CLP in SICStus Exercise Sheet 3

Constraint Programming in SICStus

Goals:

– Constraint Programming and Optimization

1. Lazy Mailman

Milo, the mailman, has very few letters to deliver daily, and he sets the goal of taking as long as possible to deliver the mail in the last street of his round. We know the following:

- It is a straight street, with ten houses, all ten meters apart from each other;
- He always walks at ten meters per minute, and wants to finish at house 6 (the person living there always offers him coffee and cake);
- He has a (greedy) solution, starting in house 1, then going to house 10, then house 2, …, and finally 6, which results in 45 minutes (9+8+7+6+5+4+3+2+1);

Model this problem using constraint programming to find a better solution (i.e., one that takes even longer than 45 minutes). Note: consider that the mailman enters the street orthogonally and time starts counting from the first house he visits.

2. Exam Questions Revisited

Recall question 4 from the previous exercise sheet. Consider now that Emma also has an estimated time to solve each exercise (the total estimated time may be more than the time available). Implement *questions(+Valuations, +Times, +TotalTime, +MinGrade, -Right)*, that receives the list of question values, the estimated times (lists of integer values with the same length), the total time of the exam, the desired minimum grade, and returns in *Right* a list of binary variables representing the combination of wright (1) or wrong (0) answers allowing to obtain the desired minimum grade. Example:

```
| ?- questions([1,2,2,3,5,1,2,4], [5,10,10,15,20,5,10,20], 60, 10, R).
R = [0,0,0,1,1,1,0,1] ? ;
no
```

3. Connecting Things

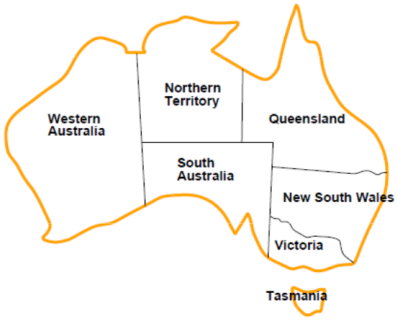
When designing a building's fire detection system, there is a central control panel to which we have to connect a number of sensors, using network cable to form a loop – a netweork connection starting at the control panel, passing through all the sensors, and returning to the control panel. Panel and sensors have unique consecutive IDs (the panel ID is 1), with the positions of the equipment in X and Y coordinates:

```
% coord(ID, Nome, X, Y).
coord(1, 'panel', 2, 4).
coord(2, 'sensor 1', 3, 3).   coord(3, 'sensor 2', 6, 5).
coord(4, 'sensor 3', 5, 4).   coord(5, 'sensor 4', 7, 7).
```

To save on communication time and money, the network cable should be as short as possible. Implement ***loop(Order, Distance)*** that returns in Order the order by which the sensors should be connected (starting with the control panel) and in Distance the total length of network cable necessary to make all the connections. Note: round the distances to the nearest integer value. Example:

```
| ?- loop(Order, Distance).
Order = [1, 2, 4, 3, 5],
Distance = 12 ?
```

4. Coloring Australia



Map Coloring is a classic problem, with the goal of coloring a map with N different colors such that no two adjacent areas have similar colors.

Solve the problem for Australia using the minimum amount of colors possible (and 5 colors at most).

5. Holiday Bus Company

A bus company has several buses that can be used to ferry several groups of tourists to their vacation destination. Different buses have different capacities, and each group of tourists has a different size. The goal is to allocate groups of tourists to buses such that:

- Each group is not separated (the entire group travels in the same bus)
- The number of used buses is minimized (each bus may ferry several groups)

Implement *bus_company(+Buses, +Groups, -Assignments)*, that receives the list of existing bus capacities, the list of group sizes, and returns in Assignments the assignment of each group to an existing bus. Example:

```
| ?- busCompany( [11, 14, 10, 20], [5, 5, 7, 4, 3], A).
A = [1,1,2,2,2] ? ;
no
```

6. Golomb Ruler

a) Implement *golomb(N, Values)*, that finds a set of *N* distinct numbers, from 0 to *N*x*N*, such that the differences between all pairs of values are different. This problem is known as the Golomb Ruler, and its resolution has practical applications in radio-astronomy. Example:

```
| ?- golomb(5, 11, V).
V = [0,1,4,9,11] ? ;
V = [0,2,7,8,11] ? ;
V = [0,2,7,10,11] ? ;
V = [0,3,4,9,11] ? ;
no
```

a) A solution X to the Golomb Ruler with N values is considered optimal if there is no other solution whose maximum value is lower than the maximum value of X.

Implement golomb_opt(N, MaxVal, Values), that finds optimal solutions to the problem.

Hint: try to avoid non-incremental solutions to improve efficiency.

7. House Chores Assignment

A young couple, Steve and Stephanie, intend to divide some domestic chores so that each has only 2 tasks, minimizing the total time spent. Their efficiency varies with the task, with the times for each task shown in the table below.

	Shopping	Cooking	Cleaning	Vacuuming
Stephanie	4.5	7.8	3.6	2.9
Steven	4.9	7.2	4.3	3.1

8. Movie Downloads

Mavis found a way to download movies from the internet using streaming services so she can take the movies and TV show episodes with her on vacation. She can program her computer to download a movie or episode, but is subject to some limitations:

The used streaming services only allow movie download in real-time, i.e., a 93-minute movie takes 93 minutes to download;

Each streaming service only allows downloading one video at a time and it's not possible to start a new process until the previous one is complete;

Each video may only be available in some streaming services;

Since Mavis likes quality TV, all videos are in Full HD, so it takes 6Mbps of bandwidth; since Mavis has a 24Mbps connection, and also uses the internet connection for other things, she wants to limit the number of simultaneous downloads to 3.

Help Mavis to schedule the movie download process in order to download all movies and episodes in the shortest time possible. Model the problem and obtain a generic implementation for this class of problems. Example, considering 5 streaming services, 6 videos to download (three left columns), a solution can be seen in the three right columns.

ID	Duration	Servers		Start Time	End Time	Server
1	22	1,2,5		0	22	1
2	23	1,2,5		0	23	2
3	23	1,2,5		0	23	5
4	24	1,3,4		0	24	3
5	23	1,3,4		0	23	4
6	22	1,3,4		22	44	1

9. Tetris

Tristan is plying a variant of Tetris where he needs to fit several shapes into the 10x10 playing board so that the bounding box of the pieces occupies the least amount of space possible. The pieces are as follows:

```
a) XX      b) XXXX      c) X          d) XXX     e)  X      f) XXX
   XX                       XXX            X       XXX        X X
```

a) Implement a predicate that helps him with this problem, considering the pieces in the same position as presented above.

b) Can you generalize the problem to allow more pieces of different types?

c) Can you generalize the problem to allow piece rotations?

d) Can you generalize the problem to consider a 3D board and 3D pieces?