Groove - quick reference

Terminology

Graph: A set of nodes and edges. In Groove, graphs have directed, labeled edges. Nodes can have a (sub)type, flags and attributes. Attributes can be integers, reals, booleans or strings.

Graph grammar: In Groove, this is a collection of graph transformation rules, host graphs, type graphs and control programs; these collectively form a Groove project.
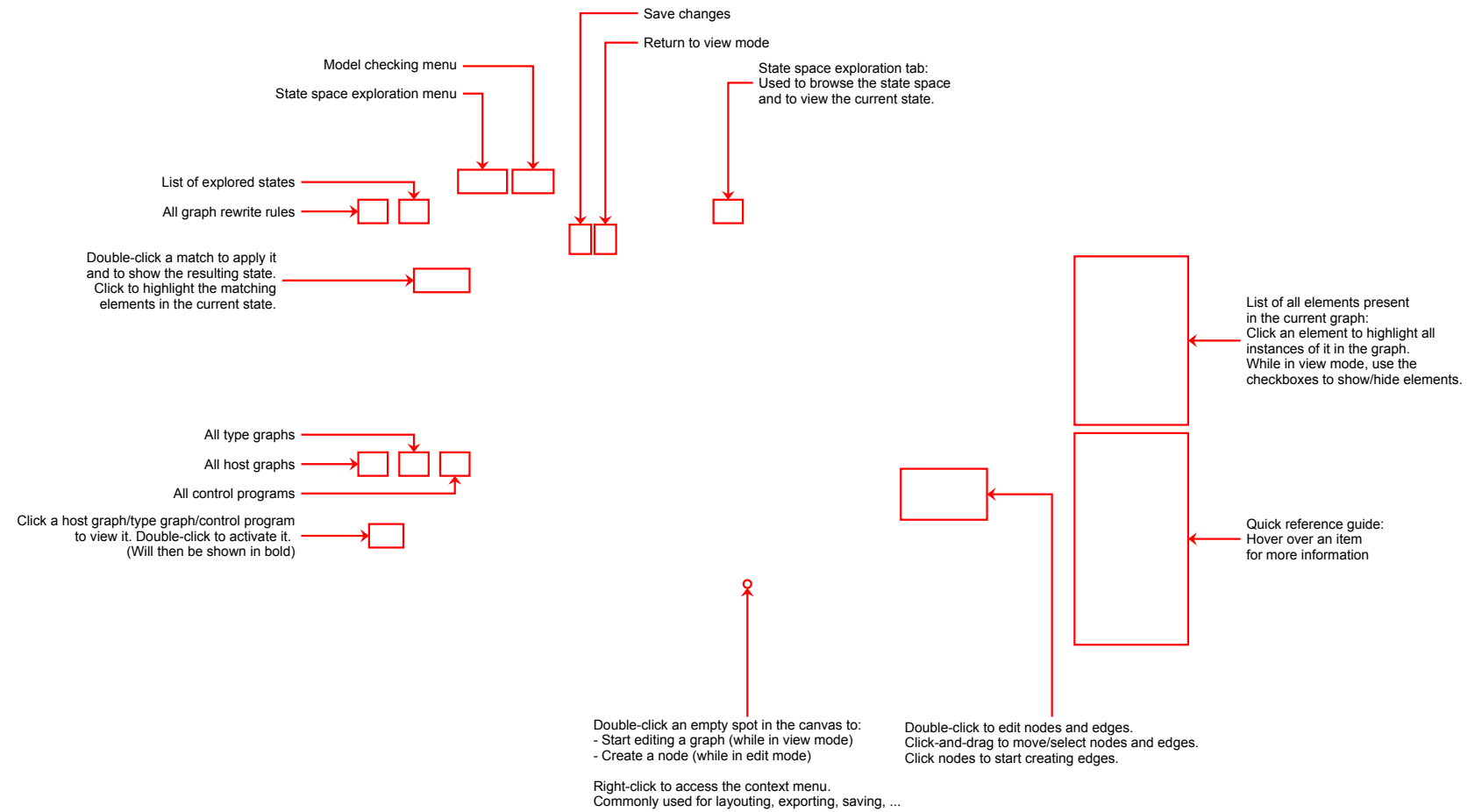
Host graph: The graph being transformed by applications of graph transformation rules.

