Zbirka nalog za predmet Osnove umetne inteligence

Jure Žabkar, Martin Možina

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani
1. junij, 2017

Kazalo

Predgovor

Pričujoča zbirka nalog je pripomoček za predmet Osnove umetne inteligence, ki se predava v 3. letniku univerzitetnega študija na Fakulteti za računalništvo in informatiko, Univerza v Ljubljani. Razdeljena je na štiri sklope: Preiskovanje (Jure Žabkar), Planiranje (Martin Možina), Strojno učenje (Jure Žabkar) in Bayesove mreže (Jure Žabkar). Skripta se navezuje na knjigo akad. prof. dr. Ivana Bratka, Prolog Programming for Artificial Intelligence, 4. izdaja, Addison-Wesley, 2012.

Hvaležna sva študentom, ki so zbirko med njenim nastajanjem uporabljali in opozarjali na nejasnosti in pomanjkljivosti. Recenzirala sta jo akad. prof. dr. Ivan Bratko in prof. dr. Marko Robnik Šikonja, za kar sva jima iskreno hvaležna.

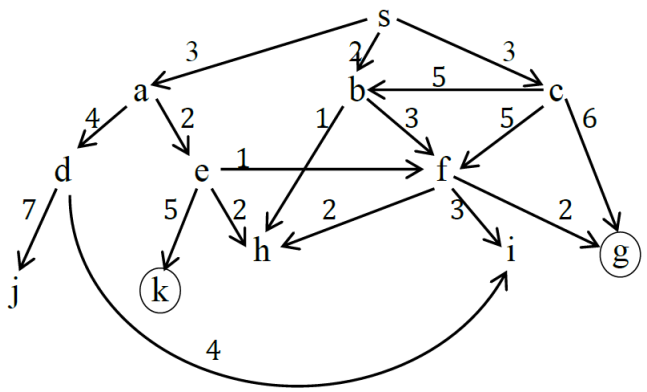
Ljubljana, 1.6.2017 Jure Žabkar, Martin Možina

Poglavje 1

Preiskovanje

1.1 Osnovni algoritmi preiskovanja

Prostor stanj in hevristična funkcija h sta podana na sliki 1.1. Pri reševanju upoštevajte vrstni red pri generiranju vozlišč; generirajo se po abecednem vrstnem redu. Ciljni vozlišči sta obkroženi. Vozlišče s je začetno vozlišče.



Vozlišče	s	a	b	c	d	e	f	g	h	i	j	k
h	7	5	5	4	8	4	1	0	2	3	7	0

Slika 1.1

Simulirajte preiskovanje grafa z naslednjimi postopki:

a) Iskanje v globino (iterativno in rekurzivno).

b) Iskanje v širino.

c) Iterativno poglabljanje.

d) A*. Med enakovrednimi kandidati razvijte najprej tistega, ki je bil prej generiran.

e) IDA*.

Rešitev:

a) Iskanje v globino (iterativno):

Raz.	Gen.	Vrsta
$\emptyset$	s	s
s	a, b, c	a, b, c
a	d, e	d, e, b, c
d	i, j	i, j, e, b, c
i	/	j, e, b, c
j	/	e, b, c
e	f, h, k	f, h, k, b, c
f	g, h, i	g, h, i, h, k, b, c
$\boldsymbol{g}$		

Iskanje v globino (rekurzivno):

Generirano vozlišče takoj razvijemo. Vrstni red razvijanja: s, a, d, i (nima naslednikov, zato sestopimo na d), j (sestopimo na a), e, f, g.

Rešitev: $s \to a \to e \to f \to g$; cena: 8.

b) Iskanje v širino:

Raz.	Gen.	Vrsta
$\emptyset$	s	s
s	a, b, c	a, b, c
a	d, e	b, c, d, e
b	f, h	c, d, e, f, h
c	b, f, g	d, e, f, h, b, f, g
d	i, j	e, f, h, b, f, g, i, j
e	f, h, k	$f, h, b, f, g, i, j, f, h, k$
f	g, h, i	$h, b, f, g, i, j, f, h, k, g, h, i$
h	/	$b, f, g, i, j, f, h, k, g, h, i$
b	f, h	$f, g, i, j, f, h, k, g, h, i, f, h$
f	g, h, i	$g, i, j, f, h, k, g, h, i, f, h, g, h, i$
$\boldsymbol{g}$		

Rešitev: $s \to c \to g$; cena: 9.

c) Iterativno poglabljanje.

Raz.	Gen.	Vrsta
$\emptyset$	s	s
s	/	
$\emptyset$	s	s
s	a, b, c	a, b, c
a	/	b, c
b	/	c
c	/	
$\emptyset$	s	s
s	a, b, c	a, b, c
a	d, e	d, e, b, c
d	/	e, b, c
e	/	b, c
b	f, h	f, h, c
f	/	h, c
h	/	c
c	b, f, g	b, f, g
b	/	f, g
f	/	g
$\boldsymbol{g}$		

Rešitev: $s \to c \to g$; cena: 9.

d) A*. Med enakovrednimi kandidati razvijte najprej tistega, ki je bil prej generiran.

Raz.	Gen.	Vrsta
$\emptyset$	s	$s(0+7)$
s	a, b, c	$b(2+5), c(3+4), a(3+5)$
b	f, h	$h(3+2), f(5+1), c(3+4), a(3+5)$
h	/	$f(5+1), c(3+4), a(3+5)$
f	g, h, i	$c(3+4), g(7+0), a(3+5), h(7+2), i(8+3)$
c	b, f, g	$g(7+0), a(3+5), h(7+2), f(8+1), g(9+0), i(8+3), b(8+5)$
$\boldsymbol{g(7+0)}$		

Rešitev: $s \to b \to f \to g$; cena: 7.

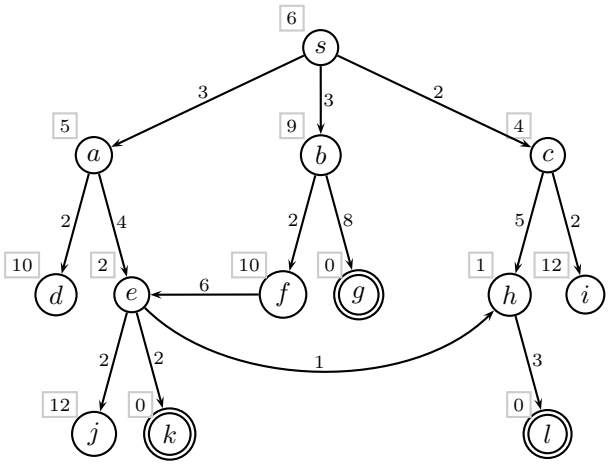
e) IDA*

F	Raz.	Gen.	Vrsta
0	$\emptyset$	s	$\underline{s(0+7)}$
7	$\emptyset$	s	$s(0+7)$
	s	a, b, c	$\underline{a(3+5)}, b(2+5), c(3+4)$
	b	f, h	$f(5+1), h(3+2), c(3+4)$
	f	g, h, i	$g(7+0), \underline{h(7+2)}, \underline{i(8+3)}, h(3+2), c(3+4)$
	$\boldsymbol{g(7+0)}$		

Rešitev: $s \to b \to f \to g$; cena: 7.
V tabeli so podčrtana tista generirana vozlišča, katerih ocena presega trenutno mejo F. Takih vozlišč seveda ne smemo razvijati; z njihovimi ocenami si pomagamo pri določanju nove vrednosti F.

1.2 A* in IDA*

Izvajajte A* in IDA* na grafu na sliki 1.2. Pri reševanju upoštevajte vrstni red pri generiranju vozlišč; generirajo se po abecednem vrstnem redu. Med enakovrednimi kandidati razvijte najprej tistega, ki je bil prej generiran. Hevristične ocene so podane v kvadratkih ob pripadajočih vozliščih.



Slika 1.2

Rešitev:

- Simulacija A*

Raz.	Gen.	Vrsta
$\emptyset$	s	$s(0+6)$
s	a, b, c	$c(2+4), a(3+5), b(3+9)$
c	h, i	$a(3+5), h(7+1), b(3+9), i(4+12)$
a	d, e	$h(7+1), e(7+2), b(3+9), d(5+10), i(4+12)$
h	l	$e(7+2), l(10+0), b(3+9), d(5+10), i(4+12)$
e	h, j, k	$h(8+1), k(9+0), l(10+0), b(3+9), d(5+10),$ $i(4+12), j(9+12)$
h	l	$k(9+0), l(10+0), b(3+9), d(5+10), i(4+12), j(9+12)$
$\boldsymbol{k}$		

Rešitev: $s \to a \to e \to k$; cena: 9.

- Simulacija IDA*

F	Raz.	Gen.	Vrsta
0	$\emptyset$	s	$\underline{s(0+6)}$
6	$\emptyset$	s	$s(0+6)$
	s	a, b, c	$\underline{a(3+5)}, \underline{b(3+9)}, c(2+4)$
	c	h, i	$\underline{h(7+1)}, \underline{i(4+12)}$
8	$\emptyset$	s	$s(0+6)$
	s	a, b, c	$a(3+5), \underline{b(3+9)}, c(2+4)$
	a	d, e	$\underline{d(5+10)}, \underline{e(7+2)}, c(2+4)$
	c	h, i	$h(7+1), \underline{i(4+12)}$
	h	l	$\underline{l(10+0)}$
9	$\emptyset$	s	$s(0+6)$
	s	a, b, c	$a(3+5), \underline{b(3+9)}, c(2+4)$
	a	d, e	$\underline{d(5+10)}, e(7+2), c(2+4)$
	e	j, k	$h(8+1), \underline{j(9+12)}, k(9+0), c(2+4)$
	h	l	$\underline{l(11+0)}, \underline{j(9+12)}, k(9+0), c(2+4)$
	$\boldsymbol{k}$		

Rešitev: $s \to a \to e \to k$; cena: 9.

1.3 Prostor stanj malo drugače

Prostor stanj je podan z naslednjim programom:

$$s(a,b).\ s(a,c).\ s(b,d).\ s(b,e).\ s(c,f).\ s(c,g).\ s(d,h).\ s(e,i).\ s(f,j).$$

$$goal(h).\ goal(i).\ goal(j).$$

$$f(a,2).\ f(b,8).\ f(c,5).\ f(d,9).\ f(e,7).$$

$$f(f,10).\ f(g,12).\ f(h,4).\ f(i,10).\ f(j,6).$$

Predikat $s(n,n')$ pomeni, da je vozlišče n' naslednik vozlišča n, predikat $f(n,v)$ pa podaja vrednost ocenitvene funkcije f za vozlišče n. Cene vseh povezav so enake 0.

Naj bo a začetno vozlišče preiskovanja.

a) Kako se spreminja meja med izvajanjem IDA* za ta graf?

b) Katero rešitveno pot vrne IDA*?

c) V kakšnem vrstnem redu IDA* razvija vozlišča (vključite tudi ponovno razvita vozlišča)?

d) Kaj pomeni, da preiskovalni algoritem *spoštuje prioritetno zaporedje*? Ali ga IDA* v zgornjem primeru spoštuje?

Rešitev:

Izvajanje IDA*:

F	Raz.	Gen.	Vrsta
0	$\emptyset$	a	$\underline{a(2)}$
2	/	a	$a(2)$
	a	b,c	$\underline{b(8)}, \underline{c(5)}$
5	/	a	$a(2)$
	a	b,c	$\underline{b(8)}, c(5)$
	c	f,g	$\underline{f(10)}, \underline{g(12)}$
8	/	a	$a(2)$
	a	b,c	$b(8), c(5)$
	b	d,e	$\underline{d(9)}, e(7), c(5)$
	e	i	$\underline{i(10)}, c(5)$
	c	f,g	$\underline{f(10)}, \underline{g(12)}$
9	/	a	$a(2)$
	a	b,c	$b(8), c(5)$
	b	d,e	$d(9), e(7), c(5)$
	d	h	$h(4), e(7), c(5)$
	h		

a) Spreminjanje meje $F : 0, 2, 5, 8, 9$

b) Rešitvena pot: $a \to b \to d \to h$

c) Vrstni red razvijanja vozlišč: $a; a, c; a, b, e, c; a, b, d, h$

d) Algoritem spoštuje prioritetno zaporedje, če razvija vozlišča po naraščajoči vrednosti ocenitvene funkcije f. IDA* v zgornjem primeru ne spoštuje prioritetnega zaporedja, ker vozlišče d razvije pred h, čeprav ima h manjšo vrednost ocenitvene funkcije kot d.

1.4 Neskončno dvojiško drevo

Prostor stanj je neskončno dvojiško drevo, kjer so cene vseh povezav enake 1. Hevristično funkcijo $h(n)$, kjer je n vozlišče, definiramo takole:

$$h(n) = 2d(n),$$

kjer je $d(n)$ globina vozlišča n v drevesu (koren drevesa ima globino 0). Začetno vozlišče preiskovanja je koren drevesa, ciljna vozlišča pa so na globini 5. Preiskujemo z algoritmom IDA*.

a) Kako se spreminja meja med izvajanjem IDA*?

b) Kolikšna je cena rešitvene poti?

c) Izračunajte koliko najmanj in koliko največ vozlišč generira IDA*, preden najde rešitev. Upoštevajte tudi morebitna ponovno generirana vozlišča.

Rešitev:

a) Vedno začnemo z mejo $F = 0$. V korenu je $f = 0$, za njegova naslednika na globini 1 pa velja $f = 1 + 2 * 1 = 3$, zato je naslednja meja $F = 3$. Ugotovimo, da meja raste z globino: sinovi vozlišč na globini 1 imajo $f = 2 + 2 * 2 = 6$ itn. Meja F se torej spreminja takole $0, 3, 6, 9, 12, 15$.

b) Cena rešitve je vsota cen poti: $1 + 1 + 1 + 1 + 1 = 5$

c) V prvi iteraciji generiramo koren in njegova naslednika, se pravi $2^0 + 2^1 = 3$ vozlišča. V drugi iteraciji generiramo vsa vozlišča iz prve iteracije in vse $2^2 = 4$ njihove naslednike: $2^0 + 2^1 + 2^2 = 7$. V tretji iteraciji ponovno generiramo vse iz prvih dveh iteracij in $2^3 = 8$ njihovih naslednikov: $2^0 + 2^1 + 2^2 + 2^3 = 15$. V i-ti iteraciji tako generiramo $\sum_{i=0}^{d} 2^i$ vozlišč. Šele v peti iteraciji generiramo morebitna končna vozlišča na globini 5 (pazite, začeli smo z globino 0, zato se število iteracij in globina ne ujemata). Do zdaj smo torej generirali $3 + 7 + 15 + 31 + 63 = 119$ vozlišč. V najboljšem primeru moramo generirati še 6 novih - to je skrajno leva veja drevesa, ob predpostavki, da IDA* izvajamo rekurzivno (generirano vozlišče takoj razvijemo) in da je skrajno levo vozlišče na globini 5 končno. Ni nam potrebno generirati njegovih naslednikov, saj ob razvijanju najprej preverimo, če je končno in naslednike generiramo samo, če ugotovimo, da ni. Generiramo torej najmanj $119 + 6 = 125$ vozlišč. Iterativna implementacija IDA* potrebuje nekoliko več generiranih vozlišč, namesto 6 jih moramo prišteti 11: $119 + 11 = 130$. Največ vozlišč generiramo takrat, ko je končno vozlišče skrajno desno

na globini 5. V tem primeru generiramo skoraj vsa vozlišča na globini 6, razen seveda zadnjih dveh, ki sta naslednika končnega vozlišča: $119 + 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 - 2 = 119 + 127 - 2 = 244$.

1.5 Teoretična

Naj bo s začetno vozlišče in f ocenitvena funkcija oblike $f(n) = g(n)+h(n)$, kjer je g cena optimalne poti od s do n, h pa hevristična ocena.

a) Kdaj je ocenitvena funkcija f monotona? Navedite preprost primer, kjer f ni monotona. Predlagajte optimistično hevristiko za problem preiskovanja grafa slovenskih cest, ki dobro usmerja preiskovanje? Podajte primer optimistične hevristike, ki slabo usmerja preiskovanje.

b) Ali iz monotonosti sledi dejstvo, da hevristika h zadošča izreku o popolnosti?

c) Ali velja: če h zadošča izreku o popolnosti, je f monotona?

d) Ali A* razvija generirana vozlišča v prioritetnem vrstnem redu, če f ni monotona? Kaj pa IDA*?

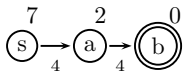
Rešitev:

a) Funkcija f je monotona, če za vsak par vozlišč n, n' velja: $s(n, n') \Rightarrow f(n) \leq f(n')$. Pri tem predikat $s(n, n')$ pomeni, da je n' naslednik vozlišča n. Primer je na sliki 1.3a. Zračna razdalja med kraji je primer optimistične hevristike za cestno omrežje, ki dobro usmerja preiskovanje. Hevristika $h(n) = 0$ za vsak n, je tudi optimistična, vendar slabo usmerja preiskovanje.

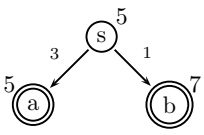
b) Ne, iz monotonosti **ne sledi**, da hevristika zadošča izreku o popolnosti. Primer je graf na sliki 1.3b.

c) Ne. Hevristika h je lahko optimistična (torej zadošča izreku o popolnosti), f pa ni monotona (slika 1.3a).

d) A* razvija generirana vozlišča v prioritetnem vrstnem redu ne glede na monotonost f, pri IDA* pa je monotonost pogoj za razvijanje v prioritetnem vrstnem redu.



(a) f ni monotona: $f(s) = 7$, $f(a) = 6$ in $f(b) = 8$. h je optimistična.

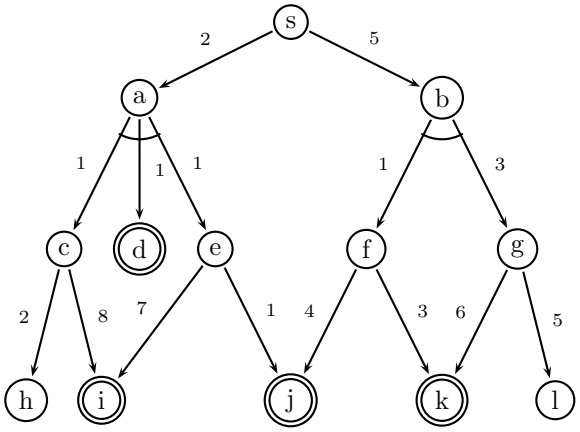


(b) f je monotona ($f(s) = 5$, $f(a) = 8$ in $f(b) = 8$), h ni optimistična.

