SAARLAND UNIVERSITY
Prof. Jana Koehler
Prof. Jörg Hoffmann

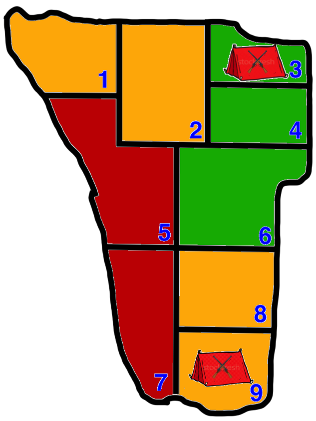
ARTIFICIAL INTELLIGENCE

Theoretical Exercise Sheet 7.

Solutions due Tuesday, June 23, 23:59 uploaded in the AI CMS.

Exercise 1: Formulating a CSP — 6 Points

You are a ranger and own a piece of land, depicted below. You want to make sure that all of your animals survive. This is only possible if no animals that kill each other live in the same or neighboring regions. You own rhinos (R), elephants (E), cheetahs (C), lions (L), jackals (J), springbok (S), zebras (Z), hippos (H) and giraffes (G). Your piece of land contains three different climatic regions: bush (green), desert (red), savannah (orange). Some animals need specific climatic environments. Unfortunately there are also two camps of poachers on the land, who are selling rhino horn and elephant tusk. Poachers treacherously shoot animals in their own and neighboring regions.



- Poachers kill rhinos and elephants
- Jackals eat springbok
- Cheetahs eat springbok and zebras
- Lions eat springbok, zebras and giraffes, and sometimes even kill elephants

- Jackals and cheetahs can live in the desert and in the savannah
- Elephants can only live in the bush
- Lions and zebras can live in the bush and in the savannah
- Springbok and hippos only live in the savannah regions

Formulate the problem as a constraint network. Use the set of variables $V = \{v_1, ..., v_9\}$, where variable v_i models which animal can live in region i.

(i) Specify the domain D_{v_i} of each variable $v_i \in V$. That is, for each region type give the set of animals that can live in that region, and state which variables have this domain.

(ii) Specify the binary constraints $C_{v_i v_j} \subseteq D_{v_i} \times D_{v_j}$ for all pairs of variables. That is, for each combination of regions give the set of animal-pairs that the region pair can inhabit, and state which variable pairs have this constraint.

(iii) Specify the unary constraints $C_{v_i} \subseteq D_{v_i}$ for all relevant variables. That is, for each region affected by the poachers, give the set of animals that can inhabit that region.

Hint: You can specifying constraints with those pairs that are not contained:
$C_{v_i,v_j} = D_{v_i} \times D_{v_j} \setminus \{(d_{v_{i,k}}, d_{v_{j,l}}), ...\}$
Solution:

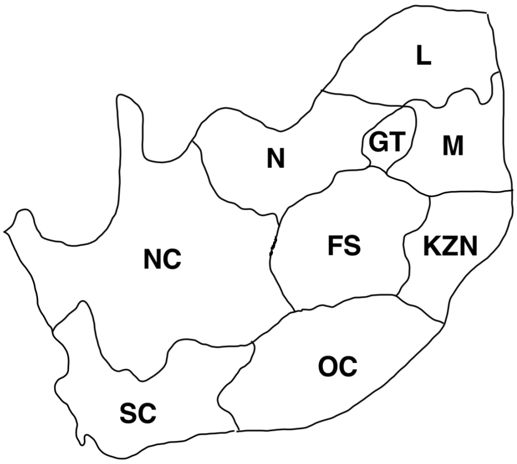
(i)
- $D_S = D_{v_1} = D_{v_2} = D_{v_8} = D_{v_9} = \{R, J, C, S, Z, L, G, H\}$
- $D_B = D_{v_3} = D_{v_4} = D_{v_6} = \{R, E, Z, L, G\}$
- $D_D = D_{v_5} = D_{v_7} = \{R, J, C, G\}$