Graph transformation rule: A graph that specifies how the host graph should be transformed: A graph transformation rule first looks for a particular subgraph within the host graph. If this subgraph is found, the rule matches; the rule is then applied by adding/removing elements within this subgraph.

Type graph: Specifies the allowed structure of your graphs, as well as the node type hierarchy, similar to a class diagram.
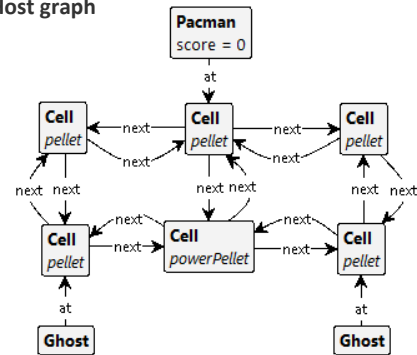
Control program: A program that schedules/governs your graph transformation rules. If no such program is active, Groove's default behaviour is to try to apply rules non-deterministically (taking into account rule priorities).

User interface

Save changes

Return to view mode

Model checking menu

State space exploration menu

State space exploration tab:
Used to browse the state space
and to view the current state.

List of explored states

All graph rewrite rules

Double-click a match to apply it
and to show the resulting state.
Click to highlight the matching
elements in the current state.

List of all elements present
in the current graph:
Click an element to highlight all
instances of it in the graph.
While in view mode, use the
checkboxes to show/hide elements.

All type graphs

All host graphs

All control programs

Click a host graph/type graph/control program
to view it. Double-click to activate it.
(Will then be shown in bold)

Quick reference guide:
Hover over an item
for more information

Double-click an empty spot in the canvas to:
- Start editing a graph (while in view mode)
- Create a node (while in edit mode)

Right-click to access the context menu.
Commonly used for layouting, exporting, saving, ...

Double-click to edit nodes and edges.
Click-and-drag to move/select nodes and edges.
Click nodes to start creating edges.

Sample graph grammar

Host graph

Type graph

Control program

```
1  while (gameInProgress) {
2      ghostMove | pacmanMove;
3      try {eatPellet | eatPowerPellet;}
4      try {ghostDies | pacmanDies;}
5  }
```
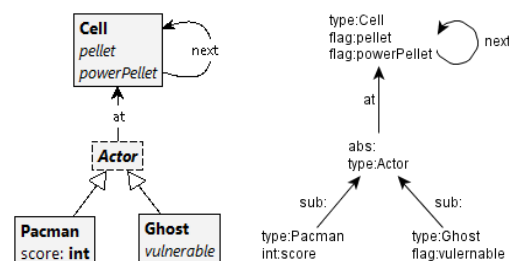
Graph transformation rules

These rules form part of a simple Pacman graph grammar.
The rules are shown both in view and edit mode.

Green elements are created if the rule matches.
Blue elements are removed if the rule matches.
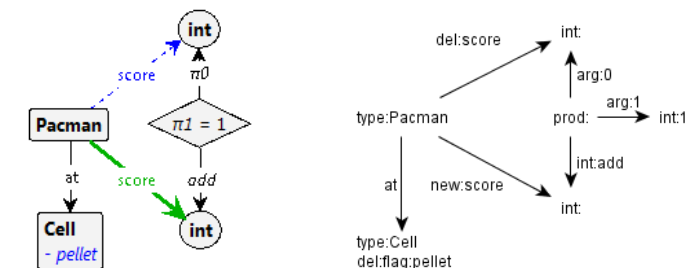Red elements may not exist in the host graph.

pacmanMove - Pacman will move to a neighbouring cell
if there is no Ghost on it, unless it is vulnerable.
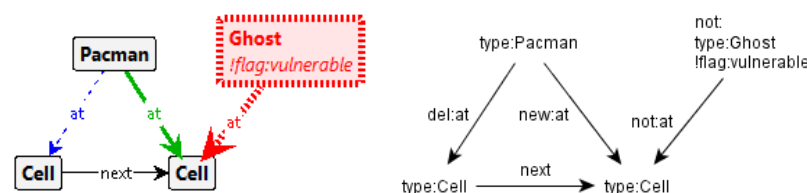
eatPellet - If Pacman is on a cell with a pellet,
he will eat it and his score increases by 1.

eatPowerPellet - If Pacman is on a cell with a power pellet,
he will eat it and all ghosts become vulnerable.

Creator: Tim Molderez, University of Antwerp