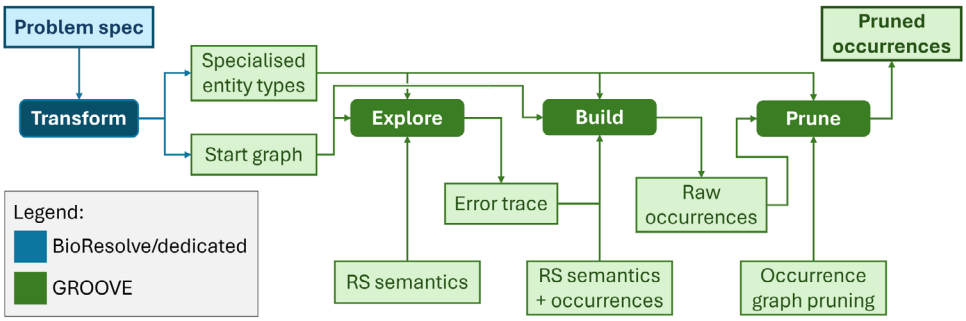

Fig. 5 Reaction system exploration and analysis using GROOVE

into GROOVE syntax. This is achieved by running the `main_do(rs2gts)` directive of BioResolve, which produces two artefacts: firstly, an additional type graph, complementary to the one shown in Fig. 3, which specifies one subtype of **Entity** for each of the entities in the problem at hand (essentially for performance reasons: relying on dedicated types speeds up the matching step of GROOVE); and secondly (more importantly) a start graph in which the entire BioResolve system is encoded as suggested by Fig. 3. For the example system, the additional types as well as two self-explanatory fragments of the start graph are shown in Fig. 6.

We claim that this transformation is semantics-preserving; Appendix A gives a sketch of the argument. A fully formal statement and proof of semantic correspondence, however, is outside of the scope of this paper.

Explore. The dynamics of Reaction Systems is encoded as a combination of two rules, context and react, which are scheduled to fire in alternation. The rule context encodes the simultaneous firing of all context processes (nondeterministically selecting an enabled **Step** from every **State** with a **Token**), whereas react encodes the (deterministic) simultaneous firing of all enabled **Reaction**s, while simultaneously erasing all **Entity**s that were not just produced. The production or erasure of an **Entity** is encoded through the creation or deletion of a *present* flag on a (persistent) **Entity** node, *not* by the creation or deletion of the node itself. In addition, to keep track of which nondeterministic choices were actually taken, the context rule marks the **Step**s that were selected with a *fired* flag, which is subsequently erased by the react rule.

Figure 7 shows the first (and most intricate) of these rules, viz. the one for the context firing. This is a quantified rule, which can be read as follows: For all **State**s with a **Token**, there is a next **Step** such that for all inhibitors there is no *present* flag whereas for all reactants there is a *present* flag; moreover, when the rule is applied, all products of the selected **Step**s receive a *present* flag, the **Step**s themselves receive a *fired* flag, and all **Token**s move to the successor **State**s. Colour coding is used in the visual representation to distinguish the quantifier nodes ∀ and ∃ (both in purple), as well as the mandatory absence (red), deletion (blue) and creation (green) of edges and flags.[3]

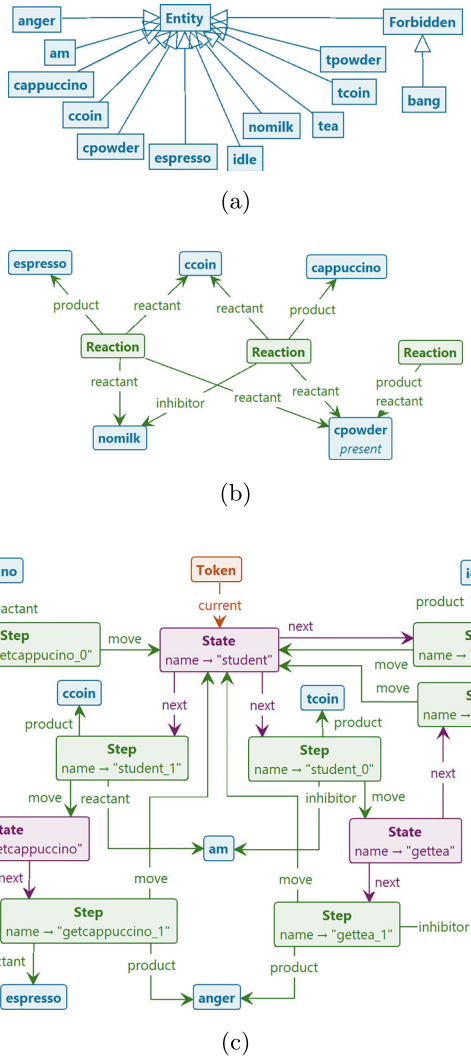

Fig. 6 Graph representation of running example. **a** Specialised entity types. **b** Start graph fragment: Three reactions from VM. **c** Start graph fragment: The Student context process

[3] This colour coding is GROOVE-specific and entirely separate from the problem-specific colouring of the graph nodes in Figs. 3 and 6; in