement (Z1) referenziert werden kann. Siehe Bild 2 das Attribut *parentattr*.

- b.) Es soll aus dem Ereignisbaum die Wahrscheinlichkeit eines Zustands (z.B. Z3, Z4 etc.) bei Eintritt des Zustands Z0 ermittelt werden, und zwar mit der Methode:

```
Z4.getprop()
```

Zusätzlich soll die Ausgabe bei Eintritt von Z4 der folgenden String ausgeben:

```
Pr( Z4 | Z1 )*Pr( Z1 | Z0 )*Pr( Z0 )
```

- c.) Ergänzen Sie die Programmierung mit einer zusätzlichen XOR-Klasse, siehe Bild 3. Instanzen von XOR-Klassen dürfen nur an den Blättern des Ereignisbaumes mit der Methode `add()` angehängt werden. Die Methode `getprop()` berechnet die Wahrscheinlichkeit, dass einer der zugeordneten Ereignisse der XOR-Instanz eintritt.

```
class XORNODE:
    def __init__(self,name):
        self.name = name
        self.nodes=[]
    def add(self,node):
        #.....
        return
    def getname(self):
        return self.name
    def getprop(self):
        #.....|
        return prop
```

Bild 3: Klasse XORNODE

- d.) Programmieren Sie den Graphen mit Hilfe von Graphviz/Matplotlib und drucken Sie diesen aus, sodass die Darstellung eines Ereignisbaumes gleicht, siehe Bild 1.
- e.) Verändern Sie bei der Abnahme den Ereignisbaum und berechnen Sie in Anwesenheit des Dozenten die neuen Wahrscheinlichkeiten. Der gedruckte Graph soll sich dabei anpassen. Der Ereignisbaum wird vom Dozenten vorgegeben.