Graphen coden statt zeichnen mit graphviz

Tobias Rehbein / blabber

07. Mai 2013

Philosophie

- Knoten und ihre Beziehungen zueinander werden in einer Texdatei beschrieben
- Graphviz berechnet die optimale Anordung der Knoten und Krümmung der Kanten

Features

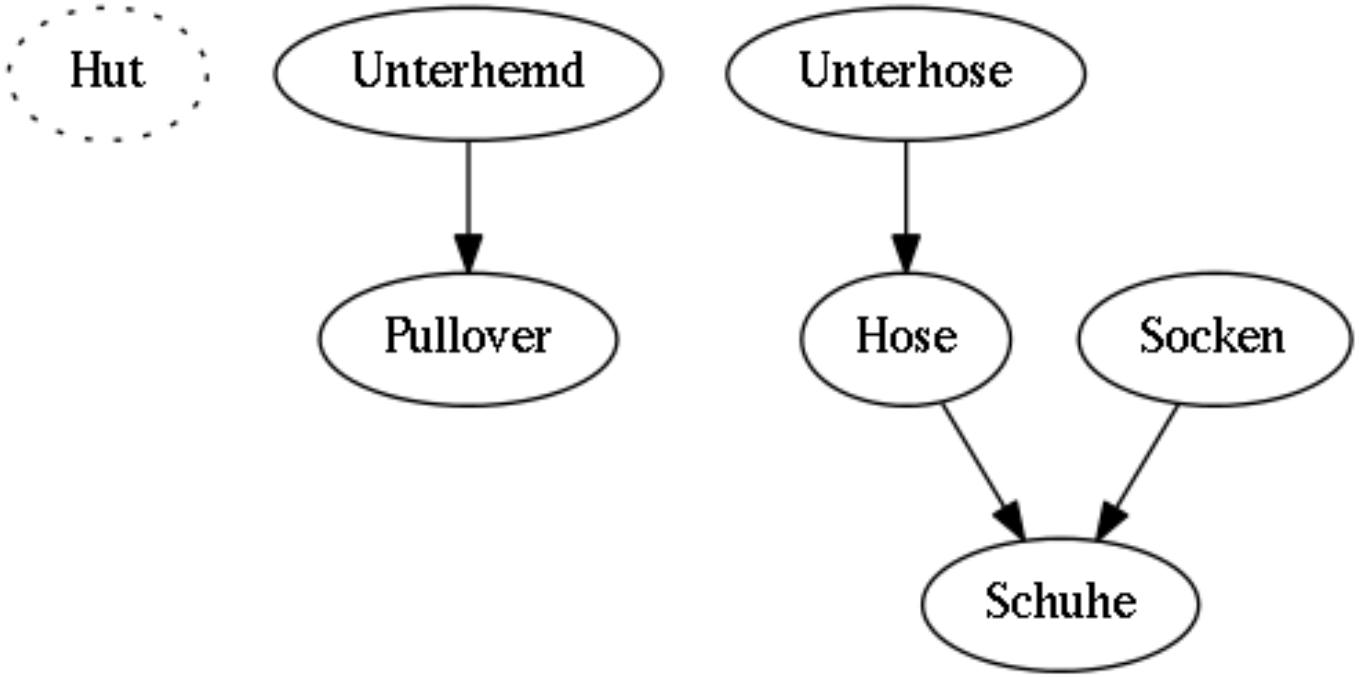
- Eigene Beschreibungssprache für Graphen (dot)
- Layout von gerichteten und ungerichteten Graphen
- Ausgabe von generierten Graphen in verschiedensten Grafikformaten
- Eigene Skriptsprache zur Weiterverarbeitung von Graphen (gvpr)

Graphen gerichtet

```
digraph Anziehen {
    Hut [style=dotted];
    Unterhemd -> Pullover;
    Unterhose -> Hose;
    Socken -> Schuhe;
    Hose -> Schuhe;
}

// dot -Tpng -o anziehen.png anziehen.dot
```