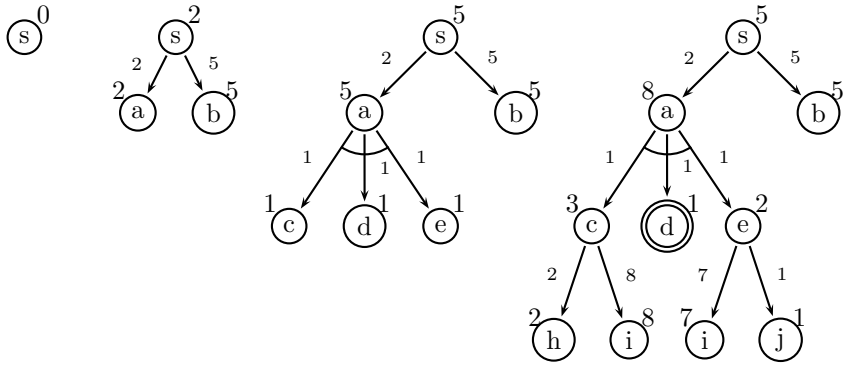
Slika 1.3

1.6 AND-OR graf

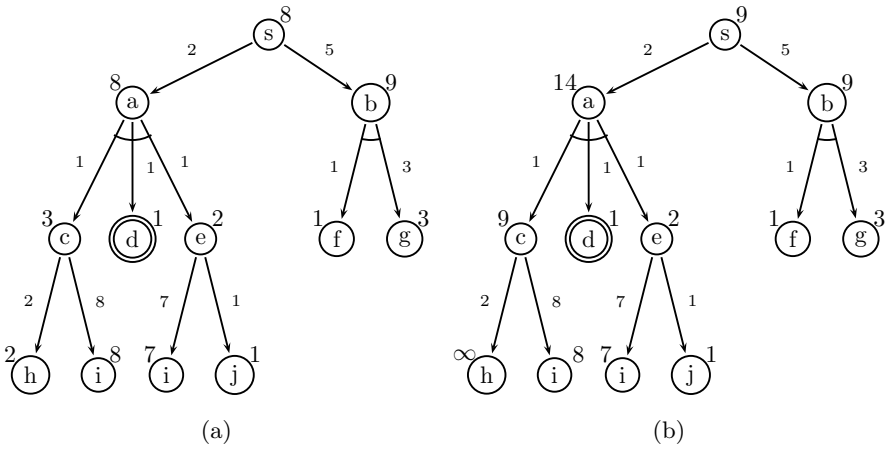
Dan je AND-OR graf (slika spodaj). Naj bodo hevristične ocene vozlišč enake 0, t.j. $h(N) = 0$ za vsak N. Naj bo s začetno vozlišče. Naslednike generiramo po abecednem redu, v AND vozlišču pa vedno razvijemo vse naslednike. Simuliraj algoritem AO*.



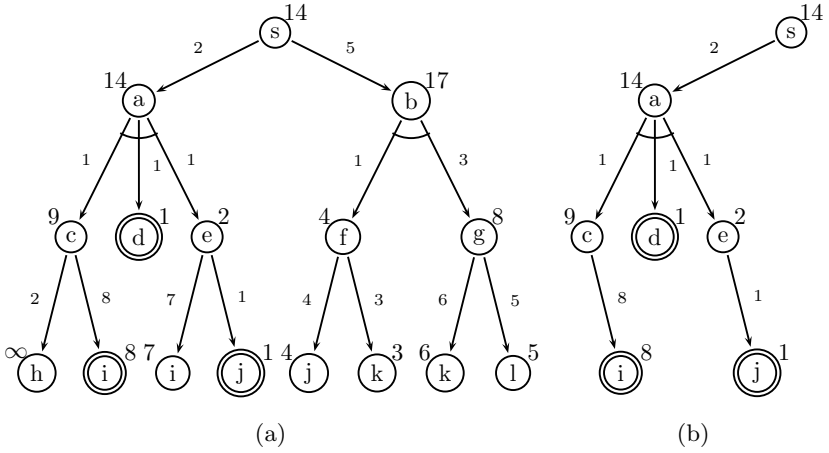
Rešitev:



Slika 1.4: Najprej generiramo s, ki ima oceno $F(s) = 0$, in ga razvijemo. Popravimo oceno $F(s) = 0 + \min\{F(a) = 2, F(b) = 5\} = 2$. Nadaljujemo z razvijanjem vozlišča a, ki ima manjšo oceno F. Generiramo vozlišča c, d in e ter popravimo oceni za a in s: $F(a) = 5$ in $F(s) = 5$. Ker $F(a)$ ne presega $F(b)$, nadaljujemo z razvijanjem v istem poddrevesu, torej razvijemo c, d in e ter popravimo njihove ocene: $F(c) = 1 + \min\{F(h) = 2, F(i) = 8\} = 3$, d je končno vozlišče ($F(d) = 1$), $F(e) = 1 + \min\{F(i) = 7, F(j) = 1\} = 2$. Popravimo še oceno vozlišča a, $F(a) = 2 + (3 + 1 + 2) = 8$ in vozlišča s, $F(s) = 0 + \min\{F(a) = 8, F(b) = 5\} = 5$.



Slika 1.5: Nadaljujemo v poddrevesu s korenom b: razvijemo b, pri čemer generiramo f in g. Ocena b se pri tem spremeni: $F(b) = 5 + (1 + 3) = 9$. (slika 1.5a). Zato se vrnemo v levo poddrevo a, kjer zdaj lahko razvijemo vozlišče h, ki ni niti končno niti nima naslednikov, zato je $F(h) = \infty$. Zaradi tega se spremeni ocena vozlišča $c, F(c) = 9$. Popravijo se tudi ocene $F(a) = 14$ in $F(s) = 9$ (slika 1.5b).



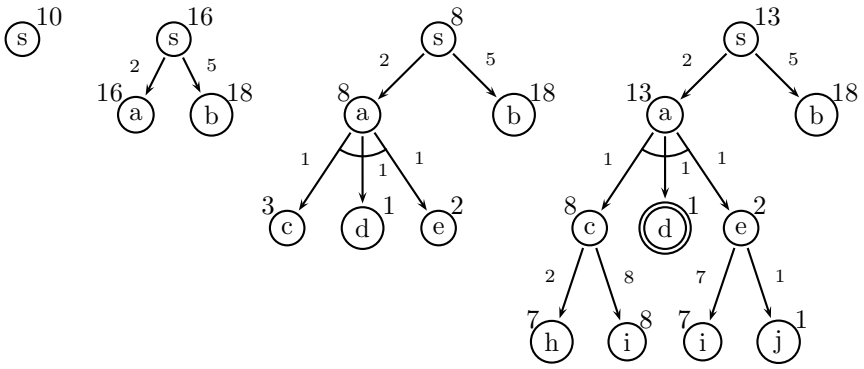
Slika 1.6: Nadaljujemo v poddrevesu s korenom v b in razvijemo f in g, kjer je $F(f) = 4$ in $F(g) = 8$. Popravimo $F(b) = 17$ in $F(s) = 14$. Ponovno gremo v poddrevo s korenom v a, kjer razvijemo i, pri čemer ugotovimo, da je i končno, ocene vozlišč pa ostanejo iste. Zato nadaljujemo z razvijanjem j (ima nižjo oceno kot i) in prav tako ugotovimo, da je končno. S tem smo končali. Rešitveno drevo je prikazano na sliki 1.6b. Cena rešitve je vsota cen vseh povezav v rešitvenem drevesu in je kar enaka oceni v korenu, $F(s) = 14$.

1.7 AND-OR graf z netrivialno hevristiko

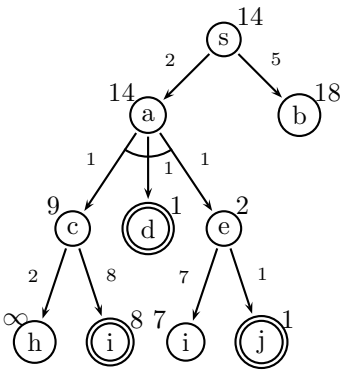
Za AND-OR graf iz prejšnje naloge podamo še hevristično funkcijo. Simuliraj algoritem AO*.

N	s	a	b	c	d	e	f	g	h	i	j	k	l
$h(N)$	10	14	13	2	0	1	4	5	5	0	0	0	7

Rešitev:



Slika 1.7: Najprej generiramo s, ki ima oceno $F(s) = 0 + 10 = 10$, in ga razvijemo. Vozlišči a in b sta lista, zato sta njuni hevristični oceni enaki $H(a) = h(a)$ in $H(b) = h(b)$, njuni F oceni pa: $F(a) = 2 + 14 = 16$ in $F(b) = 5 + 13 = 18$. Popravimo oceno $F(s) = 0 + \min\{F(a) = 16, F(b) = 18\} = 16$. Nadaljujemo z razvijanjem a in generiramo c, d in e z F ocenami $F(c) = 3$, $F(d) = 1$ in $F(e) = 2$. Popravimo še $F(a) = 8$ in $F(s) = 8$. Še vedno je $F(a) < F(b)$, zato nadaljujemo z razvijanjem c, d in e. Ustrezno popravimo ocene od listov navzgor: $F(c) = 8$, $F(d) = 1$ (ugotovimo, da je d končno vozlišče), $F(e) = 2$, $F(a) = 2 + 8 + 1 + 2 = 13$ in $F(s) = 13$.



Slika 1.8: Še vedno ostajamo v levem poddrevesu. Razvijemo najprej h, ugotovimo, da nima sinov niti ni končno, zato dobi oceno ∞ (gotovo ne vodi do rešitve). Takoj popravimo ocene njegovih prednikov: $F(c) = 1 + \min\{F(h), F(i)\} = 9$, $F(a) = 14$ in $F(s) = 14$. Vztrajamo v istem poddrevesu, kjer je naslednji kandidat za razvijanje i. Ugotovimo, da je i končno vozlišče, ocene pa se ne spremenijo, zato nadaljujemo z razvijanjem j, ki je spet končno. Ugotovimo, da obstaja poddrevo, katerega listi so vsi končna vozlišča, zato končamo. Rešitev je isto drevo kot prej, le pot do rešitve je bila krajša, ker je preiskovanje usmerjala dobra hevristična funkcija. Tudi cena rešitve je ista, saj cen povezav nismo spreminjali, hevristične ocene pa na ceno rešitve ne vplivajo.

Poglavje 2

Planiranje

2.1 STRIPS

Imamo 3 kocke (a, b, c) in 4 možne pozicije (1,2,3,4), kot kaže slika 2.1. Robot, ki premika kocke, ima na voljo le operacijo *move*. S to akcijo lahko prime kocko na vrhu in jo premakne ter odloži na drugi kocki ali na označeni pozici na tleh. Stanja in akcijo opišemo z relacijama *clear* in *on*.

a) V STRIPS jeziku opišite začetno stanje s slike in akcijo *move*.

b) V STRIPS jeziku opišite cilj planiranja. Želimo sestaviti stolp, kjer je c spodaj, b na sredini in a na vrhu.

c) Na kratko opišite osnovne korake pri planiranju s sredstvi in cilji.

d) Simulirajte postopek planiranja. Uporabite preiskovanje v globino. Kaj je problem preiskovanja v globino?

e) Uporabite iterativno poglablanje. Pri klasičnem preiskovanju iterativno poglablanje zagotavlja optimalno rešitev. Kaj pa tu?

f) Kaj je osnovni princip regresiranja ciljev. Napišite formulo.

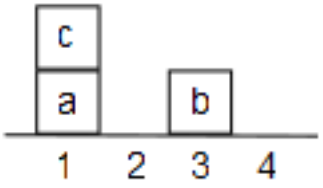
g) Simulirajte planiranje z regresijo ciljev. Uporabite iterativno poglablanje; začnite z globino $D = 3$.

Rešitev: Pri prvi nalogi bodo odgovori precej bolj podrobni, kot pri nalogah, ki sledijo. Čeprav razumevanje nekaterih rešitev zahteva malo več truda, lahko le na ta način podrobno prikažemo delovanje algoritmov planiranja.

Naloga a: opis akcije *move*

Stanje na sliki 2.1 opišemo z relacijama *clear* in *on*:

$state = [clear(2), clear(4), clear(b), clear(c), on(a, 1), on(c, a), on(b, 3)]$.



Slika 2.1: Začetno stanje v svetu kock

Za imena relacij bomo, zaradi konsistentnosti z opisi v knjigi, uporabljali angleške izraze. Z relacijo *clear* označimo prost objekt, *on* pa pomeni, da prvi objekt stoji na drugem.

Preostane nam še opis akcije *move*. Akcija ima 3 argumente:

$move(X, From, To)$,

kjer X označuje kocko, ki jo bomo premaknili, $From$ in To sta začetni in končni položaj. Akcije bomo opisali s štirimi atributi:

conditions; pogoji, ki morajo veljati v trenutnem stanju, da je akcija izvedljiva,

adds; relacije, ki jih akcija doda v stanje (temu pravimo tudi pozitivni učinki akcije),

dels; relacije, ki jih akcija izbriše iz stanja (negativni učinki),

constraints; omejitve pri določanju vrednosti spremenljivk, s katerimi lahko občutno zmanjšamo prostor preiskovanja.

Omejitve (*constraints*) in pogoji (*conditions*) imajo na prvi pogled enako vlogo. Običajno v pogoje pišemo relacije, ki jih v nadaljevanju lahko dosežemo, npr. $on(b, 2)$. V omejitvah pa so fiksni pogoji, npr. "a je kocka", in pogoji, ki preprečujejo nesmiselne akcije, npr. premik kocke na samo sebe.

Akcijo *move* lahko izvedemo pri naslednjih pogojih:

$conditions(move(X, From, To)) = [clear(X), clear(To), on(X, From)]$.

Kocka X mora biti prosta (sicer je ne moremo prijeti), To mora biti prost, da lahko kocko postavimo tja in X mora biti na $From$.

Izvedba akcije v trenutnem stanju doda relacije:

$adds(move(X, From, To)) = [clear(From), on(X, To)]$.

Objekt $From$ se po akciji sprosti in X je na mestu To. Negativni učinki akcije so: $clear(To)$ in $on(X, From)$:

$dels(move(X, From, To)) = [clear(To), on(X, From)]$.

Omejitve:

$constraints(move(X, From, To)) = [X \neq From, X \neq To, To \neq From, block(X)]$.

Omejitev $X \neq To$ preprečuje, da bi kocko X premaknili na samo sebe, $To \neq From$ pa preprečuje, da bi kocko premaknili nazaj na isto mesto. Z $block(X)$ zahtevamo, da je X kocka (sicer bi algoritem lahko poskušal premikati tudi polja 1,2,3,4).

Naloga b: opis cilja

Stolp, kjer je c spodaj in a na vrhu:

$goals = [on(a, b), on(b, c)]$.

Naloga c: princip sredstev in ciljev

Osnovni princip planiranja je sredstva-cilji (*angl.* means-ends), kjer izbiramo sredstva (akcije), ki najbolj verjetno vodijo k zadanemu cilju. Postopek ima štiri korake:

1. Izberi še nerešen cilj v $goals$. Cilje izbiramo po vrsti, kot so napisani.
2. Izberi akcijo, ki ta cilj doda v stanje. Vse akcije, ki dodajo izbran cilj v stanje, predstavljajo možnosti za nadaljevanje. Za potrebe preiskovanja akcije uredimo glede na število nerešenih predpogojev.
3. Omogoči izbrano akcijo tako, da obravnavaš njene predpogoje kot nove cilje.
4. Izvedi akcijo, ki doda izbran cilj v trenutno stanje.
5. Če obstajajo nerešeni cilji, se vrni na korak 1.

Za preiskovanje lahko uporabimo poljuben algoritem; v globino, iterativno poglabljanje, A*, itd.

Naloga d: preiskovanje v globino

V naši domeni sveta kock bi postopek potekal takole:

1. Izberemo cilj $on(a, b)$.

2. Za cilj poiščemo akcijo, ki ga vzpostavi. To informacijo vsebuje množica $adds$. Akcija, ki doda relacijo $on(a, b)$ je: $move(a, From, b)$. Predpogoji zanjo so:

$conditions(move(a, From, b)) = [clear(a), clear(b), on(a, From)]$.

Zaradi nedoločene vrednosti $From$ imamo 5 možnih akcij: $move(a, 1, b)$, $move(a, 2, b)$, $move(a, 3, b)$, $move(a, 4, b)$ in $move(a, c, b)$. Zaradi omejitev a in b nista dovoljeni vrednosti spremenljivke $From$. Najprej poskusimo s $From = 1$, ker sta potem dva predpogoja že izpolnjena v začetnem stanju.

3. Predpogoji $[clear(a), clear(b), on(a, 1)]$ postanejo trenutni cilji in rekurzivno kličemo program za planiranje (zato znak ' pri oznaki koraka).

1'. Izberemo cilj $clear(a)$.

2'. Akcija: $move(X, a, To)$. Torej, nekaj moramo premakniti iz a in postaviti drugam, da bo a prost. Predpogoji za to akcijo so:

$conditions(move(X, a, To)) = \{clear(X), clear(To), on(X, a)\}$.

Vrednosti za X in To, izpolnjene že v začetnem stanju, so $X = c$ in $To = 2$.

3'. Korak ni potreben, vsi pogoji so izpolnjeni v trenutnem stanju.

4'. Izvedemo akcijo

$$1.move(c, a, 2)$$

in dobimo novo stanje tako, da iz stanja *initial_state* izbrišemo vse iz *dels*: $clear(2), on(c, a)$ in dodamo vse iz *adds*: $clear(a), on(c, 2)$. Novo stanje je

$$state = [clear(4), clear(a), clear(b), clear(c), on(a, 1), on(c, 2), on(b, 3)].$$

5'. Vsi cilji (predpogoji) so izpolnjeni, zaključimo.

3. Zdaj so izpolnjeni vsi predpogoji za akcijo $move(a, 1, b)$.

4. Izvedemo akcijo:

$$2.move(a, 1, b)$$

in dobimo stanje:

$$state = [clear(1), clear(4), clear(a), clear(c), on(a, b), on(c, 2), on(b, 3)].$$

5. Cilj $on(b, c)$ še ni izpolnjen. Vrnemo se na korak 1.

1. Izpolnili smo prvi cilj $on(a, b)$ in se lotimo drugega, $on(b, c)$.

2. Ta cilj dosežemo z akcijo $move(b, From, c)$, ki ima predpogoje:

$$conditions(move(b, From, c)) = [clear(b), clear(c), on(b, From)].$$

Izberemo $From = 3$.

3. Rešujemo predpogoje $[clear(b), clear(c), on(b, 3)]$.

1'. Izberemo cilj $clear(b)$.

2'. Akcija: $move(X, b, To)$ in njeni predpogoji so

$$conditions(move(X, b, To)) = [clear(X), clear(To), on(X, b)].$$

Že izpolnjene vrednosti za X in To so $X = a$ in $To = 1$.

3'. Vsi pogoji so izpolnjeni.

