

Utilisation des graphes rtl

Christophe Lohr

15 septembre 2003

Table des matières

1	Les types de graphes produits par rtl	2
2	Sortie standard	2
3	Sortie pour Auto/Graph	3
4	Sortie pour Caesar/Aldebaran	5
A	Préparation à la projection	7
B	Projection avec Aldebaran	8
C	Manipulation des fichiers PostScript	8

```

specification medium_example[iu_d] : noexit

type natural is
  sorts nat
  opns
    + :nat,nat->nat
    - :nat,nat->nat
endtype

behaviour

  hide iu_s in

  stream_sender [iu_s] (30)
  |[iu_s]|
  medium [iu_s, iu_d] (14,20)

where
  process stream_sender [iu_s] (period:nat) : noexit :=
    iu_s{0} !1 !2;
    delay(period) stream_sender [iu_s] (period)
  endproc

  process medium [iu_s,iu_d] (dmin, dmax : nat) : noexit :=
    iu_s ?n1:nat ?n2:nat;
    delay(dmin,dmax) iu_d;
    medium [iu_s, iu_d] (dmin, dmax)
  endproc

endspec

```

FIG. 1 – medium.lot

1 Les types de graphes produits par rtl

À partir d'une spécification rt-lotos (par exemple un médium très simple Fig. 1), rtl peut produire un DTA ou bien un graphe d'accessibilité (ou graphe de régions).

2 Sortie standard

1. Génération du DTA :


```
rtl -TG1 -p1 -d2 medium.lot > medium.dta
```
2. Génération du graphe d'accessibilité :


```
rtl -TG2 -p1 -d2 medium.lot > medium.rg
```

 ou


```
rtl -TG3 -p1 -d2 medium.lot > medium.rg
```

rtl, avec les options -TGn -p1 -d2, affiche sur sa sortie standard les graphes selon sa propre syntaxe (Fig. 2).

Utilisation : Les outils `dta2daVinci` et `dta2dot` agissent comme des filtres et permettent de convertir les graphes au format `rtl` dans le format `daVinci`¹

¹<http://www.tzi.de/~davinci/>

<pre> State 0: 1 clocks State 1: 2 clocks State 2: 1 clocks Memory = 1184/1524 kbytes Execution time: 0:0s Speed in lotos machine: 4 states/s (4 states) Speed in graph generation: 3 states/s (3 states) 3 states, 3 arcs (0, i(iu_s<1,2>) K={0<=c1<=0} U={0<=c1<=0} C={1,2}, 1) (1, iu_d K={14<=c2} U={20<=c2} theta={(1,1)}, 2) (2, i(iu_s<1,2>) K={30<=c1<=30} U={30<=c1<=30} C={1,2}, 1) </pre>	<pre> State 0: 1 clocks State 1: 2 clocks State 2: 1 clocks Memory = 1204/1566 kbytes END step 1, 6 marked nodes, 10 nodes END OK Memory = 1212/1564 kbytes END in level 4, 21 marked nodes, 27 nodes 0-(0) URG c1=0 1-(14 14) 0<=c1<30 c2>=14 c2-c1>-16 1-(0 0) 0<=c1<30 0<=c2<14 -16<c2-c1<14 2-(30) URG c1=30 2-(14) 0<=c1<30 (0-(0), i(iu_s<1,2>), 1-(0 0)) (1-(14 14), iu_d, 2-(14)) (1-(0 0), t, 1-(14 14)) (2-(30), i(iu_s<1,2>), 1-(0 0)) (2-(14), t, 2-(30)) st0=1 st1=2 st2=2 5 classes 3 reachable DTA states 1cl(2st) 2cl(1st) 1s (0:1min) Memory = 1232/1572 kbytes </pre>
medium.dta	medium.rg

