Appendix A

Ejemplos

Example A.1. La Figura A.1 muestra un grafo dirigido que corresponde a una representación pequeña de tres rutas de evacuación en una zona de riesgo. Podemos definir el siguiente conocimiento asociado a este grafo de la manera siguiente:

```
route(2). route(1). route(0). risk(0). risk(1). risk(2). risk(3).

% nodo(punto,ruta,riesgo)
node(1,1,3).   node(2,0,3).  node(2,1,3).  node(4,0,2). node(5,0,1).
node(11,0,2).  node(8,1,1).  node(9,1,0). node(12,0,3).
node(12,2,3).  node(15,0,2).  node(16,0,1). node(16,0,1).
node(13,0,3).  node(13,2,3). node(17,2,2). node(19,2,0).

% segment (ini,fin,routa)
segment(1,2,1).  segment(2,11,0).  segment(2,4,0). segment(4,5,0).
segment(4,9,0). segment(2,8,1). segment(8,9,1). segment(12,15,0).
segment(12,17,2). segment(15,16,0). segment(16,19,0).
segment(13,15,0). segment(13,17,2). segment(17,19,2).

% townAt(pueblo, nodo)
townAt(p1,1).   townInRisk(p1). townAt(p2,12).  townInRisk(p2).
townAt(p3,13).  townInRisk(p3).
```

```
% busIniAt(autobus,punto).
bus(b1).   busIniAt(b1,p1). bus(b2).   busIniAt(b2,p2). bus(b3).
busIniAt(b3,p3).

shelther(9). shelter(19).
```

