VisualizeSLE

A Visual Editor for Separation Logic Entailments

Aquinas Hobor, Soe Lin Myat, and Bimlesh Wadhwa

School of Computing
National University of Singapore

Background

Separation logic has seen widespread use in program verification, both for hand proofs [Rey02] and in automated tools [BCO06]. Verifying programs often requires many applications of the rule of consequence, but proving entailments by hand can be tricky. Automated tools work well for simple patterns, but often break down quickly when the formulae become complex (e.g., by using). Mechanical checkers have some advantages over paper proofs, but can be extremely painful to use.

Proposal

We propose VisualizeSLE, a visual proof editor for separation logic entailments. **VSLE combines the advantage of machine checking while enabling a more visual and intuitive "whiteboard-style proof"**. A key goal of our tool is to shift the cognitive load for user by using graphical structures and to support the user by encouraging organization and management of the intermediate proof steps (e.g., save, retrieval, replay).

Coginitive Load

A human's working memory is limited in terms of the amount of information it can hold and the operations it can perform on that information [Swe88]. Working memory load can be divided into **intrinsic load** , which is due to the inherent complexity of the problem; **germane load** , which in our case relates to the size of the separation logic proof we are trying to construct; and **extraneous load** , which is the organizational and representational overhead of the tools being used [Kir02]. Our tool attempts to reduce germane load through appropriate visual metaphors [CG12] and to reduce extraneous load [Kir02] through effective interface design.We believe that these advances will free up cognitive resource to process the intrinsic and remaining germane load.

References

[BCO06] Josh Berdine, Cristino Calcagno, and Peter O'Hearn. Smallfoot: Modular automatic assertion checking with separation logic. In FMCO 2005, 2006.

[CG12] J. Cheon and Michael M. Grant. The effects of metaphorical interface on germane cognitive load in web-based instruction. In Educational Technology Research and Development, volume 60, pages 399-420, 2012.

[Kir02] Paul A. Kirschner. Cognitive load theory: implications of cognitive load theory on the design of learning. In Learning and Instruction, volume 12, pages 1-10, 2002.

[Rey02] John Reynolds. Separation logic: A logic for shared mutable data structures. In LICS 2002: IEEE Symposium on Logic in Computer Science, pages 55-74, July 2002.

[Swe88] J Sweller. Cognitive load during problem solving: Effects on learning. In Cognitive Science, volume 12, pages 257-285, 1988.

Interface

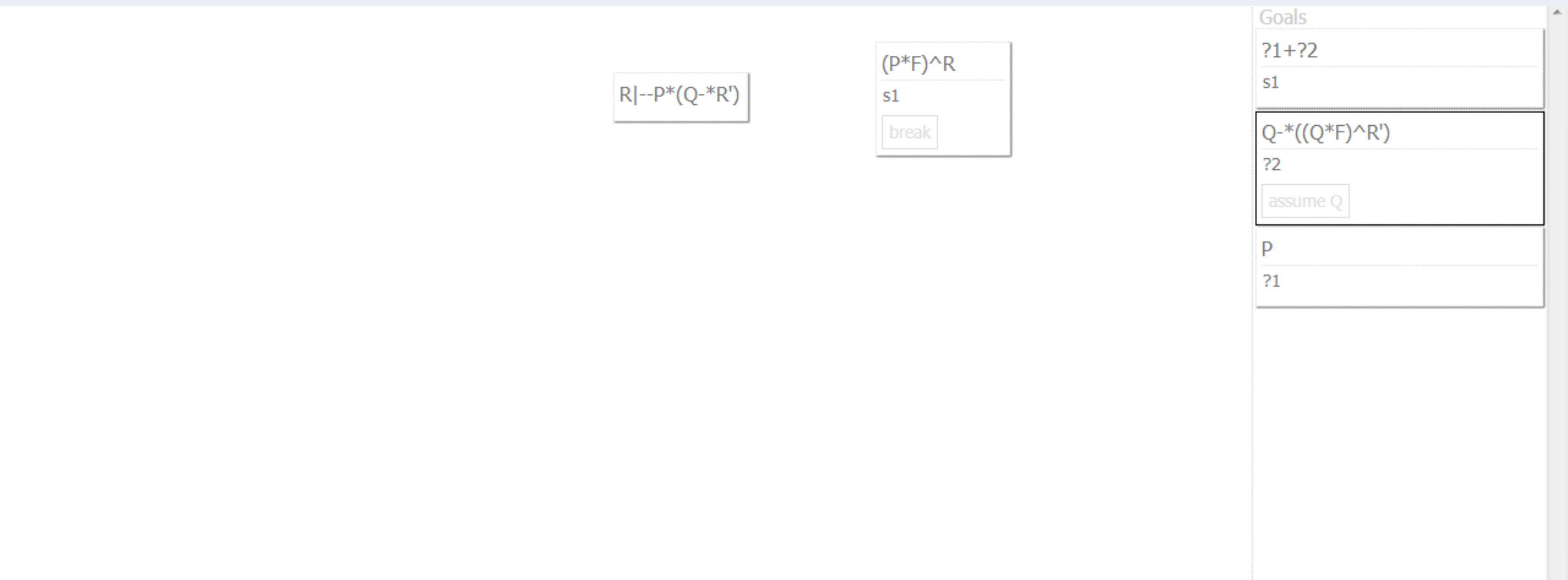
The VSLE interface is divided into two parts