FIG. 2 –

ou dans le format *dot*². *daVinci* et *dotty* sont deux logiciels qui permettent de visualiser, d'éditer et de faire du placement automatique des graphes (Fig. 3, Fig. 4).

```

rtl -TG1 -p1 -d2 medium.lot | dta2daVinci > medium.dta.daVinci
rtl -TG2 -p1 -d2 medium.lot | dta2daVinci > medium.rg.daVinci

```

ou

```

cat medium.dta | dta2daVinci > medium.dta.daVinci
cat medium.rg | dta2daVinci > medium.rg.daVinci

```

```

rtl -TG1 -p1 -d2 medium.lot | dta2dot > medium.dta.dot
rtl -TG2 -p1 -d2 medium.lot | dta2dot > medium.rg.dot

```

ou

```

cat medium.dta | dta2dot > medium.dta.dot
cat medium.rg | dta2dot > medium.rg.dot

```

3 Sortie pour Auto/Graph

1. Génération du DTA :

```

rtl -TG1 -ATG -p1 medium.lot

```

produit un fichier `medium.lot.tg0.fc2` qui décrit le DTA au format *FC2*³

2. Génération du graphe d'accessibilité :

```

rtl -TG2 -ATG -p1 medium.lot

```

ou

²<http://www.research.att.com/sw/tools/graphviz/>

³<http://www-sop.inria.fr/meije/verification/index.html>

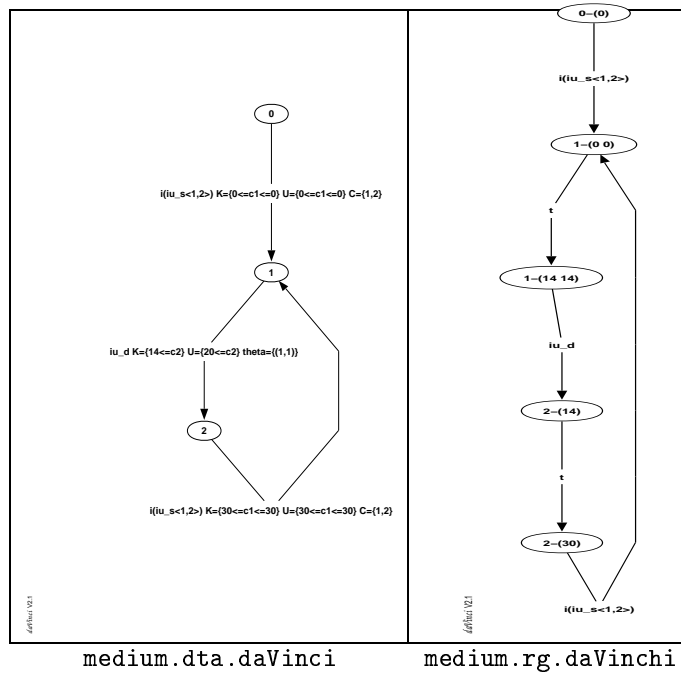


FIG. 3 -

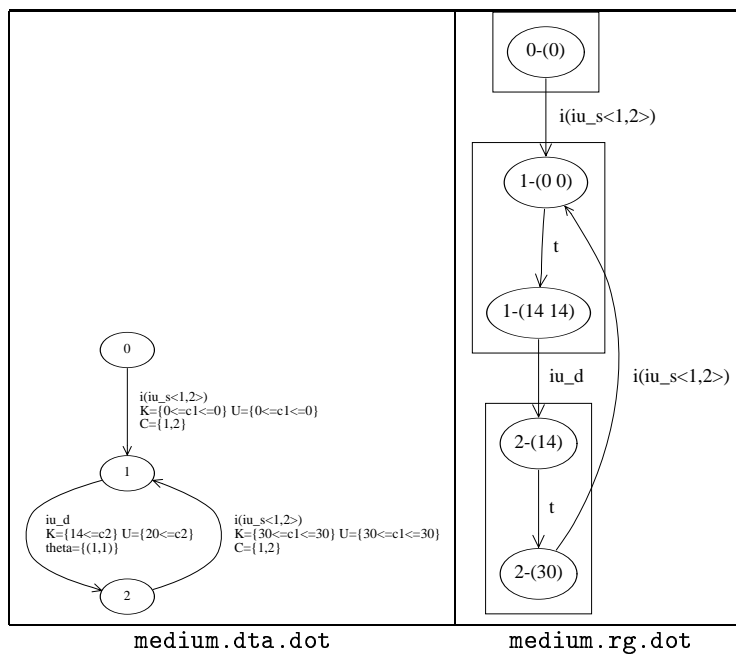


