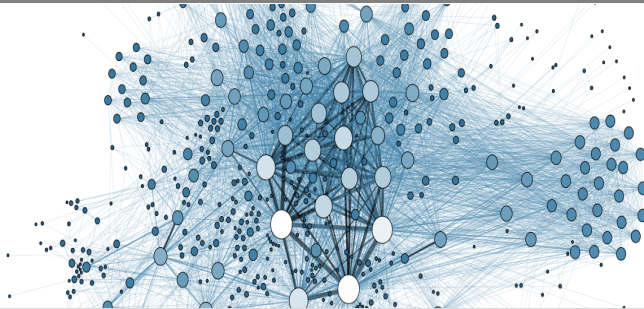


Grundbegriffe der Informatik

Tutorium 38

Relationen, Altklausur

Patrick Fetzer, uxkln@student.kit.edu | 07.02.2018



Schon bekannte Eigenschaften von Relationen:

- transitiv ($\forall x, y, z \in M : (x, y) \in R \wedge (y, z) \in R \rightarrow (x, z) \in R$)
- reflexiv ($\forall x \in M : \rightarrow (x, x) \in R$)
- symmetrisch ($\forall x, y \in M : (x, y) \rightarrow (y, x) \in R$)
- Äquivalenzrelation ist reflexiv, transitiv und symmetrisch

Neue Eigenschaft:

- antisymmetrisch ($\forall x, y \in M : (x, y) \in R \wedge (y, x) \in R \rightarrow x = y$)
- Halbordnung ist reflexiv, transitiv und antisymmetrisch

Welche Eigenschaften sind hier erfüllt?

■ \subset

■ $<$

■ \leq

■ Relation R auf A^* mit $v R w \leftrightarrow \exists u : vu = w$

Auflösung des Mastertheorem

Fall 1: Wenn $f \in \mathcal{O}(n^{\log_b a - \varepsilon})$ für ein $\varepsilon > 0$ ist, dann ist
 $T \in \Theta(n^{\log_b a})$.

Fall 2: Wenn $f \in \Theta(n^{\log_b a})$ ist, dann ist $T \in \Theta(n^{\log_b a} \log n)$.

Fall 3: Wenn $f \in \Omega(n^{\log_b a + \varepsilon})$ für ein $\varepsilon > 0$ ist, und wenn es eine Konstante d gibt mit $0 < d < 1$, so dass für alle hinreichend großen n gilt $af(n/b) \leq df$, dann ist $T \in \Theta(f)$.

```
funktion mergesort(liste);  
  falls (Größe von liste <= 1) dann antworte liste  
  sonst  
    halbiere die liste in linkeListe, rechteListe  
    linkeListe = mergesort(linkeliste)  
    rechteListe = mergesort(rechteListe)  
    antworte merge(linkeliste, rechteListe)
```

```
funktion merge(linkeliste, rechteListe);  
  neueListe  
  solange (linkeliste und rechteListe nicht leer)  
    | falls (erstes Element der linkeliste <= erstes Element der rechteListe)  
    | dann füge erstes Element linkeListe in die neueListe hinten ein und entferne es aus linkeliste  
    | sonst füge erstes Element rechteListe in die neueListe hinten ein und entferne es aus rechteListe  
  solange_ende  
  solange (linkeliste nicht leer)  
    | füge erstes Element linkeliste in die neueListe hinten ein und entferne es aus linkeliste  
  solange_ende  
  solange (rechteListe nicht leer)  
    | füge erstes Element rechteListe in die neueListe hinten ein und entferne es aus rechteListe  
  solange_ende  
  antworte neueListe
```



That's all Folks!