(ii)
- $C_{sav,sav} = C_{v_1 v_2} = C_{v_8 v_9}$
  $= D_S \times D_S \setminus \{(S, J), (J, S), (C, S), (S, C), (C, Z), (Z, C),$
  $(L, S), (L, Z), (L, G), (G, L), (Z, L), (S, L)\}$
- $C_{des,des} = C_{v_5 v_7}$
  $= D_D \times D_D$
- $C_{bush,bush} = C_{v_3 v_4} = C_{v_4 v_6}$
  $= D_B \times D_B \setminus \{(L, Z), (L, G), (L, E), (Z, L), (G, L), (E, L)\}$
- $C_{sav,des} = C_{v_1 v_5} = C_{v_2 v_5} = C_{v_8 v_7} = C_{v_9 v_7}$
  $= D_S \times D_D \setminus \{(S, J), (S, C), (Z, C), (L, G)\}$
- $C_{sav,bush} = C_{v_2 v_3} = C_{v_2 v_4} = C_{v_8 v_6} = C_{v_2 v_6}$
  $= D_S \times D_B \setminus \{(C, Z), (L, Z), (L, G), (L, E), (S, L), (Z, L), (G, L)\}$
- $C_{des,bush} = C_{v_5 v_6}$
  $= D_D \times D_B \setminus \{(C, Z), (G, L)\}$

(iii)
- $D_2 = D_8 = D_9 = D_S \setminus \{R\} = \{J, C, S, Z, L, G, H\}$
- $D_4 = D_3 = D_B \setminus \{R, E\} = \{Z, L, G\}$
- $D_7 = D_D \setminus \{R\} = \{J, C, G\}$

Exercise 2: Naive Backtracking 6 Points

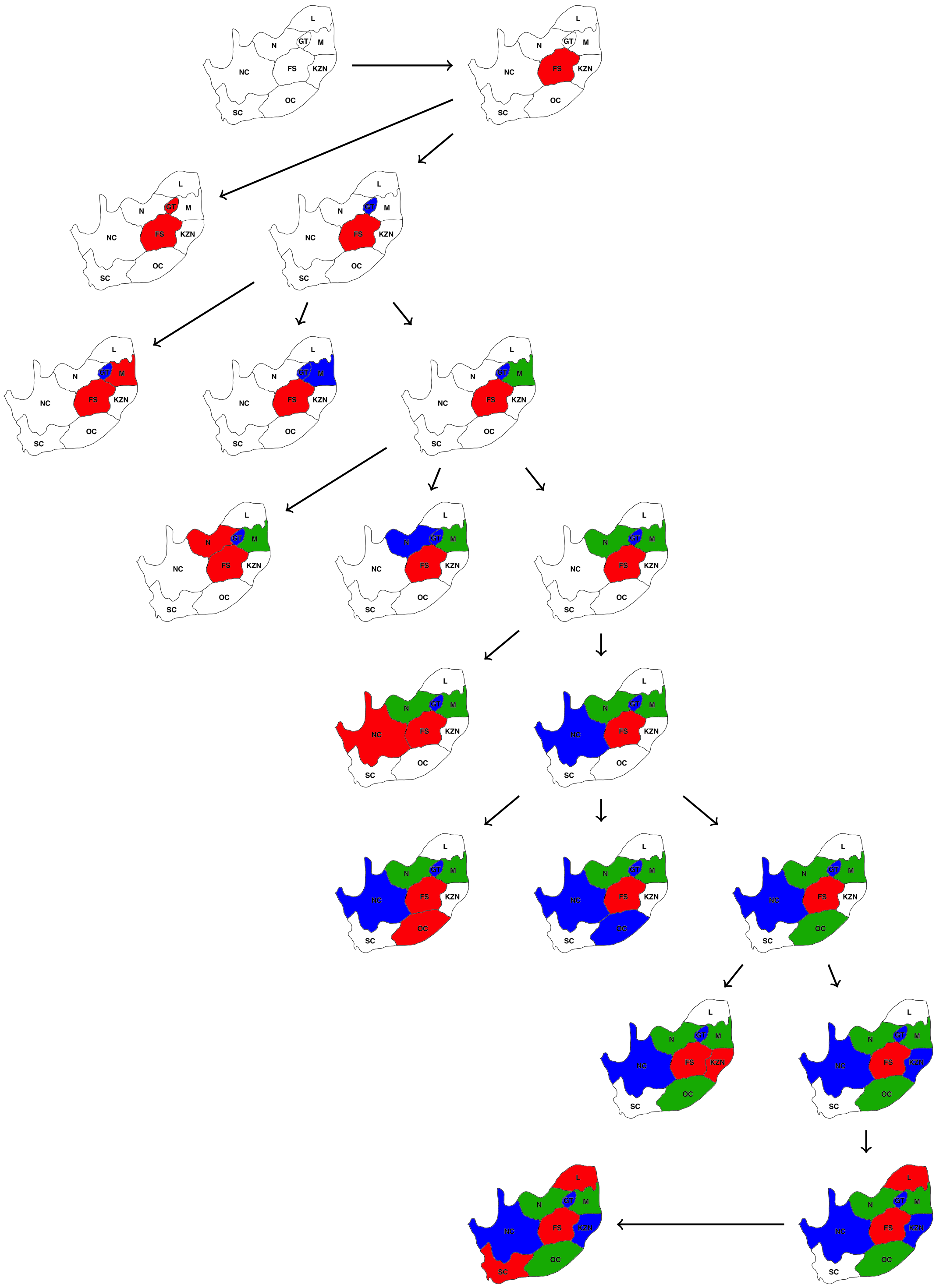
Your country has a population of three different breeds of antelopes: oryx (O), kudu (K), and impala (I). You want to mix the populations, such that each region inhabits only one of the breeds but no region has the same breed as any of their neighboring regions.