□

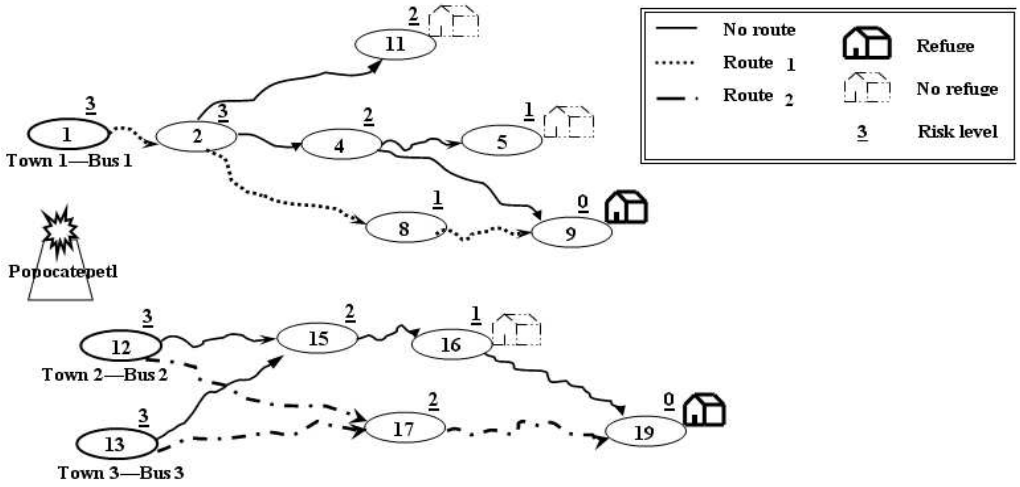

Figure A.1: Dos rutas de evacuación

Example A.2. Consideremos el Ejemplo A.1, la Figura A.1 y el conjunto de observaciones acerca del estado inicial $O = \{$**initially** $position(b1, 9, 1)$; **initially** $position(b2, 12, 2)$; **initially** $position(b3, 13, 2)\}$; y la meta $G = \{end(b1).\ end(b2).\ end(b3).\}$. Entonces una posible codificación $\pi(D, O, G, l)$ en Answer Set Programming para el problema de las rutas de evacuación es el siguiente:

```
% inicialmente el autobus B esta en el nodo N de la ruta R.
initially(position(B, N, R)).

% meta: finalmente del autobus B en un nodo final.
```

```
finally(end(B)).

% fluentes:
% posicion del autobus B es en el nodo Q de la ruta R.
fluent(position(B,Q,R)).

% camino del nodo P al nodo  Q de la ruta R esta bloqueado.
fluent(blocked (P,Q,R)).

% autobus B en un nodo final.
fluent(end(B)):- shelter(B).

% accion travel:
% autobus B  viaja por el camino del nodo  P al nodo Q de la ruta  R,
% donde  R<>0, es decir, los autobuses solo viajan por la ruta de evacuaci\'on.
action(travel(B,P,Q,R)).

% Reglas de cause dinamicas:
% si el autobus B  viaja por el camino del nodo P al nodo Q
% de la ruta R con R<>0, entonces B esta en la posicion Q de la ruta R.
caused(position(B,Q,R),travel(B,P,Q,R)) .

% si el autobus B  viaja por el camino del nodo P al nodo Q
% de la ruta R  entonces B no esta en la posicion P de la ruta R.
caused(neg(position(B,P,R)),travel(B,P,Q,R)).

% Reglas de causa estaticas:
% si el autobus B  esta en la posicion P de la ruta R tal que
% en el nodo P hay un refugio, entonces B esta en una posicion final.
caused(end(B), position(B,P,R)).
```

```
% Condiciones de ejecutabilidad:
% autobus B no puede viajara por un camino del nodo P al nodo Q de la ruta R,
% si B no esta en la posicion P de la ruta R.
noaction_if(travel(B,P,Q,R),neg(position(B,P,R))).

% autobus B no puede viajara por el  camino del nodo P al nodo Q de la ruta R,
% si el camino esta bloqueado.
noaction_if(travel(B,P,Q,R),blocked(P,Q,R)).
```

Entonces, los planes que corresponden a los answer sets de $\pi(D, O, G, l)$ con $l = 3$ son los siguientes:

time 0	time 1	time 2
travel(b1,1,2,1)	travel(b1,2,8,1)	travel(b1,8,9,1)
travel(b2,12,17,2)	travel(b2,17,19,2)	—
travel(b3,13,17,2)	travel(b3,17,19,2)	—

La codificación $\pi(D, O, G, l)$ obtiene todos los caminos posibles para los autobuses desde su posición inicial a su posición final. En este ejemplo todos los autobuses deben seguir sus rutas de evacuación predefinidas, exactamente como se espera cuando dichas rutas son definidas. En los planes se asume que cada acción emplea una unidad de tiempo. □

Example A.3. Consederemos nuevamente el Ejemplo A.2, supongamos que parte de la ruta 2 de evacuación está bloqueada porque se bloqueo $road(12, 17, 2)$, es decir, agregamos **initially** $blocked(12, 17)$ a $\pi(D, O, G, l)$. Puesto que no es posible que el autobús $b2$ sigua la ruta predefinida de evacuación, entonces $\pi(D, O, G, l)$ es inconsistente y no se pueden definir planes de evacuación. Para restaurar consistencia y obtener un plan alternativo agregamos al programa $\pi(D, O, G, l)$ la CR-regla:

$$r_2 : action(travel(B, P, Q, R')) \overset{+}{\leftarrow} bus(B), road(P, Q, R').$$

Esta CR-regla debe ser utilizada solamente si no hay manera de obtener un plan cuando parte de la ruta de la evacuación se bloquea. Entonces, podemos reescribir el programa $\pi(D, O, G, l)$ como sigue:

```
% inicialmente el autobus B esta en el nodo N de la ruta R.
initially(position(B, N, R)).

% meta: finalmente del autobus B en un nodo final.
finally(end(B)).

% fluentes:
% posicion del autobus B es en el nodo Q de la ruta R.
fluent(position(B,Q,R)).

% camino del nodo P al nodo  Q de la ruta R esta bloqueado.
fluent(blocked (P,Q,R)).

% autobus B en un nodo final.
fluent(end(B)):- shelter(B).

% Accion REGULAR travel:
% accion travel:
% autobus B  viaja por el camino del nodo  P al nodo Q de la ruta  R,
% donde  R<>0, es decir, los autobuses solo viajan por la
% ruta de evacuacion.
action(travel(B,P,Q,R)).

% CR-regla donde R' puede ser cero o no,
r_2: action(travel(B,P,Q,R')) :- + bus(B), road(P,Q,R').

% Reglas de cause dinamicas:
```

```
% si el autobus B  viaja por el camino del nodo P al nodo Q
% de la ruta R con R<>0, entonces B esta en la posicion Q de la ruta R.
caused(position(B,Q,R),travel(B,P,Q,R)) .

% si el autobus B  viaja por el camino del nodo P al nodo Q
% de la ruta R  entonces B no esta en la posicion P de la ruta R.
caused(neg(position(B,P,R)),travel(B,P,Q,R)).

% Reglas de causa estaticas:
% si el autobus B  esta en la posicion P de la ruta R tal que
% en el nodo P hay un refugio, entonces B esta en una posicion final.
caused(end(B), position(B,P,R)).

% Condiciones de ejecutabilidad:
% autobus B no puede viajara por un camino del nodo P al nodo Q de la ruta R,
% si B no esta en la posicion P de la ruta R.
noaction_if(travel(B,P,Q,R),neg(position(B,P,R))).

% autobus B no puede viajara por el  camino del nodo P al nodo Q de la ruta R,
% si el camino esta bloqueado.
noaction_if(travel(B,P,Q,R),blocked(P,Q,R)).
```

