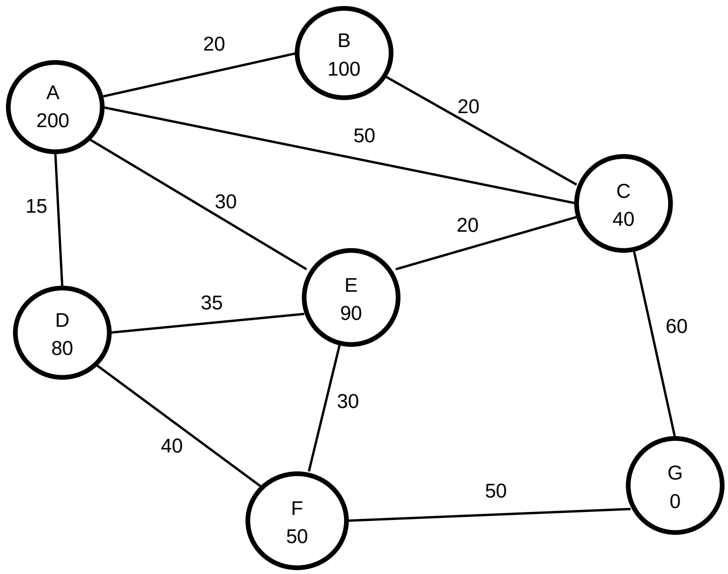
# Sztuczna Inteligencja i Systemy Doradcze
# Egzamin 11.09.2007

## Zadanie 1

Na rysunku poniżej podana jest przestrzeń stanów dla problemu znalezienia drogi o najmniejszym koszcie. $A$ jest stanem początkowym, $G$ jest stanem docelowym. Na łukach łączących poszczególne stany podany jest koszt rzeczywisty przejścia pomiędzy stanami. Wartości funkcji heurystycznej dla poszczególnych stanów podane są w węzłach grafu.




Podaj kolejne kroki wykonania algorytm A* z unikaniem powtarzających się stanów. Podaj wartość funkcji użyteczności dla każdego stanu.

## Zadanie 2

Dany jest zbiór przykładów treningowych z atrybutem decyzyjnym *Przyjety*:

| *Wiek* | *Merytoryczne* | *Formalne* | *Przyjety* |
|---|---|---|---|
| 30 | NIE | TAK | TAK |
| 40 | NIE | NIE | NIE |
| 35 | NIE | NIE | NIE |
| 55 | TAK | TAK | TAK |
| 37 | TAK | NIE | TAK |
| 48 | TAK | TAK | TAK |
| 28 | NIE | TAK | TAK |

Narysuj deterministyczne drzewo decyzyjne indukowane z powyższego zbioru przykładów z wyborem podziałów w poszczególnych węzłach drzewa optymalizującym miarę zysku *DiscernibilityGain* i podaj wartość tej miary w każdym wewnętrznym węźle drzewa. Dla danego podziału obiektów $P_1, P_2$ ($P_1 \cup P_2 = P, P1 \cap P_2 = \emptyset$):

$$DiscernibilityGain(P_1, P_2) = |\ \{ (x, y) \in P_1 \times P_2 \ : \ Przyjety(x) \neq Przyjety(y) \}\ |$$

## Zadanie 3

Gra Dzielenie Kupek jest dwuosobowa. Do gry używa się kuponów, które na początku podzielone są na pewną liczbą kupek. Gracze wykonują ruchy naprzemiennie, pojedynczy ruch polega na wybraniu kupki i podzieleniu jej na dwie mniejsze. Podziału można dokonać na dwa sposoby: z wybranej kupki można wziąć $\lfloor \frac{K}{3} \rfloor$ lub $\lfloor \frac{K}{3} \rfloor + 1$ kuponów ($K$ to liczba kuponów w kupce przed podziałem) i utworzyć z nich nową kupkę. Pozostałe kupony pozostają w drugiej kupce. Kupek złożonych z jednego lub dwóch kuponów nie można już dzielić. Gracz, który nie może podzielić już żadnej kupki, przegrywa.

Przykład rozgrywki zaczynającej się od dwóch kupek, jednej zawierającej 3 kupony, i drugiej zawierającej 7 kuponów (w nawiasach podany jest ostatni podział):
Start: 3+7
Łukasz: 3+(2+5)
Kuba: 3+2+(2+3)
Lukasz: 3+2+2+(2+1)
Kuba: (2+1)+2+2+2+1
Lukasz nie może już podzielić żadnej kupki i przegrywa.

Rozważmy grę rozpoczynającą się od dwóch kupek, jednej zawierającej 26 kuponów i drugiej zawierającej 18 kuponów. Który z graczy: rozpoczynający czy drugi w kolejności ma strategię, która zagwarantuje mu wygraną? Odpowiedz na to pytanie i opisz strategię wygrywającą.

## Zadanie 4

Dany jest następujący zbiór formuł $\Gamma$:

$$\begin{array}{l} \forall x, y, z \, (p(z) \vee q(y, x) \vee \neg r(y, z)) \\ \forall x, y \, (p(x) \rightarrow s(x, y)) \\ \forall x, y \, (p(x) \vee r(x, y)) \\ \forall x \exists y \, (q(x, y) \rightarrow p(x)) \end{array}$$

Dowieść metodą rezolucji, że formuła $\varphi = \forall x \exists y \, (p(x) \wedge s(y, x))$ jest logiczną konsekwencją zbioru $\Gamma$ (tzn. $\Gamma \models \varphi$).

## Zadanie 5

Zaprojektuj 2 warstwową (składającą się z warstwy wejściowej, warstwy ukrytej i warstwy wyjściowej) sieć neuronową, która reprezentuje dodawanie w układzie dwójkowym. Warstwa wejściowa składa się z neuronów $x_1, x_0$ oraz $y_1, y_0$, które przyjmują wartości 0 lub 1. Warstwa wyjściowa składa się z neuronów $z_2, z_1, z_0$. Każde $z_i$ zwraca 0 lub 1 tak, że spełniona jest równość:

$$\sum_{i=0}^{2} 2^i x_i + \sum_{i=0}^{2} 2^i y_i = \sum_{i=0}^{3} 2^i z_i.$$

Funkcja aktywacji $\sigma$ ma postać:

$$\sigma(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$