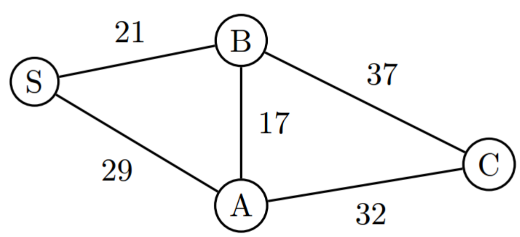
Practical Assignment Part 2
Automated Reasoning 2IMF25

Technische Universiteit Eindhoven
Jiahuan Zhang 0896785 (j.4.zhang@student.tue.nl)
Hector Joao Rivera Verduzco 0977393 (h.j.rivera.verduzco@student.tue.nl)

Problem 1

Three non-self-supporting villages A, B and C in the middle of nowhere consume one food package each per time unit. The required food packages are delivered by a truck, having a capacity of 300 food packages. The locations of the villages are given in the following picture, in which the numbers indicate the distance, more precisely, the number of time units the truck needs to travel from one village to another, including loading or delivering. The truck has to pick up its food packages at location S containing an unbounded supply. The villages only have a limited capacity to store food packages: for A and B this capacity is 120, for C it is 200. Initially, the truck is in S and is fully loaded, and in A, B and C there are 40, 30 and 145 food packages, respectively.



(a) Show that it is impossible to deliver food packages in such a way that each of the villages consumes one food package per time unit forever.

(b) Show that this is possible if the capacity of the truck is increased to 320 food packages. (Note that a finite graph contains an infinite path starting in a node v if and only if there is a path from v to a node w for which there is a non-empty path from w to itself.)

(c) Figure out whether it is possible if the capacity of the truck is set to 318.

Solution:

We generalize this problem for n number of villages and a truck with capacity T. We introduce $n \times (m+1)$ integer variables a_{ij} for $i = 1, ..., n$ and $j = 0, ..., m$, where m is the number of travels that the truck has performed, and a_{ij} represents the number of food packages in the village i after performing j number of travels. We also introduce $m+1$ integer variables p_j,

t_j and d_j for $j = 0, ..., m$, where p_j and t_j represent the position of the truck and the number of food packages in it respectively after j number of travels. Finally, d_j depicts the amount of food packages that are delivered to a village just after performing j number of travels.

First we define the boundaries of each variable. To do so, we define $max(i)$ as the maximum number of food packages that village i can store. Then we have the following formula that expresses that the number of food packages in each village should be in the range of zero and its store's capacity:

$$\bigwedge_{i=1}^{n} \bigwedge_{j=0}^{m} 0 \leq a_{ij} \leq max(i).$$

Similarly, we construct the formula for the boundaries of the reminding variables.

$$\bigwedge_{j=0}^{m} 0 \leq p_j \leq n \ \wedge$$

$$\bigwedge_{j=0}^{m} 0 \leq t_j \leq T \ \wedge$$

$$\bigwedge_{j=0}^{m} 0 \leq d_j.$$

It is worth to mention that $p_j = 0$ means that the truck is in location S (Supplier), whereas $p_j = i$ for $i = 1, ..., n$ depicts that the truck is in village i. Hence, the boundaries of p_j must be in the range of zero and n.

The problem specifies the condition that initially the truck is fully loaded and in position S. Additionally, some villages already have an amount of food packages, lets define this initial number of packages as A_i for village i. Then we have the following formula:

$$p_0 = 0 \ \wedge$$

$$t_0 = T \ \wedge$$

$$\bigwedge_{i=0}^{n} a_{i0} = A_i.$$

Next, we express the condition that the number of delivered packages d in village i is limited by the capacity of this village. Also, when the truck is in position zero (Supplier) it should not deliver any package.

$$\bigwedge_{j=0}^{m} \bigwedge_{i=1}^{n} (p_j = i) \rightarrow (d_j \leq (max(i) - a_{ij})) \ \wedge$$

$$\bigwedge_{j=0}^{m} (p_j = 0) \rightarrow (d_j = 0).$$

Finally, when the truck is in a given position, the next position should be a neighboring village. Also, when moving to another position the amount of food in every village will decrement depending on the travel time. In order to express this conditions, we introduce $\mathbf{C}_i$ as the sets of neighbors of village i, and the mappings $f_i : \mathbf{C}_i \to \mathbf{N}$ to determine the time required to travel from i to one of its neighbor villages, eg. lets say that the set of neighbors of village 1 is $\mathbf{C}_1 = \{0, 2\}$ then villages 0 and 2 are connected to village 1, and $f_1(0)$ is the time required to go from village 1 to village 0.

