

Darstellung von und Algorithmen zu Graphen

Teil 2

7.07.05

Vorlesung Bioinformatik II

Studiengang Molekulare Biotechnologie

Rainer König

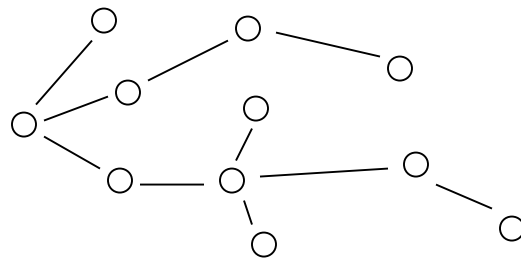
Uni-Heidelberg

1

3. Tiefensuche

3.1 Prinzip, Unterschiede und Gemeinsamkeiten zur Breitensuche

- gehe in den Graph so weit wie möglich



2

- es entstehen Wälder: wenn bei einem Vertex alles abgegrast ist, wird bei einem anderen von Neuem gestartet

- Färben: weiss: unentdeckt
 grau: entdeckt, noch "aktiv"
 schwarz: beendet, alle Nachbarn abgearbeitet

3

- neu: Zeitpunkte und Zeitstempel werden definiert

- Anfang: $t = 0$

- Zeit := Zeitschritte:

jedesmal, wenn ein neuer Vertex entdeckt wird, (weiss -> grau)

oder beendet wird (grau -> schwarz), wird die Zeit um eins erhöht

- jeder Vertex bekommt die

Entdeckungszeit $d(t)$, und die

Beendungszeit $f(t)$

- diese Zeitpunkte sind für viele Anwendungen wichtig (wir lernen eine)

4

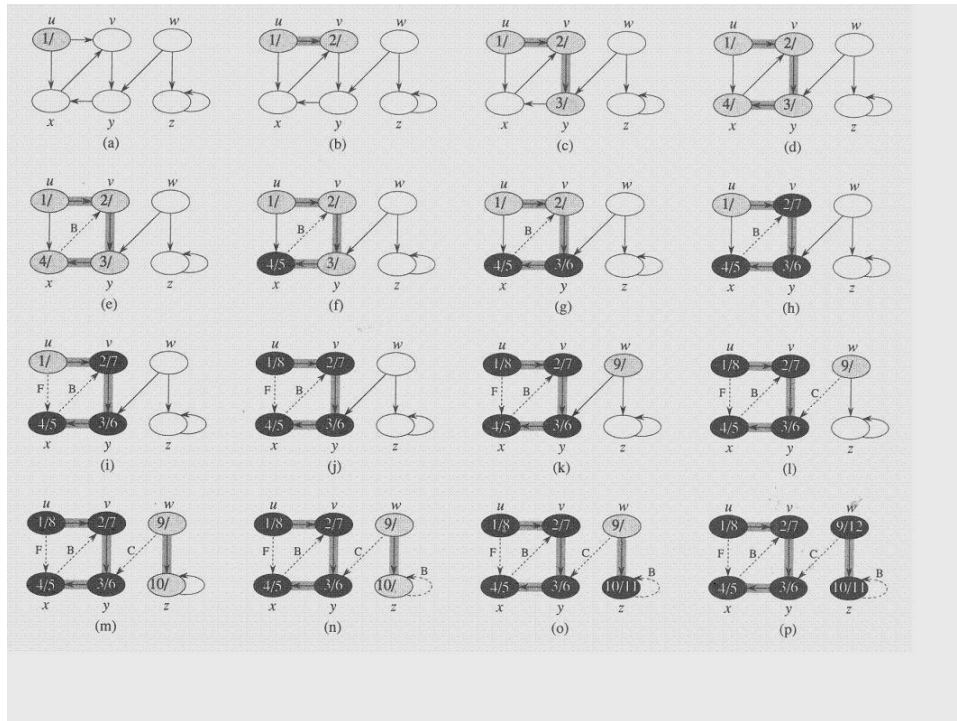
3.2 Algorithmus

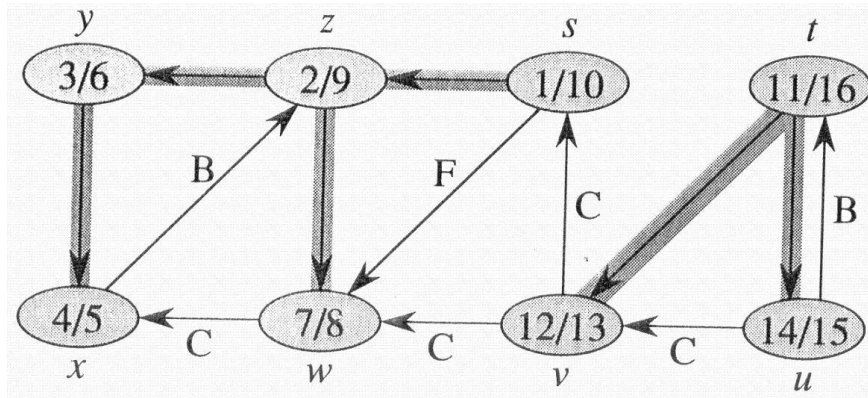
DFS(G)