Entonces, obtenemos cuatro planes de evacuación alternativos, es decir, cuatro answer sets:

Plan1:

tiempo 0	tiempo 1	tiempo 2
travel(b1,1,2,1)	travel(b1,2,4,0)	travel(b1,4,9,0)
travel(b2,12,15,2)	travel(b2,15,16,0)	travel(b2,16,19,0)
travel(b3,13,15,2)	travel(b3,15,16,0)	travel(b3,16,19,0)

Plan2:

tiempo 0	tiempo 1	tiempo 2
travel(b1,1,2,1)	travel(b1,2,8,1)	travel(b1,8,9,1)
travel(b2,12,15,0)	travel(b2,15,16,0)	travel(b2,16,19,0)
travel(b3,13,15,0)	travel(b3,15,16,0)	travel(b3,16,19,0)

Plan3:

tiempo 0	tiempo 1	tiempo 2
travel(b1,1,2,1)	travel(b1,2,8,1)	travel(b1,8,9,1)
travel(b2,12,15,0)	travel(b2,15,16,0)	travel(b2,16,19,0)
travel(b3,13,17,2)	travel(b3,17,19,2)	—

Plan4:

tiempo 0	tiempo 1	tiempo 2
travel(b1,1,2,1)	travel(b1,2,4,0)	travel(b1,4,9,0)
travel(b2,12,15,0)	travel(b2,15,16,0)	travel(b2,16,19,0)
travel(b3,13,17,2)	travel(b3,17,19,2)	—

La codificación $\pi(D, O, G, l)$ en el Ejemplo A.3 obtiene todos los posibles caminos de los autobuses desde sus posiciones inicial hasta sus posiciones finales. Por ejemplo, el Plan 3 dice que el autobus $b1$ y el autobus $b3$ deben seguir sus rutas de evacuación predefinidas mientras el autobus $b2$ debe viajar por caminos fuera de la ruta de evacuación. □

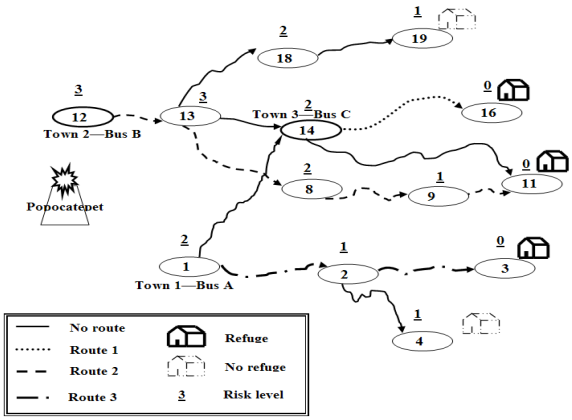

Figure A.2: Tres rutas de evacuación.

Example A.4. Consideremos el ejemplo A.2 pero ahora la acción de viajar permite viajar por segmentos que son o no parte de una ruta de evacuación, es decir, la acción *travel(B,P,Q,R)* ahora es como sigue:

```
% action travel:
% autobus B viaja por el camino del nodo P al nodo Q de la ruta R,
% donde R es igual a  0, 1, 2 o 3, es decir, los autobuses pueden
% viajar por ruta de evacuacion o no:

                    action(travel(B,P,Q,R)).
```

Consideremos en este ejemplo el grafo dirigido de la Figura A.2 podiramos definir los siguientes deseos básicos:

