UNIVERSITÀ
DI TRENTO

Formal Method Mod. 1 (Automated Reasoning)

Laboratory 6

Giuseppe Spallitta
giuseppe.spallitta@unitn.it

Università degli studi di Trento

April 21, 2021

UNIVERSITÀ DEGLI STUDI DI TRENTO

Q&A: define-fun using functions

- We can use define-fun not only to initialize the value of a variable, but also to initialize the value of a function accepting input arguments!
- Let's define $f(x, y)$ as the function computing the remainder between x and y:

```
(declare-const x Int)
(declare-const y Int)
(define-fun f ((x Int) (y Int)) (mod x y))
```

UNIVERSITÀ DEGLI STUDI DI TRENTO

Q&A: generic functions

- The only way to define generic functions (given a generic input, the output is constrained by a formula) would be using the *foall* function:
  (declare-fun isEven (Int) Bool)
  (assert (forall ((y Int)) (= (isEven y) (= 0 (mod y 2)))))
- Currently MathSAT does not support it, but other SMT solvers enable it, such as z3 (try it using this link `https://compsys-tools.ens-lyon.fr/z3/index.php`).

UNIVERSITÀ DEGLI STUDI
DI TRENTO

Outline

UNIVERSITÀ DEGLI STUDI DI TRENTO

OptiMathSAT

- OptiMathSAT is an extension of MathSAT 5.
- OptiMathSAT allows for incremental multi-objective optimization over linear arithmetic objective functions
- OptiMathSAT supports a wide range of theories (including e.g. equality and uninterpreted functions, linear arithmetic, bit-vectors, and arrays).
- More information can be found here:
  `http://optimathsat.disi.unitn.it/index.html`

OptiMathSAT input format: Extended SMT-LIB

- OptiMathSAT accepts SMT-LIB as input format, enriching it with some additional commands and instructions to enable the optimization tasks.
- All the available extensions are listed in the following page: `http://optimathsat.disi.unitn.it/pages/smt2reference.html`.
- OptiMathSAT also accept a second input format, a subset of a constraint programming paradigm called FlatZinc, that we won't cover in this course.

UNIVERSITÀ DEGLI STUDI
DI TRENTO

Outline

Selling apples

UNIVERSITÀ DEGLI STUDI DI TRENTO

Exercise 6.1: farming

There are 50 apple trees in an orchard. Each tree produces 800 apples. For each additional tree planted in the orchard, the output per tree drops by 10 apples. How many trees should be added to the existing orchard in order to maximize the total output of trees ?

Selling apples: variables

UNIVERSITÀ DEGLI STUDI DI TRENTO

As always, we first define the variables that efficiently describe the problem:

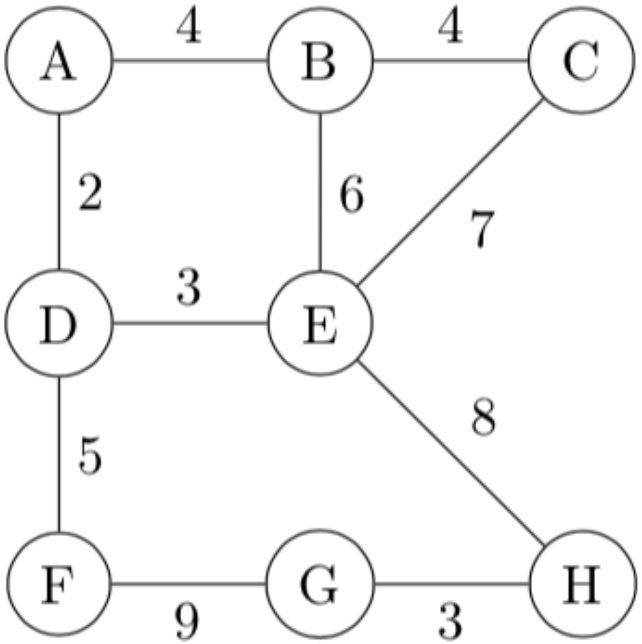
- We are interested in knowing the number of additional trees we can plant without losing profit. We call this variable n.
- n should be instanciated as `Int`.

