1 APPENDIX

1.1 Constraint solver

To benchmark the Light replay technique, the Z3 constraint solver is used with *write-read* interleavings from the Light record phase as inputs. We do this in accordance with the steps stated in [2]. A constraint solver accepts as input a set of statements which are encoded as constraints. In our experiments, we implement a constraint solver using the Z3Py library of Z3 [1] theorem solver for python programming. We present an example of the constraint solving approach as follows:
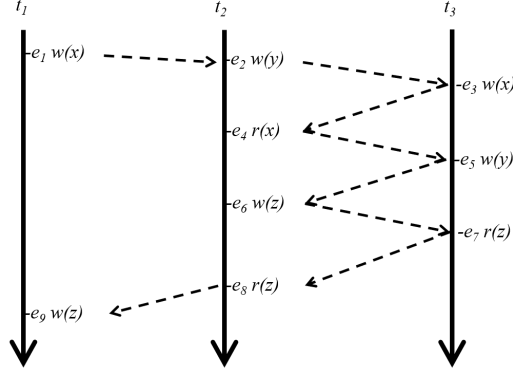

Fig. 1. Running example: An execution trace of a multithreaded program t_1 t_2, t_3. (Dashed arrows represent the global trace σ)

The trace $\sigma = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9 \rangle$ represents the execution shown in Figure 1. The set of thread interleavings from σ_2, $\{e_1 \rightsquigarrow e_3, e_3 \rightsquigarrow e_4, e_2 \rightsquigarrow e_5, e_6 \rightsquigarrow e_7, e_6 \rightsquigarrow e_8, e_8 \rightsquigarrow e_9, e_7 \rightsquigarrow e_9\}$ is produced.

Light records the write-read interleaving set
$\{e_3 \rightsquigarrow e_4, e_6 \rightsquigarrow e_7, e_6 \rightsquigarrow e_8\}$. We encode each interleaving for the constraint solver in the form $\{a - b - c - d\}$ where:

- a represents the thread id from the write event.
- b represents the program count for the interleaving write event.
- c represents the thread id for the read event.
- d represents the program count for the interleaving read event.

The set of write-read interleavings for σ is encoded as $\{\{3 - 3 - 2 - 4\}, \{2 - 6 - 3 - 7\}, \{2 - 6 - 2 - 8\}\}$.
The constraint solver also accepts the intra-thread interleaving constraint set $\{\{3 - 3 - 3 - 7\}, \{2 - 6 - 2 - 8\}\}$.

The constraint solver checks the satisfiability of the constraints and generates the set of possible schedules $\{\{3 - 3, 2 - 4, 2 - 6, 2 - 8, 3 - 7\}, \{3 - 3, 2 - 4, 2 - 6, 3 - 7, 2 - 8\}\}$.

The Z3Py code for the constraint solver is available online[1].

REFERENCES

[1] De Moura, L., and Bjørner, N. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (Berlin, Heidelberg, 2008), TACAS'08/ETAPS'08, Springer-Verlag, pp. 337–340.

[2] Liu, P., Zhang, X., Tripp, O., and Zheng, Y. Light: Replay via tightly bounded recording. *SIGPLAN Not. 50*, 6 (June 2015), 55–64.

---

[1] https://github.com/testrepo007/Constraint-solver