4'. Izvedemo akcijo:

$$3.move(a, b, 1)$$

in dobimo stanje

$$state = [clear(4), clear(a), clear(b), clear(c), on(a, 1), on(c, 2), on(b, 3)].$$

5'. Vsi cilji so izpolnjeni, zaključimo.

3 in 4. Predpogoji so izpolnjeni in izvedemo četrto akcijo, s katero dosežemo $on(b, c)$:

$$4.move(b, 3, c)$$

in dobimo stanje

$$state = \{clear(3), clear(4), clear(a), clear(b), on(a, 1), on(c, 2), on(b, c)\}.$$

5. Med reševanjem $on(b, c)$ smo podrli cilj $on(a, b)$ in ga moramo ponovno reševati. Vrnemo se na korak 1.

1. Cilj: $on(a, b)$

2. Akcija: $move(a, From, b)$ izpolni ta cilj in, ker predpogoji zanjo pri $From = 1$ v danem stanju veljajo, jo lahko (4. korak) izvedemo:

$$5.move(a, 1, b)$$

Končno stanje je:

$state = [clear(1), clear(3), clear(4), clear(a), on(a, b), on(c, 2), on(b, c)]$.

5. Vsi cilji so doseženi, končamo postopek.

S planiranjem smo našli rešitev:

1. $move(c, a, 2)$,
2. $move(a, 1, b)$,
3. $move(a, b, 1)$,
4. $move(b, 3, c)$,
5. $move(a, 1, b)$.

Rešitev ni optimalna, saj je problem rešljiv v le treh potezah. V nadaljevanju si bomo pogledali, kako lahko najdemo krajše rešitve z a) uporabo interativnega poglabljanja in b) regresijo ciljev.

Naloga e: preiskovanje z iterativnim poglabljanjem

Pri iterativnem poglabljanjem bomo postopoma povečevali dolžino D dovoljenega plana. Zaradi preglednosti bomo v simulaciji nekaj korakov izpustili.

Imamo isti cilj:

$goals = [on(a, b), on(b, c)]$

v naši začetni poziciji:

$initial\_state =$
$[clear(2), clear(4), clear(b), clear(c), on(a, 1), on(c, a), on(b, 3)]$.

$$D = 1$$
(največja dolžina plana je 1)

(d=0) Rešujemo cilje $on(a,b), on(b,c)$**.**

Izberemo cilj $on(a,b)$. Akcija, ki doda $on(a,b)$ je $move(a, From, b)$. Predpogoji so:

$conditions(move(a, From, b)) = [clear(a), clear(b), on(a, From)]$.

Ker $clear(a)$ v začetnem stanju ni resničen, potrebujemo še eno akcijo. Cilja torej ne moremo rešiti samo z eno akcijo.

Izberemo cilj $on(b,c)$. Akcija: $move(b, From, c)$, predpogoji so:

$conditions(move(b, From, c)) = [clear(b), clear(c), on(b, From)]$.

Možne vrednosti za $From$ so: $3, 1, 2, 4, a$. Pri vrednosti $From = 3$ lahko izvedemo akcijo $move(b, 3, c)$ in dosežemo $on(b,c)$. V tem stanju še vedno ni dosežen $on(a,b)$, torej to ni rešitev. Vse druge vrednosti za $From$ prav tako zahtevajo dodatne akcije za uresničitev predpogojev in jih zaradi omejitve $D = 1$ ne moremo izpolniti. Pri $D = 1$ torej ne moremo najti rešitve.

$$D = 2$$

(d=0) Rešujemo cilje $on(a,b), on(b,c)$**.**

Akcija: $move(a, From, b)$, predpogoji so

$conditions(move(a, From, b)) = [clear(a), clear(b), on(a, From)]$.

Možne vrednosti za $From$ so: $1, 2, 3, 4, c$. Nastavimo $From = 1$, rešiti moramo:

$[clear(a), clear(b), on(a, 1)]$.

(d=1) Rešujemo cilje $[clear(a), clear(b), on(a, 1)]$

Zdaj smo na globini 1 in lahko uporabimo le še eno akcijo, saj bomo v vsakem primeru izvedli $move(a, 1, b)$. Izberemo cilj $clear(a)$, akcija: $move(X, a, To)$. Predpogoji so:

$conditions(move(X, a, To)) = [clear(X), clear(To), on(X, a)]$,

kjer so možne vrednosti za X in To:

$$(c,2), (c,4), (c,b), (c,1), (c,3), (b,2), (b,4), (b,c), (b,1), (b,3).$$

Prvi trije nabori $(c,2), (c,4), (c,b)$ izpolnjujejo vse pogoje in so zato na začetku, $(c,1), (c,3)$ izpolnjujeta le dva pogoja, itd.

Izberimo $X = c$ in $To = 2$. Zdaj lahko izvedemo prvo akcijo $1.move(c, a, 2)$ in takoj nato še akcijo $2.move(a, 1, b)$ iz prejšnjega koraka. Dobimo stanje:

$state = [clear(1), clear(4), clear(a), clear(c), on(a, b), on(c, 2), on(b, 3)]$.

V tem stanju ni izpolnjen cilj $on(b, c)$ in ker smo naredili že dva koraka, se vrnemo in poskusimo drugo pot.

Postopek bi nadaljevali z naslednjim naborom vrednosti za X in To. Hitro opazimo, da z uporabo zgoraj naštetih vrednosti ne bi našli rešitve in se moramo vrniti na globino d=0.

(d=0) Rešujemo cilje $on(a, b), on(b, c)$, nadaljevanje.

V množici ciljev $[clear(a), clear(b), on(a, 1)]$ smo poskušali rešiti $clear(a)$, a nam ni uspelo priti do rešitve. Preostala dva cilja sta že izpolnjena, zato jih nima smisla reševati. Poskusimo naslednjo nastavitev za $From$; $From = 2$. Nov nabor ciljev je:

$$[clear(a), clear(b), on(a, 2)].$$

Najprej bi poskusili še enkrat rešiti $clear(a)$ z istim rezultatom kot zgoraj. Potem bi poskušali reševati $on(a, 2)$, a brez uspeha, saj za premik kocke a na drugo mesto potrebujemo $clear(a)$. Enako se zgodi pri ostalih vrednostih $From$.

Ne preostane nam drugega, kot da izberemo cilj: $on(b, c)$. Akcija je: $move(b, From, c)$, predpogoji pa:

$conditions(move(b, From, c)) = [clear(b), clear(c), on(b, From)]$.

Možne vrednosti za $From$ so: $3, 1, 2, 4, a$. Nastavimo $From = 3$ in dobimo množico ciljev:

$[clear(b), clear(c), on(b, 3)]$.

Vsi cilji so izpolnjeni, torej so izpolnjeni predpogoji za $1.move(b, 3, c)$.

Algoritem bi zdaj izračunal novo stanje in potem nadaljeval z reševanjem $on(a, b)$. Ugotovil bi, da cilj ni rešljiv z eno akcijo. Nato bi se vrnil in poskušal s $From = 1$, kar pomeni, da bi najprej premaknil kocko b na polje 1 in jo potem dal na c. Spet se ne bi izšlo. To bi ponavljal še s preostalimi vrednostmi za $From$ in vendar neuspel. S tem bi se zaključilo preiskovanje pri $D = 2$.

$$D = 3$$

Vemo, da je problem rešljiv s tremi premiki. Torej bi rešitev morali najti pri $D = 3$. Pa jo res?

(d=0) Rešujemo cilje $on(a,b), on(b,c)$.

Izberemo cilj $on(a,b)$, akcija: $move(a, From, b)$, predpogoji so

$conditions(move(a, From, b)) = [clear(a), clear(b), on(a, From)]$.

Možne vrednosti za $From$ so: $1, 2, 3, 4, c$. Nastavimo $From = 1$ in dobimo novo množico ciljev:

$[clear(a), clear(b), on(a, 1)]$.

(d=1) Rešujemo cilje $[clear(a), clear(b), on(a, 1)]$.

Izberemo cilj $clear(a)$, akcija: $move(X, a, To)$. Predpogoji so:

$conditions(move(X, a, To)) = [clear(X), clear(To), on(X, a)]$,

kjer so možne vrednosti za X in To: $(c, 2), (c, 4), (c, b), ...$ Izvedemo akciji $1.move(c, a, 2)$ in $2.move(a, 1, b)$, kot pri $D = 2$. Zdaj ostane še reševanje $on(b, c)$.

Očitno $on(b, c)$ ne dosežemo z eno akcijo, zato postopka ne bomo nadaljevali. Vprašanje je, ali je možno doseči rešitev po drugi poti; npr. če bi v koraku A spremenili vrednost $From$? Izkaže se da ne, saj algoritem poskuša najprej doseči $on(a, b)$, a pri tem ne uspe, saj bi moral najprej poskrbeti za $on(b, c)$.

(d=0) Rešujemo cilje $on(a,b), on(b,c)$, **nadaljevanje.**

Izberemo $on(b, c)$, akcija: $move(b, From, c)$ in nastavimo $From = 3$, kot pri $D = 2$. Dobimo cilje:

$[clear(b), clear(c), on(b, 3)]$.

Vsi cilji so izpolnjeni, izvedemo akcijo $1.move(b, 3, c)$. Spet hitro vidimo, da ne bomo uspeli z dvema akcijama, saj potrebujemo dva koraka za dosego $clear(a)$, ki je predpogoj za akcijo, ki bi dosegla $on(a, b)$.

$$D = 4$$

(d=0) Rešujemo cilje $on(a,b), on(b,c)$.

Izberemo cilj $on(a, b)$. Akcija: $move(a, From, b)$, predpogoji so

$conditions(move(a, From, b)) = [clear(a), clear(b), on(a, From)]$.

Možne vrednosti za $From$ so: $1, 2, 3, 4, c$. Nastavimo $From = 1$ in dobimo cilje:

$[clear(a), clear(b), on(a, 1)]$.

(d=1) Rešujemo cilje $[clear(a), clear(b), on(a, 1)]$.

Izberemo cilj $clear(a)$. Akcija: $move(X, a, To)$. Predpogoji so:

$conditions(move(X, a, To)) = [clear(X), clear(To), on(X, a)]$,

vrednosti za X in To so:

$$(c, 2), (c, 4), (c, b), (c, 1), (c, 3), (b, 2), (b, 4), (b, c), (b, 1), (b, 3).$$

Pri izbranih vrednostih $(c, 2)$ lahko izvedemo akciji $move(c, a, 2)$ in $move(a, 1, b)$ in dosežemo stanje:

$state = [clear(1), clear(4), clear(a), clear(c), on(a, b), on(c, 2), on(b, 3)]$.

Ostane še reševanje $on(b, c)$. S preostalima dvema akcijama nam to ne uspe, saj moramo najprej umakniti a iz b, potem premakniti b na c in še enkrat a na b. Vrnimo se in poskusimo z drugimi vrednostmi za X in To.

Pri $(c, 4)$,(c, b),$(c, 1)$,$(c, 3)$,$(b, 2)$ in $(b, 4)$ naletimo na podoben problem. Šele pri $X = b$ in $To = c$ najdemo rešitev. Naša akcija je torej $move(b, a, c)$, predpogoji so:

$conditions(move(b, a, c)) = [clear(b), clear(c), on(b, a)]$,

(d=2) Rešujemo cilje $[clear(b), clear(c), on(b, a)]$.

Izberemo $on(b, a)$, akcija: $move(b, From, a)$, predpogoji:

$conditions(move(b, From, a)) = [clear(b), clear(a), on(b, From)]$,

Poskusimo s $From = 3$, predpogoji za akcijo $move(b, 3, a)$ so:

$[clear(b), clear(a), on(b, 3)]$.

(d=3) Rešujemo cilje $[clear(b), clear(a), on(b, 3)]$.

Izberemo $clear(a)$. Akcija: $move(X, a, To)$. Predpogoji so:

$conditions(move(X, a, To)) = [clear(X), clear(To), on(X, a)]$,

vrednosti za X in To so:

$$(c, 2), (c, 4), (c, b), (c, 1), (c, 3), (b, 2), (b, 4), (b, c), (b, 1), (b, 3).$$

Tokrat rešitev najdemo že kar pri prvem naboru vrednosti $(c, 2)$. Zdaj lahko izvedemo

$$1.move(c, a, 2)$$

in dobimo novo stanje:

$state = [clear(4), clear(a), clear(b), clear(c), on(a, 1), on(b, 3), on(c, 2)]$.

(d=2) Rešujemo cilje $[clear(b), clear(c), on(b, a)]$, **nadaljevanje.**

Predpogoji $[clear(b), clear(a), on(b, 3)]$ so izpolnjeni, lahko izvedemo akcijo

$$2.move(b, 3, a).$$

Novo stanje:

$state = [clear(3), clear(4), clear(b), clear(c), on(a, 1), on(b, a), on(c, 2)].$

(d=1) Rešujemo cilje $[clear(a), clear(b), on(a, 1)]$, **nadaljevanje.**

Izpolnjeni predpogoji $[clear(b), clear(c), on(b, a)]$, izvedemo akcijo:

$$3.move(b, a, c)$$

Novo stanje:

$state = [clear(3), clear(4), clear(a), clear(b), on(a, 1), on(b, c), on(c, 2)].$

(d=0) Rešujemo cilje $on(a, b), on(b, c)$.

Izpolnjeni predpogoji $[clear(a), clear(b), on(a, 1)]$, izvedemo akcijo:

$$3.move(a, 1, b)$$

Novo stanje:

$state = [clear(1), clear(3), clear(4), clear(a), on(a, b), on(b, c), on(c, 2)].$

Oba cilja sta dosežena, zaključimo postopek!

Našli smo rešitev:

1. $move(c, a, 2)$,
2. $move(b, 3, a)$,
3. $move(b, a, c)$,
4. $move(a, 1, b)$.

Ta naloga kaže, da je planiranje zelo povezano s preiskovanjem. Mnogokrat se namreč zgodi, da prva izbrana akcija, ki cilj uresniči, ni primerna. V koraku z $d = 1$ smo šele v sedmem poskusu z akcijo $move(b, a, c)$ prišli do cilja.

Naloga f: planiranje z regresijo ciljev

Pri metodi sredstva-cilji se posamezni cilji rešujejo lokalno, kar mnogokrat onemogoča poiskati najkrajšo rešitev. To imenujemo Sussmanova anomalija. Regresiranje ciljev se problema loti globalno in poskuša doseči vse cilje naenkrat. S tem omogoča iskanje optimalnih planov.

Postopek začnemo tako, da iz množice ciljev izberemo cilj in ustrezno akcijo, s katero bi ta cilj dosegli. V naslednjem koraku se vprašamo, kaj vse bi moralo veljati, da bi bili po izvedeni izbrani akciji doseženi vsi cilji (ne samo izbrani). Odgovor na to vprašanje je v regresiji ciljev, ki iz prejšnjih ciljev, pogojev za akcijo in učinkov akcije izračuna nove cilje. Od tu naprej nas zanimajo le ti novi cilji, saj če jih uresničimo, bomo na koncu z izbrano akcijo rešili vse začetne cilje. Postopek se rekurzivno ponavlja dokler ne dobimo množice ciljev, ki so izpolnjeni že v začetni poziciji.

Algoritem (na nivoju i; na začetku so cilji $goals(0)$ enaki končnim ciljem):

1. Če so vsi cilji v $goals(i)$ resnični v začetnem stanju, končamo in vse akcije izvedemo v obratnem vrstnem redu. Če $goals(i)$ ni možno doseči (nemogoči cilji ali ni primerne akcije), se vrnemo v prostoru stanj in poskušamo najti rešitev po drugi poti.

2. Če cilji $goals(i)$ še niso uresničeni in so izvedljivi, izberemo cilj iz $goals(i)$ in akcijo A, ki doda ta cilj in regresiramo cilje po naslednji formuli:

$$goals(i+1) = goals(i) \cup conditions(A) \setminus adds(A)$$

Pri tem moramo paziti, da A ne izbriše trenutnega cilja (v $del(A)$ ni cilja iz $goals(i)$, oz. presek $del(A)$ in $goals(i)$ je prazna množica).

Naloga g: simuliranje algoritma z regresiranjem ciljev

Za zagotavljanje najkrajše rešitve uporabimo iterativno poglabljanje. Začnemo pri $D = 3$, globini $D = 1$ in $D = 2$ zaradi preglednosti izpustimo.

$goals(0) = [on(a, b), on(b, c)]$

Ali so cilji $on(a, b), on(b, c)$ izpolnjeni v začetni poziciji? Ne. Izberemo cilj: $on(a, b)$. Akcija: $move(a, From, b)$, predpogoji so:

$conditions(move(a, From, b)) = [clear(a), clear(b), on(a, From)]$.

Možne vrednosti za $From$ so: $1, 2, 3, 4, c$. Nastavimo $From = 1$.

$adds(move(a, 1, b)) = [clear(1), on(a, b)]$
$dels(move(a, 1, b)) = [clear(b), on(a, 1)]$

Akcijo lahko izvedemo, ker $dels$ ne vsebuje cilja iz $goals(0)$.

Regresija ciljev:

$goals(1) = goals(0) \cup conditions(move(a,1,b)) \setminus adds(move(a,1,b)) =$
$= [on(a,b), on(b,c)] \cup [clear(a), clear(b), on(a,1)] \setminus [clear(1), on(a,b)] =$
$= [clear(a), clear(b), on(a,1), on(b,c)].$

Regresirane cilje interpretiramo takole: če lahko pridemo v stanje, kjer velja $goals(1)$, bomo z akcijo $move(a,1,b)$ prišli v stanje, kjer velja $goals(0)$. Postopek nadaljujemo, dokler $goals(i)$ niso izpolnjeni v začetnem stanju.

$$goals(1) = [clear(a), clear(b), on(a,1), on(b,c)]$$

Ali so novi cilji $goals(1)$ izpolnjeni v začetni poziciji? Ne. Izberemo cilj: $clear(a)$ iz $goals(1)$ (po vrsti) in izberemo akcijo $move(X,a,To)$. Predpogoji so:

$$conditions(move(X,a,To)) = [clear(X), clear(To), on(X,a)],$$

možne vrednosti za X in To so: $(c,2)$,$(c,4)$, (c,b), $(c,1)$, itd. Izberemo $X = c$, $To = 2$.

$adds(move(c,a,2)) = [clear(a), on(c,2)]$
$dels(move(c,a,2)) = [clear(2), on(c,a)]$

Akcijo lahko izvedemo, ker $clear(2)$ in $on(c,a)$ nista v trenutni množici ciljev $goals(1)$.

Regresija ciljev:

$goals(2) = goals(1) \cup conditions(move(c,a,2)) \setminus adds(move(c,a,2)) =$
$= [clear(a), clear(b), on(a,1), on(b,c)] \cup [clear(c), clear(2), on(c,a)] \setminus$
$[clear(a), on(c,2)] =$
$= [clear(c), clear(2), on(c,a), clear(b), on(a,1), on(b,c)].$

Tu ne moremo nadaljevati, saj cilj ni izvedljiv! Hkrati ni možno doseči ciljev $on(b,c)$ in $clear(c)$.

