

```

1 TAD rur ES nat
2 TAD característica ES string
3
4 TAD Robot
5   generos robot
6   exporta id, cars, instanciar
7   igualdad observacional
8     ( $\forall r_1, r_2 : \text{robot}$ ) ( $r_1 =_{\text{obs}} r_2 \leftrightarrow \text{id}(r_1) =_{\text{obs}} \text{id}(r_2)$ )
9   observadores basicos
10    id: robot  $\rightarrow$  rur
11    cars: robot  $\rightarrow$  conj(característica)
12   generadores
13     instanciar: rur  $i \times$  conj(característica)  $cs \rightarrow$  robot  $\{i \geq 1 \wedge cs \neq_{\text{obs}} \emptyset\}$ 
14   axiomas
15     id(instanciar(r, cs))  $\equiv$  r
16     cars(instanciar(r, cs))  $\equiv$  cs
17 Fin TAD
18
19 TAD Restriccion
20   generos restriccion
21   exporta AND, OR, NOT, VAR, cumple
22   igualdad observacional
23     ( $\forall r_1, r_2 : \text{restriccion}$ ) ( $r_1 =_{\text{obs}} r_2 \leftrightarrow$ 
24       ( $\forall c : \text{conj(característica)}$ ) ( $\text{cumple}(c, r_1) =_{\text{obs}} \text{cumple}(c, r_2)$ )))
25   observadores basicos
26     cumple: conj(característica)  $\times$  restriccion  $\rightarrow$  bool
27   generadores
28     AND: restriccion  $\times$  restriccion  $\rightarrow$  restriccion
29     OR: restriccion  $\times$  restriccion  $\rightarrow$  restriccion
30     NOT: restriccion  $\rightarrow$  restriccion
31     VAR: característica  $\rightarrow$  restriccion
32   otras operaciones
33     TRUE:  $\rightarrow$  restriccion
34     FALSE:  $\rightarrow$  restriccion
35   axiomas
36     cumple(cs, VAR(c))  $\equiv c \in cs$ 
37     cumple(cs, NOT(c))  $\equiv \neg(\text{cumple}(cs, c))$ 
38     cumple(cs, AND(c1, c2))  $\equiv \text{cumple}(cs, c1) \wedge \text{cumple}(cs, c2)$ 
39     cumple(cs, OR(c1, c2))  $\equiv \text{cumple}(cs, c1) \vee \text{cumple}(cs, c2)$ 
40
41     TRUE  $\equiv$  OR(VAR("dummy"), NOT(VAR("dummy")))
42
43     FALSE  $\equiv$  AND(VAR("dummy"), NOT(VAR("dummy")))
44 Fin TAD
45
46 TAD estacion ES string
47 TAD conexion ES (estacion, restriccion)
48 TAD conexiones ES conj(conexion)
49
50 TAD Mapa
51   generos mapa
52   exporta estaciones, conexiones, nuevo, crearEst, conectar,
53     esBloqueante, conectadas, caminos
54   igualdad observacional
55     ( $\forall m_1, m_2 : \text{mapa}$ ) ( $m_1 =_{\text{obs}} m_2 \leftrightarrow$ 
56       estaciones( $m_1$ )  $=_{\text{obs}}$  estaciones( $m_2$ )  $\wedge$ 
57       ( $\forall e \in \text{estaciones}(m_1)$ ) ( $\text{conexiones}(m_1, e) =_{\text{obs}} \text{conexiones}(m_2, e)$ )))
58   observadores basicos
59     estaciones : mapa  $\rightarrow$  conj(estacion)
60     conexiones : mapa  $m \times$  estacion  $e \rightarrow$  conj(conexion)  $\{e \in \text{estaciones}(m)\}$ 
61   generadores
62     nuevo :  $\rightarrow$  mapa
63     crearEst : mapa  $m \times$  estacion  $a \rightarrow$  mapa  $\{a \notin \text{estaciones}(m)\}$ 
64     conectar : mapa  $m \times$  estacion  $a \times$  estacion  $b \times$  restriccion  $\rightarrow$  mapa  $\{a, b \in \text{estaciones}(m)\}$ 
65   otras operaciones
66     esBloqueante : mapa  $m \times$  conj(característica)  $\times$  estacion  $e \rightarrow$  bool  $\{e \in \text{estaciones}(m)\}$ 
67     esBloqueante' : conj(conexion)  $\times$  conj(característica)  $\rightarrow$  bool
68
69     conectadas : mapa  $m \times$  estacion  $a \times$  estacion  $b \rightarrow$  bool  $\{a, b \in \text{estaciones}(m)\}$ 
70     conectadas' : estacion  $\times$  conj(conexion)  $\rightarrow$  bool
71
72     caminos : mapa  $m \times$  estacion  $a \times$  estacion  $b \rightarrow$  conj(restriccion)
73                $\{a, b \in \text{estaciones}(m) \wedge \text{conectadas}(m, a, b)\}$ 

```

```

74   caminos'      : estacion × conj(conexion) → conj(restriccion)
75 axiomas
76   conexiones(crearEst(m, e), k) ≡ ∅
77   conexiones(conectar(m, a, b, r), e) ≡
78     (if e ≡ a then
79       {b, r}
80     else
81       if e ≡ b then
82         {a, r}
83       else
84         ∅
85       fi
86     fi) u conexiones(m, e)
87
88   estaciones(nuevo) ≡ ∅
89   estaciones(crearEst(m, e)) ≡ { e } u estaciones(m)
90   estaciones(conectar(m, a, b)) ≡ estaciones(m)
91
92   esBloqueante(m, r, e) ≡.
