**AI – CSP, definitions and modelling**

CSP algos take advantage of the structure of states, with the aim of identifying the value/variable combinations that **violate** the constraints

Text, letter

Description automatically generatedWe can therefore define:

Where:

* A domain is the set of admittable values
* A constraint is a pair **<scope, relation>**
  + The scope is made of the variables that participate in the constraint(s)
  + The relation defines the values that those variables can take on, that satisfy the constraint(s)

For example, if X1 and X2 both have the domain {1,2,3}, then the constraint saying that X1 must be greater than X2 can be written as:

* 〈(X1, X2),{(3,1),(3,2),(2,1)} 〉 or
* 〈(X1, X2), X1 > X2〉

Types of assignments:

* **Consistent assignment**: the assignment doesn't violate any constraints
* **Complete assignment**: every variable is assigned a value
* **Partial assignment**: leaves some variables unassigned
* **Partial solution**: Partial assignment that is consistent

**The solution to a CSP is a complete, consistent assignment**

CSPs admit the usage of backtracking during the search phase (evolving from informed and uninformed search problems)

CSPs provide a **factored** representation of states, as opposed to FSAs, which represent states **atomically** (i.e. in an indivisible manner)

***Between declarative and procedural paradigms***

Declarative programming is a paradigm describing WHAT the program does, without explicitly specifying its control flow

Imperative programming is a paradigm describing HOW the program should do something by explicitly specifying each instruction (or statement) step by step, which mutate the program's state

***Constraint graphs – The AUS Map Coloring problem***

Given:

* The variables:
  + X = WA, NT, Q, NSW, V, SA, T
* The domains:
  + D = {red, green, blue}
* The constraints:
  + C = {SA≠WA, SA ≠NT, SA ≠Q, SA ≠NSW, SA ≠V, WA ≠NT, NT ≠Q, Q ≠NSW, NSW ≠V}

Find a solution for the CSP

A picture containing text, diagram, screenshot, circle

Description automatically generatedWe should use a **constraint graph** to solve the problem

We can see that any assignment for T would work, as it’s not connected to the graph, and the assignments are done using search algos: whenever we find that a partial assignment has violated a constraint, we immediately discard further processing on it

The constraint graph is inadequate with more than binary constraints; in that case, we use an **hypergraph**, with ordinary nodes and hypernodes – which represent the *n-*ary constraints involving *n* variables

Constraints can be:

* **Unary →** only one variable involved
* **Binary →** two variables involved
* **Ternary →** three variables involved
* **A picture containing text, screenshot, diagram, circle

  Description automatically generatedGlobal →** arbitrary arity