Zdaj lahko poskusimo z drugimi vrednostmi spremenljivk X in To, vendar ne bi našli rešitve v treh ali manj korakih. Vrnemo se korak nazaj; izbrati moramo nov cilj.

Cilja $clear(b)$ in $on(a,1)$ v $goals(1)$ sta v začetnem stanju že resnična, izberemo cilj: $on(b,c)$. Akcija $move(b,From,c)$, predpogoji:

$$conditions(move(b,From,c)) = [clear(b), clear(c), on(b,From)].$$

Možne vrednosti za $From$ so: $3, 1, 2, 4, a$. Nastavimo $From = 3$.

$adds(move(b,3,c)) = [clear(3), on(b,c)]$
$dels(move(b,3,c)) = [clear(c), on(b,3)]$

V *dels* ni relacije, ki bi bila tudi v trenutnih ciljih, torej lahko izvedemo akcijo.

Regresija ciljev:

$goals(2) = goals(1) \cup conditions(move(b,3,c)) \setminus adds(move(b,3,c)) =$
$= [clear(a), clear(b), on(a,1), on(b,c)] \cup [clear(b), clear(c), on(b,3)] \setminus$
$[clear(3), on(b,c)] = = [clear(a), clear(b), clear(c), on(a,1), on(b,3)].$

$$goals(2) = [clear(a), clear(b), clear(c), on(a,1), on(b,3)]$$

Ali so cilji $goals(2)$ resnični v začetnem stanju? Ne. Izberemo cilj: $clear(a)$, izberemo akcijo $move(X,a,To)$, njeni predpogoji so:

$$conditions(move(X,a,To)) = [clear(X), clear(To), on(X,a)],$$

vrednosti za X in To so: $(c,2), (c,4), (c,b), ...$ Nastavimo vrednosti $X = c$, $To = 2$.

$adds(move(c,a,2)) = [clear(a), on(c,2)]$
$dels(move(c,a,2)) = [clear(2), on(c,a)]$

Med cilji v $goals(2)$ in $dels(move(c,a,2))$ ni konflikta .

Regresija ciljev:

$goals(3) = goals(2) \cup conditions(move(c,a,2)) \setminus adds(move(c,a,2)) =$
$= [clear(a), clear(b), clear(c), on(a,1), on(b,3)] \cup$
$[clear(c), clear(2), on(c,a)] \setminus [clear(a), on(c,2)] =$
$= [clear(b), clear(c), on(a,1), on(b,3), clear(2), on(c,a)].$

$$goals(3) = [clear(b), clear(c), on(a,1), on(b,3), clear(2), on(c,a)]$$

Opazimo, da so ti cilji resnični v začetnem stanju. Zdaj le še izvedemo akcije v obratnem vrstnem redu. Rešitev je:

1. $move(c,a,2)$
2. $move(b,3,c)$
3. $move(a,1,b)$

2.2 Razširjen svet kock

V tej nalogi bomo svet kock iz prejšnje naloge razširili z dodatnimi akcijami.

a) Predpostavimo, da lahko robot porine celoten stolp levo ali desno. Pri tem mora imeti prosto ustrezno polje (levo oz. desno), kamor premika stolp. Opišite akciji $pushLeft$ in $pushRight$, ki predstavljata levi in desni premik.

b) Imejmo spretnega robota, ki zna zagrabiti dve zgornji kocki na istem stolpcu in ju zamenjati. Definirajte akcijo $swap(X, Y, From)$, ki kocki zamenja.

c) Robot lahko prime dve kocki in ju odnese na drugo polje. Definirajte akcijo $move2(X, Y, From, To)$.

d) Kako bi posodobljen robot z novimi akcijami reševal problem opisan na začetku tega poglavja? Napišite vse možne akcije, s katerimi bi poskušal pri prvem cilju $on(a, b)$? Kakšni so pogoji za izbrane akcije? Določite smiselne vrednosti spremenljik, ki naj jih algoritem najprej poskusi. Uporabljajte planiranje brez regresije ciljev.

e) Poišči množico regresiranih ciljev $rgoals$ skozi akcije $move(a, 1, b)$, $swap(b, a, 1)$ in $move2(c, a, 3, b)$, začetni cilji so $goals = [on(a, b), on(b, c)]$. Rešujemo cilj $on(a, b)$. Ali lahko akcije izvedemo?

Rešitev:

Naloga a: premikanje levo in desno

Akcija: $pushLeft(X, From, To)$
$conditions = [on(X, From), clear(To)]$
$adds = [on(X, To), clear(From)]$
$dels = [on(X, From), clear(To)]$
$constraints = [not\ block(To), not\ block(From), From = To + 1, From > 1]$

Akcija: $pushRight(X, From, To)$
$conditions = [on(X, From), clear(To)]$
$adds = [on(X, To), clear(From)]$
$dels = [on(X, From), clear(To)]$
$constraints = [not\ block(To), not\ block(From), From = To - 1, From < 4]$

Naloga b: zamenjava dveh kock

Akcija: $swap(X, Y, From)$
$conditions = [on(X, Y), clear(X), on(Y, From)]$

$adds = [on(Y, X), clear(Y), on(X, From)]$
$dels = [on(X, Y), clear(X), on(Y, From)]$
$constraints = [block(X), block(Y)]$

Naloga c: premikanje dveh kock

Akcija: $move2(X, Y, From, To)$
$conditions = [on(X, Y), clear(X), clear(To), on(Y, From)]$
$adds = [clear(From), on(Y, To)]$
$dels = [clear(To), on(Y, From)]$
$constraints = [X \neq Y, X \neq From, Y \neq From, X \neq To, Y \neq To, To \neq From, block(X), block(Y)]$

Naloga d: planiranje

Poskušal bi z vsemi akcijami, ki dodajo relacijo $on(a, b)$. To so akcije $move$, $swap$ in $move2$. Akciji $pushLeft$ in $pushRight$ nista primerni, čeprav imata med svojimi pozitivnimi učinki $on(X, To)$, saj drugi argument ne sme biti kocka.

Akcija $move(a, From, b)$, pogoji za akcijo so:

$[clear(a), clear(b), on(a, From)]$

Algoritem bi najprej poskusil s $From = 1$, saj je a v začetni poziciji na mestu 1.

Akcija $swap(b, a)$, pogoji so: $[on(b, a), clear(b)]$.

Akcija $move2(X, b, From, a)$, pogoji: $[on(X, b), clear(X), clear(a), on(b, From)]$. Najprej bi izbral vrednosti ($X = c, From = 3$). $From = 3$ bi izbral, ker je b na začetku na 3, c, ker je edina preostala kocka (X ne more biti ne a in ne b).

Naloga e: planiranje z regresijo ciljev

1. $move(a, 1, b)$
$rgoals = [clear(a), clear(b), on(a, 1), on(b, c)]$

Če dosežemo $rgoals$, bomo z akcijo $move(a, 1, b)$ dosegli $goals$.

2. $swap(b, a, 1)$
$rgoals = [on(b, a), on(a, 1), clear(b), on(b, c)]$

Akcije ne bomo mogli izvesti, saj so cilji neizvedljivi! Kocka b ne more biti hkrati na a in c.

3. $move2(c, a, 3, b)$
$rgoals = [on(c, a), clear(c), clear(b), on(a, 3), on(b, c)]$

2.3 Falcon

Falcon je vladno letalo[1], ki se ne uporablja prav dosti. Naš cilj bo narediti načrt leta, da bi s čim manj leti prepeljali slovenske ministre po evropskih državah. Na letalu je hkrati lahko poljubno število ministrov, a so ministri včasih skregani in takrat nočejo leteti skupaj. Skregana ministra ne smeta biti hkrati na letalu.

a) Opis akcij: V domeni imamo tri možne akcije: fly (premakne Falcona iz ene države v drugo), *embark* (vkrcanje ministra) in *disembark* (izkrcanje ministra). Opišite vse akcije z jezikom STRIPS. Na voljo imate relacije $incountry(Where, X)$, ki označuje, da je X v državi $Where$, $country(Where)$ vrne True, če je X država, $minister(X)$, če je X minister, $dislikes(X, Y)$ je True, če se ministra X in Y ne marata, in $onplane(X)$ označuje, da je minister X na letalu.

b) Planiranje Falcona: Imejmo začetno stanje: $\{incountry(france, m1), incountry(slovenia, falcon)\}$ in cilje $goals = [incountry(slovenia, m1), incountry(slovenia, falcon)]$. Rešite problem s planiranjem po principu sredstva-cilji in pri tem uporabite regresiranje ciljev. Poskusite nalogo rešiti s ščitenjem ciljev in brez. Ščitenje ciljev (*angl.* protecting goals) je preprečevanje algoritmu planiranja, da bi "porušil" že dosežene cilje.

c) Naj bo začetno stanje $[incountry(slovenia, m1), incountry(slovenia, falcon), onplane(m2), onplane(m3), dislikes(m2, m1)]$ in cilj je $onplane(m1)$. Zakaj s planiranjem ne moremo doseči cilja? Kako bi problem odpravili?

Rešitev:

Naloga a: opis akcij

Akcija: $fly(From, To)$
$conditions = [incountry(From, falcon)]$
$adds = [incountry(To, falcon)]$
$dels = [incountry(From, falcon)]$
$constraints = [country(From), country(To)]$

Akcija: $embark(Where, Who)$
$conditions = [incountry(Where, falcon), incountry(Where, Who)]$
$adds = [onplane(Who)]$
$dels = [incountry(Where, Who)]$

[1] V času pisanja te skripte je slovenska vlada vsekakor še imela to letalo.

$constraints =$
$[land(Where), minister(Who), not\ dislikes(Who, X), onplane(X)]$

Akcija: $disembark(Where, Who)$
$conditions = [incountry(Where, falcon), onplane(Who)]$
$adds = [incountry(Where, Who)]$
$dels = [onplane(Who)]$
$constraints = [land(Where), minister(Who)]$

Dodatno vprašanje: ali bi znali to predstavitev posplošiti za več vladnih letal?

Naloga b: planiranje poletov s Falconom

S ščitenjem ciljev. Izberimo cilj: $incountry(slovenia, m1)$. To relacijo doda le akcija $disembark(slovenia, m1)$, ki ima predpogoje:

$conditions(disembark(slovenia, m1)) =$
$[incountry(slovenia, falcon), onplane(m1)]$

Regresirani cilji so $[incountry(slovenia, falcon), onplane(m1)]$.

Izberemo cilj $onplane(m1)$, ki še ni izpolnjen. Akcija $embark(Where, m1)$ s predpogoji

$conditions(embark(Where, m1)) =$
$[incountry(Where, falcon), incountry(Where, m1)]$

Za $Where$ bi bilo tu najbolje vzeti $france$, saj se minister $m1$ tam nahaja. Problem je, da potem dobimo neizvedljive regresirane cilje

$[incountry(france, falcon), incountry(france, m1), incountry(france, falcon)$.

Letalo ne more biti hkrati na dveh mestih.

Alternativna možnost je, da izberemo cilj $incountry(slovenia, falcon)$, vendar to s ščitenjem ciljev ni možno. S ščitenjem ciljev ne najdemo rešitve.

Brez ščitenja ciljev. Izberemo cilj $incountry(slovenia, falcon)$ in akcijo: $fly(From, slovenia)$ s predpogoji:

$conditions(fly(From, slovenia)) = [incountry(From, falcon)]$

$From$ je lahko katerakoli država, rešitev bomo našli pri $From = france$.

Regresirani cilji so $[incountry(france, falcon), onplane(m1)]$

Izberemo cilj $onplane(m1)$. Akcija $embark(Where, m1)$, kjer je $Where = france$. Novi regresirani cilji so $[incountry(france, falcon), incountry(france, m1)]$.

Preostane nam le še cilj $incountry(france, falcon)$. Akcija: $fly(From, france)$. Pri vrednosti $From = slovenia$ lahko akcijo izvedemo v začetnem stanju. Rezultat planiranja je:
$fly(slovenia, france)$,
$embark(france, m1)$,
$fly(france, slovenia)$,
$disembark(slovenia, m1)$.

Naloga c

Cilj $onplane(m1)$. Akcija: $embark(slovenia, m1)$

$conditions = [incountry(slovenia, falcon), incountry(slovenia, m1)]$

Pogoji so izpolnjeni, a akcije ne moremo izvesti zaradi omejitev
$not\ dislikes(Who, X), onplane(X)$.

Omejitev bi morali zapisati kot pogoj. Potem bi pogoji ne bili izpolnjeni in algoritem planiranja bi jih poskusil izpolniti. Opis akcije *embark* spremenimo v:

Akcija: $embark(Where, Who)$
$conditions = [incountry(Where, falcon), incountry(Where, Who),$
$dislikes(Who, X), not\ onplane(X)]$
$adds = [onplane(Who)]$
$dels = [incountry(Where, Who)]$
$constraints = [land(Where), minister(Who)]$

Poglavje 3

Strojno učenje

3.1 Dva atributa in razred

Dana je tabela učnih primerov z dvojiškima atributoma A in B in dvojiškim razredom R.

A	B	R
0	0	0
0	0	1
0	0	0
0	1	1
0	1	0
0	1	1
1	0	0
1	0	0
1	0	0
1	1	1
1	1	1
1	1	1

a) Brez računanja določi informacijski prispevek in razmerje inf. prispevka atributa A. Odgovor utemelji.

b) Oceni ali sta inf. prispevek in razmerje inf. prispevka atributa B večja, manjša ali enaka kot pri atributu A.

c) Izračunaj klasifikacijsko napako drevesa, ki je ekvivalentno pravilu: IF $B = 1$ THEN $R = 1$ ELSE $R = 0$. Uporabi Laplaceovo oceno verjetnosti.

d) Iz atributov A in B naredimo nov atribut AB, ki ga definiramo takole: $AB = 1$, kadar je $A = B = 1$, sicer je $AB = 0$. S pomočjo tega atributa si kot drevo lahko predstavljamo naslednje pravilo: IF $A = 1 \wedge B = 1$ THEN $R = 1$ ELSE $R = 0$. Ocenite njegovo klasifikacijsko točnost z uporabo m-ocene. Naj bo $m = 5$, apriorne verjetnosti pa izračunajte z relativno frekvenco na celotni učni množici.

Rešitev:

a) Porazdelitev obeh razredov, $R = 0$ in $R = 1$ je pri $A = 0$ in $A = 1$ enaka:
$A = 0$: $[3, 3]$
$A = 1$: $[3, 3]$
Informacijski prispevek atributa A je torej 0, iz tega pa sledi, da je tudi razmerje inf. prispevka A enako 0.

b) Ker je inf. prispevek A enak 0, ima B lahko le večji ali enak, gotovo pa ne manjši inf. prispevek. Pogledamo porazdelitve razreda pri obeh vrednostih atributa B:
$B = 0$: $[5, 1]$
$B = 1$: $[1, 5]$
Iz neenakomerne porazdelitve vidimo, da inf. prispevek B gotovo ni 0, iz česar sledi, da ima B večji inf. prispevek kot A in posledično tudi večje razmerje inf. prispevka.

c) Napaka pravila IF $B = 1$ THEN $R = 1$ ELSE $R = 0$ je:

$$e = 1-(P(B = 1)\ P_{\text{Laplace}}(R = 1|B = 1)+P(B = 0)\ P_{\text{Laplace}}(R = 0|B = 0)).$$

$P(B = 1) = \frac{6}{12} = \frac{1}{2}$, $P_{\text{Laplace}}(R = 1|B = 1) = \frac{5+1}{6+2}$ (imamo 6 primerov z $B = 1$, od tega jih ima 5 $R = 1$)

$P(B = 0) = \frac{6}{12} = \frac{1}{2}$, $P_{\text{Laplace}}(R = 0|B = 0) = \frac{5+1}{6+2}$ (imamo 6 primerov z $B = 0$, od tega jih ima 5 $R = 0$)

Napaka je: $e = 1 - (\frac{1}{2}\frac{6}{8} + \frac{1}{2}\frac{6}{8}) = \frac{1}{4} = 0.25$.

d) Točnost izračunamo kot uteženo vsoto verjetnosti napovedanega razreda v posameznem listu drevesa:

$$\begin{aligned} t &= P(A = 1 \wedge B = 1)P(R = 1|A = 1 \wedge B = 1) + \\ &\quad +(1 - P(A = 1 \wedge B = 1))P(R = 0|\neg(A = 1 \wedge B = 1)) \end{aligned}$$

$P(A = 1 \wedge B = 1) = \frac{3}{12} = \frac{1}{4}$
$P(R = 1|A = 1 \wedge B = 1) = \frac{3+5*6/12}{3+5} = 0.74$
$P(R = 0|\neg(A = 1 \wedge B = 1)) = \frac{6+5*6/12}{9+5} = \frac{6+5/2}{14} = 0.607$

$t = 0.25 * 0.74 + 0.75 * 0.607 = 0.64$

3.2 Mišmaš

Kot vemo, pri Mišmašu že od nekdaj pečejo kruh miši. Včasih je bilo enostavno: črne miši so pekle črn kruh, bele belega, sive polbelega in rumene koruznega. Kriza tudi Miševu ni prizanesla. Polbeli in koruzni kruh so začasno prenehali peči, sive in rumene miši pa je Mišmaš skrivnostno prerazporedil na beli in črni kruh. S pomočjo spodnje tabele ugotovi, kako se je odločal. Zgradi odločitveno drevo z uporabo informacijskega prispevka (*angl. Information Gain*).

BARVA	REP	KLOBUK	KRUH
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
siva	dolg	ima	črn
siva	dolg	nima	črn
rumena	dolg	nima	črn
rumena	dolg	nima	črn
siva	dolg	ima	črn
siva	dolg	nima	črn
rumena	dolg	ima	bel
rumena	kratek	ima	bel
črna	dolg	ima	bel
siva	kratek	nima	bel
bela	dolg	ima	bel
bela	dolg	nima	bel
bela	dolg	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel

Rešitev:
Entropija razreda je:

$$H(\text{KRUH}) = -p(\text{bel})\log_2(\text{bel}) - p(\text{črn})\log_2(\text{črn}) = 1\text{bit},$$

kar sicer lahko izračunamo tudi na pamet, saj je razred enakomerno porazdeljen.