93     if ∅?(conexiones(m, e)) then
94       False
95     else
96       esBloqueante'(conexiones(m, e), r)
97     fi
98
99   esBloqueante'(c, r) ≡
100     if ∅?(c) then
101       True
102     else
103       ¬(cumple(r, π2(dameUno(c)))) ∧1 esBloqueante'(sinUno(c), r)
104     fi
105
106   conectadas(m, a, b) ≡ conectadas'(b, conexiones(m, a))
107
108   conectadas'(a, c) ≡
109     if ∅?(c) then
110       False
111     else
112       π1(dameUno(c)) ≡ a) ∨1 conectadas'(a, sinUno(c))
113     fi
114
115   caminos(m, a, b) ≡ caminos'(b, conexiones(m, a))
116
117   caminos'(a, c) ≡
118     if ∅?(c) then
119       ∅
120     else
121       (if π1(dameUno(c)) ≡ a then
122         π2(dameUno(c))
123       else
124         ∅
125       fi) u caminos'(a, sinUno(c))
126     fi
127 Fin TAD
128
129 TAD Ciudad
130   generos ciudad
131   exporta nueva, agregar, mover, borrar,
132     mapeo, robots, historial, buscar, ultimoId, inspeccion
133   igualdad observacional
134     (∀ c1, c2 : ciudad) (c1 =obs c2 ↔
135       (mapeo(c1) =obs mapeo(c2) ∧ robots(c1) =obs robots(c2)) ∧1
136       (∀ r ∈ robots(c1)) (historial(c1, r) =obs historial(c2, r)) ∧1
137       ((robots(c1) ≠obs ∅) ⇒ ultimoId(c1) =obs ultimoId(c2)))
138   observadores basicos
139     mapeo      : ciudad → mapa
140     robots     : ciudad → conj(robot)
141     historial  : ciudad c × robot r → pila(conexion) {r ∈ robots(c)}
142   generadores
143     nueva      : mapa → ciudad
144     agregar    : ciudad c × conj(caracteristica) cs × estacion e → ciudad
145                 {cs ≠obs ∅ ∧1 e ∈ estaciones(mapeo(c)) ∧1 ¬esBloqueante(mapeo(c), cs, e)}
146     mover      : ciudad c × robot r × conexion x → ciudad {esConexionValida(c, r, x)}
147     borrar     : ciudad c × robot r → ciudad {r ∈ robots(c)}

```

```

148 otras operaciones
149   esConexionValida: ciudad × robot × conexion → bool
150
151   buscar           : ciudad c × rur i → robot {(∃r ∈ robots(c)) (id(r) =obs i)}
152   buscar'          : conj(robot) c × rur i → robot {(∃r ∈ c) (id(r) =obs i)}
153
154   ultimoId         : ciudad c → rur {robots(c) ≠obs ∅}
155   ultimoId'        : ciudad c × rur i × conj(robot) k → rur
156                     {k ⊆ robots(c) ∧ (∃r ∈ robots(c)) (id(r) =obs i)}
157
158   inspeccion       : ciudad c × estacion e → ciudad {e ∈ estaciones(mapeo(c))}
159   inspeccion'      : ciudad c × estacion e × (robot, nat) r × conj((robot, nat)) k → ciudad
160                     {e ∈ estaciones(mapeo(c)) ∧1 k ⊆ obtenerRobots(c, e) ∧1 r ∈ obtenerRobots(c, e)}
161
162   infracciones     : ciudad c × robot r → nat {r ∈ robots(c)}
163   infracciones'    : conj(caracteristica) × pila(conexion) → nat
164
165   obtenerRobots    : ciudad c × estacion e → conj((robot, nat)) {e ∈ estaciones(mapeo(c))}
