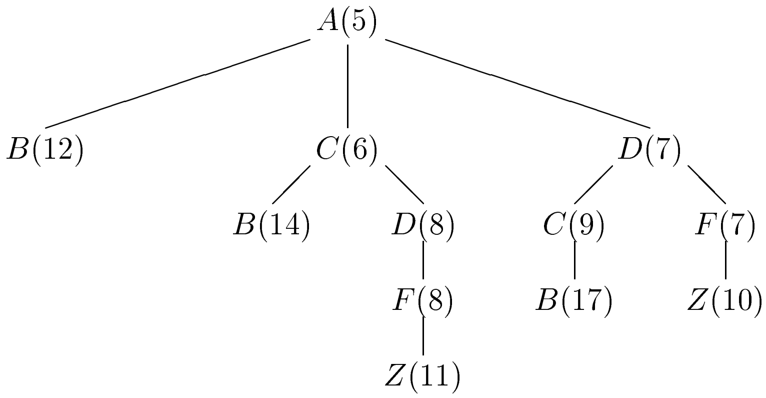
Uitwerking tentamen Kunstmatige intelligentie vrijdag 1 juni 2001

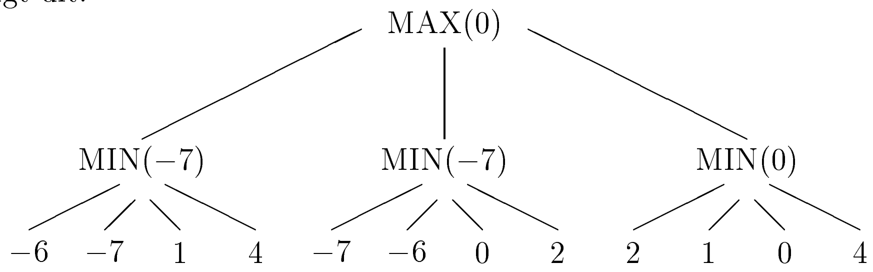
Voor een meer uitgebreide uitwerking van enkele (onderdelen van) opgaven wordt verwezen naar het boek van Russell en Norvig, en voor de opgave over het Apriori-algoritme naar `http://www.liacs.nl/home/kosters/AI/datam.html`.

Opgave 1:
Uiteindelijk wordt onderstaande boom opgebouwd. Knopen die al eerder op een pad voorkwamen worden niet aangegeven, er zijn dus geen kringen (bijvoorbeeld onder $C(6)$ komt niet opnieuw A te hangen). Tussen haakjes staat naast de knopen hun f-waarde. Achtereenvolgens worden geopend = ontwikkeld = geexpandeerd $A(5)$, $C(6)$, $D(7)$, $F(7)$, $D(8)$ (let op, hoewel al een doelknoop gevonden is, worden eerst nog knopen met lagere f-waarde ontwikkeld), $F(8)$, $C(9)$ en tot slot $Z(10)$. Bij $F(8)$ staat dankzij de *pathmax equation* 8 in plaats van 7, bij $F(7)$ staat evenzo 7 in plaats van 6.



De effectieve vertakkingsgraag b^* laat zich berekenen uit $1 + b^* + (b^*)^2 + \cdots + (b^*)^d = N$, met $d = 3$ de diepte van de oplossing, en $N = 8$ het aantal geopende knopen.

Opgave 2:
Denk er aan de waardes in de bladeren (in dit geval eindsituaties; in het algemeen wordt hier een evaluatie-functie aangeroepen) goed te "kiezen": als speler 1 de MAX-speler is, geef dan de situaties met uitslag groter dan 5 een min-teken. Het α-β-algoritme snoeit de drie meest rechtse kinderen van de middelste knoop, en dat is te begrijpen: de MAX-speler kan via zijn linker kindknoop al een antwoord -7 bewerkstelligen, dus als (dankzij het meest linkse kind van de middelste knoop) de middelste knoop -7 of nog lager oplevert, heeft het geen zin nog te kijken naar die drie meest rechtse kinderen. De spelboom ziet er als volgt uit:



Opgave 3:
Denk er aan ook te vertellen wat er gebeurt als er geen voorbeelden meer zijn voor een knoop, wat als er geen attributen meer over zijn (kies de meerderheidswaarde, zie ook **b**), en wat als alle voorbeelden dezelfde classificatie hebben (dan heb je een ja- of nee-knoop). Geef ook de formule voor Gain(attribuut).
Voor het probleem met zeer veel mogelijke attribuutwaarden kan bijvoorbeeld de Gain()-waarde gedeeld worden door iets wat de groottes van de verschillende klassen "meet", bijvoorbeeld:

$$SplitInfo(A) = -\sum_{i=1}^{\nu} \frac{|S_i|}{|S|} \log(\frac{|S_i|}{|S|}),$$

als de betreffende voorbeelden S door het attribuut A in ν klassen $S_1, \ldots, S_\nu$ verdeeld wordt.

Opgave 4:
Een bitstring moet een compleet rooster coderen. Start bijvoorbeeld met een matrix met als rijen de uren van de week en als kolommen de klassen. Zet per matrixelement een leraar (bijvoorbeeld VRIJ)—even aannemend dat leraren en vakken uniek gekoppeld zijn. Zet deze matrix met rijen achter elkaar in een string. Ook is het mogelijk een rij van tupels (vak,klas,uur,leraar,lokaal) te maken, als rijen maar altijd even lang zijn.
Een evaluatiefunctie moet als het ware een rapportcijfer geven aan zo'n rooster. Er zijn harde constraints: de klassen moeten de hun toebedeelde vakken krijgen, leraren kunnen niet op meerdere plaatsen tegelijk aanwezig zijn, en per lokaal en tijd mag er maximaal één klas zijn. Dit soort conflicten leveren strafpunten op. Evenzo, maar dan minder hard, zijn er (minder) strafpunten voor tussenuren en slechte vakverdelingen. Al deze punten opgeteld geven een evaluatiefunctie.
Met behulp van selectie, crossover en mutatie werkt het Genetisch algoritme zich een weg naar een goede oplossing.
Speciale mutaties kunnen bijvoorbeeld twee leraren verwisselen, of een VRIJ uur toevoegen. Bij crossover kun je met daggrenzen of klassen rekening houden.
Een Genetisch algoritme heet *elitair* als de beste x individuen steeds naar de volgende generatie gaan.

Opgave 5:
De frequente 1-itemsets zijn: $\{3\}$ (met een *support* van 5), $\{7\}$ (met een *support* van 2), $\{8\}$, $\{9\}$ en $\{10\}$; de frequente 2-itemsets zijn: $\{3,7\}$, $\{3,8\}$, $\{3,9\}$, $\{3,10\}$, $\{7,9\}$ en $\{9,10\}$ (de door Apriori gegenereerde sets $\{7,8\}$, $\{7,10\}$, $\{8,9\}$ en $\{8,10\}$ vervallen omdat ze de support-drempel niet halen); de frequente 3-itemsets zijn: $\{3,7,9\}$ en $\{3,9,10\}$ (de door Apriori gegenereerde sets $\{3,7,8\}$, $\{3,7,10\}$, $\{3,8,9\}$ en $\{3,8,10\}$ worden "gepruned" omdat ze een niet frequente deelverzameling, bijvoorbeeld $\{7,8\}$ in het geval van $\{3,7,8\}$, bevatten).