V koren drevesa bomo postavili tisti atribut, ki najbolj zmanjša nedoločenost H. Za vsak atribut izračunamo zmanjšano entropijo:

$$\text{IG(X)} = H(\text{KRUH}) - H_{\text{res}}(\text{X}).$$

$H_{\text{res}}(X)$ je preostala nedoločenost, če podatke razdelimo glede na atribut X. Izračunamo jo kot uteženo vsoto entropij posameznih vrednosti atributa X:

$H_{\text{res}}(X) = \sum_v p(v)H(v)$, kjer je: $H(v) = -\sum_r p(r|X = v)\log_2 p(r|X = v)$ prek vseh razredov r.
Imamo 5 črnih, 4 rumene, 6 belih in 5 sivih miši, torej:
$H_{\text{res}}(\text{BARVA}) = \frac{5}{20}H(\text{črna}) + \frac{4}{20}H(\text{rumena}) + \frac{6}{20}H(\text{bela}) + \frac{5}{20}H(\text{siva})$
$H(\text{črna}) = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.722$
$H(\text{rumena}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$
$H(\text{bela}) = -\frac{6}{6}\log_2\frac{6}{6} - \frac{0}{6}\log_2\frac{0}{6} = 0$
$H(\text{siva}) = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.722$
$H_{\text{res}}(\text{BARVA}) = 0.561$ in
$\text{IG}(\text{BARVA}) = H(\text{KRUH}) - H_{\text{res}}(\text{BARVA}) = 1 - 0.561 = 0.439.$

Podobno izračunamo IG(REP) in IG(KLOBUK):
$H_{\text{res}}(\text{REP}) = \frac{11}{20}H(\text{dolg}) + \frac{9}{20}H(\text{kratek})$
$H(\text{dolg}) = -\frac{5}{11}\log_2\frac{5}{11} - \frac{6}{11}\log_2\frac{6}{11} = 0.994$
$H(\text{kratek}) = -\frac{5}{9}\log_2\frac{5}{9} - \frac{4}{9}\log_2\frac{4}{9} = 0.991$
$H_{\text{res}}(\text{REP}) = 0.993$ in
$\text{IG}(\text{REP}) = H(\text{KRUH}) - H_{\text{res}}(\text{REP}) = 1 - 0.993 = 0.007.$

$H_{\text{res}}(\text{KLOBUK}) = \frac{10}{20}H(\text{ima}) + \frac{10}{20}H(\text{nima})$
$H(\text{ima}) = -\frac{8}{10}\log_2\frac{8}{10} - \frac{2}{10}\log_2\frac{2}{10} = 0.722$
$H(\text{nima}) = -\frac{2}{10}\log_2\frac{2}{10} - \frac{8}{10}\log_2\frac{8}{10} = 0.722$
$H_{\text{res}}(\text{KLOBUK}) = 0.722$ in
$\text{IG}(\text{KLOBUK}) = H(\text{KRUH}) - H_{\text{res}}(\text{KLOBUK}) = 1 - 0.722 = 0.278.$

Največji informacijski prispevek (IG) ima BARVA, zato ta atribut izberemo za koren drevesa, ki ga gradimo. Trenutno drevo je globine 1 in ima eno notranje vozlišče (BARVA) in štiri liste, v katerih se nahajajo učni primeri z ustreznimi vrednostmi atributa BARVA. Postopek računanja informacijskih prispevkov in izbire atributa ponovimo v vsakem od listov. Pri tem upoštevamo samo učne primere v vsakem listu. Ker smo barvo že določili, bomo izbirali samo med atributoma REP in KLOBUK. Za vsako barvo si naredimo ustrezno tabelo primerov in izračunajmo informacijska prispevka preostalih atributov:

BARVA = črna:

BARVA	REP	KLOBUK	KRUH
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	dolg	ima	bel

Porazdelitev razreda je 4 : 1 (4 črne miši pečejo črn kruh, 1 pa belega). Entropija razreda je:
$H(\text{KRUH}) = -p(\text{bel})\log_2(\text{bel}) - p(\text{črn})\log_2(\text{črn}) = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5} =$ 0.722bit

Informacijska prispevka atributov REP in KLOBUK za črne miši sta:

$H_{\text{res}}(\text{REP}) = \frac{1}{5}H(\text{dolg}) + \frac{4}{5}H(\text{kratek})$
$H(\text{dolg}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$
$H(\text{kratek}) = -\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4} = 0$
$H_{\text{res}}(\text{REP}) = 0$ in
$\text{IG}(\text{REP}) = H(\text{KRUH}) - H_{\text{res}}(\text{REP}) = 0.722 - 0 = 0.722.$

Ker je porazdelitev vrednosti pri obeh atributih enaka, je račun za KLOBUK isti: $\text{IG}(\text{KLOBUK}) = H(\text{KRUH}) - H_{\text{res}}(\text{KLOBUK}) = 0.722 - 0 = 0.722$.

Atributa REP in KLOBUK sta torej enakovredna in vseeno je, katerega izberemo za koren tega poddrevesa. To bi pravzaprav lahko ugotovili že brez računanja, ker sta atributa praktično identična.

BARVA = rumena:

BARVA	REP	KLOBUK	KRUH
rumena	dolg	nima	črn
rumena	dolg	nima	črn
rumena	dolg	ima	bel
rumena	kratek	ima	bel

Tudi pri rumenih miših zadostuje že metoda ostrega pogleda: rumene miši s klobukom pečejo bel kruh, tiste brez pa črnega. Glede na rep očitno ne moremo natančno določiti vrste kruha. Opaženo potrdimo z računanjem:

Porazdelitev razreda je 2 : 2 – tega še ne bomo računali, vemo že, da je njegova entropija 1bit.
$H_{\text{res}}(\text{REP}) = \frac{3}{4}H(\text{dolg}) + \frac{1}{4}H(\text{kratek})$

$H(\text{dolg}) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$
$H(\text{kratek}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$
$H_{\text{res}}(\text{REP}) = 0.689$ in
$\text{IG}(\text{REP}) = H(\text{KRUH}) - H_{\text{res}}(\text{REP}) = 1 - 0.689 = 0.311.$

$H_{\text{res}}(\text{KLOBUK}) = \frac{2}{4}H(\text{ima}) + \frac{2}{4}H(\text{nima})$
$H(\text{ima}) = -\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2} = 0$
$H(\text{nima}) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0$
$H_{\text{res}}(\text{KLOBUK}) = 0$ in
$\text{IG}(\text{KLOBUK}) = H(\text{KRUH}) - H_{\text{res}}(\text{REP}) = 1 - 0 = 1.$

BARVA = bela:

BARVA	REP	KLOBUK	KRUH
bela	dolg	ima	bel
bela	dolg	nima	bel
bela	dolg	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel

Bele miši očitno pečejo samo bel kruh. Iz tega sklepamo, da je entropija oz. nedoločenost enaka 0. Z drugimi besedami, nesmiselno bi bilo nadalje vejiti to vozlišče drevesa. Naše sklepanje potrdi tudi račun:

$$H(\text{KRUH}) = -p(\text{bel})\log_2(\text{bel}) - p(\text{črn})\log_2(\text{črn}) = -\frac{6}{6}\log_2\frac{6}{6} - \frac{0}{6}\log_2\frac{0}{6} = 0\text{bit}.$$

BARVA = siva:

BARVA	REP	KLOBUK	KRUH
siva	dolg	ima	črn
siva	dolg	nima	črn
siva	dolg	ima	črn
siva	dolg	nima	črn
siva	kratek	nima	bel

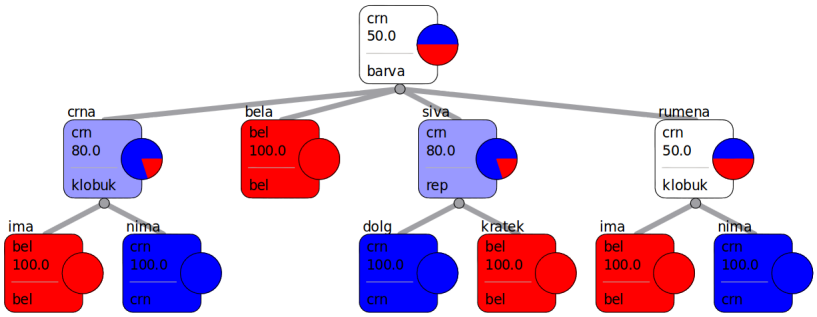
Ostale so še sive miši. Nedoločenost pri njih je $H(\text{KRUH}) = -p(\text{bel})\log_2(\text{bel}) - p(\text{črn})\log_2(\text{črn}) = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5} = 0.722\text{bit}.$

$H_{\text{res}}(\text{REP}) = \frac{4}{5}H(\text{dolg}) + \frac{1}{5}H(\text{kratek})$
$H(\text{dolg}) = -\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4} = 0$

$H(\text{kratek}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$
$H_{\text{res}}(\text{REP}) = 0$ in
$\text{IG}(\text{REP}) = H(\text{KRUH}) - H_{\text{res}}(\text{REP}) = 0.722 - 0 = 0.722.$

$H_{\text{res}}(\text{KLOBUK}) = \frac{2}{5}H(\text{ima}) + \frac{3}{5}H(\text{nima})$
$H(\text{ima}) = -\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2} = 0$
$H(\text{nima}) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$
$H_{\text{res}}(\text{KLOBUK}) = 0.551$ in
$\text{IG}(\text{KLOBUK}) = H(\text{KRUH}) - H_{\text{res}}(\text{KLOBUK}) = 0.722 - 0.551 = 0.171.$

Atribut REP ima višji informacijski prispevek, zato ga damo v koren poddrevesa. V naslednjem koraku spet razdelimo množico učnih primerov glede na trenutno drevo(slika 3.1). Ugotovimo, da so listi tega drevesa čisti in da nadaljna vejitev drevesa ni smiselna. Zato je drevo na sliki 3.1 že končno drevo. Še enkrat poudarimo, da sta REP in KLOBUK pri črnih miših enakovredna in da je pravilno tudi drevo, ki ima v korenu poddrevesa pri črnih miših atribut REP.



Slika 3.1: Mišmaš info gain in gini

3.3 Mišmaš z razmerjem inf. prispevka

Zgradite odločitveno drevo za 2. nalogo z uporabo razmerja informacijskega prispevka (*angl. Information Gain Ratio*).

Rešitev:

Večino dela za izračun razmerja informacijskega prispevka smo opravili že zgoraj. Izračunati moramo le še entropije atributov, s katerimi bomo delili njihove informacijske prispevke. Entropija atributa X je $H(X) = \sum_v p(X = v) \log_2 p(X = v)$, kjer je v vrednost atributa X.

$H(\text{BARVA}) = -p(\text{BARVA} = \text{črna}) \log_2 p(\text{BARVA} = \text{črna}) - p(\text{BARVA} = \text{rumena}) \log_2 p(\text{BARVA} = \text{rumena}) - p(\text{BARVA} = \text{bela}) \log_2 p(\text{BARVA} = \text{bela}) - p(\text{BARVA} = \text{siva}) \log_2 p(\text{BARVA} = \text{siva}) =$
$= -\frac{5}{20} \log_2 \frac{5}{20} - \frac{4}{20} \log_2 \frac{4}{20} - \frac{6}{20} \log_2 \frac{6}{20} - \frac{5}{20} \log_2 \frac{5}{20} = 1.985\text{bit}$
$H(\text{REP}) = -p(\text{REP} = \text{dolg}) \log_2 p(\text{REP} = \text{dolg}) - p(\text{REP} = \text{kratek}) \log_2 p(\text{REP} = \text{kratek}) = -\frac{11}{20} \log_2 \frac{11}{20} - \frac{9}{20} \log_2 \frac{9}{20} = 0.993\text{bit}$
$H(\text{KLOBUK}) = -p(\text{KLOBUK} = \text{ima}) \log_2 p(\text{KLOBUK} = \text{ima}) - p(\text{KLOBUK} = \text{nima}) \log_2 p(\text{KLOBUK} = \text{nima}) = -\frac{10}{20} \log_2 \frac{10}{20} - \frac{10}{20} \log_2 \frac{10}{20} = 1\text{bit}$
Razmerje informacijskega prispevka atributa X, $IGR(X)$, je

$$IGR(X) = IG(X)/H(X).$$

Na celotni množici podatkov je:
$IGR(\text{BARVA}) = IG(\text{BARVA})/H(\text{BARVA}) = 0.439/1.985 = 0.221$
$IGR(\text{REP}) = IG(\text{REP})/H(\text{REP}) = 0.007/0.993 = 0.007$
$IGR(\text{KLOBUK}) = IG(\text{KLOBUK})/H(\text{KLOBUK}) = 0.278/1 = 0.278$

Glede na IGR je za koren najbolj primeren atribut KLOBUK. V nadaljevanju postopamo kot prej. Začetno množico podatkov razdelimo glede na vrednosti atributa KLOBUK. Tako za KLOBUK=ima kot za KLOBUK=nima moramo izračunati informacijska prispevka atributov BARVA in REP.

KLOBUK = ima:

BARVA	REP	KLOBUK	KRUH
siva	dolg	ima	črn
siva	dolg	ima	črn
rumena	dolg	ima	bel
rumena	kratek	ima	bel
črna	dolg	ima	bel
bela	dolg	ima	bel
bela	dolg	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel

$H(\text{KRUH}) = -p(\text{bel})\log_2(\text{bel}) - p(\text{črn})\log_2(\text{črn}) = -\frac{8}{10}\log_2\frac{8}{10} - \frac{2}{10}\log_2\frac{2}{10} = 0.722\text{bit}$.

$H_{\text{res}}(\text{BARVA}) = \frac{1}{10}H(\text{črna}) + \frac{2}{10}H(\text{rumena}) + \frac{5}{10}H(\text{bela}) + \frac{2}{10}H(\text{siva})$
$H(\text{črna}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$
$H(\text{rumena}) = -\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2} = 0$
$H(\text{bela}) = -\frac{5}{5}\log_2\frac{5}{5} - \frac{0}{5}\log_2\frac{0}{5} = 0$
$H(\text{siva}) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0$
$H_{\text{res}}(\text{BARVA}) = 0$ in
$\text{IG}(\text{BARVA}) = H(\text{KRUH}) - H_{\text{res}}(\text{BARVA}) = 0.722 - 0 = 0.722$.

$H(\text{BARVA}) = -\frac{2}{10}\log_2\frac{2}{10} - \frac{2}{10}\log_2\frac{2}{10} - \frac{1}{10}\log_2\frac{1}{10} - \frac{5}{10}\log_2\frac{5}{10} = 1.761\text{bit}$
$IGR(\text{BARVA}) = IG(\text{BARVA})/H(\text{BARVA}) = 0.722/1.761 = 0.410$

$H_{\text{res}}(\text{REP}) = \frac{6}{10}H(\text{dolg}) + \frac{4}{10}H(\text{kratek})$
$H(\text{dolg}) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.918$
$H(\text{kratek}) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$
$H_{\text{res}}(\text{REP}) = 0.551$ in
$\text{IG}(\text{REP}) = H(\text{KRUH}) - H_{\text{res}}(\text{REP}) = 0.722 - 0.551 = 0.171$.

$H(\text{REP}) = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10} = 0.971\text{bit}$
$IGR(\text{REP}) = IG(\text{REP})/H(\text{REP}) = 0.171/0.971 = 0.176$

Izračun pokaže, da je po kriteriju razmerja informacijskega prispevka BARVA boljši atribut, zato ga damo v koren poddrevesa.

KLOBUK=nima:

BARVA	REP	KLOBUK	KRUH
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
siva	dolg	nima	črn
rumena	dolg	nima	črn
rumena	dolg	nima	črn
siva	dolg	nima	črn
siva	kratek	nima	bel
bela	dolg	nima	bel

$H(\text{KRUH}) = -p(\text{bel})\log_2(\text{bel}) - p(\text{črn})\log_2(\text{črn}) = -\frac{2}{10}\log_2\frac{2}{10} - \frac{8}{10}\log_2\frac{8}{10} = 0.722\text{bit}$.

$H_{\text{res}}(\text{BARVA}) = \frac{4}{10}H(\text{črna}) + \frac{2}{10}H(\text{rumena}) + \frac{1}{10}H(\text{bela}) + \frac{3}{10}H(\text{siva})$
$H(\text{črna}) = -\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4} = 0$
$H(\text{rumena}) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0$
$H(\text{bela}) = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$
$H(\text{siva}) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$
$H_{\text{res}}(\text{BARVA}) = 0.276$ in
$\text{IG}(\text{BARVA}) = H(\text{KRUH}) - H_{\text{res}}(\text{BARVA}) = 0.722 - 0.276 = 0.446$.

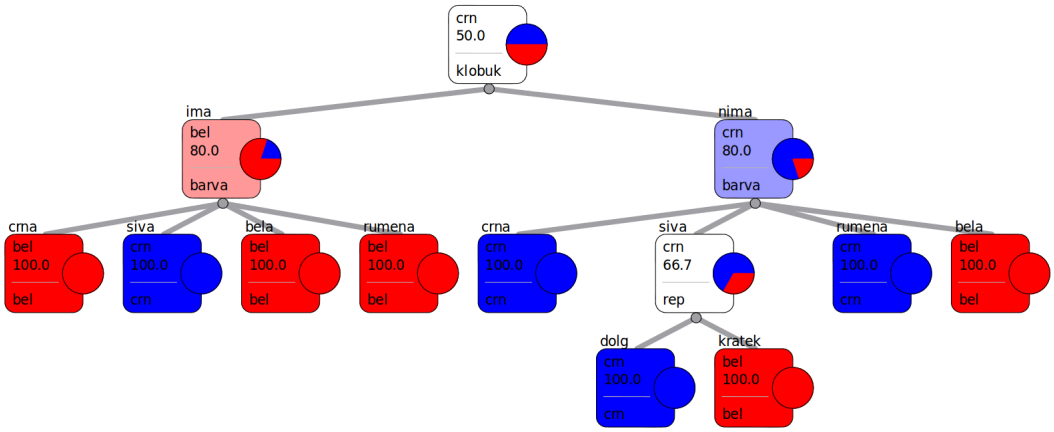
$H(\text{BARVA}) = -\frac{4}{10}\log_2\frac{4}{10} - \frac{3}{10}\log_2\frac{3}{10} - \frac{2}{10}\log_2\frac{2}{10} - \frac{1}{10}\log_2\frac{1}{10} = 1.846\text{bit}$
$IGR(\text{BARVA}) = IG(\text{BARVA})/H(\text{BARVA}) = 0.446/1.846 = 0.242$

$H_{\text{res}}(\text{REP}) = \frac{5}{10}H(\text{dolg}) + \frac{5}{10}H(\text{kratek})$
$H(\text{dolg}) = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.722$
$H(\text{kratek}) = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.722$
$H_{\text{res}}(\text{REP}) = 0.722$ in
$\text{IG}(\text{REP}) = H(\text{KRUH}) - H_{\text{res}}(\text{REP}) = 0.722 - 0.722 = 0$.