Run the naïve backtracking algorithm to find an assignment. The corresponding constraint satisfaction problem is $\gamma = (V, D, C)$ with

- The variables are $V = \{SC, NC, OC, FS, N, KZN, GT, M, L\}$ for each of the regions, each with domain $D = \{O, K, I\}$, for each of the breeds.
- The constraints are such that no pair of adjacent regions can be inhabited by the same breed, i.e., $C_{\{u,v\}} \in C$ for all adjacent u and v, $u \neq v$. For example, $C_{\{\text{GT,FS}\}} \in C$ because GT and FS are adjacent and consequently have to be inhabited by two different breeds, but $C_{\{\text{GT,SC}\}} \notin C$.
- Use the combination strategy described on slide 42 (bottom) of Chapter 9, i.e., **among the most *constrained* variables, pick the most *constraining* first. As a tie-breaker, use the alphabetical order on the region labels**.
- The value order should first assign oryx (O), then kudu (K), and eventually impala (I). **Draw the search graph and mark inconsistent assignments.**

If you want to use colors, use the following colors:
oryx: red, kudu: blue, impala: green.


L
N
GT
M
NC
FS
KZN
OC
SC

Solution:

Exercise 3: AC-3, AcyclicCG 14 Points

On your piece of land you additionally own a lodge for tourists and a cat sanctuary. You want to show your tourists all your nice animals and everything the region has to offer. You have 5 different time slots: $1, 2, 3, 4, 5$, and want to fit all the tasks (see below) in, such that all of the constraints (see below) are satisfied.

Tasks:

- (GD) go for game drive with tourists
- (DW) go to the desert with tourists
- (HF) make helicopter flight with tourists
- (HB) have braai with tourists
- (PB) prepare braai
- (PC) pet cheetahs
- (FL) feed lions
- (FC) feed cheetahs
- (FZ) feed zebra

Constraints:

- Cheetahs need to be fed immediately before you can pet them ($FC = PC - 1$)
- Lions and cheetahs have to be fed at the same time, otherwise they will get jealous ($FC = FL$)
- The desert walk has to be (at least) two time slots before the game drive, otherwise your tourists will get to exhausted ($DW \leq GD - 2$)
- In order to have braai with your tourists you first have to exhaust them on the game drive and on the desert walk, so they can enjoy their beer ($GD < HB$; $DW < HB$)
- You have to prepare braai before having the braai with your tourists ($PB < HB$)

- You can only find lions on the game drive if they have been fed two time slots earlier ($GD = FL + 2$)
- In order to give your tourists an impression of the area first, the heli flight is done just before the desert walk ($HF = DW - 1$)
- In order to save time you want to feed the zebra on the game drive ($FZ = GD$)
- Variables: $V = \{DW, FC, FL, FZ, GD, HB, HF, PC, PB\}$
- Domains: For all $v \in V$: $D_v = \{1, 2, 3, 4, 5\}$

