Appendix A: Menu Item Shortcuts

The following shortcuts to menu items are supported in GDCT.

- ALT+F: File Menu
  - ALT+N: New Category
  - ALT+O: Open Category
  - ALT+D: Download Category
  - ALT+F: Download Functor
  - ALT+X: Exit Program
- ALT+S: Settings Menu
  - ALT+S: Server Settings
  - ALT+E: Endomorphism Limit
- ALT+W: Window Menu
  - ALT+C: Close All Windows
- ALT+H: Help Menu
  - ALT+H: Help Topics Index
  - ALT+B: Bug Report
  - ALT+G: About GDCT
  - ALT+A: About Authors

Appendix B: Miscellaneous Information

Version Information

The Graphical Database for Category Theory is a research project at Mount Allison University that now includes over 70 Java source files and over 30000 lines of code.

Version 1.0: Released March 20, 2000

This is the preliminary version of Graphical Database for Category Theory

Version 1.1: Released Summer 2000

This version of GDCT contained fixes and enhancements described in its user guide. See: http://mathcs.mta.ca/research/rosebrugh/gdct/v1release.htm

Version 2.0: Released May 2002

This version of GDCT contained fixes and enhancements described in its user guide. See: http://mathcs.mta.ca/research/rosebrugh/gdct/gdct2.0/index.html

Version 3.0: Released February 2006/Alpha

This version of GDCT contained the following fixes and enhancements:

- Moved to a desktop-style interface, where multiple categories and functors can be opened in their own respective windows

- Completely removed Borland from the software.
- Converted most graphic components from AWT to Swing (lightweight)
- Divided each tool into two separate classes: an interface class and an algorithm class
- Implemented pullback, pushout, create product, create sum, and partial order category tools.
- Converted most lingering C based code to object oriented.

Appendix C: File Format Examples

An example .cat file:

```
#a commutative triangle
category
triangle.cat
objects
A, B, C.

arrows
ab:A->B, ac:A->C, bc:B->C.
relations
bc*ab = ac.
```

An example .cgl file:

```
# A parallel pair plus equalizing arrow.
category
equalizer.cgl
objects
A, B, C.

arrows
f:B->A, g:B->A, h:C->B.
relations
g*h = f*h.
gml
graph [
   directed 1
   node [
      id 0
      label "A"
      graphics [
         Image [
            Type ""
            Location ""
         ]
         center [
            x 125.0
            y -1.0
            z 0.0
         ]
         width 15.94
         height 28.72
         depth 20.0
      ]
      vgj [
```

```
      color "black"
      labelPosition "in"
      shape "Oval"
   ]
]
node [
   id 1
   label "B"
   graphics [
      Image [
         Type ""
         Location ""
      ]
      center [
         x 2.0
         y -1.0
         z 0.0
      ]
      width 17.36
      height 28.72
      depth 20.0
   ]
   vgj [
      color "black"
      labelPosition "in"
      shape "Oval"
   ]
]
node [
   id 2
   label "C"
   graphics [
      Image [
         Type ""
         Location ""
      ]
      center [
         x -101.0
         y -1.0
         z 0.0
      ]
      width 18.78
      height 28.72
      depth 20.0
   ]
   vgj [
      color "black"
      labelPosition "in"
      shape "Oval"
   ]
]
node [
   id 3
   label "relations\nf*h = g*h"
   graphics [
      Image [
         Type ""
         Location ""
      ]
      center [
         x 0.0
```

```
            y -150.0
            z 0.0
         ]
         width 74.16
         height 51.44
         depth 20.0
      ]
      vgj [
         color "black"
         labelPosition "in"
         shape "Oval"
      ]
   ]
   edge [
      linestyle "solid"
      label "h"
      source 2
      target 1
   ]
   edge [
      linestyle "solid"
      label "g,f"
      source 1
      target 0
   ]
]
```

An example .fun file:

```
category
csquare
objects
A, B, C, D.
arrows
f:A->B, g:A->C, h:B->D, k:C->D.
relations
k*g = h*f.
category
coequalize
objects
A, B, C.
arrows
f:A->B, g:A->B, h:B->C.
relations
h*g = h*f.
functor
 cscoeq
objects
A |--> A, B |--> B, C |--> B, D |--> C.
arrows
f |--> f, g |--> g, h |--> h, k |--> h.
```