Using all this elements, the formula that expresses this condition is the following:

$$\bigwedge_{j=0}^{m-1} \bigwedge_{i=0}^{n} (p_j = i) \to (\bigvee_{k \in \mathbf{C}_i} (p_{j+1} = k \ \wedge \ a_{ij+1} = a_{ij} + d_j - f_i(k) \ \wedge$$

$$\bigwedge_{1 \leq l \leq n : i \neq l} a_{lj+1} = a_{lj} - f_i(k) \ \wedge$$

$$(k = 0) \to (t_{j+1} \geq t_j) \ \wedge$$

$$(k \neq 0) \to (t_{j+1} = t_j - d_{j+1}))).$$

It is worth to mention that the last two expressions of this big formula express that when the selected neighbor is the Supplier ($k = 0$), then the next amount of packages in the truck t can only be bigger or equal to the previous one, because in this position the truck is being filled again. When $k \neq 0$ the packages in the truck will decrement since some packages will be delivered to the selected neighbor k.

The total formula now consists of the conjunction of all these ingredients. We can find a particular solution for this problem choosing $n = 3$, $T = 300$, $max(1) = max(2) = 120$, $max(3) = 200$, $A_1 = 40$, $A_2 = 30$, $A_3 = 145$, $\mathbf{C}_0 = \{1, 2\}$, $\mathbf{C}_1 = \{0, 2, 3\}$, $\mathbf{C}_2 = \{0, 1, 3\}$, $\mathbf{C}_3 = \{1, 2\}$ and the values of $f_i(k)$ as depicted in the picture of the villages.

The complete formula expressed in SMT syntax is as follow:

```
;Practical Assignment - Automated_ Reasoning 2IMF25
;Problem 1
(benchmark test.smt
:logic QF_UFLIA
:extrafuns
((a1_0 Int) (a2_0 Int) (a3_0 Int)
(a1_1 Int) (a2_1 Int) (a3_1 Int)
(a1_2 Int) (a2_2 Int) (a3_2 Int)
......
(p_0 Int) (t_0 Int) (d_0 Int)
(p_1 Int) (t_1 Int) (d_1 Int)
(p_2 Int) (t_2 Int) (d_2 Int)
......
)
:formula
(and
;the initial values for each village, the truck and position
(= p_0 0)
(= a1_0 40)
(= a2_0 30)
```

```
(= a3_0 145)
(= t_0 300)
;Bound of each variable
(>= a1_0 0) (<= a1_0 120) (>= a2_0 0) (<= a2_0 120) (>= a3_0 0) (<= a3_0 200)
(>= a1_1 0) (<= a1_1 120) (>= a2_1 0) (<= a2_1 120) (>= a3_1 0) (<= a3_1 200)
(>= a1_2 0) (<= a1_2 120) (>= a2_2 0) (<= a2_2 120) (>= a3_2 0) (<= a3_2 200)
......
(>= p_0 0) (<= p_0 3) (>= t_0 0) (<= t_0 300) (>= d_0 0)
(>= p_1 0) (<= p_1 3) (>= t_1 0) (<= t_1 300) (>= d_1 0)
(>= p_2 0) (<= p_2 3) (>= t_2 0) (<= t_2 300) (>= d_2 0)
......
;Step 1
(implies (= p_0 0) (and (= d_0 0)
(or (and (= p_1 1) (= a1_1 (- a1_0 29)) (= a2_1 (- a2_0 29)) (= a3_1 (- a3_0 29)) (= t_1 (- t_0 d_1)))
(and (= p_1 2) (= a1_1 (- a1_0 21)) (= a2_1 (- a2_0 21)) (= a3_1 (- a3_0 21)) (= t_1 (- t_0 d_1))))))

(implies (= p_0 1) (and (<= d_0 (- 120 a1_0))
(or (and (= p_1 0) (= a1_1 (- (+ a1_0 d_0) 29)) (= a2_1 (- a2_0 29)) (= a3_1 (- a3_0 29)) (>= t_1
t_0))
(and (= p_1 2) (= a1_1 (- (+ a1_0 d_0) 17)) (= a2_1 (- a2_0 17)) (= a3_1 (- a3_0 17)) (= t_1 (- t_0
d_1)))
(and (= p_1 3) (= a1_1 (- (+ a1_0 d_0) 32)) (= a2_1 (- a2_0 32)) (= a3_1 (- a3_0 32)) (= t_1 (- t_0
d_1))))))

(implies (= p_0 2) (and (<= d_0 (- 120 a2_0))
(or (and (= p_1 0) (= a1_1 (- a1_0 21)) (= a2_1 (- (+ a2_0 d_0) 21)) (= a3_1 (- a3_0 21)) (>= t_1
t_0))
(and (= p_1 1) (= a1_1 (- a1_0 17)) (= a2_1 (- (+ a2_0 d_0) 17)) (= a3_1 (- a3_0 17)) (= t_1 (- t_0
d_1)))
(and (= p_1 3) (= a1_1 (- a1_0 37)) (= a2_1 (- (+ a2_0 d_0) 37)) (= a3_1 (- a3_0 37)) (= t_1 (- t_0
d_1))))))

(implies (= p_0 3) (and (<= d_0 (- 200 a3_0))
(or (and (= p_1 1) (= a1_1 (- a1_0 32)) (= a2_1 (- a2_0 32)) (= a3_1 (- (+ a3_0 d_0) 32)) (= t_1 (-
t_0 d_1)))
(and (= p_1 2) (= a1_1 (- a1_0 37)) (= a2_1 (- a2_0 37)) (= a3_1 (- (+ a3_0 d_0) 37)) (= t_1 (- t_0
d_1))))))
......
))
```