166   obtenerRobots'   : ciudad c × conj(robot) k × estacion e → conj((robot, nat))
167                     {e ∈ estaciones(mapeo(c)) ∧ k ⊆ robots(c)}
168
169 axiomas
170   mapeo(nueva(m)) ≡ m
171   mapeo(agregar(c, r, e)) ≡ mapeo(c)
172   mapeo(mover(c, r, e)) ≡ mapeo(c)
173   mapeo(borrar(c, r)) ≡ mapeo(c)
174
175   robots(nueva(m)) ≡ ∅
176   robots(agregar(c, cs, e)) ≡
177     { instanciar(if ∅?(robots(c)) then 1 else ultimoId(c) + 1 fi, cs) } ∪ robots(c)
178   robots(mover(c, r, e)) ≡ robots(c)
179   robots(borrar(c, r)) ≡ robots(c) - { r }
180
181   historial(agregar(c, cs, e), r) ≡
182     if id(r) ≡ ultimoId(c) + 1 then
183       apilar((e, cs), vacia)
184     else
185       historial(c, r)
186     fi
187
188   historial(mover(c, r', x), r) ≡
189     if r ≡ r' then
190       apilar(x, historial(c, r))
191     else
192       historial(c, r)
193     fi
194
195   esConexionValida(c, r, x) ≡
196     r ∈ robots(c) ∧1
197     π1(x) ∈ estaciones(mapeo(c)) ∧1
198     conectadas(mapeo(c), tope(historial(c, r)), π1(x)) ∧1
199     π2(x) ∈ caminos(mapeo(c), tope(historial(c, r)), π1(x))
200
201   buscar(c, i) ≡ buscar'(robots(c), i)
202
203   buscar'(cs, i) ≡
204     if id(dameUno(cs)) ≡ i then
205       dameUno(cs)
206     else
207       buscar'(sinUno(cs), i)
208     fi
209
210   ultimoId(c) ≡ ultimoId'(id(dameUno(robots(c))), sinUno(robots(c)))
211
212   ultimoId'(m, c) ≡
213     if ∅?(c) then
214       m
215     else
216       ultimoId'(if id(dameUno(c)) > m then id(dameUno(c)) else m fi, sinUno(c))
217     fi
218
219   inspeccion(c, e) ≡
220     if ∅?(obtenerRobots(c, e)) then
221       c
222     else

```

```

222     inspeccion'(c, e, dameUno(obtenerRobots(c, e)), sinUno(obtenerRobots(c, e)))
223     fi
224
225     inspeccion'(c, e, r, cs) ≡
226     if  $\emptyset?$ (cs) then
227         if  $\pi_2(r) > 0$  then
228             borrar(c, r)
229         else
230             c
231         fi
232     else
233         inspeccion'(c, e, if  $\pi_2(\text{dameUno}(cs)) > \pi_2(r)$  then dameUno(cs) else r fi, sinUno(cs))
234     fi
235
236     obtenerRobots(c, e) ≡ obtenerRobots'(c, robots(c), e)
237
238     obtenerRobots'(c, cs, e) ≡
239     if  $\emptyset?$ (cs) then
240          $\emptyset$ 
241     else
242         (if tope(historial(c, dameUno(cs))) ≡ e then
243             (dameUno(cs), infracciones(c, dameUno(cs)))
244         else
245              $\emptyset$ 
246         fi) u obtenerRobots'(c, sinUno(cs), e)
247     fi
248
249     infracciones(c, r) ≡ infracciones'(cars(r), historial(c, r))
250
251     infracciones'(c, p) ≡
252     if tamaño(p) ≤ 1 then
253         0
254     else
255          $\beta(\neg \text{cumple}(c, \pi_2(\text{tope}(p)))) + \text{infracciones}'(c, \text{desapilar}(p))$ 
256     fi
257 Fin TAD

```