FIG. 4 -

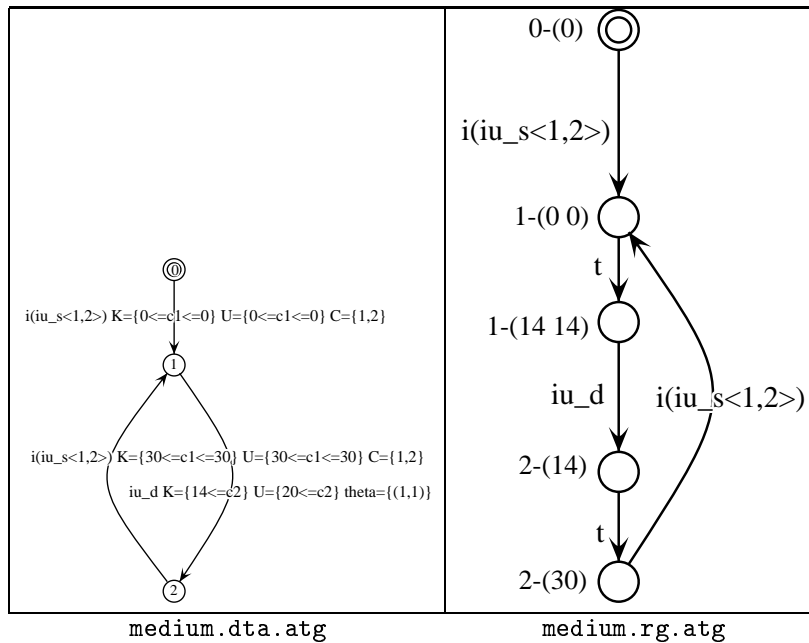


FIG. 5 –

```
rtl -TG3 -ATG -p1 medium.lot
produit en plus du fichier de DTA medium.lot.tg0.fc2 un fichier me-
dium.lot.rg0.fc2 qui décrit le graphe d'accessibilité au format FC2
```

Le logiciel `atg` permet d'éditer ce fichier et de placer le graphe à la main (Fig. 5). Il produit alors un fichier `.atg`.

4 Sortie pour Caesar/Aldebaran

1. Génération du DTA :

```
rtl -TG1 -ATG -AUT -p1 medium.lot
produit un fichier medium.lot.tg0.aut (en plus du fichier medium.lot.tg0.fc2)
qui décrit le DTA au format ALDEBARAN4
```

2. Génération du graphe d'accessibilité :

```
rtl -TG2 -ATG -AUT -p1 medium.lot
```

ou

```
rtl -TG3 -ATG -AUT -p1 medium.lot
produit un fichier medium.lot.rg0.aut (en plus des fichiers medium.lot.tg0.fc2
medium.lot.tg0.aut et medium.lot.rg0.fc2) qui décrit le graphe d'ac-
cessibilité au format ALDEBARAN
```

Il peut alors être utile de convertir ce fichier au format `.bcg` par la com-
mande : `bcg_io medium.lot.tg0.aut medium.dta.bcg`

L'outil `bcg_edit` permet de placer automatiquement les états et d'éditer le
graphe (Fig. 6).

