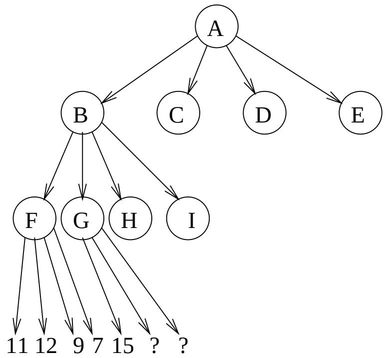
Entwurf und Analyse von Algorithmen

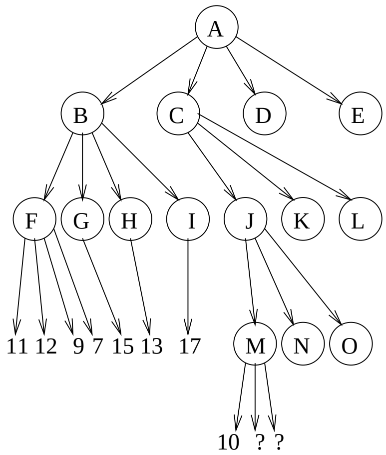
WS 1999/2000
Prof. Dr. K. Madlener
Übungsblatt 10

Aufgabe 10.1. a) Die Alpha-Beta-Prozedur aus der Vorlesung besteht aus den Funktionen *AlphaBetaMax* und *AlphaBetaMin*. Modifizieren Sie das Programm für *AlphaBetaMax* so, dass die Funktion *AlphaBetaMin* nicht mehr benötigt wird.

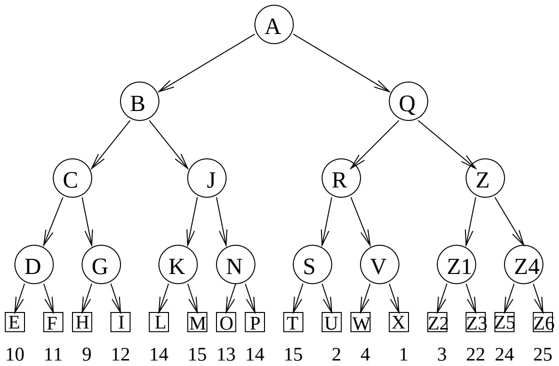
b) Wir betrachten einen Spielbaum während des Ablaufs der Alpha-Beta-Prozedur aus der Vorlesung. Warum kann bei folgender Situation auf die weitere Expansion des Knotens G (d. h. auf die weitere Betrachtung seiner Söhne) verzichtet werden?



c) Warum kann bei folgender Situation auf die weitere Expansion des Knotens M verzichtet werden?

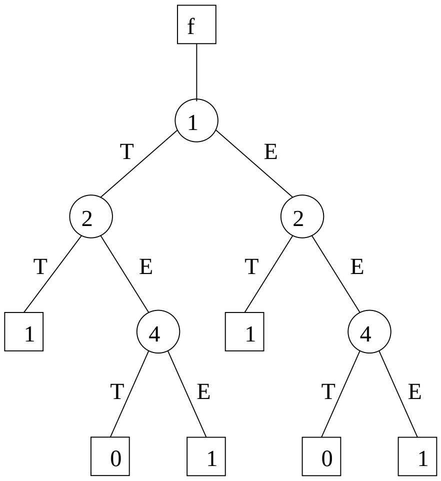


d) Führen Sie die Alpha-Beta-Prozedur aus der Vorlesung für den folgenden Spielbaum aus und kommentieren Sie ausführlich:



Aufgabe 10.2. a) Geben Sie jeweils BDDs für die Boole'schen Funktionen $\wedge$ (Konjunktion), $\vee$ (Disjunktion), $\rightarrow$ (Implikation), $\oplus$ (exklusives Oder) und $\leftrightarrow$ (Äquivalenz) an.

b) Die Boole'sche Funktion f sei durch folgendes BDD gegeben:



Geben Sie ein entsprechendes reduziertes BDD an. Geben Sie auch eine disjunktive Normalform für f an.

c) Finden Sie eine „gute" Ordnung der Variablen a, b, c und d für die Boole'sche Funktion $f = abc + bc\overline{d}$ und erklären Sie, warum Ihre Ordnung gut ist. Zeichnen Sie das entsprechende geordnete reduzierte BDD. Wiederholen Sie dies für eine „schlechte" Ordnung der Variablen.

Aufgabe 10.3. Wir wollen noch einmal die Greedy-Heuristik für das 0/1-Rucksack-Problem betrachten, die versucht, Elemente (nach nicht-steigender Profitdichte geordnet) in den Rucksack aufzunehmen, wenn sie noch passen; ansonsten wird das Element ignoriert und das nächste probiert. Zeigen Sie, dass diese Heuristik beliebig schlechte Ergebnisse liefern kann, d. h. zeigen Sie, dass es für jedes $k \in \mathbb{N}$ eine Eingabe für diesen Algorithmus gibt, so dass der optimale Profit mindestens k-mal größer ist als der Profit, der mit diesem Algorithmus erreicht wird. Mit welchem einfachen weiteren Schritt kann das Ergebnis dieses Algorithmus so verbessert werden, dass die erzielte Näherungslösung für jede Eingabe nur noch um einen Faktor kleiner als 2 vom optimalen Profit abweicht?

Informationen zur Vorlesung:
http://www-madlener.informatik.uni-kl.de/ag-madlener/lectures/eaa/ws1999-2000/eaa.html