

## Shutherland-Hodgman [1]

**Remark.** Originally, this protocol has one choice where the destination is not the same across all branches (when choosing , or ). This may or may not be allowed depending on the specific formal variant of MPST you choose. In this benchmark, we reordered them so that the order is always  $P \rightarrow R \{ \ell. P \rightarrow C \{ \ell. G \} \}$ .

$$P \rightarrow R \left\{ \begin{array}{l} \textcolor{red}{Plane}(\langle \textcolor{brown}{point}, \textcolor{brown}{point}, \textcolor{brown}{point}, \textcolor{brown}{point} \rangle). \mu \textcolor{blue}{t}. P \rightarrow R \left\{ \begin{array}{l} \textcolor{red}{IsAbove}(\textcolor{brown}{point}). R \rightarrow P \left\{ \begin{array}{l} \textcolor{red}{Res}(\textcolor{brown}{bool}). P \rightarrow R \left\{ \begin{array}{l} \textcolor{red}{IsAbove}(\textcolor{brown}{point}). R \rightarrow P \left\{ \begin{array}{l} \textcolor{red}{Res}(\textcolor{brown}{bool}). \left\{ \begin{array}{l} P \rightarrow R \{ \textcolor{blue}{BothIn}. P \rightarrow C \{ \textcolor{red}{BothIn}(\textcolor{brown}{Point}). \textcolor{blue}{t} \} \} \\ P \rightarrow R \{ \textcolor{blue}{BothOut}. P \rightarrow C \{ \textcolor{red}{BothOut}(\textcolor{brown}{Point}). \textcolor{blue}{t} \} \} \\ P \rightarrow R \left\{ \textcolor{red}{Intersect}(\langle \textcolor{brown}{Point}, \textcolor{brown}{Point} \rangle). R \rightarrow P \left\{ \textcolor{red}{Res}(\textcolor{brown}{point}). P \rightarrow C \left\{ \begin{array}{l} \textcolor{red}{SecOut}(\textcolor{brown}{point}). \textcolor{blue}{t} \\ \textcolor{red}{SecIn}(\langle \textcolor{brown}{point}, \textcolor{brown}{point} \rangle). \textcolor{blue}{t} \end{array} \right\} \right\} \right\} \right\} \right\} \right\} \right\} \\ \textcolor{red}{Close}. P \rightarrow C \{ \textcolor{red}{Close}. \textcolor{blue}{end} \} \end{array} \right. \\ \end{array} \right.$$

## References

- [1] Rumyana Neykova, Raymond Hu, Nobuko Yoshida, and Fahd Abdeljallal. A session type provider: Compile-time API generation of distributed protocols with refinements in F#. In *Proceedings of the 27th International Conference on Compiler Construction*, pages 128–138, Vienna Austria, February 2018. ACM.