⁴<http://www.inrialpes.fr/vasy/cadp.html>

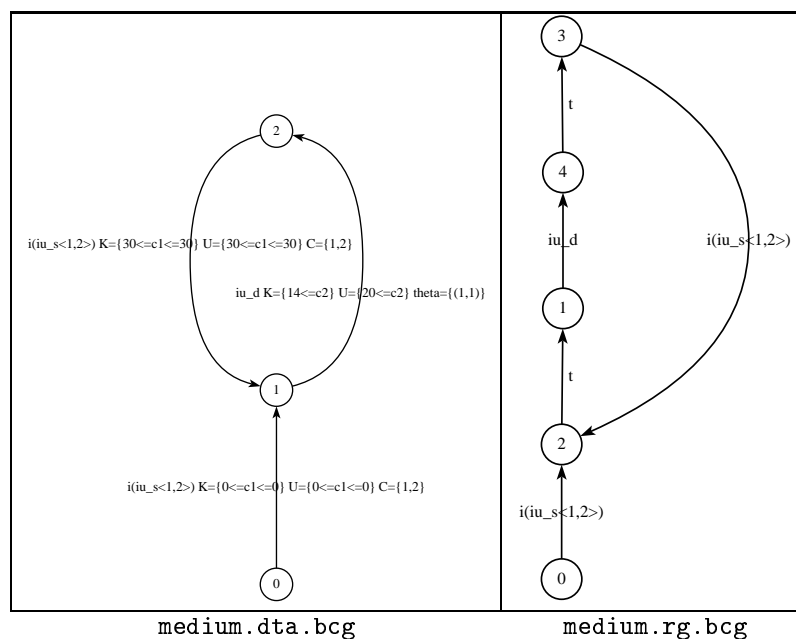


FIG. 6 –

L'outil `aldebaran` permet de réaliser un certain nombre de manipulations sur un graphe et notamment de faire des projections en ne considérant que les arcs que l'on souhaite observer :

```
aldebaran -imin medium.proj.aut > medium.imin.aut
ou5
aldebaran -omin medium.proj.aut > medium.omin.aut
```

Le fichier `medium.proj.aut` est obtenu à partir du fichier `.aut` du graphe initial dans lequel on a renommé les arcs que l'on voulait internes en “ i ”⁶.

Remarques : Avec `aldebaran` on trouve d'autres outils :

`bcg_min` implémente des algorithmes de minimisation. La minimisation par équivalence de branchement est assez semblable à la projection observationnelle.

`bcg_label` permet de renommer les labels des arcs.

⁵L'annexe B montre la différence entre `-imin` et `-omin`

⁶Le script Perl proposé en annexe A effectue ce renommage

A Préparation à la projection

```
#!/usr/local/bin/perl

$outfile = $ARGV[0];
$obsfile = $ARGV[1];

if ( open(OBS,$obsfile) ) {
  while (<OBS>) {
    while (/[^\s]*\s*(.*)?$/;
    push(@obsactions,$1);
    push(@map,$2);
  }
}

if ( open(AUT,$outfile) ) {
  while (<AUT>) {
    if (/^\((\d+),"(w-)?i",(\d+)\)$/ )
    { print "$1,$3\n" }
    elsif (/^\((\d+),"t",(\d+)\)$/ )
    { &compare ($1,"t",$2,$_); }
    elsif (/^\((\d+),"w-)?i\([[\w\d]*\])",(\d+)\)$/ )
    { &compare ($1,$3,$4,$_); }
    elsif (/^\((\d+),"w-)*(\w+)",(\d+)\)$/ )
    { &compare ($1,$3,$4,$_); }
    else{print}
  }
}

sub compare {
  local($arg1,$arg3,$arg5,$line)=0.;
  local($found)= 0;
  local($i)=0;
  foreach $obsact (@obsactions) {
    if ($arg3 eq $obsact)
    { $found = 1 }
    elsif ($line eq $obsact)
    { $found = 1 }
    if($found == 0)
    { $i++ }
  }
  if ($found == 1) {
    $action=@map[$i];
    print "($arg1,\"$action\",$arg5)\n"
  }
  else
  { print "($arg1,$i,$arg5)\n" }
}
}
```