Now we try to find a solution for each interrogant of the original problem:

(a) Show that it is impossible to deliver food packages in such a way that each of the villages consumes one food package per time unit forever.

Applying `yices-smt part2_1a.smt`, it yields SAT when choosing $m = 20$ but yields UNSAT when generating the code for $m = 21$, hence the truck can make at most 20 travels between villages before one consumes all its food. We conclude that is impossible to deliver food packages in such a way that each of the villages consumes one food package per time unit forever.

(b) Show that this is possible if the capacity of the truck is increased to 320 food packages.

In order to find a solution where the truck can deliver food forever, we first try to find a state that can be reached again in the future, so for this case the truck can perform the same route forever always passing for the same states. We find this adding conditions to the yices' code to compare two different states and yields SAT if they are equal. Performing some experiments we found out that after adding the following formula it yields satisfiable:

$$a_{1\_10} = a_{1\_3} \;\wedge$$

$$a_{2\_10} = a_{2\_3} \;\wedge$$

$$a_{3\_10} = a_{3\_3} \;\wedge$$

$$a_{1\_10} = a_{1\_3} \;\wedge$$

Choosing $T = 320$ and applying `yices-smt -m part2_1b.smt` to the generated code, the tool yields the following result:

```
sat
(= t_0 320)
(= a1_0 40)
(= a2_0 30)
(= a3_0 145)
(= d_0 0)
(= p_0 0)
(= t_1 295)
(= a1_1 19)
(= a2_1 9)
(= a3_1 124)
(= d_1 25)
(= p_1 2)
(= t_2 177)
(= a1_2 2)
(= a2_2 17)
(= a3_2 107)
(= d_2 118)
(= p_2 1)
......
```

Hence we conclude that for the case when the truck has a capacity of 320 food packages, it is possible to deliver food in such a way that each village consumes food forever. The final result for this special case is depicted in the following table:

variables/state	0	1	2	3	4	5	6	7	8	9	10
Truck position (p)	S	B	A	B	S	A	C	B	S	A	B
Food's pkgs in truck (t)	320	295	177	58	320	253	67	0	296	177	58
Food's pkgs delivered (d)	0	25	118	119	0	67	186	67	0	119	119
Food's pkgs in village A ($a1$)	40	19	2	103	82	53	88	51	30	1	103
Food's pkgs in village B ($a2$)	30	9	17	0	98	69	37	0	46	17	0
Food's pkgs in village C ($a3$)	145	124	107	90	69	40	8	157	136	107	90

As can be observed, state 10 is exactly the same as state 3, therefor we have found a route that satisfies the requirement.

(c) Figure out whether it is possible if the capacity of the truck is set to 318.

Similarly, we prove that it is possible to deliver food forever for this case too. We generate the yices' code choosing $T = 318$ and adding the same extra condition as in b), after applying `yices-smt -m part2_1c.smt` it yields the following result:

```
sat
(= t_0 318)
(= a1_0 40)
(= a2_0 30)
(= a3_0 145)
(= d_0 0)
(= p_0 0)
(= t_1 293)
(= a1_1 19)
(= a2_1 9)
(= a3_1 124)
(= d_1 25)
(= p_1 2)
(= t_2 175)
(= a1_2 2)
(= a2_2 17)
(= a3_2 107)
(= d_2 118)
(= p_2 1)
......
```

The following table shows the solution for this problem:

variables/state	0	1	2	3	4	5	6	7	8	9	10
Truck position (p)	S	B	A	B	S	A	C	B	S	A	B
Food's pkgs in truck (t)	318	293	175	55	318	252	66	0	295	175	55
Food's pkgs delivered (d)	0	25	118	120	0	66	186	66	0	120	120
Food's pkgs in village A ($a1$)	40	19	2	103	82	53	87	50	29	0	103
Food's pkgs in village B ($a2$)	30	9	17	0	99	70	38	1	46	17	0
Food's pkgs in village C ($a3$)	145	124	107	90	69	40	8	157	136	107	90

Remark:

Generalization:

Problem 2

Solution:

Remark:

Generalization:

Problem 3

Solution:

Remark:

Generalization:

Problem 4

Solution:

Remark:

Generalization: