A review on "A Decision Procedure for (Co)datatypes in SMT Solvers" by Nikolay Amiantov

Alexander Grebenyuk

March 25, 2017

In his project the author provides an implementation of the theory of co-datatypes for a satisfiability modulo theory(SMT) solver. To achieve this, the author shows how to define the satifiability on the theory of codatatypes and proves that his definition is correct, using derivation trees.

The reviewer was able to understand the whole project. The project has a clear structure and is split into several logically independent sections. In each section the author gives an informal definition of the contents of the section at first, and then provides a clear formal definition, which helps the reader to understand the material better. The project sections are ordered in a way to gradually build up formalism to achieve the goal of proving the refutation and the solition soundness of the author's theory of codatatypes in the final section, and avoid unnecessary complex definitions in the preliminary ones.

Some parts of the project were unclear, however. The comments and the suggestions how to improve these parts are listed by page below. The list of English grammar and puctuation errors, typos and word ordering problems is presented in the final section of the review, listed by page in a form of a table, with each row showing an errorneous combination of words in bold, a suggestion to improve it, a reason to consider it an error and a severity of this error to the reviewer's mind.

Logic defects and errors

- Page 1:
  - The author does not give a formal definition of "type" in general and rather resorts to a highly philosophical one, e.g. "a meaning of expression". It is advised to give a more formal definition of a type, or to not give any one at all.
  - The use of Haskell notation for the definitions of (co)datatypes is understandable, applicable and makes it clear for readers with a programming background. To the reviewer's mind it can be found confusing by other readers, so it is strongly advised to give a proper explaination to this notation, mostly for the "SCons" operation and the symbol "|".
  - Peano numbers are mentioned without a reference to a source.

- Page 2:
  - A problem with the Haskell notation again: it would be better to explain the meaning of the "_" symbol.
  - "Recall that a logic is defined through a signature that lists all non-logic symbols ..." – it is not clear where to recall from. It is advised to add a reference to a source.
  - Definitions of sytactic equality, term equality and α-equality on this page and the following ones are buried in a lot of text. It is hard for a reader to refer to these definitions throughout the course of the paper. Consider highlighting those in some way. The same advise is given for the definition of the set of (co)datatypes.
- Page 3:
  - The bracket notation is inclear in the "codata" formal definition. Please, clarify the meaning of $[\mathrm{s}_{ij}^{k} :]$.
  - The mathematical notation is incorrect in $\mathscr{F} = \mathscr{F}_{ctr} \uplus \mathscr{F}_{sel} \uplus true, false$, a correct one is $\mathscr{F} = \mathscr{F}_{ctr} \uplus \mathscr{F}_{sel} \uplus \{true, false\}$.
  - $\mathrm{d}_{i1}(x) \vee \cdots \vee \mathrm{d}_{im_i}(x)$ – it is not stated what this disjunction must be equal to, which can be naturally guessed from the infornal explaination to be 1 however. Still, it had better be stated clearly in the formal definition.
  - The "induction property" and the "dual property" definitions are buried in a lot of text, but are referenced in several proofs, and thus are considered highly important. Highlighting the properties in a similar to the "shared basic properties" way is advised, as the paper's clarity would benefit from it.
  - The formal definition for an "interpretation" of a formula is absent, but is referenced in the defintion of DC-satisfiability: "formula is called DC-satisfiable, if there exists an interpretation ... that satisfies it".
- Page 4:
  - The lemma 1 definition and proof have several defects:
    * "Corecursive codatatypes" – by definition of "codatatypes" from p. 1 they are already corecursive and their corecursivity need not be mentioned again in the lemma.
    * In the proof of the lemma the author states that "cross product of any domain to a singleton set is the same domain", which is technically incorrect, as a cross product of a domain to a singleton set would produce another domain, consisting of pairs of elements: $A \times S = \{(a, s) \mid a \in A \text{ and } S = \{s\}\}$, but it obviously will not affect the new domain's cardinality. This minor mathematical issue should be fixed, nonetheless.
- Page 5:
  - The operation "<<" is not defined formally by the author but is widely used in definitions of rules (e.g. Trans, Cong).

  - In the definition of the injection closure, the notation used in the conclusion of the rule is unclear, mainly "$E+$". Also it is unclear why the syntactic equality operator is used between a closure E and a set of rules on the right side.
  - The termin "free variable" occurs first on this page and is widely used by the author thereafter, but the paper lacks its formal definition.
  - Highlighting the α-equality is strongly advised, as it is used in the key theorems in the paper.