- 1 **for** each vertex $u \in V[G]$
- 2 **do** $color[u] \leftarrow \text{WHITE}$
- 3 $\pi[u] \leftarrow \text{NIL}$
- 4 $time \leftarrow 0$
- 5 **for** each vertex $u \in V[G]$
- 6 **do if** $color[u] = \text{WHITE}$
- 7 **then** DFS-VISIT(u)

DFS-VISIT(u)

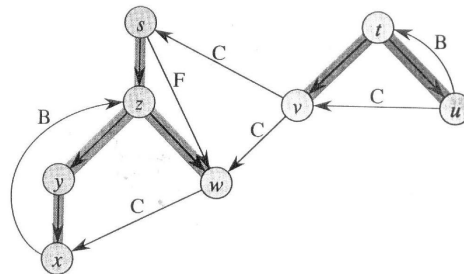
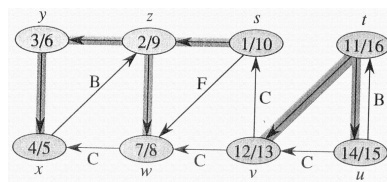
- 1 $color[u] \leftarrow \text{GRAY}$ \triangleright White vertex u has just been discovered.
- 2 $d[u] \leftarrow time \leftarrow time + 1$
- 3 **for** each $v \in Adj[u]$ \triangleright Explore edge (u, v) .
- 4 **do if** $color[v] = \text{WHITE}$
- 5 **then** $\pi[v] \leftarrow u$
- 6 DFS-VISIT(v)
- 7 $color[u] \leftarrow \text{BLACK}$ \triangleright Blacken u ; it is finished.
- 8 $f[u] \leftarrow time \leftarrow time + 1$





3.3 Eigenschaften

- Tiefensuchbäume bilden Wald



Bezeichnungen im unteren Graphen:

- dicke Kanten = Baum-Kanten

(gehen immer abwärts)

- B = Back/Rückwärtskanten,

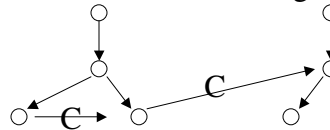
verbinden einen Vertex mit einem seiner Vorfahren

- F = Foreward/Vorwärtskante

Nicht-Baum-Kante, die einen Vertex mit einem seiner Nachkommen verbindet

- C = Cross-Kante

Sonstige Kante, i.e. Kante zwischen Vertices im gleichen Baum
oder zwischen 2 Bäumen



9

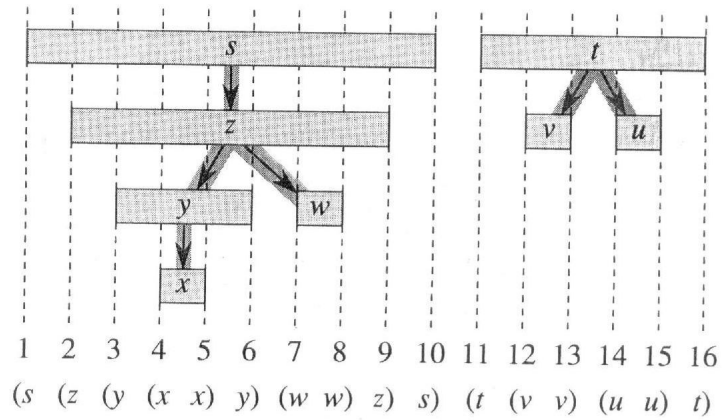
- im ungerichteten Graphen gibt es keine Sonstigen-Kanten

□

(hier ist Rückwärts bzw. Vorwärtskante Definitionssache, man entscheidet sich für Rückwärts)

10

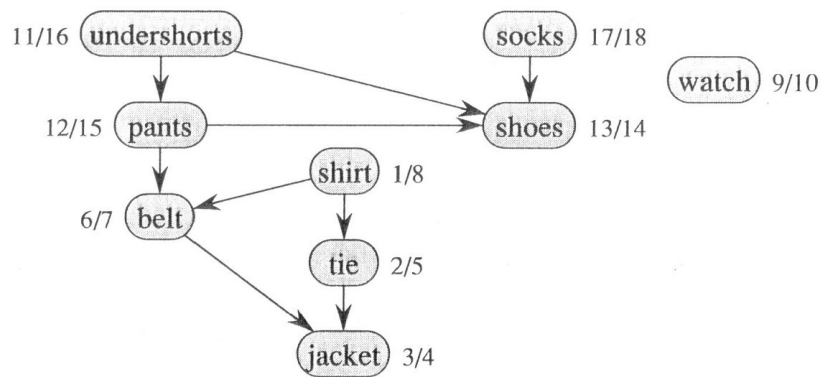
- Klammern-Struktur



- Nachkommen werden später
 entdeckt und früher beendet als
 ihre Vorfahren

11

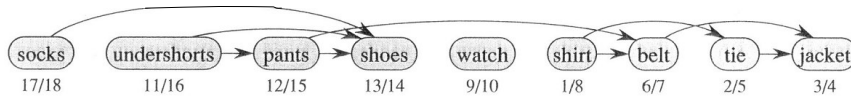
3.3 Topologisches Sortieren



In welcher
 Reihenfolge darf er
 sich anziehen?

- geg. DAG

- topologisches Sortieren ordnet linear alle Vertices, s.d. es nur Rechts-Pfeile gibt



13

Algorithmus:

- lasse Tiefensuchalgorithmus laufen
- gebe Vertices nach Beendigungszeit geordnet aus

14