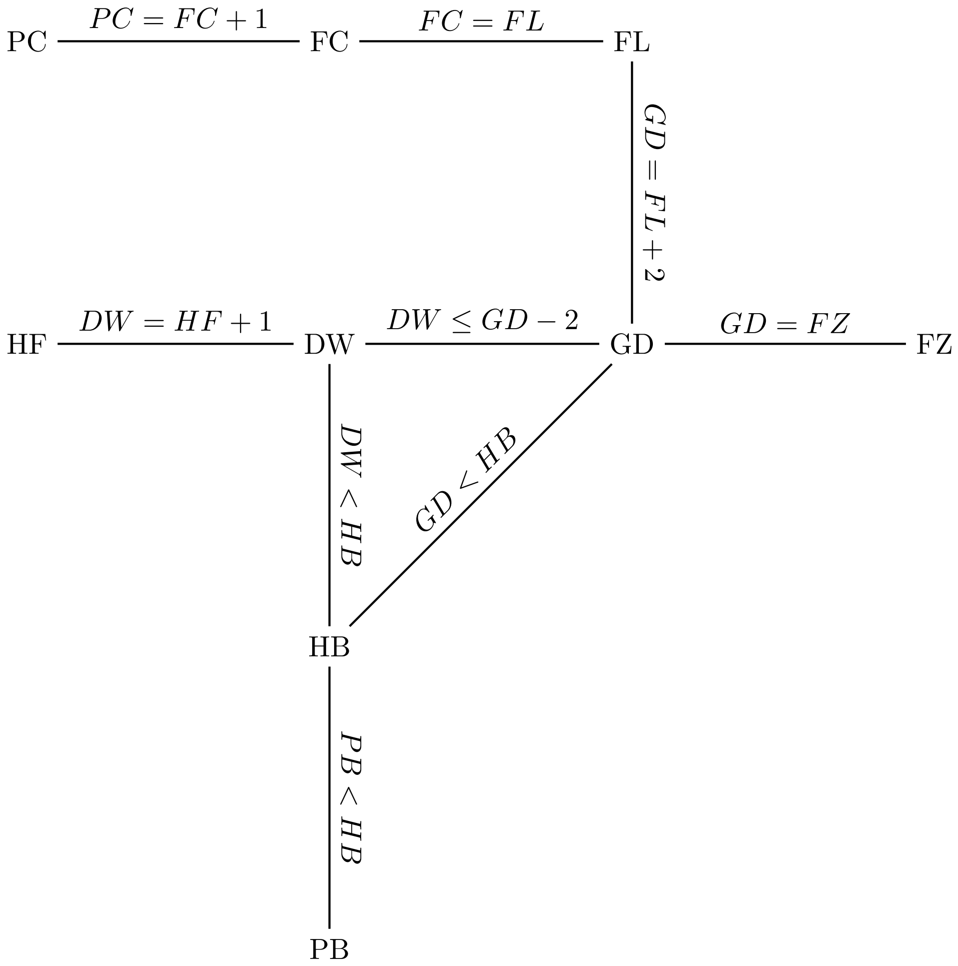
(i) Draw the constraint graph of γ. Label each edge with the corresponding constraint.

(ii) Run AC-3. Initialize M with an alphabetical list. For each step: Give the selected pair, the updated domain, and the pairs that are added to M (give all pairs that would be added and mark those that are not already contained). You do not have to give the updated M in each iteration.

(iii) Run AcyclicCG:

1. Assume that the constraint $DW < HB$ does not exist (remove this edge from the graph). Pick DW as the root and draw the directed tree.
2. Give the resulting variable order obtained by step 2 of the algorithm. If the ordering of variables is not unique, break ties by using the alphabetical order.
3. List the calls to $Revise(\gamma, v_{parent(i)}, v_i)$ in the order executed by step 3 of the algorithm, and for each of them give the resulting domain of $v_{parent(i)}$.
4. Run BacktrackingWithInference with Forward Checking. For each recursive call to BacktrackingWithInference during step 4 of the algorithm, give the domain D'_{v_i} of the selected variable v_i after Forward Checking, and give the value $d \in D'_{v_i}$ assigned to v_i.