—*travelERass* para expresar que se prefiere que los autobúses viajan por la ruta de evacuación asignada por el gobierno *hasta* que ellos lleguen al refugio.

$$
\begin{aligned}
&\mathit{travelERass} := \\
&\mathbf{until}(\mathbf{occ}(\mathit{travel}(\mathit{busB}, 12, 13, 2)) \vee \mathbf{occ}(\mathit{travel}(\mathit{busB}, 13, 8, 2)) \vee \\
&\qquad \mathbf{occ}(\mathit{travel}(\mathit{busB}, 8, 9, 2)) \vee \mathbf{occ}(\mathit{travel}(\mathit{busB}, 9, 11, 2)) \ , \ \mathit{position}(\mathit{busB}, 11, 2)) \wedge \\
&\mathbf{until}(\mathbf{occ}(\mathit{travel}(\mathit{busC}, 14, 16, 1)) \ , \mathit{position}(\mathit{busC}, 16, 1)) \wedge \\
&\mathbf{until}(\mathbf{occ}(\mathit{travel}(\mathit{busA}, 1, 2, 3)) \vee \mathbf{occ}(\mathit{travel}(\mathit{busA}, 2, 3, 3)) \ , \ \mathit{position}(\mathit{busA}, 3, 3))
\end{aligned}
$$

—*travelER* para expresar que es preferido que los autobuses viajen por caminos que pertenecen a alguna ruta de evacuación y no es importante si ellos viajan o no por su

ruta de evacuación asignada *hasta* que lleguen a su refugio asignado.

$$
\begin{aligned}
&\mathit{travelER} := \\
&\quad \textbf{until}(\\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busB}, 12, 13, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busB}, 13, 8, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busB}, 8, 9, 2)) \vee \\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busB}, 9, 11, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busB}, 14, 16, 1)) \vee \textbf{occ}(\mathit{travel}(\mathit{busB}, 1, 2, 3)) \vee \\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busB}, 2, 3, 3)) \vee \textbf{occ}(\mathit{travel}(\mathit{busC}, 12, 13, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busC}, 13, 8, 2)) \vee \\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busC}, 8, 9, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busC}, 9, 11, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busC}, 14, 16, 1)) \vee \\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busC}, 1, 2, 3)) \vee \textbf{occ}(\mathit{travel}(\mathit{busC}, 2, 3, 3)) \vee \textbf{occ}(\mathit{travel}(\mathit{busA}, 12, 13, 2)) \vee \\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busA}, 13, 8, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busA}, 8, 9, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busA}, 9, 11, 2)) \vee \\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busA}, 14, 16, 1)) \vee \textbf{occ}(\mathit{travel}(\mathit{busA}, 1, 2, 3)) \vee \textbf{occ}(\mathit{travel}(\mathit{busA}, 2, 3, 3)) \ , \\
&\qquad\qquad \mathit{position}(\mathit{busC}, 16, 1) \ \wedge \ \mathit{position}(\mathit{busB}, 11, 2) \ \wedge \ \mathit{position}(\mathit{busA}, 3, 3) \ )
\end{aligned}
$$

—*travelBlSh* para expresar que los autobuses viajen por su ruta de evacuación asignada *hasta que eventualmente* parte de su ruta de evacuación este bloqueda y entonces ellos viajen fuera de la ruta de evacuación *hasta* que ellos lleguen a sus refugios asignados.

