

```

1  #----- Domain Class-----
2  class Domain():
3      def __init__(self, name, start, end):
4          self.name = name
5          self.values = range(start,end+1)
6          self.vars = []
7
8
9
10     def addVariable(self,variable):
11         self.vars.append(variable)
12
13     def getValues(self):
14         return self.values
15
16     def getVars(self):
17         return vars
18
19     def strSelectVariables(self):
20         print("vars"+str(self.vars))
21         res = ""
22         for v in self.vars:
23             res += "?" + v + " "
24         return res
25
26     def __str__(self):
27         res = ":" + self.name + " rdf:type :Domain ;\n" + "\t:values "
28         for val in self.values[:-1]:
29             res += str(val) + ", "
30         res += str(self.values[-1]) + " ;\n\t:variables"
31         for var in self.vars[:-1]:
32             res += " : " + str(var) + ", "
33         res += " : " + self.vars[-1] + ".\n\n"
34         return res
35
36
37  #----- Constraint Class-----
38  class Constraint:
39      def __init__(self):
40          self.typeCons = ""
41          self.vars = []
42          self.values = []
43          self.first = ""
44          self.second = ""
45          self.third = ""
46
47      def addVar(self, var):
48          self.vars.append(var)
49
50      def addValue(self, value):
51          self.values.append(value)
52
53      def setTypeCons(self,typeCons):
54          self.typeCons = typeCons
55          if self.typeCons == "Reject:\n":
56              self.first = " != "
57              self.second = " || "
58              self.third = " && "
59          elif self.typeCons == "Accept:\n":
60              self.first = " = "
61              self.second = " && "
62              self.third = " || "
63
64      def __str__(self):
65          print(self.vars)
66          print(self.values)
67          res = "\t\t( "
68          for i in range(len(self.values)):
69              res += "("
70              for j in range(len(self.vars)):
71                  res += "?" + str(self.vars[j]) + self.first + str(self.values[i][j])
72                  if j == len(self.vars) - 1:

```

```

73         res += ")"
74     else:
75         res += self.second
76
77         if i != len(self.values) - 1:
78             res += self.third
79     res += " )\n"
80     return res
81
82
83
84
85 #----- Script itself -----
86 class MainRun:
87     def __init__(self):
88         self.inFileName = ""
89         self.outFileName = ""
90         self.domains = {}
91         self.fileOutRDF = None
92         self.fileOutSPAQRL = None
93
94     def writeDomains(self):
95         for d in self.domains.keys():
96             self.fileOutRDF.write(str(self.domains[d]))
97
98     def parseConstraints(self, file, nConst):
99         nConstParsed = 0
100         for line in file:
101             constraint = Constraint()
102             if("Vars:" in line):
103                 if nConstParsed < nConst:
104                     nVars = int(file.readline())
105                     for x in range(nVars):
106                         var = file.readline().rstrip('\n')
107                         constraint.addVar(var)
108                     typeCons = file.readline()
109                     constraint.setTypeCons(typeCons)
110                     nValues = int(file.readline())
111                     for x in range(nValues):
112                         lineValue = file.readline().rstrip('\n')
113                         constraint.addValue(lineValue.split())
114                     if nConstParsed + 1 < nConst:
115                         self.fileOutSPAQRL.write(str(constraint) + "\t\t&& \n")
116                     else:
117                         self.fileOutSPAQRL.write(str(constraint))
118                     nConstParsed += 1
119
120     def writeSelect(self):
121         self.fileOutSPAQRL.write("SELECT ")
122         for _ , value in self.domains.items():
123             self.fileOutSPAQRL.write(value.strSelectVariables())
124         self.fileOutSPAQRL.write("\n")
125
126     def writeWhere(self):
127         self.fileOutSPAQRL.write("WHERE {\n")
128         for key , value in self.domains.items():
129             listVar = value.vars
130             for v in listVar:
131                 self.fileOutSPAQRL.write("\t:" + key + " :values ?" + v + " .\n")
132         self.fileOutSPAQRL.write("\tFILTER {\n")
133
134     def run(self):
135         self.inFileName = input("Enter F2CSP file name:")
136         self.outFileName = input("Enter output file name:")
137         self.fileOutRDF = open(self.outFileName + ".ttl", "w+")
138         self.fileOutRDF.write("@prefix : <http://www.w3.org> .\n")
139         self.fileOutRDF.write("@prefix rdf:
140         <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .\n\n")
141         self.fileOutSPAQRL = open(self.outFileName + ".rq", "w+")
142         self.fileOutSPAQRL.write("PREFIX : <http://www.w3.org>\n")
143         self.fileOutSPAQRL.write("PREFIX rdf:
144         <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n")

```

```

143
144     fileIn = open(self.inFileName, "r")
145     for line in fileIn:
146         if("Domains:" in line):
147             nDomains = int(fileIn.readline())
148             for _ in range(nDomains):
149                 currD = fileIn.readline()
150                 d = currD.split()
151                 self.domains[d[0]] = Domain(d[0], int(d[1][0]),int(d[1][-1]))
152         if("Variables:" in line):
153             nVars = int(fileIn.readline())
154             for _ in range(nVars):
155                 currV = fileIn.readline()
156                 v = currV.split()
157                 self.domains[v[1]].addVariable(v[0])
158             self.writeDomains()
159         if("Constraints:" in line):
160             self.writeSelect()
161             self.writeWhere()
162             self.parseConstraints(fileIn,int(fileIn.readline()))
163             self.fileOutSPAQRL.write("\t)\n")
164             self.fileOutSPAQRL.write("}")
165     fileIn.close()
166     self.fileOutRDF.close()
167     self.fileOutSPAQRL.close()
168     print("SCRIPT END")
169
170
171
172 scriptRun = MainRun()
173 scriptRun.run()

```