Solution:



(i)

(ii) $M = \{(DW, GD), (DW, HB), (DW, HF), (FC, FL), (FC, PC), (FL, FC), (FL, GD), (FZ, GD), (GD, DW), (GD, FL), (GD, FZ), (GD, HB), (HB, DW), (HB, GD), (HB, PB), (HF, DW), (PB, HB), (PC, FC)\}$

(1) **selected**:(DW, GD); $D_{DW} = \{1, 2, 3\}$; **add**: (HB, DW), (HF, DW)

(2) **selected**:(DW, HB); no changes

(3) **selected**:(DW, HF); $D_{DW} = \{2, 3\}$; **add**: (GD, DW), (HB, DW)

(4) **selected**:(FC, FL); no changes

(5) **selected**:(FC, PC); $D_{FC} = \{1, 2, 3, 4\}$; **add**: (FL, FC)

(6) **selected**:(FL, FC); $D_{FL} = \{1, 2, 3, 4\}$; **add**: (GD, FL)

(7) **selected**:(FL, GD); $D_{FL} = \{1, 2, 3\}$; **add**: (FC, FL)

(8) **selected**:(FZ, GD); no changes

(9) **selected**:(GD, DW); $D_{GD} = \{4, 5\}$; **add**: $\underline{(FL, GD)}$, $\underline{(FZ, GD)}$, (HB, GD)

(10) **selected**:(GD, FL); no changes

(11) **selected**:(GD, FZ); no changes

(12) **selected**:(GD, HB); $D_{GD} = \{4\}$; **add**: $\underline{(DW, GD)}$, (FL, GD), (FZ, GD)

(13) **selected**:(HB, DW); $D_{HB} = \{3, 4, 5\}$; **add**: $\underline{(GD, HB)}$, (PB, HB)

(14) **selected**:(HB, GD); $D_{HB} = \{5\}$; **add**: $\underline{(DW, HB)}$, (PB, HB)

(15) **selected**:(HB, PB); no changes

(16) **selected**:(HF, DW); $D_{HF} = \{1, 2\}$; no adds

(17) **selected**:(PB, HB); $D_{PB} = \{1, 2, 3, 4\}$; no adds

(18) **selected**:(PC, FC); $D_{PC} = \{2, 3, 4, 5\}$; no adds

(19) **selected**:(FC, FL); $D_{FC} = \{1, 2, 3\}$; **add**: $\underline{(PC, FC)}$

(20) **selected**:(FL, GD); $D_{FL} = \{2\}$; **add**: $\underline{(FC, FL)}$

(21) **selected**:(FZ, GD); $D_{FZ} = \{4\}$; no adds

(22) **selected**:(DW, GD); $D_{DW} = \{2\}$; **add**: $\underline{(HB, DW)}$, $\underline{(HF, DW)}$

