|  |
| --- |
| ARTIFICIAL INTELLIGENCE AND KNOWLEDGE ENGINEERING |
| CSP Project “Time Equation” |
| Second assignment |
| **Team 31**  **Álvaro Martín Orive 78955100**  **Sonia Piérola Abraldes 78907376** |
|  |
| **24/04/2011** |

|  |
| --- |
|  |

# Problem Formulation and Design

1. Define the problem as a Constraint Satisfaction Problem (CSP)

### Variables

In this CSP the variables are the digits of the equation. It can be defined as

* Ten minute digit (A)
* Unit minute digit (B)
* Ten second digit (C)
* Unit second digit (D)
* Multiplier (M)
* Result ten minute digit (E)
* Result unit minute digit (F)
* Result ten second digit (G)
* Result unit second digit (H)

### Value Domains

For this CSP there is just one domain for all the variables:

D = {1 ,2, 3, 4, 5, 6, 7, 8 ,9}

Another option that may be interesting to take into account is to specify a domain D1 = {0,1,2,3,4,5} for the ten seconds digit, but this domain is a subset of our main domain, so this particularity is going to be controlled by the specification of the constraints

### Constraints

Unary constraints

* C<6
* G<6
* H = Constant
* M = Multiplier
* A <= Ten of the Max Minutes
* F <= Ten of the Max Minutes

Binary constraints

* FE <= Max Minutes
* AB <= Max Minutes (it could be optimized to AB <= Max Minutes / Multiplier + X2 )

Global constraints

* No variable can take the same value as the Multiplier. (they will be converted into unary constraints)
* No variable can take the same value as the Constant. (they will be converted into unary constraints)
* ( ( 10 \* C + D ) \* Multiplier ) % 60 == 10 \* G + Constant
* ( 10 \* A + B ) \* Multiplier + ( ( 10 \* C + D ) \* Multiplier ) / 60 = 10 \* E + F

## Type of CSP

The type of CSP chosen in this task is an incremental state formulation. It has been selected in this way due to there is a digit which has a value assigned already, so it could be considered as if the search of the algorithm is partially solved based in an incremental state problem.

**Initial state:** The time equation has assigned the unit second digit of the result and the multiplier.

**Goal test: is the current assignment complete?**

**Final state: A state in which each variable has a value and their constraints are fulfilled.**

**State: Any possible state in which there are assigned values to the variables.**

**Actions: Assign a value to an unassigned variable having checked that every constraint with already assigned variables is satisfied.**

**Transition model: A variable has a value assigned**

## The most suitable search algorithm for this CSP

### Simple Backtracking

This search algorithm and the one with heuristics are the most suitable algorithms for a incremental state problem. This algorithm is the one that has been selected for solving the Time Equation problem. The reason is that is the only algorithm implemented in the CSP library.

### Simple Backtracking with heuristics, which heuristics and why.

It could be interesting to use a Backtracking with heuristic, specifically the MRV heuristic or the Degree heuristic. There are almost two digits that have just a single value in their domain (M – Multiplier and H – Constant), so assigning these values first and then go through the variables that have the smallest domain it is an interesting way to solve the CSP Problem. This method has not been implemented in the practice due to there is no backtracking algorithm with heuristics implemented in the template.

### Minimum Conflicts

In order to use Minimun Conflicts the problem state formulation must be a complete state problem. This algorithm is not applicable because we are facing an incremental state problem.

## UML Class Diagram

In order to see the UML Class Diagrams open the pdf file.