$H(\text{REP}) = -\frac{5}{10}\log_2\frac{5}{10} - \frac{5}{10}\log_2\frac{5}{10} = 1\text{bit}$
$IGR(\text{REP}) = IG(\text{REP})/H(\text{REP}) = 0/1 = 0$

Atribut BARVA je spet bolje ocenjen od atributa REP, vendar sive miši brez klobuka še vedno pečejo različen kruh, zato gradnja našega drevesa še ni končana. Ker pa smo drevo vejili že po dveh atributih, nam ostane samo še eden, REP. Zato računanje ni potrebno. Vidimo, da siva miš brez klobuka in

s kratkim repom peče bel kruh, oni dve z dolgim repom pa črnega. Končno drevo za naš primer je na sliki 3.2. Prav lahko bi se zgodilo, da bi v listih ne dobili čistih porazdelitev – če zmanjka atributov za nadaljne vejitve, se gradnja drevesa ne glede na to ustavi.



Slika 3.2: Mišmaš gain ratio

Drevesi 3.1 in 3.2 sta različni zato, ker informacijski prispevek in razmerje informacijskega prispevka različno rangirata atribute. Razlog za to je, da informacijski prispevek daje prednost večvrednostnim atributom, torej atributu BARVA pred atributom REP.

3.4 Mišmaš z Gini indeksom

Rešite 2. nalogo še z uporabo Gini indeksa.
Rešitev:
Gini razreda je: $Gini(\text{KRUH}) = 1-p(\text{bel})^2-p(\text{črn})^2 = 1-0.5^2-0.5^2 = 0.5$.

V koren drevesa bomo postavili tisti atribut, ki najbolj zmanjša *Gini* razreda. Za vsak atribut izračunamo njegov *Gini* prispevek:
$Gini(\text{BARVA}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{BARVA})$
Če podatke razdelimo glede na atribut X, je $Gini_{\text{res}}(X) = \sum_{v \in X} p(v)Gini(v)$, kjer je: $Gini(v) = 1 - \sum_r p(r|X = v)^2$ prek vseh razredov r.

Imamo 5 črnih, 4 rumene, 6 belih in 5 sivih miši, torej:
$Gini_{\text{res}}(\text{BARVA}) = \frac{5}{20}Gini(\text{črna})+\frac{4}{20}Gini(\text{rumena})+\frac{6}{20}Gini(\text{bela})+\frac{5}{20}Gini(\text{siva})$
$Gini(\text{črna}) = 1 - (\frac{1}{5})^2 - (\frac{4}{5})^2 = 0.320$
$Gini(\text{rumena}) = 1 - (\frac{2}{4})^2 - (\frac{2}{4})^2 = 0.5$
$Gini(\text{bela}) = 1 - (\frac{6}{6})^2 - (\frac{0}{6})^2 = 0$
$Gini(\text{siva}) = 1 - (\frac{1}{5})^2 - (\frac{4}{5})^2 = 0.320$
$Gini_{\text{res}}(\text{BARVA}) = 0.260$ in
$GiniGain(\text{BARVA}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{BARVA}) = 0.5 - 0.260 = 0.240$.

$Gini_{\text{res}}(\text{REP}) = \frac{11}{20}Gini(\text{dolg}) + \frac{9}{20}Gini(\text{kratek})$
$Gini(\text{dolg}) = 1 - (\frac{5}{11})^2 - (\frac{6}{11})^2 = 0.496$
$Gini(\text{kratek}) = 1 - (\frac{5}{9})^2 - (\frac{4}{9})^2 = 0.494$
$Gini_{\text{res}}(\text{REP}) = 0.495$ in
$GiniGain(\text{REP}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{REP}) = 0.5 - 0.495 = 0.005$.

$Gini_{\text{res}}(\text{KLOBUK}) = \frac{10}{20}Gini(\text{ima}) + \frac{10}{20}Gini(\text{nima})$
$Gini(\text{ima}) = 1 - (\frac{8}{10})^2 - (\frac{2}{10})^2 = 0.320$
$Gini(\text{nima}) = 1 - (\frac{2}{10})^2 - (\frac{8}{10})^2 = 0.320$
$Gini_{\text{res}}(\text{KLOBUK}) = 0.320$ in
$GiniGain(\text{KLOBUK}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{KLOBUK}) = 0.5 - 0.320 = 0.180$.

Največji Gini prispevek ima BARVA, zato jo damo v koren drevesa. Kot v prejšnjih dveh nalogah postopek ponovimo na listih trenutnega drevesa.

BARVA = črna:

BARVA	REP	KLOBUK	KRUH
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	kratek	nima	črn
črna	dolg	ima	bel

$$Gini(\text{KRUH}) = 1 - (p(\text{bel}))^2 - (p(\text{črn}))^2 = 1 - (\frac{1}{5})^2 - (\frac{4}{5})^2 = 0.320$$

$Gini_{\text{res}}(\text{REP}) = \frac{1}{5}Gini(\text{dolg}) + \frac{4}{5}Gini(\text{kratek})$
$Gini(\text{dolg}) = 1 - (\frac{1}{1})^2 - (\frac{0}{1})^2 = 0$
$Gini(\text{kratek}) = 1 - (\frac{0}{4})^2 - (\frac{4}{4})^2 = 0$
$Gini_{\text{res}}(\text{REP}) = 0$ in
$GiniGain(\text{REP}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{REP}) = 0.320 - 0 = 0.320.$

Ker je porazdelitev vrednosti pri obeh atributih enaka, je račun za KLOBUK isti: $GiniGain(\text{KLOBUK}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{KLOBUK}) = 0.320 - 0 = 0.320$.

Za atributa REP in KLOBUK velja enako kot prej – vseeno je, katerega izberemo za koren tega poddrevesa.

BARVA = rumena:

BARVA	REP	KLOBUK	KRUH
rumena	dolg	nima	črn
rumena	dolg	nima	črn
rumena	dolg	ima	bel
rumena	kratek	ima	bel

$$Gini(\text{KRUH}) = 1 - (p(\text{bel}))^2 - (p(\text{črn}))^2 = 1 - (\frac{2}{4})^2 - (\frac{2}{4})^2 = 0.5$$

$Gini_{\text{res}}(\text{REP}) = \frac{3}{4}Gini(\text{dolg}) + \frac{1}{4}Gini(\text{kratek})$
$Gini(\text{dolg}) = 1 - (\frac{1}{3})^2 - (\frac{2}{3})^2 = 0.444$
$Gini(\text{kratek}) = 1 - (\frac{1}{1})^2 - (\frac{0}{1})^2 = 0$
$Gini_{\text{res}}(\text{REP}) = 0.333$ in
$GiniGain(\text{REP}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{REP}) = 0.5 - 0.333 = 0.167.$

$Gini_{\text{res}}(\text{KLOBUK}) = \frac{2}{4}Gini(\text{ima}) + \frac{2}{4}Gini(\text{nima})$
$Gini(\text{ima}) = 1 - (\frac{2}{2})^2 - (\frac{0}{2})^2 = 0$
$Gini(\text{nima}) = 1 - (\frac{0}{2})^2 - (\frac{2}{2})^2 = 0$
$Gini_{\text{res}}(\text{KLOBUK}) = 0$ in
$GiniGain(\text{KLOBUK}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{KLOBUK}) = 0.5 - 0 = 0.5.$

BARVA = bela:

BARVA	REP	KLOBUK	KRUH
bela	dolg	ima	bel
bela	dolg	nima	bel
bela	dolg	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel
bela	kratek	ima	bel

Sklepamo tako kot v nalogi 3.2.

BARVA = siva:

BARVA	REP	KLOBUK	KRUH
siva	dolg	ima	črn
siva	dolg	nima	črn
siva	dolg	ima	črn
siva	dolg	nima	črn
siva	kratek	nima	bel

$$Gini(\text{KRUH}) = 1 - p(\text{bel})^2 - p(\text{črn})^2 = 1 - (\frac{1}{5})^2 - (\frac{4}{5})^2 = 0.320$$

$Gini_{\text{res}}(\text{REP}) = \frac{4}{5}Gini(\text{dolg}) + \frac{1}{5}Gini(\text{kratek})$
$Gini(\text{dolg}) = 1 - (\frac{0}{4})^2 - (\frac{4}{4})^2 = 0$
$Gini(\text{kratek}) = 1 - (\frac{1}{1})^2 - (\frac{0}{1})^2 = 0$
$Gini_{\text{res}}(\text{REP}) = 0$ in
$GiniGain(\text{REP}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{REP}) = 0.320 - 0 = 0.320.$

$Gini_{\text{res}}(\text{KLOBUK}) = \frac{2}{5}Gini(\text{ima}) + \frac{3}{5}Gini(\text{nima})$
$Gini(\text{ima}) = 1 - (\frac{2}{2})^2 - (\frac{0}{2})^2 = 0$
$Gini(\text{nima}) = 1 - (\frac{1}{3})^2 - (\frac{2}{3})^2 = 0.444$
$Gini_{\text{res}}(\text{KLOBUK}) = 0.267$ in

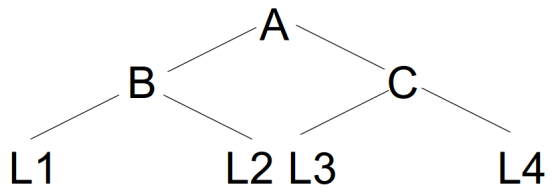
$GiniGain(\text{KLOBUK}) = Gini(\text{KRUH}) - Gini_{\text{res}}(\text{KLOBUK}) = 0.053.$

Atribut REP ima višji gini prispevek, zato ga damo v koren poddrevesa.

Tako kot v nalogi 3.2 tudi na tem mestu končamo gradnjo drevesa in drevo je natančno tako kot tisto, ki smo ga dobili z informacijskim prispevkom (slika 3.1). Ugotovimo, da sta gini in informacijski prispevek enako razvrstila atribute po pomembnosti.

3.5 Ocenjevanje verjetnosti

Dano je odločitveno drevo s seznami razredov tistih primerov, ki padejo v ustrezni list. $L_1 = [x, x, x, y, z]$, $L_2 = [x, y, y, y, y]$, $L_3 = [y, y, z, z, z, z]$, $L_4 = [w, w, w, w]$.



a) Oceni klasifikacijske točnosti v listih $L_1, \ldots, L_4$ z uporabo Laplaceove ocene verjetnosti.

b) Oceni točnost celotnega drevesa z Laplaceovo oceno verjetnosti.

c) Oceni točnost v listu L_2 z uporabo m-ocene, pri čemer naj bo m=7, apriorne verjetnosti razredov pa naj bodo: $p_0(x) = 0.1$, $p_0(y) = 0.05$, $p_0(z) = 0.8$ in $p_0(w) = 0.05$.

Opomba: *Klasifikacijska točnost* je verjetnost, da bo model pravilno klasificiral naključni testni primer. *Klasifikacijska napaka* je verjetnost napačne klasifikacijske.

Rešitev:

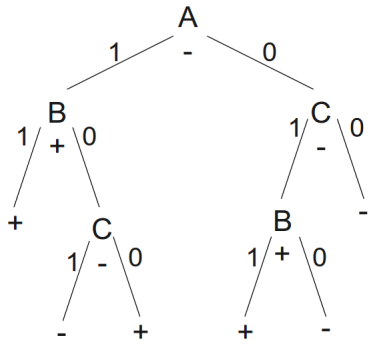
a) Iz podatkov razberemo, da je število razredov, $k = 4$. Število primerov večinskega razreda n in število vseh primerov v listu N ugotovimo za vsak list posebej. Ocene točnosti so torej:
$t(L_1) = (3 + 1)/(5 + 4) = 4/9$,
$t(L_2) = (4 + 1)/(5 + 4) = 5/9$,
$t(L_3) = (4 + 1)/(6 + 4) = 5/10$,
$t(L_4) = (4 + 1)/(4 + 4) = 5/8$.

b) Ocena točnosti celotnega drevesa, $t(A)$, je utežena vsota točnosti v listih. Utežimo jo z deležem primerov, ki padejo v posamezni list:
$t(A) = \frac{5}{20}t(L_1) + \frac{5}{20}t(L_2) + \frac{6}{20}t(L_3) + \frac{4}{20}t(L_4) = 0.525$.

c) Za vsak razred moramo izračunati oceno verjetnosti, ter nato klasificirati v najbolj verjeten razred.
razred x: $\frac{1+p_0(x)*7}{5+7} = 0.14$
razred y: $\frac{4+p_0(y)*7}{5+7} = 0.36$
razred z: $\frac{0+p_0(z)*7}{5+7} = 0.47$
razred w: $\frac{0+p_0(w)*7}{5+7} = 0.03$
Klasificiramo torej v razred z. Razlog je zelo visoka apriorna verjetnost tega razreda.

3.6 Rezanje: metoda zmanjševanja napake

Dano je klasifikacijsko drevo in rezalna tabela. Z rezanjem po metodi zmanjševanja napake poreži dano drevo.



A	B	C	R
0	0	1	−
0	0	1	+
0	1	1	−
0	0	0	−
0	1	0	−
0	1	1	−
0	1	1	−
0	1	1	−
0	0	1	+
1	1	1	+
1	1	0	+
1	0	1	+
1	0	1	+
1	0	0	+
1	0	1	−
1	0	0	−
1	0	0	−
1	0	0	−

Rešitev:

Rezanje klasifikacijskih dreves po metodi zmanjševanja napake (*angl. Reduced Error Prunning*) poteka od spodaj navzgor, t.j. od listov drevesa proti njegovemu korenu. Poleg drevesa postopek zahteva še rezalno tabelo podatkov. Pri gradnji drevesa se ti podatki ne upoštevajo, uporabljajo se samo pri rezanju. Z rezanjem začnemo v vozliščih tik nad listi. V vsakem vozlišču v izračunamo t.i. dobitek rezanja, $G(v)$, ki je definiran kot razlika med številom napačno klasificiranih primerov v poddrevesu T s korenom v v in številom napačnih klasifikacij pri v, če bi drevo tam porezali:

$$G(v) = \#\text{napak}_T - \#\text{napak}_v.$$

Režemo takrat, ko je $G(v) \geq 0$, torej če je napaka poddrevesa večja od napake v vozlišču v ali če sta napaki enaki, saj imamo ob enaki napaki raje manjše drevo.

Koristno je najprej vsakemu vozlišču drevesa pripisati porazdelitev razreda iz rezalne tabele (glej levo drevo na sliki 3.3). V oglatih oklepajih je najprej navedeno število primerov z razredom +, nato pa število primerov z razredom −. Sprotno preštevanje primerov hitro privede do napak. S pomočjo zgornje formule in izpisanih porazdelitev pa zlahka izračunamo dobitke rezanja za vsako vozlišče.

Začnimo pri vozlišču z atributom $(C, [3, 4], -)$. V tem vozlišču je torej 7 primerov, od tega 3 z razredom + in 4 z razredom −. Drevo v tem vozlišču klasificira v razred −. Poudarimo, da to ni vezano na porazdelitev primerov iz rezalne tabele, ampak je lastnost drevesa, ki je posledica porazdelitve v učnih podatkih, ki jih naloga ne navaja.

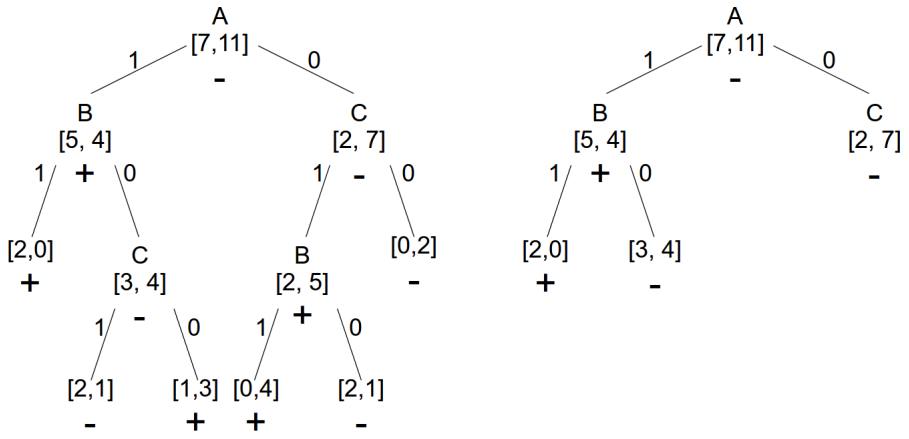
Število napak v tem vozlišču, $\#\text{napak}_v$, je 3, saj drevo napačno klasificira vse 3 primere z razredom +. Število napak v poddrevesu tega vozlišča je: $\#\text{napak}_T = 2 + 3 = 5$, 2 napaki v levem listu in 3 v desnem. Dobitek $G(v) = 5 - 3 = 2$, torej je dobro drevo porezati tako, da vozlišče $(C, [3, 4], -)$ postane list.

Naslednji kandidat za rezanje, po drevesu navzgor, je vozlišče $(B, [5, 4], +)$:
$\#\text{napak}_v = 4$
$\#\text{napak}_T = 0 + 3 = 3$
$G(v) = 3 - 4 = -1$
V tem vozlišču je bolje pustiti drevo tako kot je, kot ga porezati.

Nadaljujmo v vozlišču $(B, [2, 5], +)$:
$\#\text{napak}_v = 5$
$\#\text{napak}_T = 4 + 2 = 6$
$G(v) = 6 - 5 = 1$
Drevo tu porežemo in $(B, [2, 5], +)$ postane list.

Nadaljujemo v $(C, [2, 7], -)$:
$\#\text{napak}_v = 2$
$\#\text{napak}_T = 5 + 0 = 5$
$G(v) = 5 - 2 = 3$
Drevo porežemo in $(C, [2, 7], -)$ postane list.

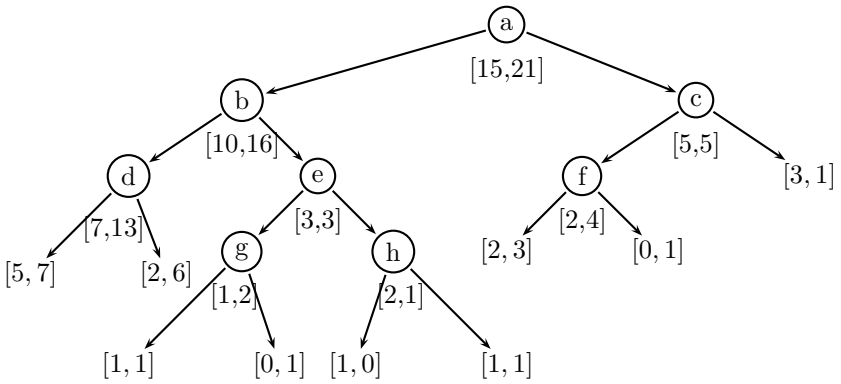
Končno drevo je na sliki 3.3 desno.



Slika 3.3: Pomožno drevo (levo) in rešitev (desno).

3.7 Rezanje: MEP

Porežite drevo na sliki 3.4 s postopkom MEP z uporabo Laplaceove ocene. V oglatih oklepajih je navedeno število primerov obeh razredov, npr. vozlišče b vsebuje 10 primerov prvega in 16 primerov drugega razreda. V vozlišču b je večinski drugi razred.