An example .fgl file:

```
#skeleton of sets with 3 or less elements
category
parpair
objects
A, B.
arrows
```

```
f:A->B, g:A->B.
relations
.
gml
graph [
   directed 1
   node [
      id 0
      label "A"
      graphics [
         Image [
            Type ""
            Location ""
         ]
         center [
            x -144.0
            y 11.0
            z 0.0
         ]
         width 15.94
         height 27.299999999999997
         depth 20.0
      ]
      vgj [
         labelPosition "in"
         shape "Oval"
      ]
   ]
   node [
      id 1
      label "B"
      graphics [
         Image [
            Type ""
            Location ""
         ]
         center [
            x 121.0
            y 13.0
            z 0.0
         ]
         width 17.36
         height 27.299999999999997
         depth 20.0
      ]
      vgj [
         labelPosition "in"
         shape "Oval"
      ]
   ]
   edge [
      linestyle "solid"
      label "g,f"
      source 0
      target 1
   ]
]

#skeleton of sets with 3 or less elements
category
set3
```

```
objects
0, 1, 2, 3.
arrows
d00:0->1, d10:1->2, d11:1->2, e00:2->1,
d20:2->3, d21:2->3, d22:2->3, e10:3->2,
e11:3->2, s00:2->2, s10:3->3, s11:3->3.
relations
e00*d10 = 1, e00*d11 = 1, d11*d00 = d10*d00, d21*d10 = d20*d10,
d22*d10 = d20*d11, d22*d11 = d21*d11, e00*e11 = e00*e10, e11*d20 =
d10*e00,
e10*d20 = 1, e11*d21 = 1, e10*d21 = 1, e11*d22 = 1,
e10*d22 = d11*e00, s00*s00 = 1, s00*d10 = d11, s10*d22 = d22*s00,
s10*d20 = d21, s11*d21 = d22, s11*d20 = d20*s00, e11*s10*s11 = s00*e10,
e10*s10 = e10, e11*s11 = e11, e00*s00 = e00, s11*s11 = 1,
s10*s10 = 1, s10*d21 = d20, s11*d22 = d21, s00*d11 = d10,
e10*s11*s10 = s00*e11, s11*s10*s11 = s10*s11*s10, e00*e10*s11 =
e00*e10, s00*e10*s11 = e11*s10,
s00*e11*s10 = e10*s11, d22*d10*d00 = d20*d10*d00, s11*d20*d10 =
d22*d10, s10*d22*d10 = d21*d11,
e11*s10*d22 = s00, s11*s10*d22 = d21*s00, e10*s11*d20 = s00,
s10*s11*d20 = d21*s00,
s10*d22*e10*s11 = d22*e11*s10, s11*d20*e10*s11 = d20*e11*s10,
s10*d22*e11*s10 = d22*e10*s11, s11*d20*e11*s10 = d20*e10*s11.
gml
graph [
   directed 1
   node [
      id 0
      label "0"
      graphics [
         Image [
            Type ""
            Location ""
         ]
         center [
            x -199.0
            y 0.0
            z 0.0
         ]
         width 15.94
         height 27.299999999999997
         depth 20.0
      ]
      vgj [
         labelPosition "in"
         shape "Oval"
      ]
   ]
   node [
      id 1
      label "1"
      graphics [
         Image [
            Type ""
            Location ""
         ]
         center [
            x -100.0
            y 0.0
            z 0.0
         ]
```

```
        width 15.94
        height 27.299999999999997
        depth 20.0
      ]
      vgj [
        labelPosition "in"
        shape "Oval"
      ]
    ]
    node [
      id 2
      label "2"
      graphics [
        Image [
          Type ""
          Location ""
        ]
        center [
          x 50.0
          y 0.0
          z 0.0
        ]
        width 15.94
        height 27.299999999999997
        depth 20.0
      ]
      vgj [
        labelPosition "in"
        shape "Oval"
      ]
    ]
    node [
      id 3
      label "3"
      graphics [
        Image [
          Type ""
          Location ""
        ]
        center [
          x 220.0
          y 0.0
          z 0.0
        ]
        width 15.94
        height 27.299999999999997
        depth 20.0
      ]
      vgj [
        labelPosition "in"
        shape "Oval"
      ]
    ]
    node [
      id 4
      label "relations\ne00*d10 = 1,e00*d11 = 1,d11*d00 =
d10*d00,d21*d10 = d20*d10,d22*d10 = d20*d11,\nd22*d11 = d21*d11,e00*e11
= e00*e10,e11*d20 = d10*e00,e10*d20 = 1,e11*d21 = 1,\ne10*d21 =
1,e11*d22 = 1,e10*d22 = d11*e00,s00*s00 = 1,s00*d10 = d11,\ns10*d22 =
d22*s00,s10*d20 = d21,s11*d21 = d22,s11*d20 = d20*s00,e11*s10*s11 =
s00*e10,\ne10*s10 = e10,e11*s11 = e11,e00*s00 = e00,s11*s11 = 1,s10*s10
```

```
= 1,\ns10*d21 = d20,s11*d22 = d21,s00*d11 = d10,e10*s11*s10 =
s00*e11,s11*s10*s11 = s10*s11*s10,\ne00*e10*s11 = e00*e10,s00*e10*s11 =
e11*s10,s00*e11*s10 = e10*s11,d22*d10*d00 = d20*d10*d00,s11*d20*d10 =
d22*d10,\ns10*d22*d10 = d21*d11,e11*s10*d22 = s00,s11*s10*d22 =
d21*s00,e10*s11*d20 = s00,s10*s11*d20 = d21*s00,\ns10*d22*e10*s11 =
d22*e11*s10,s11*d20*e10*s11 = d20*e11*s10,s10*d22*e11*s10 =
d22*e10*s11,s11*d20*e11*s10 = d20*e10*s11"
      graphics [
         Image [
            Type ""
            Location ""
         ]
         center [
            x 0.0
            y -133.0
            z 0.0
         ]
         width 1069.58
         height 219.0
         depth 20.0
      ]
      vgj [
         labelPosition "in"
         shape "Oval"
      ]
   ]
   edge [
      linestyle "solid"
      label "d22,d20,d21"
      source 2
      target 3
   ]
   edge [
      linestyle "solid"
      label "s11,s10"
      source 3
      target 3
   ]
   edge [
      linestyle "solid"
      label "d11,d10"
      source 1
      target 2
   ]
   edge [
      linestyle "solid"
      label "e11,e10"
      source 3
      target 2
   ]
   edge [
      linestyle "solid"
      label "s00"
      source 2
      target 2
   ]
   edge [
      linestyle "solid"
      label "d00"
      source 0
      target 1
```

```
   ]
   edge [
      linestyle "solid"
      label "e00"
      source 2
      target 1
   ]
]

functor
 pp2s3
objects
A |--> 1, B |--> 2.
arrows
f |--> d11, g |--> d10.
```