Appendix B

Exemples

Example B.1. La figure 8.1 montre un graphique dirigé qui correspond à une petite représentation de trois voies d'évacuation dans une zone à risque. Nous pouvons définir la connaissance suivante associée à ce graphique de telle manière :

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

% node(point,voie,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,voie)
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(ville, nodo)
townAt(p1,1).   townInRisk(p1). townAt(p2,12).  townInRisk(p2).
townAt(p3,13).  townInRisk(p3).
```

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

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

□

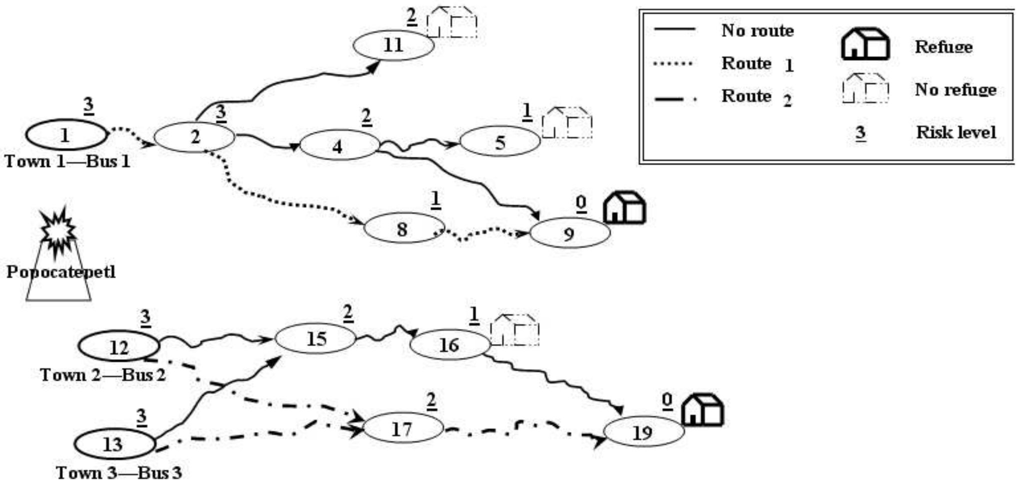

Figure B.1: Deux routes d'évacuation

Example B.2. Nous considérons l'exemple B.1, la figure 8.1 et l'ensemble des observations au sujet de l'état initial $O = \{$**initially** $position(b1, 9, 1)$; **initially** $position(b2, 12, 2)$; **initially** $position(b3, 13, 2)\}$; et le but $G = \{end(b1).\ end(b2).\ end(b3).\}$. Alors une codification possible $\pi(D, O, G, l)$ dans Answer Set Programming pour les problèmes des voies d'évacuation, est la suivante:

```
% Initialement, le bus B est au point N de la voie R.
initially(position(B, N, R)).

% but: finalement le bus B est \'{a} un point final.
```

```
finally(end(B)).

% fluentes:
% la position du bus B est au point Q de la voie R.
fluent(position(B,Q,R)).

% le chemin du point P au point  Q de la voie R est bloqu\'e.
fluent(blocked (P,Q,R)).

%  le bus B \`{a} un point final.
fluent(end(B)):- shelter(B).

% Action travel:
% le bus B  roule sur le chemin du point  P au point Q de la voie R,
% o\`{u}  R<>0, c'est \`{a} dire que les bus roulent
% seulement sur la voie d'\'evacuation.
action(travel(B,P,Q,R)).

% R\`{e}gles de cause dynamiques:
% si le bus B  roule sur le chemin du point P au point Q
% de la voie R avec R<>0, alors B est dans la position Q
% de la voie R.
caused(position(B,Q,R),travel(B,P,Q,R)) .

% si le bus B  roule sur le chemin du point P au point Q
% de la voie R  alors B n'est pas dans la position P de la voie R.
caused(neg(position(B,P,R)),travel(B,P,Q,R)).

% R\`{e}gles de cause statiques:
% si le bus B  est dans la position P de la voie R tel que
% au point P il y a un refuge, alors B est dans une position finale.
```

```
caused(end(B), position(B,P,R)).


% Condicions d'application:
% le bus B ne peut pas rouler sur le chemin du point P au point Q de la voie R,
% si B n'est pas dans la position P de la voie R.
noaction_if(travel(B,P,Q,R),neg(position(B,P,R))).