Selling apples: properties

- The total number of trees we will plant will be $50 + n$.
- For each additional tree planted, each tree will produce fewer apples, following the equation $800 - 10n$.
- The problem should be set as a maximization problem over the total number of produced apples, obtained as the product between the two numbers.
  $\Rightarrow$ Remember you could maximize complex cost functions, not only single variables.

Calcutating graph properties

Exercise 6.2: shortest path



Use OptiMathSAT to compute the shortest path between G and B.

Shortest path: variables

UNIVERSITÀ DEGLI STUDI DI TRENTO

As always, we first define the variables that efficiently describe the problem:

- For each node we require a variable, storing the value of the current shortest path from node G.
- Consequently we must declare `Int` variables.

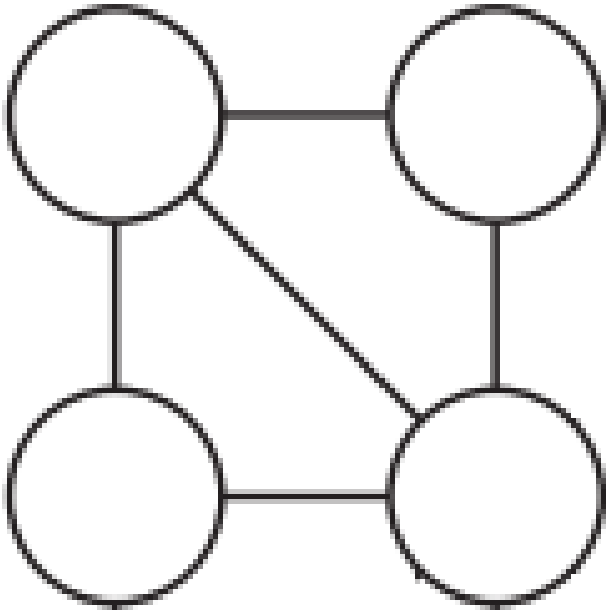
UNIVERSITÀ DEGLI STUDI DI TRENTO

Shortest path: properties

- The starting point score should be set to 0.
- For the other nodes we must define the possible constraints to update their score, considering that each edge could modify its score.
- The problem should be set as a minimization problem over B (but we can also choose the other points).
- We can extend the encoding to generalize the problem to two generics points, instead of fixing the starting point.

Calcutating graph properties

Exercise 6.3: maximum clique



Use OptiMathSAT to compute the maximum clique on the graph. A clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent.

Maximum clique: variables

UNIVERSITÀ DEGLI STUDI DI TRENTO

As always, we first define the variables that efficiently describe the problem:

- For each node we require a variable, storing if they can be used to define a clique.
- We declare `Int` variables, using them as 0-1 variables to state their membership to the clique.
- Why not using `Bool`? We will define a **PseudoBoolean (PB) SMT problem**, nut an equivalent PB SAT problem can be instantiated to achieve the same result.

UNIVERSITÀ DEGLI STUDI DI TRENTO

Maximum clique: properties

- The main idea is that if two nodes are not connected by an edge, then we must ensure that not both of them are chosen:

$$x_i + x_j \leq 1 \tag{1}$$

- Now we should find the maximum number of nodes that can satisfy the constraints, thus belonging to the clique