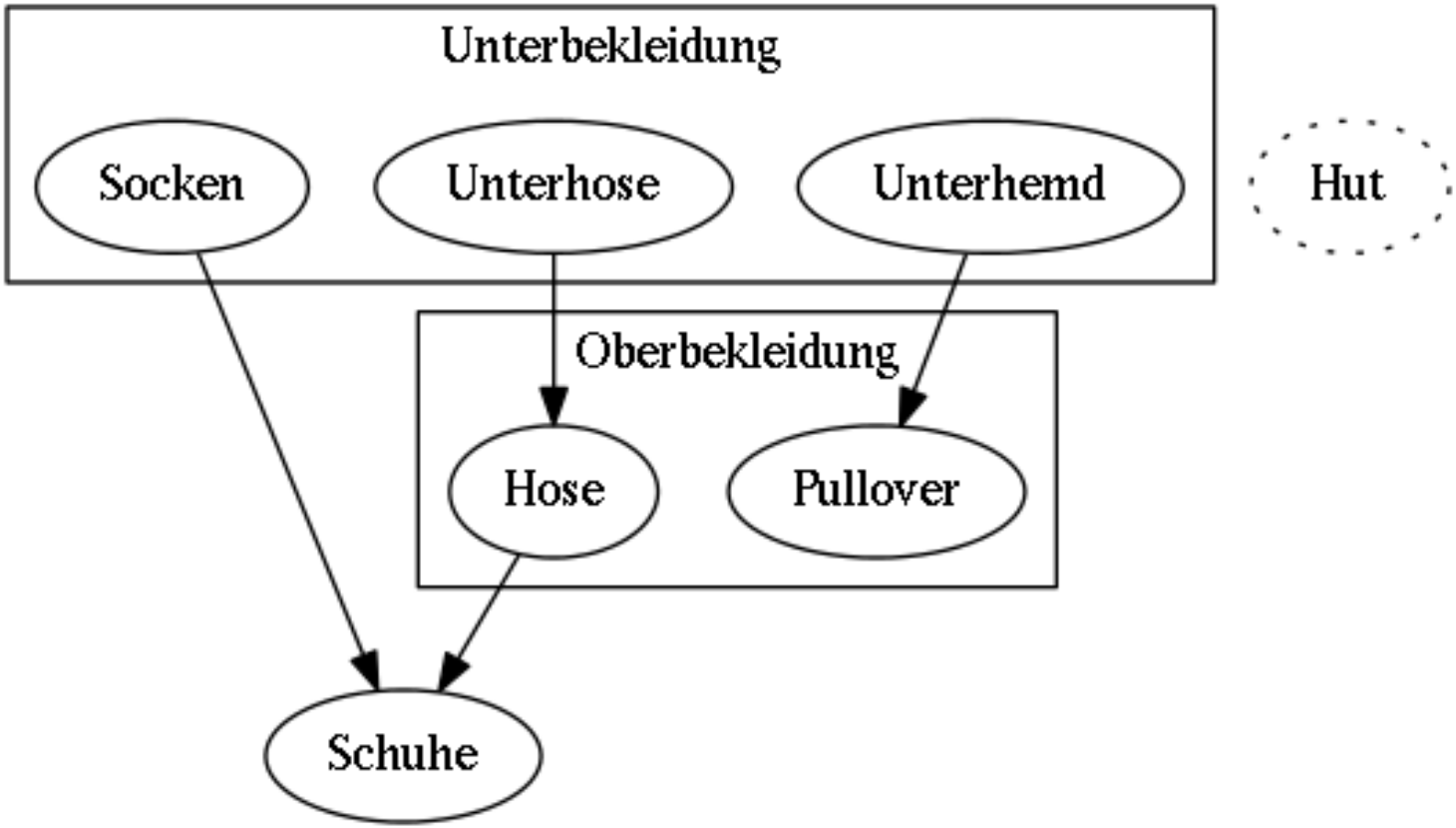
Graphen ungerichtet



Graphen gerichtet (Cluster)

```
digraph Cluster {
    subgraph cluster_oberbekleidung {
     label="Oberbekleidung";
     Pullover; Hose;
    }
    subgraph cluster_unterbekleidung {
     label="Unterbekleidung";
     Socken; Unterhemd; Unterhose;
    }
    Hut [style=dotted];
    Unterhemd -> Pullover;
    Unterhose -> Hose;
    Socken -> Schuhe;
    Hose -> Schuhe;
}
```