Slika 3.4

Rešitev:

V vsakem notranjem vozlišču v (govoriti o rezanju v listih ni smiselno) primerjamo statično napako e in vzvratno napako E. Statična napaka $e(v)$ je napaka drevesa v primeru, če v postane list (torej, če bi drevo porezali tik pod vozliščem v). Vzvratna napaka $E(v)$ je napaka, če drevesa ne porežemo. Kdaj naj torej drevo porežemo pod v? Napako hočemo minimizirati, torej ga porežemo takrat, ko je statična napaka manjša ali enaka od vzvratne napake. Zakaj tudi, če je enaka? Z vidika točnosti je vseeno ali drevo tu porežemo ali ne, ker pa je manjši model ponavadi tudi bolj razumljiv, ga porežemo.

Ponovimo najprej formulo za izračun Laplaceove ocene verjetnosti,

$$p = \frac{n+1}{N+k},$$

kjer je:

- n število primerov večinskega razreda v vozlišču
- N število vseh primerov v vozlišču
- k število razredov

V našem primeru je $k = 2$.

Vozlišče d:
Ker računamo napako, moramo ocenjeno verjetnost odšteti od 1 (glej nalogo 3). Statična napaka v vozlišču d je $e(d) = 1 - \frac{n+1}{N+2} = 1 - \frac{13+1}{20+2} = 8/22 = 0.363$. Za izračun vzvratne napake potrebujemo napako v levem in desnem listu d, ki ju bomo označili kar z d_L in d_D.
$e(d_L) = 1 - \frac{7+1}{12+2} = 6/14 = 0.428$
$e(d_D) = 1 - \frac{6+1}{8+2} = 3/10 = 0.3$
Vzvratna napaka je utežena vsota napak v obeh vozliščih. Utežimo ju z deležem primerov, ki pripadajo vsakemu listu. V levi list vozlišča d gre 12 od 20 primerov iz d, v desnega pa preostalih 8: $E(d) = \frac{12}{20}\ 0.428 + \frac{8}{20}\ 0.3 = 0.376$
Izračunali smo, da je statična napaka $e(d)$ manjša od vzvratne napake $E(d)$, torej je boljše, da drevo **porežemo** pod d.

Podobno izračunamo obe napaki za ostala vozlišča, pri čemer je pomembno, da gremo po drevesu od spodaj navzgor:
Vozlišče g:

$e(g) = 1 - \frac{2+1}{3+2} = 2/5 = 0.4$

$e(g_L) = 1 - \frac{1+1}{2+2} = 0.5$

$e(g_D) = 1 - \frac{1+1}{1+2} = 1/3 = 0.333$

$E(g) = \frac{2}{3}\ 0.5 + \frac{1}{3}\ 0.333 = 0.444$

$e(g) \leq E(g)$, torej režemo pod g.

Vozlišče h:

$e(h) = 1 - \frac{2+1}{3+2} = 2/5 = 0.4$

$e(h_L) = 1 - \frac{1+1}{1+2} = 1/3 = 0.333$

$e(h_D) = 1 - \frac{1+1}{2+2} = 0.5$

$E(h) = \frac{1}{3}\ 0.333 + \frac{2}{3}\ 0.5 = 0.444$

$e(h) \leq E(h)$, torej režemo pod h.

Vozlišče e:

$e(e) = 1 - \frac{3+1}{6+2} = 0.5$

$e(e_L) = e(g) = 0.4$

$e(e_D) = e(h) = 0.4$

$E(e) = \frac{1}{2}\ 0.4 + \frac{1}{2}\ 0.4 = 0.4$

$e(e) > E(e)$, torej **NE** režemo pod e.

Vozlišče f:

$e(f) = 1 - \frac{4+1}{6+2} = 3/8 = 0.375$

$e(f_L) = 1 - \frac{3+1}{5+2} = 3/7 = 0.429$

$e(f_D) = 1 - \frac{1+1}{1+2} = 1/3 = 0.333$

$E(f) = \frac{5}{6}\ 0.429 + \frac{1}{6}\ 0.333 = 0.412$

$e(f) \leq E(f)$, torej režemo pod f.

Vozlišče c:

$e(c) = 1 - \frac{5+1}{10+2} = 0.5$

$e(c_L) = e(f) = 0.375$

$e(c_D) = 1 - \frac{3+1}{4+2} = 1/3 = 0.333$

$E(c) = \frac{6}{10}\ 0.375 + \frac{4}{10}\ 0.333 = 0.358$

$e(c) > E(c)$, torej **NE** režemo pod c.

Vozlišče b:

$e(b) = 1 - \frac{16+1}{26+2} = 0.393$

$e(b_L) = e(d) = 0.363$ (ker smo pri d porezali, vzamemo tu statično napako vozlišča d)

$e(b_D) = E(e) = 0.4$ (pri e nismo rezali, zato vzamemo tu vzvratno napako)

$E(b) = \frac{20}{26}\ 0.363 + \frac{6}{26}\ 0.4 = 0.371$

$e(b) > E(b)$, torej **NE** režemo pod b.

Vozlišče a:

$e(a) = 1 - \frac{21+1}{36+2} = 0.421$

$e(a_L) = E(b) = 0.371$ (ker pri b nismo rezali, vzamemo tu vzvratno napako vozlišča b)

$e(a_D) = E(c) = 0.358$ (vzamemo vzvratno napako pri c, ker pod c nismo rezali)

$E(a) = \frac{26}{36}\ 0.371 + \frac{10}{36}\ 0.358 = 0.367$

$e(a) > E(a)$, torej **NE** režemo pod a.

3.8 Naivni Bayesov klasifikator

Dani so naslednji učni podatki:

X	Y	Z	R
0	1	a	0
0	1	a	0
1	0	b	0
1	0	b	0
2	1	b	0
1	1	b	1
1	1	b	1
1	0	b	1
2	0	a	1
2	1	a	1

Klasificirajte naslednje primere z naivnim Bayesom. Verjetnosti računajte z relativno frekvenco.

- $X = 1, Y = 1, Z = a$
- $X = 0, Y = 0, Z =?$
- $X = 2, Y = 0, Z = b$

Rešitev:
Z naivnim Bayesom klasificiramo tako, da za vsak razred c izračunamo produkt pogojnih verjetnosti danih vrednosti atributov in vse skupaj pomnožimo z verjetnostjo razreda. Na koncu klasificiramo v tisti razred, kjer ima zgornji produkt največjo vrednost:

$$R = argmax_{c \in R} P(c) \prod_{i=1}^{n} P(A_i|c)$$

- $X = 1, Y = 1, Z = a$

	$c = 0$	$c = 1$
$P(c)$	1/2	1/2
$P(X = 1\|c)$	2/5	3/5
$P(Y = 1\|c)$	3/5	3/5
$P(Z = a\|c)$	2/5	2/5
$\prod$	6/125	9/125

Primer $(X = 1, Y = 1, Z = a)$ klasificiramo v razred $c = 1$. Opozorimo, da izračunana vrednost R ni verjetnost $P(c = 1|X = 1, Y = 1, Z = a)$ - to verjetnost lahko izračunamo takole:

$$P(c = 1|X = 1, Y = 1, Z = a) = \frac{9/125}{6/125 + 9/125} = 0.6$$

- $X = 0, Y = 0, Z =?$

	$c = 0$	$c = 1$
$P(c)$	$1/2$	$1/2$
$P(X = 0\|c)$	$2/5$	0
$P(Y = 0\|c)$	$2/5$	$2/5$
$P(Z =?\|c)$	/	/
$\prod$	$2/25$	0

  Primer $(X = 0, Y = 0, Z =?)$ klasificiramo v razred $c = 0$. Ker vrednost atributa Z ni podana, tega atributa pri računanju ne upoštevamo.

- $X = 2, Y = 0, Z = b$

	$c = 0$	$c = 1$
$P(c)$	$1/2$	$1/2$
$P(X = 2\|c)$	$1/5$	$2/5$
$P(Y = 0\|c)$	$2/5$	$2/5$
$P(Z = b\|c)$	$3/5$	$3/5$
$\prod$	$3/125$	$6/125$

  Primer $(X = 2, Y = 0, Z = b)$ klasificiramo v razred $c = 1$.

V tej nalogi smo pogojne verjetnosti računali z relativno frekvenco, lahko pa bi uporabili Laplaceovo oceno verjetnosti ali m-oceno.

3.9 Loto

Po novoletnem žrebanju lota v Srečnem dolu nad Radovno so v loteriji s pomočjo svojih zvestih gledalcev naredili statistiko, da bi ugotovili, če so bili zadetki predvidljivi. V kratki anketi so gledalce vprašali, če so srečko kupili ali jim je bila podarjena (oz. so jo kupili in jo podarili), vprašali so jih po barvi las in pa če so spremljali žrebanje v neposrednem prenosu po televiziji. Odgovore povzema naslednja tabela:

DOBITEK	KUPIL SREČKO		BARVA LAS			TV		$\sum$
	DA	NE	SVETLA	TEMNA	RDEČA	DA	NE	
DA	3	777	650	50	80	779	1	780
NE	218	2	5	195	20	1	219	220
	221	779	655	245	100	780	220	1000

S pomočjo naivnega Bayesovega klasifikatorja ugotovite ali so bili med srečnimi dobitniki tudi:

a) Temnolasi srečnodolčan, ki ni kupil srečke, a je gledal žrebanje.

b) Blondinka, ki je kupila srečko in ni gledala žrebanja.

c) Rdečelaska, ki ni kupila srečke in ni gledala žrebanja.

Pogojne verjetnosti ocenjujte z m-oceno ($m = 5$), apriorne pa z relativno frekvenco.

Rešitev:[1]

a) P(DOBITEK | (NE, T, DA))

	DOBITEK = DA	DOBITEK = NE
apriorna	$P(DA) = 780/1000 = 0.78$	$P(NE) = 220/1000 = 0.22$
KUPIL SREČKO	$P(NE\|DA) = \frac{777+m779/1000}{780+m} = 0.994$	$P(NE\|NE) = \frac{2+m779/1000}{220+m} = 0.022$
BARVA LAS	$P(T\|DA) = \frac{50+m245/1000}{780+m} = 0.065$	$P(T\|NE) = \frac{195+m245/1000}{220+m} = 0.871$
TV	$P(DA\|DA) = \frac{779+m780/1000}{780+m} = 0.996$	$P(DA\|NE) = \frac{1+m780/1000}{220+m} = 0.018$
$\prod$	$0.78 \cdot 0.994 \cdot 0.065 \cdot 0.996 = 0.05$	$0.22 \cdot 0.022 \cdot 0.871 \cdot 0.018 = 7.6 \cdot 10^{-5}$

P(DA | (NE, T, DA)) > P(NE | (NE, T, DA))
Temnolasi srečnodolčan, ki ni kupil srečke, a je gledal žrebanje je bil bolj verjetno med srečnimi dobitniki kot ne.

[1]Zaradi utesnjenosti v tabelah krajšamo imena in vrednosti atributov.

b) P(DOBITEK | (DA, B, NE))

	DOBITEK = DA	DOBITEK = NE
apriorna	$P(DA) = 780/1000 = 0.78$	$P(NE) = 220/1000 = 0.22$
KUPIL SREČKO	$P(DA\|DA) = \frac{3+m221/1000}{780+m} = 0.005$	$P(DA\|NE) = \frac{218+m221/1000}{220+m} = 0.973$
BARVA LAS	$P(S\|DA) = \frac{650+m655/1000}{780+m} = 0.832$	$P(S\|NE) = \frac{5+m655/1000}{220+m} = 0.036$
TV	$P(NE\|DA) = \frac{1+m220/1000}{780+m} = 0.003$	$P(NE\|NE) = \frac{219+m220/1000}{220+m} = 0.978$
$\prod$	$0.78 \cdot 0.005 \cdot 0.832 \cdot 0.003 = 9.7 \cdot 10^{-6}$	$0.22 \cdot 0.022 \cdot 0.871 \cdot 0.018 = 0.008$

P(DA | (DA, B, NE)) < P(NE | (DA, B, NE))
Svetlolaska, ki je kupila srečko in ni gledala žrebanja najverjetneje ni bila med dobitniki.

c) P(DOBITEK | (NE, R, NE))

	DOBITEK = DA	DOBITEK = NE
apriorna	$P(DA) = 780/1000 = 0.78$	$P(NE) = 220/1000 = 0.22$
KUPIL SREČKO	$P(NE\|DA) = \frac{777+m779/1000}{780+m} = 0.994$	$P(NE\|NE) = \frac{2+m779/1000}{220+m} = 0.022$
BARVA LAS	$P(R\|DA) = \frac{80+m100/1000}{780+m} = 0.102$	$P(R\|NE) = \frac{20+m100/1000}{220+m} = 0.089$
TV	$P(NE\|DA) = \frac{1+m220/1000}{780+m} = 0.003$	$P(NE\|NE) = \frac{219+m220/1000}{220+m} = 0.978$
$\prod$	$0.78 \cdot 0.994 \cdot 0.102 \cdot 0.003 = 2.4 \cdot 10^{-4}$	$0.22 \cdot 0.022 \cdot 0.089 \cdot 0.978 = 4.2 \cdot 10^{-4}$

P(DA | (NE, R, NE)) < P(NE | (NE, R, NE))
Rdečelaska, ki ni kupila srečke in ni gledala žrebanja najverjetneje ni bila med dobitniki.

3.10 Nakup računalnika

Tržni analitik računalniškega podjetja raziskuje trg računalnikov. Analizirati želi dejavnike, ki vplivajo na odločitev potrošnikov za nakup bodisi računalnika PC ali MAC. Izluščil je naslednje dejavnike:

- *Navada.* V spletni anketi je bilo od 1000 vprašanih 600 takih, ki so tradicionalno uporabljali PC; od teh se je 100 odločilo, da bo njihov nov računalnik MAC. Skupaj so sodelujoči v anketi kupili 470 računalnikov MAC.
- *Trg* Raziskava svetovnega trga je pokazala, da evropejci večinoma kupujemo računalnike PC (71,5%), verjetnost, da je kupec iz EU pa je 35%. Več kupcev prihaja iz ZDA (40%), kjer pa prevladujejo računalniki MAC (95%); preostanek je azijski trg, kjer je razmerje PC:MAC = 1,5:1. V svetovnem merilu je delež računalnikov MAC 58%.
- *Izobrazba* Od tistih, ki so se pri novem računalniku odločili za PC, je študiralo 78% vprašanih, od tistih, ki so se odločili za MAC pa 91%.

Sestavi naivni Bayesov nomogram za zgornje podatke in ga komentiraj. Kateri dejavniki so bolj, kateri manj pomembni? Kaj najbolj pripomore k odločitvi za MAC in kaj k odločitvi za PC?

Rešitev:

Izberimo za ciljni razred $C_T = \text{PC}$. Za vsak atribut

$$A \in \{\text{Navada, Trg, Izobrazba}\}$$

potrebujemo pogojne verjetnosti posameznih vrednosti pri danem razredu $C \in \{\text{PC, MAC}\}$, da dobimo velikost vpliva $x_{A=v}$ na nomogramski osi atributa:

$$\ln \frac{P(A = v|C)}{P(A = v|\overline{C})}$$

- *Navada*

  Iz podatkov razberemo:

	nov = PC	nov = MAC	$\sum$
navada = PC		100	600
navada = MAC			
$\sum$		470	1000

Tabelo dopolnemo:

	nov = PC	nov = MAC	$\sum$
navada = PC	500	100	600
navada = MAC	30	370	400
$\sum$	530	470	1000

Iz tabele preberemo potrebne pogojne verjetnosti in izračunamo vrednosti

$$x_{\text{Navada=PC}} = \ln \frac{P(\text{Navada=PC}|\text{PC})}{P(\text{Navada=PC}|\text{MAC})} = \ln \frac{500/530}{100/470} = \ln(0.943/0.213) = 1.487$$

$$x_{\text{Navada=MAC}} = \ln \frac{P(\text{Navada=MAC}|\text{PC})}{P(\text{Navada=MAC}|\text{MAC})} = \ln \frac{30/530}{370/470} = \ln(0.057/0.787) = -2.625$$

- *Trg*

  Iz podatkov razberemo:
  $P(\text{PC}|\text{EU}) = 0.715$
  $P(\text{EU}) = 0.35$
  $P(\text{ZDA}) = 0.4$
  $P(\text{MAC}|\text{ZDA}) = 0.95$
  $P(\text{PC}|\text{AZIJA}) = 1.5/(1 + 1.5) = 0.6$
  $P(\text{MAC}|\text{AZIJA}) = 1/(1 + 1.5) = 0.4$
  $P(\text{MAC}) = 0.58$

  Iz teh podatkov izračunamo:

$$P(\text{PC}) = 1 - P(\text{MAC}) = 1 - 0.58 = 0.42$$
$$P(\text{EU}|\text{PC}) = P(\text{PC}|\text{EU}) \cdot P(\text{EU})/P(\text{PC}) = 0.715 \cdot 0.35/0.42 = 0.596$$
$$P(\text{MAC}|\text{EU}) = 1 - P(\text{PC}|\text{EU}) = 0.285$$
$$P(\text{EU}|\text{MAC}) = P(\text{MAC}|\text{EU}) \cdot P(\text{EU})/P(\text{MAC}) = 0.285 \cdot 0.35/0.58 = 0.172$$
$$P(\text{PC}|\text{ZDA}) = 1 - P(\text{MAC}|\text{ZDA}) = 0.05$$

$P(\text{ZDA}|\text{PC}) = P(\text{PC}|\text{ZDA}) \cdot P(\text{ZDA})/P(\text{PC}) = 0.05 \cdot 0.4/0.42 = 0.048$

$P(\text{ZDA}|\text{MAC}) = P(\text{MAC}|\text{ZDA}) \cdot P(\text{ZDA})/P(\text{MAC}) = 0.95 \cdot 0.4/0.58 = 0.655$

$$P(\text{AZIJA}) = 1 - (P(\text{EU}) + P(\text{ZDA}) = 0.25$$

$P(\text{AZIJA}|\text{PC}) = P(\text{PC}|\text{AZIJA}) \cdot P(\text{AZIJA})/P(\text{PC}) = 0.6 \cdot 0.25/0.42 = 0.357$

$P(\text{AZIJA}|\text{MAC}) = P(\text{MAC}|\text{AZIJA}) \cdot P(\text{AZIJA})/P(\text{MAC}) = 0.4 \cdot 0.25/0.58 = 0.172$

Vrednosti na nomogramski osi so:

$$x_{\text{Trg=EU}} = \ln(0.596/0.172) = 1.243$$

$$x_{\text{Trg=ZDA}} = \ln(0.048/0.655) = -2.613$$

$$x_{\text{Trg=AZIJA}} = \ln(0.357/0.172) = 0.73$$

- *Izobrazba*

  Iz podatkov razberemo:

$$P(\text{ŠTUDIRAL}|\text{PC}) = 0.78$$

$$P(\text{ŠTUDIRAL}|\text{MAC}) = 0.91$$

  Izračunati moramo še:

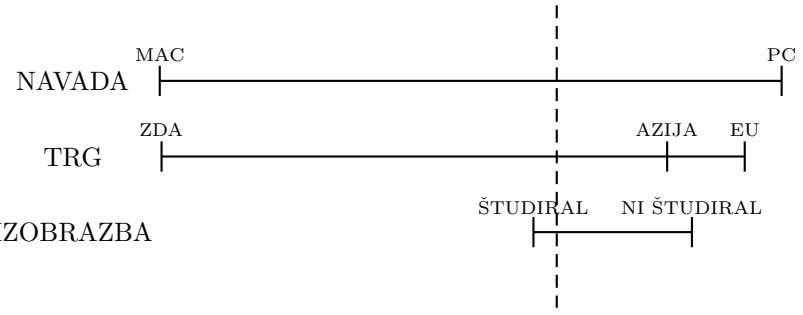
$$P(\text{NI ŠTUDIRAL}|\text{PC}) = 1 - 0.78 = 0.22$$

$$P(\text{NI ŠTUDIRAL}|\text{MAC}) = 1 - 0.91 = 0.09$$

  Vrednosti na nomogramski osi sta:

$$x_{\text{Izobrazba=ŠTUDIRAL}} = \ln(0.78/0.91) = -0.154$$

$$x_{\text{Izobrazba=NI ŠTUDIRAL}} = \ln(0.22/0.09) = 0.894$$

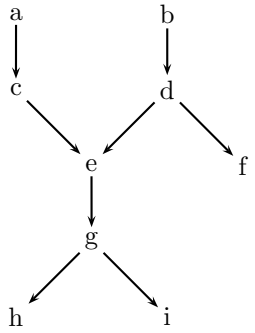


Poglavje 4

Bayesove mreže

4.1 Prva

Dan je usmerjen graf:



a) Katere verjetnosti morajo biti podane ob grafu, da bo dobro definirana Bayesova mreža?

b) Kakšen je pomen množice vozlišč, ki $d-$ločuje dani vozlišči?

c) Brez računanja oceni, katera verjetnost je večja in utemelji svojo odločitev:

$p(c|e)$ ali $p(c|de)$?
$p(c|e)$ ali $p(c|eg)$?
$p(c|e)$ ali $p(c|def)$?

