Interval-based QuickXplain Algorithm

Adrien Bisutti, Julien Alexandre Dit Sandretto, Alexandre Chapoutot, Rémi Delmas

Interval-based QuickXplain Algorithm

Adrien Bisutti[1], Julien Alexandre dit Sandretto[1], Alexandre Chapoutot[1], and Rémi Delmas[2]

[1]ENSTA ParisTech, Université Paris-Saclay,
828, bd des maréchaux, 91762 Palaiseau Cedex, France
{bisutti,alexandre,chapoutot}@ensta.fr

[2]ONERA, DTIM
2, av. Edouard Belin BP74025, 31055 Toulouse Cedex 4, France
remi.delmas@onera.fr

Keywords: SMT, Conflict Clause, Interval Analysis

Introduction

Cyber-physical systems are made of discrete-time components, *i.e.*, pieces of software, and continuous-time components, *i.e.*, plants which continuously and strongly interact each other. Such kind of systems is usually found in critical applications, *e.g.*, aircraft autopilot or cruise control mechanism in a car. In consequence, it is important to ensure safety of such systems in order to avoid the lost of human life or catastrophic consequences.

Formal safety verification techniques aim at automatically and mathematically prove that a mathematical model of a cyber-physical system is safe. One difficulty of this approach is to deal with models involving a combination of state transition systems, representing the software parts, and ordinary differential equations, representing the plant parts. Model-checking techniques based on *SAT Modulo Theory* (SMT) techniques are efficient and robust enough to deal with such heterogeneous mathematical models. More precisely, *SAT modulo ODE* techniques [3, 4] are very promising to prove safety properties of cyber-physical systems. A SMT solver aims at proving that a first-order logical formula ϕ involving terms coming from different

theories, *e.g.*, linear integer arithmetic (LIA) or non-linear real arithmetic (NRA), is *satisfiable*, *i.e.*, there is a value of the variables which make the formula ϕ true or *unsatisfiable*, *i.e.*, for all values of the variables, ϕ is false. The main algorithm used in SMT solver is known as Conflit-Driven Clause Learning (CDCL).

This article is interested in SMT with NRA theory. This theory is associated to a solver based on *Interval Constraint Propagation* (ICP) techniques [1] which is easily implementable with IBEX library. The contribution of the paper is the definition of an algorithm dedicated to the conflict analysis step. This algorithm is an adaptation of the QuickXplain algorithm [2], mainly dedicated to discrete domain *Constraint Satisfaction Problems* (CSP), to continuous or numerical CSP.

Main contribution

Basically when only one theory T is involved, a SMT solver is made of a SAT solver and a T-solver, such as IBEX for NRA theory. The combination of the two solvers works as follows, starting from a logical formula ϕ involving T-terms in *normal conjunctive form* (CNF), $\phi_{\text{CNF}} \equiv \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} \ell_{ij}$,

1. For each literal ℓ_{ij}, *e.g.*, $\cos(x) + y \leqslant 1$, a Boolean variable $b(\ell_{ij})$ is assigned.

2. SAT solver searches for an assignment α of $b(\ell_{ij})$ such that ϕ_{CNF} is true, *i.e.*, $\alpha(b(\ell_{ij})) = \text{true}$. It can also find a contradiction.

3. T-solver is started if an assignment α exists, it determines if the conjunction of constraints induced by α is not contradictory in the theory T. In particular, the constraints c_i are defined such that $c_i = \ell_{ij}$ if $\alpha(b(\ell_{ij}))$ is true or $c_i = \neg\ell_{ij}$ otherwise. If the constraints c_i are true then SMT solvers returns SAT, otherwise a *conflict clause* κ is generated and added to ϕ_{CNF} to avoid unfeasible paths during the search of a new α in Step 2.

4. If no assignment α can be found then UNSAT is returned.

In iSat [3], which tightly integrates SAT and ICP solvers, a conflict learning algorithm, to generate κ, is used based on the decision tree at the SAT-solver level. Otherwise, it seems that no conflict analysis is defined for T-solver keeping the relative independence between the two solvers. To cope with this lack, a new conflict analysis method is proposed for NRA theory and based on the QuickXplain algorithm [2]. An implementation of this algorithm has been performed in IBEX.

Algorithm 1 Continuous QuickXplain algorithm

function QUICKXPLAIN(C, U, d)
  if $\text{card}(C) \neq 0 \wedge \text{contract}(C, d) = \emptyset$ **then**
    return $\emptyset$
  end if
  if $U = \emptyset$ **then**
    return $\emptyset$
  end if
  $\alpha_0, \ldots, \alpha_{n-1}$ be an enumeration of U
  $k \leftarrow 0$; $C_s \leftarrow C$; $d_s \leftarrow d$
  while $d_s \neq \emptyset \wedge k < \text{card}(U)$ **do**
    $C_s \leftarrow C_s \cup \{\alpha_k\}$; $d_s \leftarrow \text{contract}(C_s, d)$; $k \leftarrow k + 1$
  end while
  if $d_s \neq \emptyset$ **then**
    return $\emptyset$
  end if
  $k \leftarrow k - 1$; $X \leftarrow \{\alpha_k\}$; $i \leftarrow \lfloor k/2 \rfloor$
  $U_1 \leftarrow \{\alpha_0, \ldots, \alpha_{i-1}\}$
  $U_2 \leftarrow \{\alpha_i, \ldots, \alpha_{k-1}\}$
  if $U_2 \neq \emptyset$ **then**
    $C_2 \leftarrow C \cup U_1 \cup X$
    $X_2 \leftarrow$ QUICKXPLAIN(C_2, U_2, d)
    $X \leftarrow X \cup X_2$
  end if
  if $U_1 \neq \emptyset$ **then**
    $C_1 \leftarrow C \cup X$
    $X_1 \leftarrow$ QUICKXPLAIN(C_1, U_1, d)
    $X \leftarrow X \cup X_1$
  end if
  return X
end function

Results

The new conflict analysis method is given in Algorithm 1 where C stands for the smallest set of conflicted constraints (initially $\emptyset$), U stands for the initial set of constraints, d stands for the domain of variables. The contract operations in Line 2 and 14 is implemented using the HC4 algorithm. A positive side effect of this adaptation to continuous constraints of the original work [2] is that some propagation operations have been removed.

Acknowledgement

This work benefited of the "Chair Complex Systems Engineering - École polytechnique, THALES, DGA, FX, DASSAULT AVIATION, DCNS Research, ENSTA ParisTech, Télécom ParisTech, Fondation ParisTech and FDO ENSTA" and also partially funded by DGA MRIS.

References

[1] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, 2001.

[2] U. Junker and F. Valbonne. QuickXPlain: Conflict Detection for Arbitrary Constraint Propagation Algorithms. In *Proc. of the IJCAI Workshop on Modelling and Solving problems with constraints*, 2001.

[3] A. Eggers, M. Fränzle, and C. Herde. SAT Modulo ODE: A direct sat approach to hybrid systems. In *Proc. of International Symposium on Automated Technology for Verification and Analysis*, Springer, 2008.

[4] S. Gao, S. Kong, and E. M. Clarke. dReal: An SMT Solver for Nonlinear Theories over the Reals, In *Proc. of International Conference on Automated Deduction*, Springer, 2013.