(23) **selected**:(GD, HB); no changes

(24) **selected**:(DW, HB); no changes

(25) **selected**:(PC, FC); $D_{PC} = \{2, 3, 4\}$; no adds

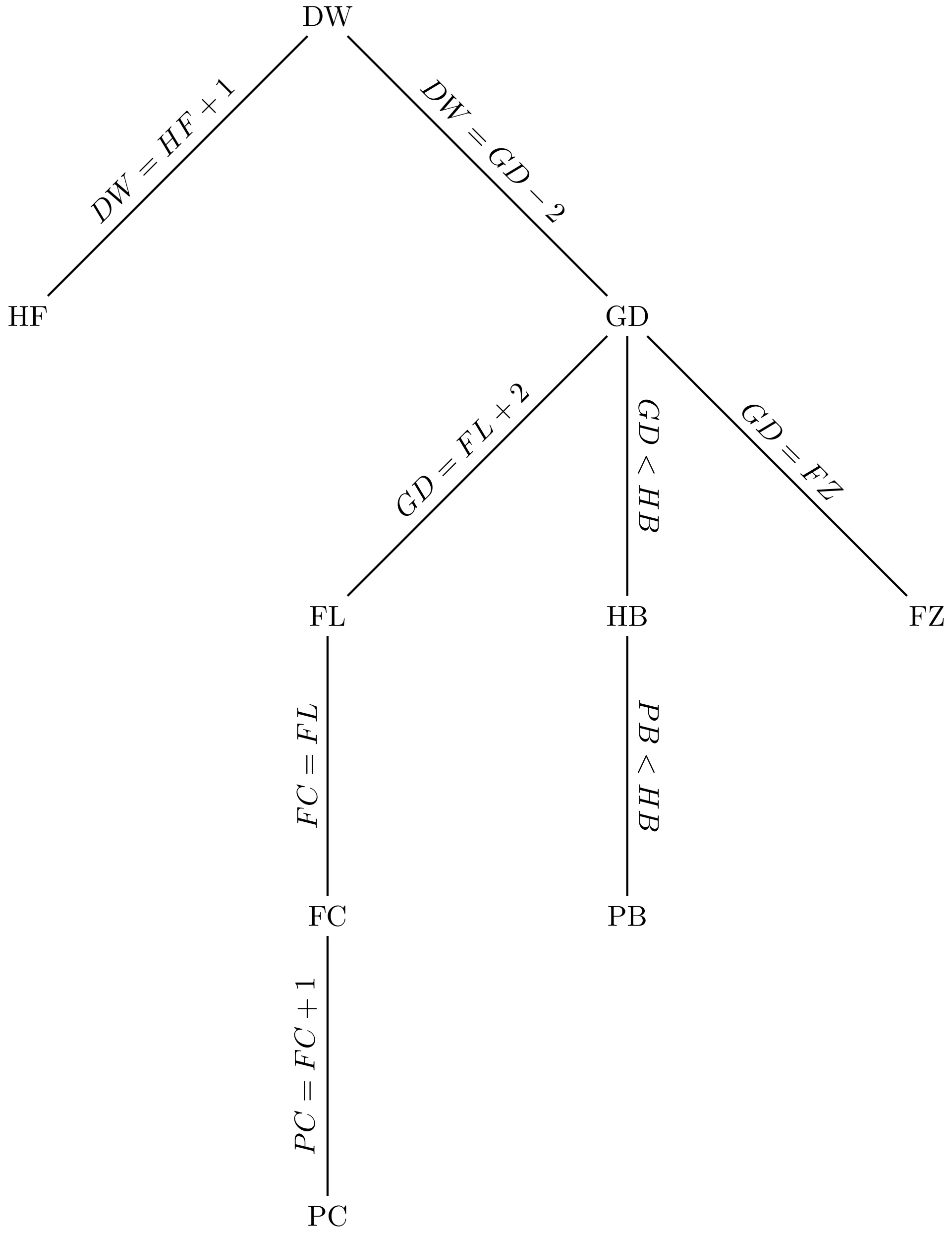
(26) **selected**:(FC, FL); $D_{FC} = \{2\}$; **add**: $\underline{(PC, FC)}$

(27) **selected**:(HB, DW); no changes

(28) **selected**:(HF, DW); $D_{HF} = \{1\}$; no adds

(29) **selected**:(PC, FC); $D_{PC} = \{3\}$; no adds

M empty; return modified γ :
$D_{DW} = \{2\}$
$D_{FC} = \{2\}$
$D_{FL} = \{2\}$
$D_{FZ} = \{4\}$
$D_{GD} = \{4\}$
$D_{HB} = \{5\}$
$D_{HF} = \{1\}$
$D_{PB} = \{1, 2, 3, 4\}$
$D_{PC} = \{3\}$



(iii) 1.

2. DW, GD, FL, FC, FZ, HB, HF, PB, PC
3. (1) Revise FC, PC: $D_{FC} = \{1, 2, 3, 4\}$
   (2) Revise HB, PB: $D_{HB} = \{2, 3, 4, 5\}$
   (3) Revise DW, HF: $D_{DW} = \{2, 3, 4, 5\}$
   (4) Revise GD, HB: $D_{GD} = \{1, 2, 3, 4\}$

(5) Revise GD, FZ: $D_{GD} = \{1, 2, 3, 4\}$
(6) Revise FL, FC: $D_{FL} = \{1, 2, 3, 4\}$
(7) Revise GD, FL: $D_{GD} = \{3, 4\}$
(8) Revise DW, GD: $D_{DW} = \{2\}$

4. (1) $D_{DW} = \{2\}$, $d = \{2\}$
(2) $D_{GD} = \{4\}$, $d = \{4\}$
(3) $D_{FL} = \{2\}$, $d = \{2\}$
(4) $D_{FC} = \{2\}$, $d = \{2\}$
(5) $D_{FZ} = \{4\}$, $d = \{4\}$
(6) $D_{HB} = \{5\}$, $d = \{5\}$
(7) $D_{HF} = \{1\}$, $d = \{1\}$
(8) $D_{PB} = \{1, 2, 3, 4\}$, $d = \{1\}$
(9) $D_{PC} = \{3\}$, $d = \{3\}$