Pri tem predpostavite, da je a pozitiven vzrok za c ($p(c|a) > p(c|\neg a)$) in b pozitiven vzrok za d ($p(d|b) > p(d|\neg b)$).

Rešitev:

a) Ob vsakem vozlišču mora biti podana njegova verjetnost pri danih starših:
$p(a)$
$p(b)$
$p(c|a), p(c|\neg a)$
$p(d|b), p(d|\neg b)$
$p(e|cd), p(e|c\neg d), p(e|\neg cd), p(e|\neg c\neg d)$
$p(f|d), p(f|\neg d)$
$p(g|e), p(g|\neg e)$
$p(h|g), p(h|\neg g)$
$p(i|g), p(i|\neg g)$

b) Naj množica S $d-$ločuje vozlišči v_1 in v_2. Potem sta v_1 in v_2 pogojno neodvisna pri dani množici S:

$$p(v_1, v_2|S) = p(v_1|S) * p(v_2|S).$$

c)
- $p(c|e) > p(c|de)$. Ker je c vzrok za e, se pri danem e poveča verjetnost za c. Če je dan tudi d, ki je alternativni vzrok za c pa se verjetnost za c zniža. Če imate glavobol (e), je bolj verjetno, da imate virus prehlada (c), kot da ga nimate. Če pa imate glavobol in dokazan meningitis (d), je verjetnost za prehlad manjša, saj je meningitis alternativni vzrok za glavobol.
- $p(c|e) = p(c|eg)$. Če poznamo e, potem prisotnost g ne vpliva na verjetnost c.
- $p(c|e) > p(c|def)$. $p(c|def) = p(c|de)$, kar pa se prevede na zgornji primer.

4.2 Izrazi pogojne verjetnosti

Za Bayesovo mrežo iz naloge 1 izrazi s podanimi verjetnostmi:

a) $p(c)$

b) $p(g|c,d)$

c) $p(g|c,i)$

d) $p(d|e,f)$

Rešitev:
Pri računanju v Bayesovih mrežah si pomagamo z naslednjimi pravili:

1. Verjetnost gotovega dogodka: $p(X|A_1,\ldots,X,\ldots,A_n) = 1$
2. Verjetnost nemogočega dogodka: $p(X|A_1,\ldots,\neg X,\ldots,A_n) = 0$
3. Verjetnost konjunkcije: $p(A \wedge B|C) = p(A|C)\; p(B|A \wedge C)$
4. Verjetnost negacije: $p(\neg X|C) = 1 - p(X|C)$
5. Bayesovo formula (če je Y naslednik od X in je Y vsebovan v pogojnem delu):
$$p(X|Y \wedge C) = p(X|C)\; \frac{p(Y|X \wedge C)}{p(Y|C)}$$
6. Če pogojni del ne vsebuje naslednika od X:
   a) če X nima staršev, je $p(X|C) = p(X)$, pri čemer je $p(X)$ podan.
   b) če ima X starše, je:
$$p(X|C) = \sum_{S \in \mathcal{P}_X} p(X|S)\; p(S|C),$$
   kjer je $\mathcal{P}_X$ množica vseh možnih stanj staršev od X.

a) $p(c) = p(a)p(c|a) + p(\neg a)p(c|\neg a)$

b) $p(g|c,d) \stackrel{6b}{=} p(g|e)p(e|c,d) + p(g|\neg e)p(\neg e|c,d)$
$p(e|c,d)$ je podana, $p(\neg e|c,d) \stackrel{4}{=} 1 - p(e|c,d)$.

c)

$$p(g|c,i) \overset{5}{=} p(g|c)\frac{p(i|g,c)}{p(i|c)}$$

$$p(g|c) \overset{6b}{=} p(g|e)p(e|c) + p(g|\neg e)p(\neg e|c)$$

$$\begin{aligned} p(e|c) \overset{6b}{=}\ & p(e|c,d)p(c,d|c) + \\ & p(e|c,\neg d)p(c,\neg d|c) + \\ & p(e|\neg c,d)p(\neg c,d|c) + \\ & p(e|\neg c,\neg d)p(\neg c,\neg d|c) \end{aligned}$$

Izrazimo še manjkajoče faktorje:

$$\begin{aligned} p(c,d|c) &\overset{3}{=} p(c|c)p(d|c) \overset{1}{=} p(d|c) = \\ &\overset{6b}{=} p(d|b)p(b|c) + p(d|\neg b)p(\neg b|c) = \\ &\overset{6a}{=} p(d|b)p(b) + p(d|\neg b)p(\neg b) \\ p(c,\neg d|c) &\overset{3}{=} p(c|c)p(\neg d|c) \overset{1,6b,6a}{=} p(\neg d|b)p(b) + p(\neg d|\neg b)p(\neg b) \\ p(\neg c,d|c) &\overset{3}{=} p(\neg c|c)p(d|c,\neg c) \overset{2}{=} 0 \\ p(\neg c,\neg d|c) &\overset{3}{=} p(\neg c|c)p(\neg d|\neg c) \overset{2}{=} 0 \end{aligned} \tag{4.1}$$

Poznamo torej vse, kar potrebujemo za izračun $p(e|c)$, po 4. pravilu pa znamo izračunati tudi $p(\neg e|c)$; $p(g|c)$ smo tako izrazili samo z znanimi verjetnostmi. Nadaljujmo s $p(i|g,c)$:

$$p(i|g,c) \overset{6b}{=} p(i|g)p(g|g,c) + p(i|\neg g)p(\neg g|g,c) \overset{1,2}{=} p(i|g).$$

Verjetnost $p(i|g)$ je dana; izraziti moramo le še $p(i|c)$:

$$p(i|c) \overset{6b}{=} p(i|g)p(g|c) + p(i|\neg g)p(\neg g|c).$$

Že zgoraj smo izračunali $p(g|c)$ in $p(\neg g|c) = 1 - p(g|c)$, ostalo pa je podano.

d)

$$p(d|e,f) \overset{5}{=} p(d|f)\frac{p(e|d,f)}{p(e|f)}$$

$$p(d|f) \overset{5}{=} p(d)\frac{p(f|d)}{p(f)}$$

$$p(d) \overset{6b}{=} p(d|b)p(b) + p(d|\neg b)p(\neg b)$$

$$p(f) \overset{6b}{=} p(f|d)p(d) + p(f|\neg d)p(\neg d)$$

$$
\begin{aligned}
p(e|d,f) &\overset{6b}{=} p(e|c,d)p(c,d|d,f) + \\
&\quad p(e|c,\neg d)p(c,\neg d|d,f) + \\
&\quad p(e|\neg c,d)p(\neg c,d|d,f) + \\
&\quad p(e|\neg c,\neg d)p(\neg c,\neg d|d,f) = \\
&\overset{3}{=} p(e|c,d)p(c|d,f) + p(e|\neg c,d)p(\neg c|d,f)
\end{aligned}
$$

$$
\begin{aligned}
p(c|d,f) &\overset{6b}{=} p(c|a)p(a|d,f) + p(c|\neg a)p(\neg a|d,f) = \\
&\overset{6a}{=} p(c|a)p(a) + p(c|\neg a)p(\neg a)
\end{aligned}
$$

4.3 d-ločevanje

Za Bayesovo mrežo iz naloge 1 zapiši vse množice, ki $d-$ločujejo naslednje pare vozlišč:

a) c in d,

b) b in f,

c) b in i,

d) h in i.

Rešitev:

Množica B $d-$ločuje V_i in V_j, če blokira vse poti med V_i in V_j.Množica B blokira pot med V_i in V_j, če obstaja tako vozlišče V_b na poti, ki blokira to pot. V_b blokira pot, če velja eden od pogojev:

1. $V_b \in B$ je divergentno vozlišče oz. skupni vzrok (V_b je v B in obe povezavi na poti kažeta iz V_b),
2. $V_b \in B$ je zaporedno vozlišče (V_b je v B in ena povezava na poti kaže v V_b in ena iz V_b),
3. $V_b \notin B$ je konvergentno vozlišče oz. skupna posledica (obe povezavi na poti kažeta v V_b) in niti V_b niti noben njegov naslednik ni v B.

a) Iz pogoja 3 sledi, da c in d $d-$ločujejo vse množice, ki ne vsebujejo vozlišč e, g, h in i:
$\{\}$,$\{a\}$,$\{b\}$,$\{f\}$,$\{a,b\}$,$\{a,f\}$,$\{b,f\}$,$\{a,b,f\}$

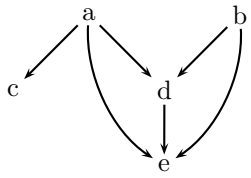
b) Na poti med b in f je samo d, ki je zaporedno vozlišče. Najmanjša množica, ki $d-$ločuje b in f je $\{d\}$. Rešitev pa je tudi vsaka množica, ki vsebuje vozlišče d, npr. $\{d,a\},\{d,g,h\}$ (ker je takih množic veliko, tu ne navajamo vseh).

c) Najmanjše množice, ki $d-$ločujejo b in i so: $\{d\},\{e\},\{g\}$. Vsako od teh vozlišč namreč blokira pot od b do i. Tudi vsaka množica, ki vsebuje najmanj eno od vozlišč d,e,g, $d-$ločuje b in i.

d) Vozlišče g blokira pot med h in i, ker je njun skupni vzrok. Zato vse množice, ki vsebujejo g $d-$ločujejo h in i.

4.4 Pre(za)pletena BM

Dana je struktura Bayesove mreže:



a) Brez računanja oceni, katera verjetnost je večja: $p(a|e)$ ali $p(a|e,b)$. Predpostavi, da velja: $p(e|a) > p(e|\neg a)$ in $p(e|b) > p(e|\neg b)$.

b) Izrazi pogojno verjetnost $p(d|c)$ z verjetnostmi, ki morajo biti podane za zgornjo strukturo mreže.

c) Zapiši vse množice, ki $d-$ločujejo c in b.

Rešitev:

a) Tako oceno lahko podamo samo pri predpostavki, da povezave med vozlišči pomenijo pozitiven vzrok. V našem primeru mora torej veljati: $p(e|a) > p(a)$ in $p(e|b) > p(b)$. Ob tej predpostavki je $p(a|e) > p(a|e,b)$, ker je b alternativni vzrok za e, zato zmanjša verjetnost a-ju.

b) Podane morajo biti naslednje verjetnosti:
$p(a), p(b)$,
$p(c|a), p(c|\neg a)$,
$p(d|a,b), p(d|a,\neg b), p(d|\neg a,b), p(d|\neg a,\neg b)$
$p(e|a,b,d), p(e|a,b,\neg d), p(e|a,\neg b,d), p(e|a,\neg b,\neg d)$
$p(e|\neg a,b,d), p(e|\neg a,b,\neg d), p(e|\neg a,\neg b,d), p(e|\neg a,\neg b,\neg d)$

$$\begin{aligned} p(d|c) &= p(d|a,b)p(a,b|c) + \\ &\quad p(d|a,\neg b)p(a,\neg b|c) + \\ &\quad p(d|\neg a,b)p(\neg a,b|c) + \\ &\quad p(d|\neg a,\neg b)p(\neg a,\neg b|c) \end{aligned}$$

$$\begin{aligned} p(a,b|c) &= p(a|c)p(b|a,c) = \\ &= p(a)\frac{p(c|a)}{p(c)}p(b|a,c) \\ p(c) &= p(c|a)p(a) + p(c|\neg a)p(\neg a) \\ p(b|a,c) &= p(b) \end{aligned}$$

Zaradi neodvisnosti b in c bi lahko sklepali tudi takole:

$$p(a,b|c) = p(a|c)p(b|c) = p(a|c)p(b).$$

$$\begin{aligned} p(a,\neg b|c) &= p(a|c)p(\neg b|a,c) = \\ &= p(a)\frac{p(c|a)}{p(c)}p(\neg b|a,c) = \\ p(\neg b|a,c) &= p(\neg b) = 1 - p(b) \end{aligned}$$

$$\begin{aligned} p(\neg a,b|c) &= p(\neg a|c)p(b|a,c) = \\ &= p(\neg a)\frac{p(c|\neg a)}{p(c)}p(b|a,c) \end{aligned}$$

$$\begin{aligned} p(\neg a,\neg b|c) &= p(\neg a|c)p(\neg b|a,c) = \\ &= p(\neg a)\frac{p(c|\neg a)}{p(c)}p(\neg b|a,c) \end{aligned}$$

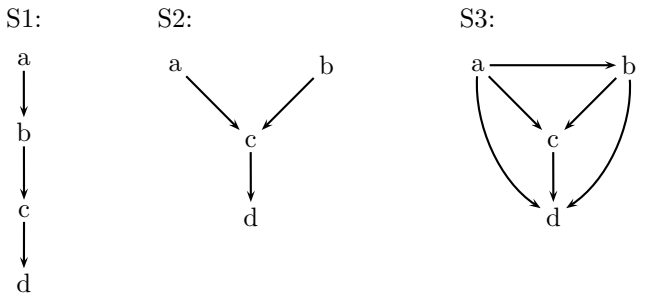
c) Od c do b vodijo 4 neusmerjene poti. Za vsako od njih bomo poiskali množice, ki jih blokirajo.

1. $c-a-d-b$: a je divergentno vozlišče, torej $\{a\}$ in vsaka množica, ki vsebuje a blokira to pot. Torej tudi: $\{a,d\},\{a,e\},\{a,d,e\}$. Vozlišče d je konvergentno, iz česar sledi, da vse množice, ki ne vsebujejo d in njegovih naslednikov (e), blokirajo to pot: $\{\}$ in $\{a\}$, ki pa jo imamo že zgoraj.
2. $c-a-e-b$: Po istem premisleku kot zgoraj, namesto vozlišča d imamo zdaj na poti e, ki pa je tako kot d konvergentno: $\{a\},\{\},\{a,d\},\{a,e\},\{a,d,e\}$.
3. $c-a-d-e-b$: Vozlišče a blokira pot in podobno kot prej, $\{a\}$ in vsaka množica, ki vsebuje a, blokira to pot. Vozlišče d na tej poti je zaporedno, torej tudi $\{d\}$ in vsaka množica, ki vsebuje d, blokira to pot. Ker je e konvergentno, tudi $\{\}$ blokira to pot. Vse množice, ki jih dobimo na tej poti so: $\{\},\{a\},\{d\},\{a,d\},\{a,e\},\{d,e\},\{a,d,e\}$.
4. $c-a-e-d-b$: Premislek in rešitev sta taka, kot za pot $c-a-d-e-b$.

Povzemimo: vozlišči c in b d-ločujejo: $\{\},\{a\},\{a,d\},\{a,e\},\{a,d,e\}$.

4.5 Zadnja

Podane so naslednje strukture Bayesovih mrež S1, S2 in S3:



a) Koliko podatkov prihrani posamezna struktura v primerjavi s tem, ko Bayesova mreža sploh ni podana?

b) Ali je možno podatke o verjetnostih za strukturo S1 prevesti v podatke za S2 tako, da bosta obe Bayesovi mreži definirali enako skupno verjetnostno porazdelitev vseh spremenljivk? Utemelji podgovor.

c) Zapiši vse množice, ki d-ločujejo a in b za vsako od podanih mrež.

Rešitev:

a) Če mreža ni podana potrebujemo $2^4 - 1 = 15$ podatkov o (pogojnih) verjetnostih. Pri S1 potrebujemo samo naslednje podatke:

$$p(a), p(b|a), p(b|\neg a), p(c|b), p(c|\neg b), p(d|c), p(d|\neg c),$$

kar nam prihrani 15-7=8 podatkov.

Po podobnem premisleku za S2 potrebujemo 8 podatkov, torej jih prihranimo 7. Pri S3 ne prihranimo ničesar.

b) Ne, ker v S1 je b odvisen od a, medtem ko sta v S2 neodvisna (velja $p(a, b) = p(a)p(b)$).

c) S1: Ne obstaja množica, ki bi d-ločevala a in b.

S2: Take so vse množice, ki ne vsebujejo c in d: $\{\}$

S3: Imamo 5 poti, ki jih moramo pregledati: za poti $a - c - b$ in $a - d - b$ je rešitev enaka zgornji. Za poti $a - c - d - b$, $a - d - c - b$ je c zaporedno, d pa konvergentno vozlišče. Ker je c zaporedno, množici $\{c\}, \{c, d\}$ blokirata obe poti. Prav tako ju blokirata $\{c\}$ in $\{\}$, ker je d konvergentno. Preostane nam še najkrajša pot, $a - b$, ki pa je ni možno d-ločiti. Ker ne obstaja množica, ki bi blokirala vse naštete poti, zaključimo, da a in b ni možno d-ločiti.