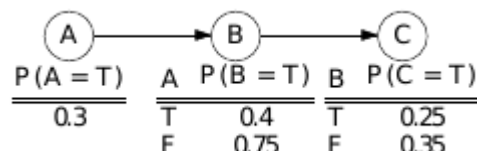


## Blatt 7 (8 Punkte)

Abgabe durch Hochladen auf der eCampus-Seite bis Sonntag, 05.06.2016, 23:59 Uhr, in Gruppen von 2-3 Personen.

### Aufgabe 7.1: Bayessche Netzwerke: Unabhängigkeitsannahmen

(1)



In der Vorlesung wurde besprochen, dass Bayessche Netzwerke durch die direkten Abhängigkeiten auch Unabhängigkeitsannahmen modellieren. Zeigen Sie für das obige Bayessche Netzwerk, ob  $P(A, C|B) = P(A|B)P(C|B)$  gilt. Nutzen Sie hierfür die durch die Topologie des Netzwerkes induzierte Faktorisierung der Verbundwahrscheinlichkeit, d.h.

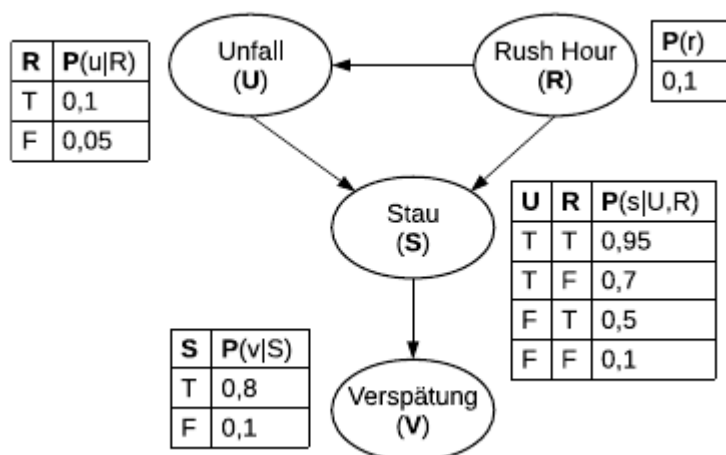
$$P(A, B, C) = \prod_{X \in \{A, B, C\}} P(X|\text{parents}(X))$$

oder geben Sie ein Gegenbeispiel an. Es sollte klar sein, dass ein Beweis für die Gültigkeit einer Formel **nicht** aus dem Rechnen eines Beispiels folgt!

### Aufgabe 7.2: Bayessche Netzwerke: Verspätung

(1 + 3 = 4)

Gegeben sei folgendes Bayessche Netzwerk:



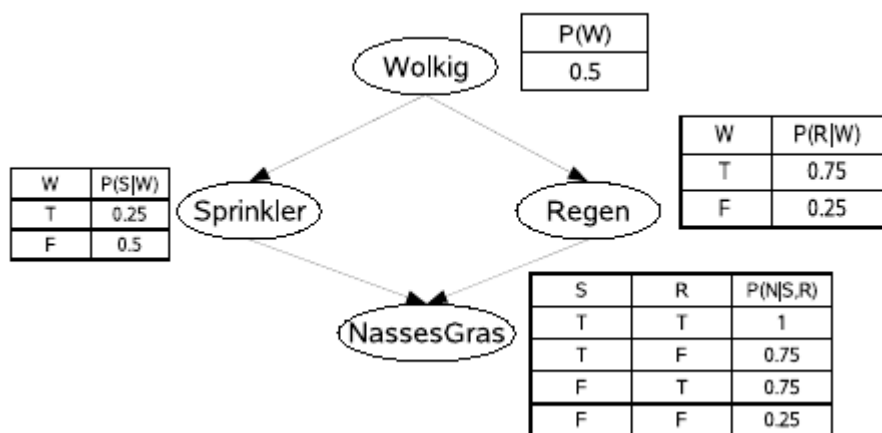
- Geben Sie die der Netztopologie entsprechende, in bedingte Wahrscheinlichkeiten faktorierte Schreibweise für  $P(V, S, U, R)$  an. Berechnen Sie mit Hilfe der CPTs die Wahrscheinlichkeiten für  $P(\neg v, s, u, r)$  und  $P(\neg v, \neg s, \neg u, \neg r)$ .
- Berechnen Sie mittels exakter Inferenz durch Aufzählen die Anfrageverteilung *Verspätung* gegeben die Evidenzen *Unfall* und *Rush Hour*, d.h  $P(V|u, r)$ . Es ist also eine vollständige Fallanalyse durchzuführen (s. Vorl. 10, Folien 29 ff. sowie Vorl. 11, Folien 18 - 20). Nutzen Sie für einfacheres Rechnen eine Normierungskonstante (s. Vorl. 10, Folie 30).