Left
The workspace, for the current objects being manipulated.

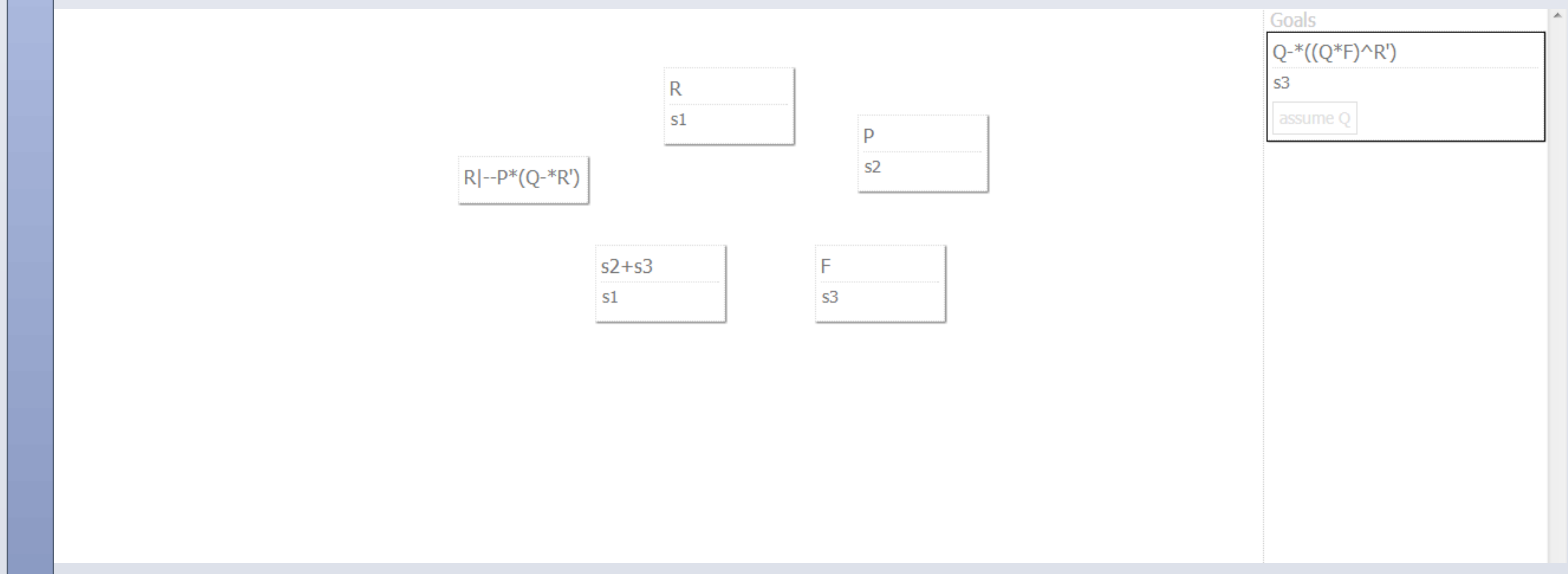
Right
The Goals

Metaphor : **Stickynote** to represent each Goal & assumption

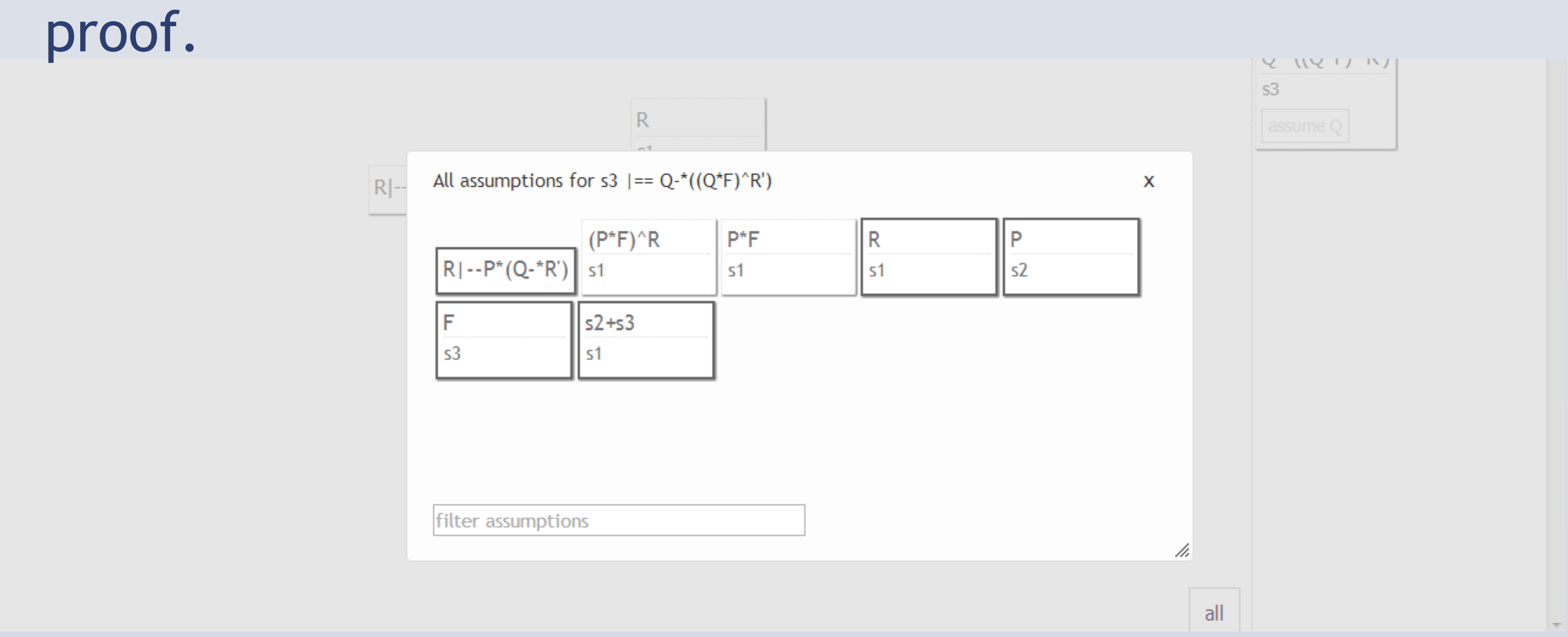
User Control **"all"** : Recently used assumptions are collapsed and can be accessed using "all" user control.



Above Screen shot shows a scenario in which we have some goals and assumptions but have not yet dragged anything into the leftmost workspace for immediate attention.



Above Screen shot shows an intermediate state in the proof.


All assumptions for s3 |== Q-*((Q*F)^R')

Above screen shot shows the "log", i.e., the current proof state.

Future Goal

Our eventual goal is to output these proofs into a nice human-readable format (via ASCII and LaTeX) as well as a machine-readable (and hence checkable) format (Coq scripts).

Contact

Aquinas Hobor: aquinashobor@live.com; http://www.comp.nus.edu.sg/~hobor/vsl