Maximum clique: properties

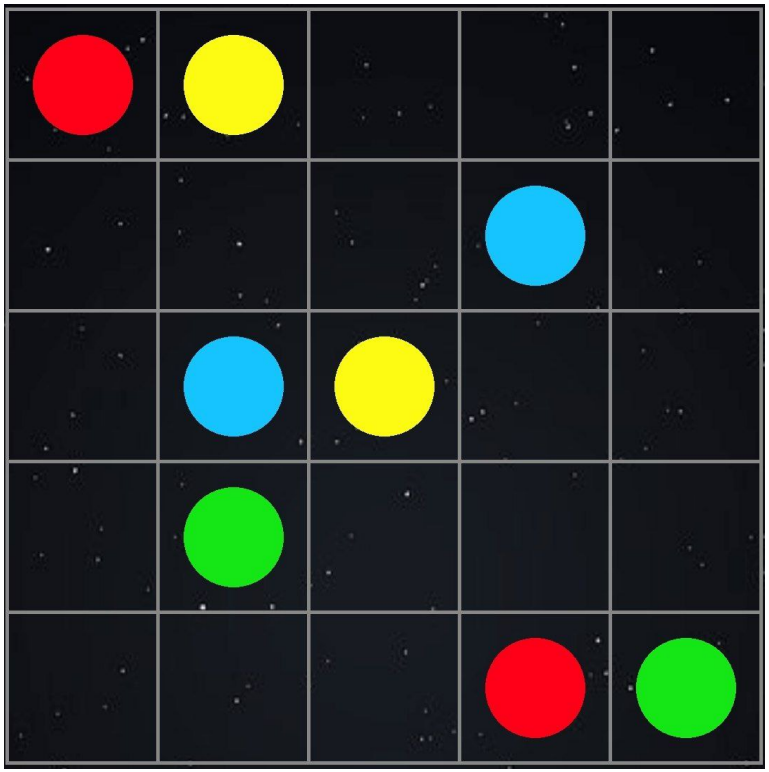
- We can define a penalty function that increases by 1 every time one variable does not belong to the clique (its value is not 1).
- Our goal will be minimizing this penalty function: the higher the better.
- Using MaxSAT will consider this condition: each assertion must be declared as a soft clause, if it is not satisfied by the assignment a penalty (chosen by the user, default is 1) is added to the final penalty score. Just be sure to use the same weight for each soft clause (there is no priority).

UNIVERSITÀ DEGLI STUDI
DI TRENTO

Outline

Connect dots

Exercise 6.4: connect dots



Use MathSAT to solve the puzzle shown in the figure. The rules are simple: you must connect dots with the same color with a single line and all cells must be used to generate a valid solution.

Connect dots: variables

UNIVERSITÀ DEGLI STUDI DI TRENTO

As always, we first define the variables that efficiently describe the problem:

- For each cell of the grid a variable
- Creating an `Int`-to-Color mapping (thus setting `Int` as variable type) is enough for our goal.

Connect dots: properties (1)

UNIVERSITÀ DEGLI STUDI DI TRENTO

- For each cell we must ensure their value is in the range of the admitted color. Setting both lower and upper bound is enough.
- In each line connecting two dots, the following properties always holds:
  - The extremes have one neighbor cell with the same color.
  - The internal nodes of the lines two neighbor cells with the same color.

Connect dots: properties (2)

UNIVERSITÀ DEGLI STUDI DI TRENTO

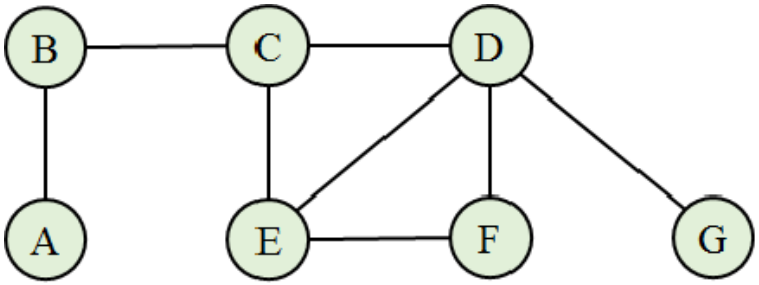
- Both conditions can be encoded using respectively *AtLeastOne* and *AtLeastTwo*.
- According to our reasoning we should use *ExactlyOne* and *ExactlyTwo*, but the *AtLeast* operators are sufficient to correctly constrain the problem in our case (and it is simpler to encode).
- If you provide the *ExactlyOne* encodings you'll obtain the same result, so feel free to implement it.

UNIVERSITÀ DEGLI STUDI
DI TRENTO

Outline

Minimum cover set

Homework 6.1: minimum vertex cover



A vertex-cover set of an undirected graph is a subset of vertices such that if an edge belongs to the graph, then at least one of the two nodes linked by this edges belong to the vertex-cover subset. Use OptiMathSAT to compute the minimum vertex-cover set of the graph in figure.

Homeworks

UNIVERSITÀ DEGLI STUDI DI TRENTO

Homework 6.2: checking professor's interpolants

Find two non-negative numbers x and y so that:

- Their sum is 9.
- The product of one number and the square of the other number is a maximum in the range $(0, 200)$.

Homeworks

Homework 6.3: graph coloring... again

Solve the color graph problem again with the following map of countries (so you must ensure adjacent countries do not have the same color). This time use OMT to retrieve the minimum number of color that satisfy the problem.