Efficient evaluation of quantitative non-functional Service Level Agreement [⋆]

Carlos G. Lopez Pombo[1,2] and Agustín E. Martinez Suñé[1]

[1] Universidad de Buenos Aires. School of science, Department of computing.
[2] CONICET–Universidad de Buenos Aires. Instituto de Investigación en Ciencias de la Computación (ICC).

Distributed software resulting from emerging paradigms such as *service-oriented computing* (SOC), Cloud/Fog computing and the Internet of Things are transforming the world of software systems in order to support applications able to respond and adapt to the changes of their execution environment, giving impulse to what is called the API's economy. The underlying idea of the API's economy is that it is possible to construct software artifacts by composing services provided by third parties and previously registered in repositories. This envisages a generation of applications running over globally available computational resources and communication infrastructure, which, at run-time, are dynamically and transparently reconfigured by the intervention of a dedicated middleware, subject to the negotiation of a *Service Level Agreement – SLA*[1].

Under this paradigm software services are accessed by their API. Thus, a key element is the availability of formal languages, together with associated analysis techniques, capable of fully expressing the API behavioral contract.

As usual, contract satisfaction is dealt with by checking whether certain judgement of the form $Pr \vdash Rq$ hold, where Pr is the provision contract and Rq is the requirement contract. While there exists a plethora of formalisms for describing and analysing both, soundness of the communication, and the functional behaviour of a software artifact; non-functional behaviour has generally been relegated to, at most, informal documentation. Many non-functional attributes, which we call quantitative attributes, can be used, whenever they are formally specified, to classify functionally equivalent services by Quality of Service – QoS they provide. This means that while services may have the same functional behavior, they might differ on their non-functional one (for example, a service may offer low speed computation at a very low cost while another, functionally equivalent one, might be faster but more onerous).

We propose an efficient procedure for evaluating quantitative non-functional SLA based on the state of the art techniques used in hybrid system verification [2], but adapted to profit from the fact that *contract reduction through convex optimization* can be done off-line when the service is registered in the repository, producing an efficiency gain when $Pr \vdash Rq$ is checked at run-time to evaluate a SLA. More over, as finding the minimum size contract is NP-complete, reduction can be performed as semantics preserving successive refinements.

⋆ Carlos G. Lopez Pombo's research is supported by Agencia Nacional de Promoción Científica y Tecnológica by grant PICT 2013-2129, and Consejo Nacional de Investigaciones Científicas y Técnicas by grant PIP 11220130100148CO.

Contract: Contracts are theory presentations whose formulae are boolean combinations of convex constraints [3]: 1) it is constructed by associating a boolean variable $v_{f(\overrightarrow{x})\ \mathcal{R}\ c}$ to each convex constraint $f(\overrightarrow{x})\ \mathcal{R}\ c$, with $\mathcal{R} \in \{\leq, <, \geq, >\}$[3], 2) the resulting theory presentation is represented by an ROBDD [4] producing an efficient representation of the assignments satisfying the boolean structure. There, an assignment γ, leading to 1, expresses a system of convex constraints $[\![\gamma]\!] = \{f(\overrightarrow{x})\ \mathcal{R}\ c \mid \gamma(v_{f(\overrightarrow{x})\ \mathcal{R}\ c}) = 1\} \cup \{\neg(f(\overrightarrow{x})\ \mathcal{R}\ c) \mid \gamma(v_{f(\overrightarrow{x})\ \mathcal{R}\ c}) = 0\}$.

Contract reduction through convex optimization: Given a contract represented as we described before: 1) the ROBDD is traversed in *BFS* to guarantee that pruning of the boolean representation happens as close to the root as possible, 2) when at a node labelled with variable $v_{f(\overrightarrow{x})\ \mathcal{R}\ c}$, with a partial assignment γ, we test whether the convex constraint $f(\overrightarrow{x})\ \mathcal{R}\ c$ is superfluous for $[\![\gamma]\!]$ by considering the feasibility of $[\![\gamma]\!] \cup \{\neg(f(\overrightarrow{x})\ \mathcal{R}\ c)\}$ using CPlex [5]; if it is not, $\mathit{Solutions}([\![\gamma]\!]) \subseteq \mathit{Solutions}(f(\overrightarrow{x})\ \mathcal{R}\ c)$, translated into a boolean constraint $\gamma \implies v_{f(\overrightarrow{x})\ \mathcal{R}\ c}$ used to reduce the ROBDD (the same is done for $[\![\gamma]\!] \cup \{f(\overrightarrow{x})\ \mathcal{R}\ c\}$) leading, in both cases, to a smaller representation of the contract. If none of them is superfluous then no reduction action is taken and the traverse continue. As usual, the smaller γ is, the more useful it becomes and, while finding the minimum subset of γ is an NP-complete problem, there are several efficient heuristics for finding minimal ones [2], 3) the resulting representation is an ROBDD in which every assignment leading to 1 characterize a convex set formed by values satisfying the original contract.

Non-functional SLA as contract implication: Determining if a given provision contract *Pr* satisfies a requirements contract *Rq* results from testing whether the contract reduction through convex optimisation of $\mathit{Pr} \wedge \neg \mathit{Rq}$ has no solution (i.e. no assignment leading to 1).

References

1. Fiadeiro, J.L., Lopes, A., Bocchi, L.: An abstract model of service discovery and binding. Formal Aspects of Computing **23**(4) (2011) 433–463
2. Shoukry, Y., Nuzzo, P., Sangiovanni-Vincentelli, A.L., Seshia, S.A., Pappas, G.J., Tabuada, P.: Smc: Satisfiability modulo convex optimization. In: Procs. of the 20th Intl. Conference on Hybrid Systems: Computation and Control, ACM (2017) 19–28
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge Univ. Press (2004)
4. Brace, K.S., Rudell, R.L., Bryant, R.E.: Efficient implementation of a BDD package. In Smith, R.C., ed.: Procs. of the 27th. ACM/IEEE conference on design automation, ACM Press (1990) 40–45
5. IBM: IBM ILOG CPLEX Optimization Studio (2004) `https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer`.

[3] As usual in convex optimization, strict relations in convex constraints are interpreted as non-strict ones but considering an appropriate threshold provided by the user.