Graphen gerichtet (Cluster)

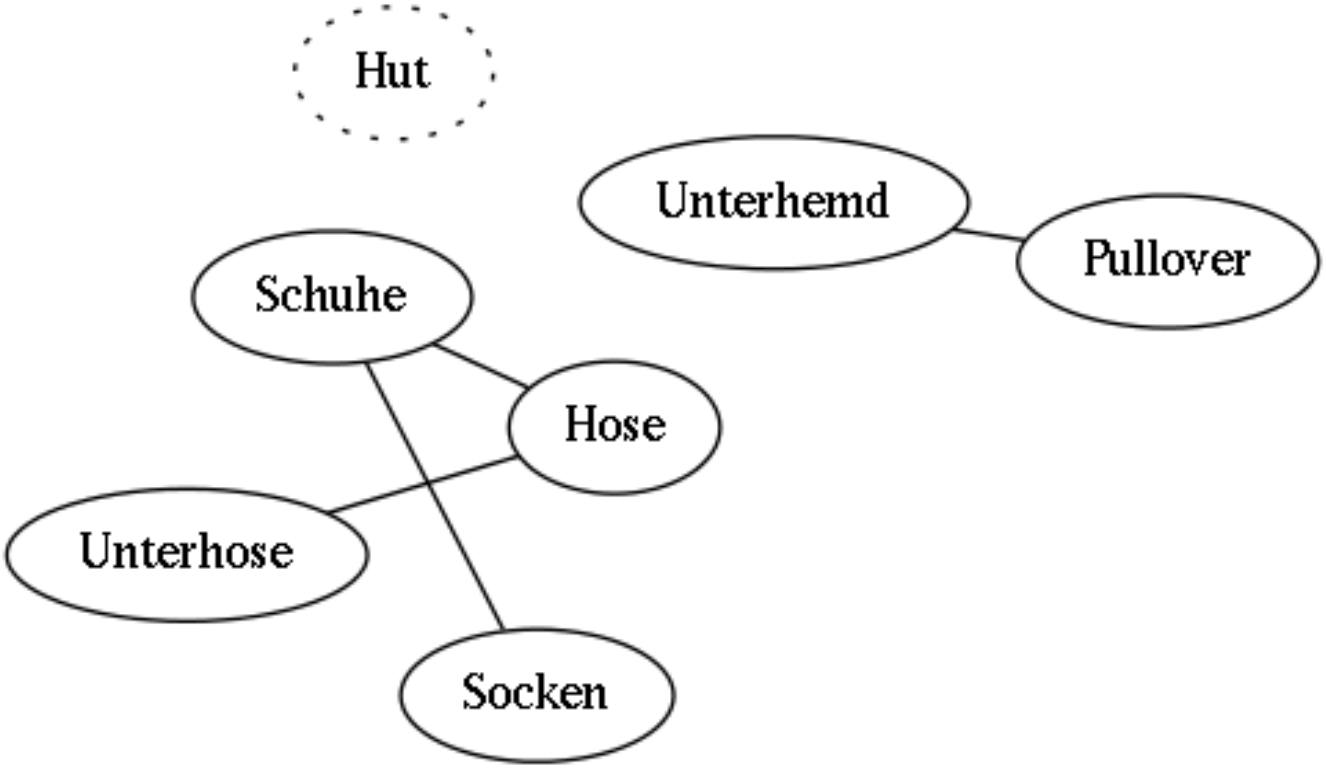


Graphen ungerichtet

```
graph Kleidung {
    Hut [style=dotted];
    Unterhemd -- Pullover;
    Unterhose -- Hose;
    Socken -- Schuhe;
    Hose -- Schuhe;
}

// fdp -Tpng -o kleidung.png kleidung.dot
```

Graphen ungerichtet



Paketabhängigkeiten

- Graph der alle Pakete meines Laptops und die Abhängigkeiten beschreibt
- Die Dotdatei hat 5594 Zeilen
- Das Rendern dauert 10 Minuten
- Auf dem gerenderten Graphen ist nichts zu erkennen

gvpr

- gvpr bietet die Möglichkeit Dotdateien skriptbasiert weiterzubearbeiten
- Die Syntax von gvpr ist eine Mischung aus awk und C

