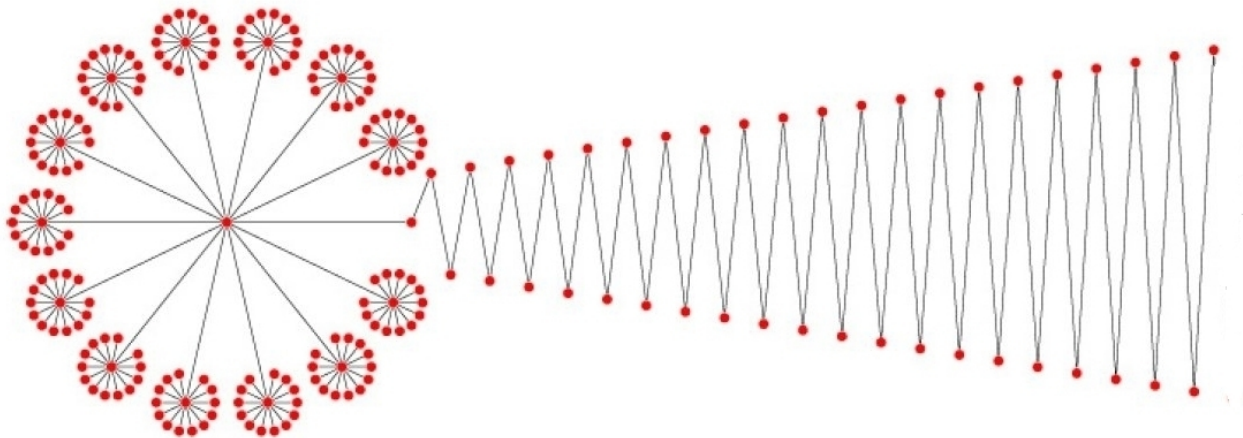


Übungen zur Vorlesung  
**Effiziente kombinatorische Algorithmen**  
 WS 2019/20  
 Blatt 9

**AUFGABE 21:**

Der folgende Graph ist der 15-Komet  $Q_{15}$ .



Für  $k \geq 2$  besteht der  $k$ -Komet  $Q_k$  aus

$$n = \underbrace{(k-1)(k-2) + 1}_{\text{Kopf}} + \underbrace{3(k-1)}_{\text{Schweif}} = k^2$$

Knoten.

- (a) Welchen Grad und wieviele Kanten in Abhängigkeit von  $k$  hat  $Q_k$ ? Aus wievielen Knoten besteht eine kleinste Knotenüberdeckung?

*Hinweis:* Die Frage nach der Anzahl der Kanten ist sehr einfach zu beantworten, wenn Sie sich klarmachen, zu welcher Familie von Graphen  $Q_k$  gehört.

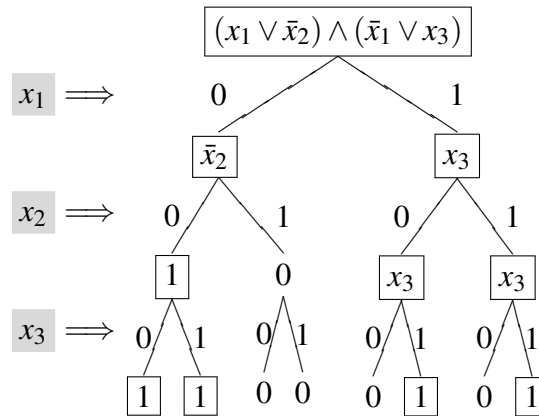
- (b) Wie verhält sich der exakte allgemeine Algorithmus MVC, wenn er mit  $Q_k$  gestartet wird?

**AUFGABE 22:**

In dieser Aufgabe entwickeln wir einen ersten deterministischen Algorithmus für das Erfüllbarkeitsproblem SAT, oder genauer, für seine Variante  $k$ -SAT, dessen Laufzeit „besser“ als  $2^n$  ist. Mathematisch korrekt muß es heißen: Seine Laufzeit ist  $o(2^n)$ .

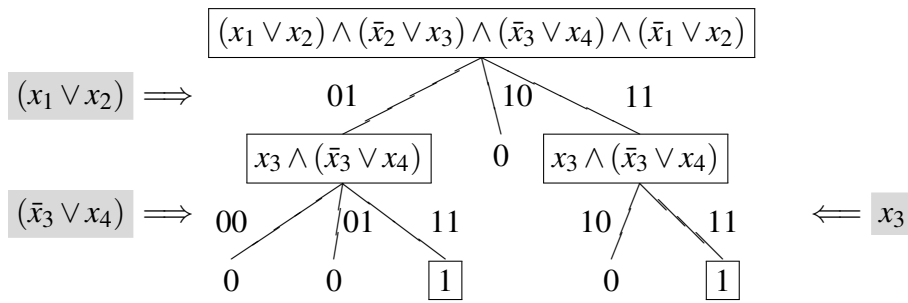
Sei also  $\Phi$  eine Boolesche Formel in Konjunktiver Normalform über den  $n$  Variablen  $\{x_1, \dots, x_n\}$ , in der jede der  $m$  Klauseln  $C_1, \dots, C_m$  aus höchstens  $k$  Literalen besteht. Wenn Sie eine konkrete Belegung der Variablen haben, dann kann  $\Phi$  in Zeit  $O(n + k \cdot m)$  ausgewertet werden.

Den Algorithmus, der alle möglichen Belegungen durchprobiert, kann man mit Hilfe eines sog. Berechnungsbaums darstellen. Im folgenden Beispiel ist eine KNF  $\Phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_3)$  für 2SAT und der zugehörige Berechnungsbaum angegeben. Ein Kasten bedeutet, daß die Formel in ihm erfüllbar ist.



Die Laufzeit des Verfahrens ist im ungünstigsten Fall, wenn der gesamte Baum durchprobiert werden muß,  $O((n + k \cdot m) \cdot 2^n) = O^*(2^n)$ .

Betrachten Sie nun den folgenden, „schlaueren“ Berechnungsbaum (für eine andere Formel).



Hier wird an einem *Knoten* eine Klausel  $C$  genommen und die eine Belegung der (Teil-)Länge  $k$ , die  $C$  nicht erfüllt, von der Betrachtung ausgeschlossen, da diese (Teil-)Belegung nicht Teil einer erfüllenden Belegung sein kann. Enthält  $C$  weniger als  $k$  Literale, so werden die zu belegenden Variablen durch noch unbelegte Variablen „aufgefüllt“.

- Beschreiben Sie den Algorithmus konkret.
- Wieviele Level hat der Berechnungsbaum im Worst Case, wenn man mit einer Formel mit je  $k$  Literalen je Klausel beginnt?
- Wie ist die Laufzeit des Algorithmus im Worst Case in Abhängigkeit von  $m$ ,  $n$  und  $k$ ? Was kommt z. B. für  $k = 3$  konkret heraus?

*Hinweis:* Geometrische Reihe  $\sum_{i=0}^t a^i = \frac{a^{t+1} - 1}{a - 1} = \Theta(a^t)$