Beachten Sie, dass von Aufgaben 7.3 und 7.4 nur eine auszusuchen ist, die abgegeben werden kann/muss. Es wird dementsprechend nur eine Aufgabe bepunktet korrigiert. Bei Abgabe beider Aufgaben bitte kenntlich machen, welche bewertet werden soll. Erfolgt in diesem Fall keine solche Kennzeichnung, wird Aufg. 7.3 bewertet.

### Aufgabe 7.3: Sampling Methoden

(entweder hier  $1 + 0,5 + 1,5 = 3$ )

Das folgende Bayessche Netzwerk ist eine modifizierte Version des Sprinkler-Netzwerkes aus der Vorlesung:



- a) i) Simulieren Sie einen Zufallsprozess mit einer Münze, die Sie zweimal werfen. Um die Wahrscheinlichkeiten abzuschätzen, benutzen Sie folgenden Fakten: die Wahrscheinlichkeit, dass der *erste* Wurf *Zahl* zeigt, beträgt 50%, dass beide Würfe *Zahl* zeigen 25% und dass mindestens einer der Würfe *Zahl* zeigt 75%. Erzeugen Sie jetzt anhand des Algorithmus PRIOR-SAMPLE drei Samples (wie z.B. [true, true, false, true]), indem Sie jede Variable in der Reihenfolge  $[W, S, R, N]$  (und zwar genau und nur in dieser Reihenfolge) über die folgenden drei Doppelwurfsequenzen sampeln:
- 1)  $[Zahl, Kopf], [Zahl, Kopf], [Kopf, Zahl], [Kopf, Kopf]$ ,
  - 2)  $[Kopf, Kopf], [Zahl, Zahl], [Kopf, Zahl], [Zahl, Kopf]$ ,
  - 3)  $[Kopf, Zahl], [Zahl, Kopf], [Zahl, Zahl], [Kopf, Zahl]$ .
- ii) Schätzen Sie aus diesen (sehr unrepräsentativen) drei Stichproben die drei Wahrscheinlichkeiten  $P(s)$ ,  $P(r)$  und  $P(n)$ .
- iii) Gegen welche Wahrscheinlichkeiten sollten die Werte bei größeren Mengen von Samples konvergieren? Schreiben Sie diese Wahrscheinlichkeiten lediglich formal als faktorisierte Verbundwahrscheinlichkeiten auf ohne sie auszurechnen.
- b) Wie groß ist die Wahrscheinlichkeit, das Sample  $[\neg w, s, \neg r, n]$  zu erhalten?
- c) Betrachten Sie jetzt den Algorithmus LIKELIHOOD-WEIGHTING und gehen Sie davon aus, dass die Evidenz  $\neg r$  (kein Regen) gegeben ist. Benennen Sie alle möglichen Samples, die vorkommen können, und geben Sie die zugehörige Gewichtung an, die dem Sample zugeordnet wird. Zeigen Sie für das Sample  $[w, s, \neg r, n]$  explizit, wie sich das Gewicht  $w$  in der Funktion WEIGHTED-SAMPLE beim Besuch eines jeden Knotens im Netz ändert.

#### Aufgabe 7.4: Programmieraufgabe: Rasensprenger

(oder hier 3)

**Vorbereitung:** Laden Sie das ZIP-Archiv `bn.zip` herunter von unserer eCampus-Seite unter Kursunterlagen >> Python und AIMA Python >> Bayesian Networks. Das ZIP-Archiv enthält den Ordner *bayes* mit dem Skript *prior\_sampling.py* sowie dem vorgegebenen Bayessches Netzwerk *gras\_wet.bayes*. Fügen Sie den Ordner *bayes* auf der gleichen Ebene wie den Ordner *aima* ein.

#### Aufgabe:

- a) Erstellen Sie auf der Basis des gegebenen Skripts *prior\_sampling.py* ein neues Skript *reject\_sampling.py*, welches das Rejection-Sampling umsetzt.
- b) Schätzen Sie nun mit Hilfe Ihres neu erstellten Skripts *reject\_sampling.py* die Verteilung  $P(\text{Gras\_wet} | \text{Cloudy} = \text{true})$  über 100 Iterationen (diese Einstellungen können Sie über den Properties-Dialog der GUI einstellen).

Die Punkte werden für die Lösung von Teil a) vergeben. Es werden jedoch keine Punkte für Teil a) vergeben ohne vernünftigen Abschluss durch zumindest den einen in Teil b) geforderten Testlauf und die entsprechende Antwort auf Teil b).

**Hinweis:** Zum Thema MDP wird es auf Blatt 8 eine Aufgabe geben.