Beispiel 1: Statistik

```
BEG_G {
    int n, e;

    n = nNodes($G);
    e = nEdges($G);

    printf("%d nodes %d edges in graph \"%s\"\n",
        n, e, $G.name);
}

// gvpr -f stats.gvpr dependencies.dot
```

Beispiel 1: Statistik

```
486 nodes 5163 edges in graph "dependencies"
```

Beispiel 2: Subgraph extrahieren (1/3)

```
BEGIN {
    graph_t g;

    void childs(node_t n, edge_t o) {
        for (o = fstout(n); o != NULL;
             o = nxtout(o)) {
            clone(g, o);
            childs(o.head, NULL);
        }
    }
```

Beispiel 2: Subgraph extrahieren (2/3)

```
    void parents(node_t n, edge_t i) {
        for (i = fstin(n); i != NULL; i = nxtin(i))
            clone(g, i);
            parents(i.tail, NULL);
        }
    }
} // BEGIN
```

Beispiel 2: Subgraph extrahieren (3/3)

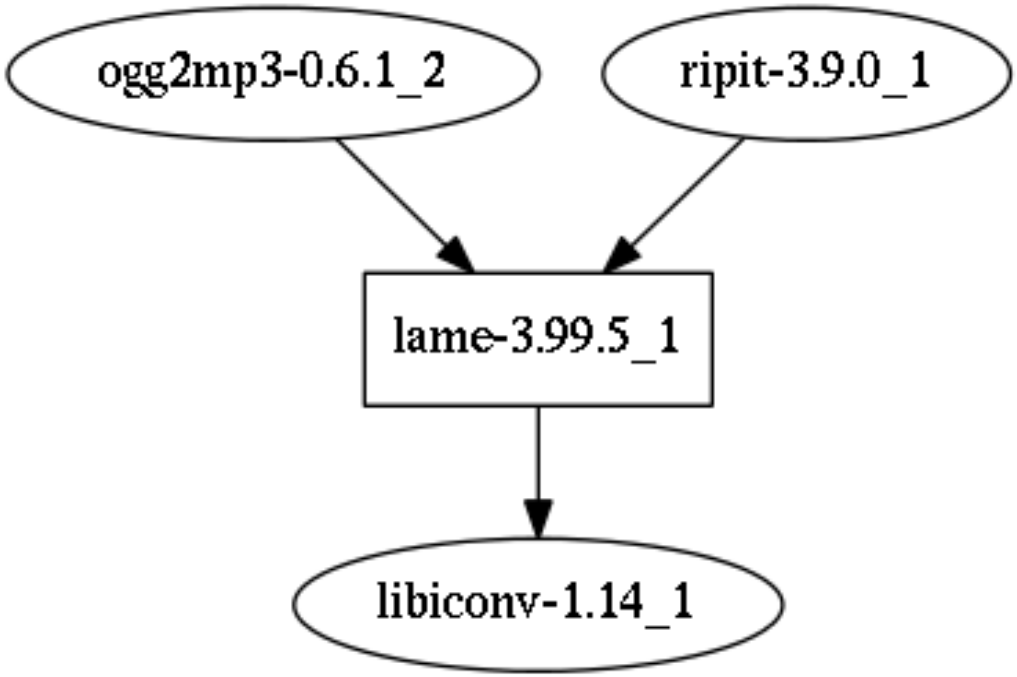
```
BEG_G { g = graph(ARGV[0], "S"); }

N [name == ARGV[0]] {
        $.shape = "box";
        clone(g, $);
        childs($, NULL);
        parents($, NULL);
}

END_G { $O = g; }

// gvpr -f extract.gvpr -a lame-3.99.5_1 \
//   dependencies.dot | dot -Tpng -o lame.png
```

Beispiel 2: Subgraph extrahieren



Vielen Dank für eure Aufmerksamkeit

Vielen Dank für eure Aufmerksamkeit!
Mehr Infos und Beispiele findet ihr hier: http://www.graphviz.org/