Data Quantification in Temporal Specification Languages

Domenico Bianculli, Giles Reger, Dmitriy Traytel

November 24, 2017

In this working group, we tried to collect and characterize the different kinds of quantification or parametrization that can be encountered in temporal specification languages used in the runtime verification community. Our selection, albeit far from being comprehensive, shows that the main semantic difference is the domain of quantification. We have identified four different groups.

Standard First-Order Quantification An early approach taken by Emerson's first-order linear temporal logic (FOLTL) [8] is to add standard first-order logic quantifiers to LTL. Thereby, the quantifiers range over a fixed domain, which is independent of the trace (sequence of structures). Chomicki's real-time extension of FOLTL, called metric first-order temporal logic (MFOTL) [6] follows this approach. The MonPoly monitoring tool [2] demonstrates how such quantifiers can be handled algorithmically.

Quantification over the Active Domain What is the active domain in a trace?

all values that occur (or will occur) in the stream
database community LTL-FO [7]
all previously seen values
values at current time-point
$\mathrm{LTL}^{\mathrm{FO}}$ [4]
$\mathrm{LTL\text{-}FO}^{+}$ [10]
Parametrized LTL [13]
LTL_4-C [11]

Freeze Quantification not to be confused with TPTL
[3]

Templates and Parametric Trace Slicing LOLA [9]
Parametric Trace Slicing [5, 12]
QEA [1]

References

[1] H. Barringer, Y. Falcone, K. Havelund, G. Reger, and D. E. Rydeheard. Quantified event automata: Towards expressive and efficient runtime monitors. In D. Giannakopoulou and D. Méry, editors, *FM 2012*, volume 7436 of *LNCS*, pages 68–84. Springer, 2012.

[2] D. A. Basin, F. Klaedtke, S. Müller, and E. Zalinescu. Monitoring metric first-order temporal properties. *J. ACM*, 62(2):15:1–15:45, 2015.

[3] D. A. Basin, F. Klaedtke, and E. Zalinescu. Runtime verification of temporal properties over out-of-order data streams. In R. Majumdar and V. Kuncak, editors, *CAV 2017*, volume 10426 of *LNCS*, pages 356–376. Springer, 2017.

[4] A. Bauer, J. Küster, and G. Vegliach. The ins and outs of first-order runtime verification. *Formal Methods in System Design*, 46(3):286–316, 2015.

[5] F. Chen and G. Rosu. Parametric trace slicing and monitoring. In S. Kowalewski and A. Philippou, editors, *TACAS 2009*, volume 5505 of *LNCS*, pages 246–261. Springer, 2009.

[6] J. Chomicki. Efficient checking of temporal integrity constraints using bounded history encoding. *ACM Trans. Database Syst.*, 20(2):149–186, 1995.

[7] A. Deutsch, L. Sui, V. Vianu, and D. Zhou. Verification of communicating data-driven web services. In S. Vansummeren, editor, *PODS 2006*, pages 90–99. ACM, 2006.

[8] E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. 1990.

[9] P. Faymonville, B. Finkbeiner, S. Schirmer, and H. Torfah. A stream-based specification language for network monitoring. In Y. Falcone and C. Sánchez, editors, *RV 2016*, volume 10012 of *LNCS*, pages 152–168. Springer, 2016.

[10] S. Hallé and R. Villemaire. Runtime monitoring of message-based workflows with data. In *ECOC 2008*, pages 63–72. IEEE Computer Society, 2008.

[11] R. Medhat, B. Bonakdarpour, S. Fischmeister, and Y. Joshi. Accelerated runtime verification of LTL specifications with counting semantics. In Y. Falcone and C. Sánchez, editors, *RV 2016*, volume 10012 of *LNCS*, pages 251–267. Springer, 2016.

[12] G. Reger and D. E. Rydeheard. From first-order temporal logic to parametric trace slicing. In E. Bartocci and R. Majumdar, editors, *RV 2015*, volume 9333 of *LNCS*, pages 216–232. Springer, 2015.

[13] V. Stolz. Temporal assertions with parametrized propositions. *J. Log. Comput.*, 20(3):743–757, 2010.