% bus B ne peut pas rouler sur le chemin du point P au point Q de la voie R,
% si le chemin est bloqu\'e.
noaction_if(travel(B,P,Q,R),blocked(P,Q,R)).
```

Alors les plans qui correspondent aux answer sets de $\pi(D, O, G, l)$ avec $l = 3$ sont les suivants:

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 codification $\pi(D, O, G, l)$ obtient tous les chemins pour les autobus de leurs positions initiales à leur position finale. Dans cet exemple, tous les autobus doivent suivre leur voie d'évacuation prédéfinie, exactement comme il est souhaité lors de la définition de ces voies. Dans les plans, on assume que chaque action emploie une unité de temps. □

Example B.3. En revenant à notre exemple B.2, nous supposons qu'une partie de la voie d'évacuation 2 est bloquée parce que s'est bloquée $road(12, 17, 2)$, c'est à dire, nous ajoutons **initially** $blocked(12, 17)$ a $\pi(D, O, G, l)$. Comme il n'est pas possible que le bus $b2$ suive la voie d'évacuation prédéfinie, alors $\pi(D, O, G, l)$ est inconsistant et on ne peut pas définir de plans d 'évacuation. Pour restaurer la consistance et obtenir un plan alternatif, nous ajoutons au programme $\pi(D, O, G, l)$ la CR-règle:

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

Cette CR-règle doit être utilisée seulement s'il n'y a aucun moyen d'obtenir un plan quand une partie de la voie d'évacuation est bloquée. Nous pouvons alors réécrire le programme $\pi(D, O, G, l)$ comme suit:

```
% Initialement, le bus B est au point N de la voie R.
initially(position(B, N, R)).

% but: finalement le bus B est \`{a} un point final.
finally(end(B)).

% fluentes:
% la position du bus B est au point Q de la voie R.
fluent(position(B,Q,R)).

% le chemin du point P au point  Q de la voie R est bloqu\'e.
fluent(blocked (P,Q,R)).

%  le bus B \`{a} un point final.
fluent(end(B)):- shelter(B).

% Action NORMALE travel:
% action travel:
% le bus B  roule sur le chemin du point  P au point Q de la voie R,
% o\`{u}  R<>0, c'est \`{a} dire que les bus roulent seulement sur
% la voie d'\'evacuation.
action(travel(B,P,Q,R)).

% CR-r\`{e}gle o\`{u} R' peut \^{e}tre z\'ero ou pas,
```

```
r_2: action(travel(B,P,Q,R')) :- + bus(B), road(P,Q,R').

% R\`{e}gles de cause dynamiques:
% si le bus B  roule sur le chemin du point P au point Q
% de la voie R avec R<>0, alors B est dans la position Q
% de la voie R.
caused(position(B,Q,R),travel(B,P,Q,R)) .

% si le bus B  roule sur le chemin du point P au point Q
% de la voie R  alors B n'est pas dans la position P de la voie R.
caused(neg(position(B,P,R)),travel(B,P,Q,R)).

% R\`{e}gles de cause statiques:
% si le bus B  est dans la position P de la voie R tel que
% au point P il y a un refuge, alors B est dans une position finale.
caused(end(B), position(B,P,R)).

% Condicions d'application:
% le bus B ne peut pas rouler sur le chemin du point P au point Q de la voie R,
% si B n'est pas dans la position P de la voie R.
noaction_if(travel(B,P,Q,R),neg(position(B,P,R))).

% bus B ne peut pas rouler sur le chemin du point P au point Q de la voie R,
% si le chemin est bloqu\'e.
noaction_if(travel(B,P,Q,R),blocked(P,Q,R)).
```

Nous obtenons alors quatre plans d'évacuation alternatifs, c'est à dire quatre 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 codification $\pi(D, O, G, l)$ dans l'exemple B.3 obtient tous les chemins possibles des bus depuis leurs positions initiales jusqu'à leurs positions finales . Par exemple, le Plan 3 indique que le bus $b1$ et le bus $b3$ doivent suivre leurs voies d'évacuation prédéfinies alors que le bus $b2$ doit emprunter des chemins hors de la zone d'évacuation □

Example B.4. Nous considérons l'exemple B.2 mais maintenant, l'action travel permet de se déplacer sur des segments qui font partie ou non d'une voie d'évacuation, c'est à dire que l'action *travel(B,P,Q,R)* est maintenant la suivante:

```
% action travel:
% le bus B roule sur le chemin du point P au point Q de la voie R,
```

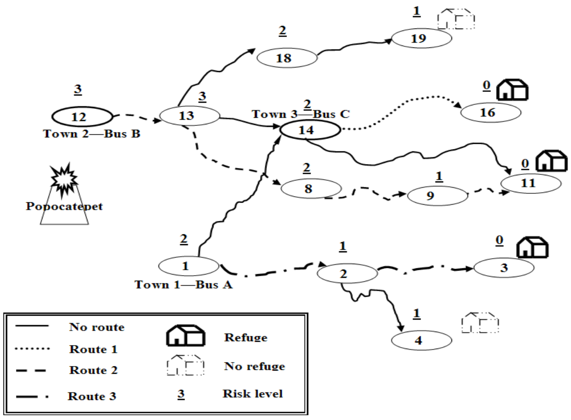

Figure B.2: Trois routes d'vacuation

```
% o\`{u} R est \'egal \`{a}  0, 1, 2 ou 3, c'est \`{a} dire
% que les bus peuvent prendre la voie d'\'evacuation ou non:

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

Si nous considérons dans cet exemple le graphique dirigé de la figure B.2, nous pourrions définir les désirs basiques suivants :

—*travelERass* pour exprimer qu'on préfère que les bus prennent la voie d'évacuation désignée par le gouvernement *jusqu'à* ce qu'ils arrivent au refuge.

