LUNDS UNIVERSITET
Lunds Tekniska Högskola

Institutionen för datavetenskap
*Krzysztof Kuchcinski*

# Tentamen i kursen
# EDAN01: Constraint-Programmering
# (Constraint programming)

2016-03-18, kl. 8-13

**Sal:**

E:2116

**Hjälpmedel:**

Inga

**Resultat anslås:**

Senast 2016-04-01

**Poänggränser:**

Max 40 p., för 4 krävs ca 25 p, för 5 ca 30 p.

**Jourhavande lärare:**

Krzysztof Kuchcinski, tel. 22 23414

*The answers to the questions can be written in Swedish or English.*

***Lycka till!***

# 1 (3 p.)

Indicate which statements are true or false:

1. A finite domain solver, such as JaCoP, combined with search finds some, but possibly not all, specific solutions using backtracking.
2. An finite domain solver, such as JaCoP, combined with search finds all specific solutions, and also some non-solutions, using backtracking.
3. An incomplete finite domain solver, such as JaCoP, combined with search finds all specific solutions using backtracking but rejects all non-solutions.

Give a short motivation for your answer.

# 2 (3 p.)

Constraint programming defines *node*, *arc* and *path* consistency methods for domain pruning based on constraints.

- Explain these methods shortly and give examples of their applications.
- Give example when arc consistency cannot determine inconsistency while path consistency can do it.
- How these methods are related to k-consistency and strong k-consistency.

# 3 (3 p.)

Use *bounds consistency* and *domain consistency* to prune the domains of FDVs involved in the constraint $x * 3 = y$, $x :: \{1..33\}, y :: \{0..3, 11..15\}$.

# 4 (4 p.)

In constraint programming over finite domain, depth-first-search algorithm is used to enumerate different solutions for a given problem. Discuss different strategies to control the search by selecting variable ordering and domain assignment.

What is the difference between the following two selection choices for finite domain variables.

- *split search*, that is the choice between $x \leq middle$ and $x > middle$, where $middle = \frac{min(x)+max(x)}{2}$, and
- *2-way choice* for x corresponding to choice between $x \leq value$ and $x > value$, where $value$ is the minimal value in the domain of variable $x$.

Illustrate this on an example of list `[x, y]` with `x :: {0..2}, y :: {0..2}` and *largest* selection principle (a first variable with the largest minimal value in the domain is selected). Draw a search tree and all values which are assigned to `x` and `y`.

# 5 (3 p.)

You need to model Boolean clause constraint `BooleanClause(vp, vn)` using constraints available in a standard finite domain constraint solver, such as JaCoP. This constraint is defined for two vectors of $0/1$ variables, `vp` and `vn`, It holds if at least one $vp_i = 1$ or one $vn_i = 0$. In other words, it defines the following constraint.

$$vp_1 \vee \cdots \vee vp_n \vee \neg vn_1 \vee \ldots \neg vn_m \tag{1}$$

# 6 (3 p.)

Explain the concept of *implied constraints*. When should we use them and when should we try to avoid them? Give an example using the scheduling problem of auto regression filter from the lab assignments.

# 7 (5 p.)

A group of $N$ people wants to take a group photo where all people stay in one row. Each person can give preferences next to whom he or she wants to be placed on the photo. Assume that it is specified as an array where each row specifies preferences (for an example in minizinc see Figure 1). Find a placement that satisfies as many preferences as possible.

```
a) array[1..N] var of set 1..N: preferences = [{2,3}, {1}, ..., {2,3,4}];

b) int[][] preferences = {{2,3}, {1}, ..., {2,3,4}};
```

Figure 1: Personal preferences for the photo in minizinc (a) and Java (b).

# 8 (5 p.)

Figure 2 depicts a task graph for a set of six jobs with precedence relations (one finished before another). The jobs are executed on two machines ($m_1$ and $m_2$) in such a way that only one job can use a machine at any time moment. Some jobs require the same machine. Determine a suitable schedule that fulfills all constraints and minimizes the schedule length.

# 9 (5 p.)

There are three warehouses with capacity of identical products 500, 300 and 400 respectively. Four clients have certain demands on these products. Client A demands 200, B demands 400, C demands 300, and D demands 100 products. Transport costs between warehouses and clients is defined by the following matrix.
Organize the supply such that the transport costs are minimal.

Write a pseudo-code or minizinc program that models and solves this problem.

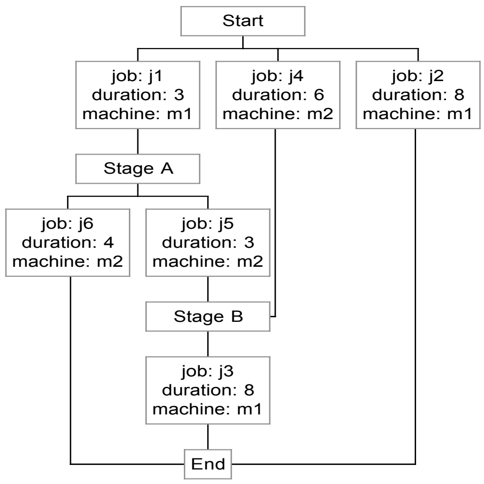


Figure 2: Example task graph for six jobs.

| Warehouse | Client A | Client B | Client C | Client D |
|---|---|---|---|---|
| 1 | 10 | 8 | 5 | 9 |
| 2 | 7 | 5 | 5 | 3 |
| 3 | 11 | 10 | 8 | 7 |

## 10 (6 p.)

The traveling salesperson needs to visit four cities, A, B, C, and D, to sell products. There is different distance between cities and the salesperson wants to optimize the route. The distance between cities is given in the table. Note, that it is non-symmetrical traveling salesperson problem, i.e. distances are different in different directions.

| | A | B | C | D |
|---|---|---|---|---|
| A | - | 3 | 6 | 41 |
| B | 4 | - | 40 | 5 |
| C | 8 | 42 | - | 4 |
| D | 37 | 6 | 2 | - |

- Write a program using FD solver which finds the minimal distance route which contains all cities.
- Assume that the salesperson distributes parcels in the cities as presented in the table below. Positive integer means that the specified number of parcels is loaded into the car and negative integer means that this number of parcels is unloaded. Find a shortest rout that fulfills car maximum capacity of 5 parcels, i.e., a route that one cannot unload from an empty car and load number of parcels that exceeds car capacity.

| | Cities | | | |
|---|---|---|---|---|
| | A | B | C | D |
| Distribution | 2 | -3 | 4 | -1 |

Write a pseudo-code or minizinc program that models and solves this problem.