pre-proj.perl

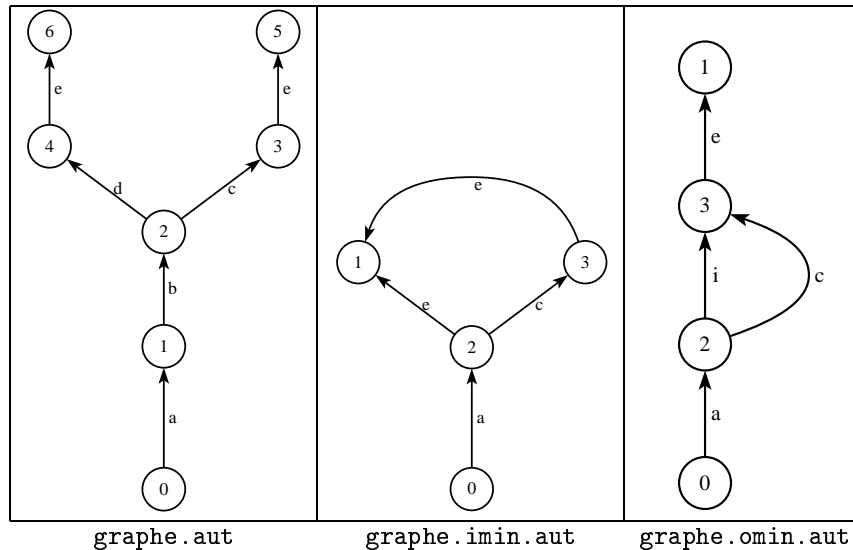
Utilisation :

pre-proj.perl graphe.aut graphe.obs > graphe.proj.aut

<pre>ds(0,6,7) (0,"a",1) (1,"b",2) (2,"c",3) (2,"d",4) (3,"e",5) (4,"e",6)</pre>	<pre>a a c c e e</pre>	<pre>ds(0,6,7) (0,"a",1) (1,i,2) (2,"c",3) (2,i,4) (3,"e",5) (4,"e",6)</pre>
medium.proj.aut	medium.obs	medium.lot.rg0.aut

Chaque ligne de medium.obs indique quelle action observable rechercher et par quel label la remplacer.

B Projection avec Aldebaran



C Manipulation des fichiers PostScript

- les fichiers PostScript produit par `atg` doivent être passé par l'outil `psfp` (fourni avec les outils `FC2Tools`)
- et éventuellement convertis en `.eps` (PostScript encapsulé) par la commande `ps2epsi` (fournie avec `ghostscript`)
- dans les fichiers PostScript produit par `bcg_edit`, les labels des arcs sont *sur* les flèches, ce qui les rend parfois peu lisibles :
 - pour les décaler les labels vers la droite, repérez les chaînes de caractères correspondantes en fin de fichier et rajoutez des espaces avant
 - pour les décaler verticalement, repérez en début de fichier la procédure `/show_text { ... }`, recopiez-la sous le nom `show_text2` en remplaçant la ligne `■text_y_shift rmoveto■` par `■-10 rmoveto■` (ou n'importe quelle valeur qui vous convient), puis, en fin des procédures `/edge { ... }` et `/spline { ... }` remplacez l'instruction `show_text` par la vôtre
- pour passer un PostScript de "landscape" à "portrait" :


```
psstops "0r@0.7(0,29.7cm)" fichier_in.ps fichier_out.ps
```
- importer les fichier PostScript produits par `atg` dans Word pose parfois des problèmes (les noeuds disparaissent) : repérez dans le fichier PostScript encapsulé les lignes contenant les commandes `vertex` et `initial` et recopiez-les juste avant la commande `showpage` en fin de fichier
- un paquetage `LATEX` intéressant pour dessiner des graphes : <http://www.informatik.uni-bremen.de/~drewe/graphs.html>