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

—*travelER* pour exprimer qu'il est préférable que les bus prennent des chemins qui font partie de l'une des voies d'évacuation et il n'est pas important qu'ils prennent ou non la voie d'évacuation désignée, jusqu'à ce qu'ils arrivent à leurs refuges désignés.

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

—*travelBlSh* pour exprimer que les bus empruntent leurs voies d'évacuation désignée *jusqu'à ce qu'éventuellement* une partie de leur voie d'évacuation soit bloquée et alors ils se déplaceront hors de la zone d'évacuation *jusqu'à* ce qu'ils arrivent à leur refuge désigné.

$$
\begin{array}{l}
\mathit{travelBlSh} := \\
\mathbf{until}(\mathbf{occ}(travel(busB, 12, 13, 2)) \ \vee \mathbf{occ}(travel(busB, 13, 8, 2)) \vee \\
\qquad \mathbf{occ}(travel(busB, 8, 9, 2)) \vee \mathbf{occ}(travel(busB, 9, 11, 2)) \ , \\
\qquad \mathbf{until}(\ \mathbf{eventually}(\mathit{blocked}(12, 13, 2) \ \vee \mathit{blocked}(13, 8, 2)) \vee \mathit{blocked}(8, 9, 2) \vee \mathit{blocked}(9, 11, 2)), \\
\qquad\qquad travel(busB, 13, 18, 0)) \vee travel(busB, 18, 19, 0)) \vee travel(busB, 13, 14, 2)) \vee \\
\qquad\qquad travel(busB, 14, 11, 2)) \vee travel(busB, 1, 14, 2)) \vee \mathit{position}(busB, 11, 2) \ ) \ ) \\
\wedge
\end{array}
$$

$$\begin{aligned}
&\textbf{until}(\textbf{occc}(travel(busC,14,16,1))\ ,\\
&\qquad \textbf{until}(\ \textbf{eventually}(blocked(14,16,1)),\\
&\qquad\qquad travel(busC,13,18,0)) \vee travel(busC,18,19,0)) \vee travel(busC,13,14,2))\vee\\
&\qquad\qquad travel(busC,14,11,2)) \vee travel(busC,1,14,2)) \vee position(busC,16,1)\ )\ )\\
&\ \wedge\\
&\textbf{until}(\textbf{occ}(travel(busA,1,2,3))\ \vee \textbf{occ}(travel(busA,2,3,3))\ ,\\
&\qquad \textbf{until}(\ \textbf{eventually}(blocked(1,2,3)\ \vee blocked(2,3,3))),\\
&\qquad\qquad travel(busA,13,18,0)) \vee travel(busA,18,19,0)) \vee travel(busA,13,14,2))\vee\\
&\qquad\qquad travel(busA,14,11,2)) \vee travel(busA,1,14,2)) \vee position(busA,3,3)\ )\ )
\end{aligned}$$

De la même manière, nous pourrions exprimer d'autres désirs basiques. Une préférence atomique possible ψ indiquant l'ordre dans lequel l'ensemble des désirs basiques doit être satisfait est la suivante: $\psi = travelERass \triangleleft travelER \triangleleft travelBlSh$
La préférence atomique ψ dit que les plans qui satisfont $travelERass$ sont préférés, mais dans d'autres cas les plans qui satisfont $travelER$ sont préférés et dans d'autres cas encore les plans satisfaisant $travelBlSh$ sont préférés. □

Example B.5. Notons que le désir basique *travelER* de l'exemple B.4 explique une disjonction consistante des différentes options que les autobus ont pour voyager sur les axes qui font partie de l'une des voies d'évacuation prédéfinies. Pourtant, nous supposons que les voies d'évacuation ont un plus grand nombre d'axes, alors, pour exprimer *travelER* d'une manière similaire, nous aurions à spécifier une disjonction plus grande, définie à partir de tous les axes dans les nouvelles voies d'évacuation. Même si dans [43] on indique que les fluents et les actions dans $\mathcal{PP}$ avec variables sont une abréviation représentative de l'ensemble de toutes ses ground instances, l'idée étant que l'utilisation des fluents et des actions avec variables n'est pas suffisant pour exprimer ce type de problèmes. Pour cela, il y a plusieurs préférences qui ne peuvent pas s'exprimer dans $\mathcal{PP}$ d'une manière simple et naturelle. Pour avoir une représentation naturelle de cette classe de préférences, dont on s'inspire dans [22], nous définissons le langage $\mathcal{PP}^{pour}$ une extension du langage de $\mathcal{PP}$ où les connectifs propositionnels et temporels permettent que nous représentions de manière compacte les préférences qui ont une caractéristique particulière. Par exemple, une représentation naturelle et compacte de la préférence *travelER* de l'exemple 8.5 qui utilise *ou paramétrique* serait :

$$\begin{aligned}\mathbf{until}(&\bigvee\{occ(travel(B,I,F,R)) : bus(B), road(I,F,R), neq(R,0)\},\\ &\qquad\bigwedge\{position(B,Fi,R) : bus(B), shelter(Fi), route(R), neq(R,0)\})\end{aligned}$$