$$
\begin{aligned}
&\mathit{travelBlSh} := \\
&\textbf{until}(\textbf{occ}(\mathit{travel}(\mathit{busB}, 12, 13, 2)) \ \vee \textbf{occ}(\mathit{travel}(\mathit{busB}, 13, 8, 2)) \vee \\
&\qquad \textbf{occ}(\mathit{travel}(\mathit{busB}, 8, 9, 2)) \vee \textbf{occ}(\mathit{travel}(\mathit{busB}, 9, 11, 2)) \ , \\
&\qquad \textbf{until}(\ \textbf{eventually}(\mathit{blocked}(12, 13, 2) \ \vee \mathit{blocked}(13, 8, 2)) \vee \mathit{blocked}(8, 9, 2) \vee \mathit{blocked}(9, 11, 2)), \\
&\qquad\qquad \mathit{travel}(\mathit{busB}, 13, 18, 0)) \vee \mathit{travel}(\mathit{busB}, 18, 19, 0)) \vee \mathit{travel}(\mathit{busB}, 13, 14, 2)) \vee \\
&\qquad\qquad \mathit{travel}(\mathit{busB}, 14, 11, 2)) \vee \mathit{travel}(\mathit{busB}, 1, 14, 2)) \vee \mathit{position}(\mathit{busB}, 11, 2) \ ) \ ) \\
&\wedge \\
&\qquad \textbf{until}(\textbf{occ}(\mathit{travel}(\mathit{busC}, 14, 16, 1)) \ , \\
&\qquad\qquad \textbf{until}(\ \textbf{eventually}(\mathit{blocked}(14, 16, 1)), \\
&\qquad\qquad\qquad \mathit{travel}(\mathit{busC}, 13, 18, 0)) \vee \mathit{travel}(\mathit{busC}, 18, 19, 0)) \vee \mathit{travel}(\mathit{busC}, 13, 14, 2)) \vee \\
&\qquad\qquad\qquad \mathit{travel}(\mathit{busC}, 14, 11, 2)) \vee \mathit{travel}(\mathit{busC}, 1, 14, 2)) \vee \mathit{position}(\mathit{busC}, 16, 1) \ ) \ ) \\
&\qquad \wedge \\
&\qquad \textbf{until}(\textbf{occ}(\mathit{travel}(\mathit{busA}, 1, 2, 3)) \ \vee \textbf{occ}(\mathit{travel}(\mathit{busA}, 2, 3, 3)) \ , \\
&\qquad\qquad \textbf{until}(\ \textbf{eventually}(\mathit{blocked}(1, 2, 3) \ \vee \mathit{blocked}(2, 3, 3))), \\
&\qquad\qquad\qquad \mathit{travel}(\mathit{busA}, 13, 18, 0)) \vee \mathit{travel}(\mathit{busA}, 18, 19, 0)) \vee \mathit{travel}(\mathit{busA}, 13, 14, 2)) \vee \\
&\qquad\qquad\qquad \mathit{travel}(\mathit{busA}, 14, 11, 2)) \vee \mathit{travel}(\mathit{busA}, 1, 14, 2)) \vee \mathit{position}(\mathit{busA}, 3, 3) \ ) \ )
\end{aligned}
$$

De manera similar podiramos expresar otros deseos basicos. Una preferencia atomica posible ψ indicando el orden en el cual el conjunto de deseos basicos deben ser satisfechos es la siguiente: $\psi = travelERass \triangleleft travelER \triangleleft travelBlSh$

La preferencia atómica ψ dice que los planes satisfaciendo $travelERass$ son preferidos, en otro caso los planes que satisfacen $travelER$ son preferidos, y finalmente se prefieren los planes que satisfacen $travelBlSh$. □

Example A.5. Notemos que el deseo básico *travelER* del Ejemplo A.4 esta especificando una disjunción consistente de las diferentes opciones que los autobuses tienen para viajar por los arcos que pertenecen a alguna ruta de evacuación predefinida. Sin embargo, supongamos que las rutas de evacuación tienen un número más grande de arcos. Entonces, para expresar *travelER* de una manera similar tendríamos que especificar una disyunción más grande consistente de todos los arcos en las rutas nuevas de evacuación. A pesar de que en [43] se indica que los fluentes y las acciones en $\mathcal{PP}$ con variables eson una abreviación representando el conjunto de todas sus instancias ground, la idea de usar fluentes y acciones con variables no es suficiente para expresar este tipo de problemas. Por lo tanto, hay algunas preferencias que no se pueden expresar en $\mathcal{PP}$ de una manera simple y natural. Para tener una representación natural de esta clase de preferencias definimos el lenguaje $\mathcal{PP}^{par}$ inspirados en [22]. $\mathcal{PP}^{par}$ es una extensión del lenguaje de $\mathcal{PP}$ donde los conectivos propositionales y temporales permiten que representemos de manera compacta las preferencias que tienen una característica particular. Por ejemplo, una representación natural y compacta de la preferencia *travelER* del ejemplo A.4 que usa un *o paramétrico* sería:

$$\textbf{until}(\bigvee\{occ(travel(B,I,F,R)) : bus(B), road(I,F,R), neq(R,0)\},$$
$$\bigwedge\{position(B,Fi,R) : bus(B), shelter(Fi), route(R), neq(R,0)\})$$