Exercise 4: Cutset Conditioning — 6 Points

You now want to feed all the animals in your animal sanctuary. Your sanctuary inhabits cheetah (C), leopard (Le), lions (L), kudu (K), springbok (S), steenbok (St), impala (I), oryx (O), karakal (Kk), zebra (Z), giraffes (G) and elephants (E). Unfortunately some of your animals might get eaten by the others if you call them all together, so you have to make sure that not all of them eat at the same time. Additionally, some of your animals will get jealous if you feed certain other animals before them. You have 3 possible time slots: 1, 2 and 3

Constraints:

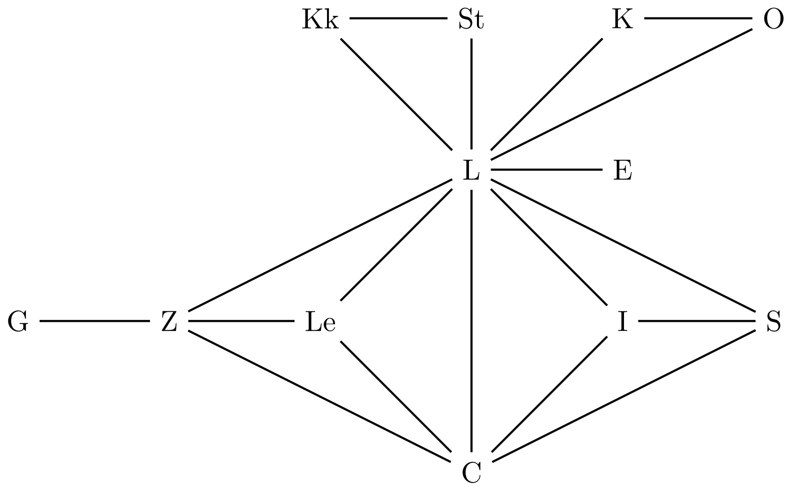
- Your big cats have to eat together ($C = L$, $C = Le$, $L = Le$)
- Lions eat steenbok, springbok, impala, kudu, oryx and zebra ($L \neq St$, $L \neq S$, $L \neq I$, $L \neq K$, $L \neq O$, $L \neq Z$)
- Impalas eat together with springbok ($I = S$)
- Kudu eat together with oryx ($K = O$)
- Karakals eat steenbok ($Kk \neq St$)
- Cheetah eat impalas, springbok and zebras ($C \neq I$, $C \neq S$, $C \neq Z$)
- Giraffes and zebras eat together ($G = Z$)
- Lions have to be fed before elephants ($L < E$)
- Karakals have to be fed before lions ($Kk < L$)
- Leopards eat zebra ($Le \neq Z$)

(i) Draw the constraint graph of γ.

(ii) What is the optimal cutset V_0 for γ?

(iii) If the CutsetConditioning algorithm from the lecture is called with such a minimal cutset V_0, how many calls to AcyclicCG will be performed in the worst case? Justify your answer.

Solution:

(i) 

(ii) $V_0 = \{L, C\}$

(iii) The only possible assignment is $L = 2$ and $C = 2$. In the worst case, we go over all possibilities until we get to the correct one, so that we call AcyclicCG 9 times. However, be aware that this is only a worst case scenario. Since we use forward checking during the calculation, we may end up with less calls.

Submission Instructions

Solutions need to be packaged into a `.zip` file and uploaded in the AI CMS. The `.zip` file has to contain a single folder with name:
`AI2020_TE7_mat1_mat2_mat3` where `mat1, mat2, mat3` are the matriculation numbers of the students who submit together. This folder must contain the following files:

- `authors.txt` listing the names and matriculation numbers of all students who submit together. Use one line per student and no spaces: Name;Matriculation number.
- The `.pdf` file containing your solutions.

Do not add any other folder or sub folder, this means place all files directly into `AI2020_TE7_mat1_mat2_mat3`. Do not place any file outside of `AI2020_TE7_mat1_mat2_mat3`.

Only one student of each group needs to do the submission! Remember that this sheet can be submitted in groups of up to three members (all members of the group must however be assigned to the same tutorial).