- Page 6:
  - The paper lacks an informal explaination of the reasons why complex rules (e.g. Acyclic, Unique, Single and Split) are constructed the way they are defined formally. The paper's clarity would benefit from providing it.
  - In the formal definition of the Split rule, the right part of the sub-conjuction on the right side of the disjuncton, namely "δ is finite" is mathematically incorrect and should be written as "$|\delta|$ is finite" instead.
  - The renaming part of a rule conclusion "$\mathcal{A} := \mathcal{A}\left[\tilde{u} \to \mu\tilde{u}.\ C(\tilde{t_1}, \ldots, \tilde{t_n})\right]$" makes use of an undefined symbol ":=". It is advised to clarify the renaming notation and make it consistent with a previously given one for terms.
  - The figure numbering is incorrect across the whole paper. Figures are numbered subsequently, not depending on the secton number, but are referenced by a combined section and figure number. For example the sentence "Rules for this phase are listed on figure 3.2" refers to the figure with caption "Figure 2: Derivation rules for acyclicity and uniqueness" in section 3.
  - "Hence, $\mathcal{A}(t^\tau)$ ... can take in models of E" – it is unclear what the "model" of E is, as the paper lacks the definition of a model of a closure of rules.
  - The termin "derivation trees" had better be defined formally by the section 4, as it is widely used across the paper in the key theorems and is mentioned as "closed" on this page without defining what a closed derivation tree is in either formal or informal way.

- Page 7:
  - $S_t^\infty = \lim_{i\to\infty} S_t^i$ – a formal definition or an informal one for a limit, defined on sets, is needed.
  - "S_t^∞ ... is a finite set because all chains of selectors in input are finite" – this part of the proof of the theorem 1 needs clarification.
  - The whole proof of the termination theorem (theorem 1) is given in a rather sketchy way, and thus is unclear to the reviewer. The induction process is not shown and is just informally described. If the calculations, needed to formally describe it, are bulky, it should at least be explained to the reader, that they are left outside the scope of the paper due to their cumbersomeness.

– The same problem holds for the proof of refutation soundness (theorem 2).

  Firstly, the statement "If Split is applied on some t^τ, by the previous step, then $E \cup \left\{t \approx C_j\left(s_j^1(t), \ldots, s_j^{n_j}(t)\right)\right\}$ is $\mathcal{DC}$-unsatisfiable for all $C_j \in \mathscr{F}_{ctr}^\tau$, and all possible models require this equality" needs clarification.

  Secondly, the induction process should be described using a proper mathematical induction algorithm, clearly specifying the base and the step.

– In the definition of the expansion rule, namely the "$\langle x\rangle^B_{t=\mu y.\ D(\bar{v})}$" case, it is unclear what happens if $x \notin B$ and $x \neq y$ at the same time.

- Page 8:

  – The author provides a good and clear example of a not normal μ-term. However, it is advised to provide some calculations, showing that $\mu y.\ C(y)$ and $\mu x.\ C(\mu y.\ C(y))$ are indeed α-equvalent.

  – In the point 2 of the definiton of a "normal interpretation", namely "…where x is fresh" the termin "fresh" is undefined. The same termin is used in the proof of solution soundness. It is asvised to provide a definition of a "fresh" term.

- Page 9:

  – An "assignment" notation is used here again, but is not defined.

- Page 10:

  – An undefined equality notation is used in "$t_1 \equiv t_2 \in F$". The symbol "$\approx$" should be used instead.

English errors, typos and readability issues

P.	Context	Suggestion	Reason
1	Introduction **And** Definitions	and	readability
	and so is expression 42∗2	use TeX cdot	readability
2	**and we** use capitalized names for constructors	We	readability
	starting on **data further from a base case**	data from a special case	readability
	for **an** SMT solver	a	grammar
	We use *many-sorted* first-order logic, that is, **logic** with	a logic	grammar
3	A type δ depends on another type ϵ **is** ϵ **is** the type of	ϵ is	repetition
	We also need **auxilitary** functions	auxiliary	typo
	Distinctness: . . . **no two constructors** are equal	no two constructors of the same type	readability
	We then define theory of datatypes and codatatypes $\mathcal{DC}$	We then define $\mathcal{DC}$ – the theory . . .	readability
	It defines a class of **Σ-interpretations which**	possibly the symbol for this class is missing	notation
	A formula is called *DC-satisfiable* if there exists an interpretation **in that** satisfies it	something is missing	readability
	An instance of degenerated codatatype . . . with **an** unique value	a	grammar
4	Lemma 1. . . . a singleton **(cardinality equals 1)**	with cardinality equal to 1	readability
	In the latter case, δ necessarily **as** a single constructor	has	typo
	$C(1, C(0, C(0, \dots)))$, $C(0, C(1, C(0, \dots)))$ **etc**	etc.	grammar
	The procedure **builts closure** of E under given rules which are assigned a **phase; rules** belonging . . .	builds; a closure; phase. Rules	grammar
	where **conclusion** either specifies	the conclusion	grammar
	or is ⊥ (**contradiction**)	a contradiction	grammar
	and **children** of a node are **results** of rule application	the children; the results	grammar
	one by one **in spirit** of depth-first search	in a spirit	grammar
	with leaf nodes ⊥ **allowed**	need to rephrase “contradiction allowed”	readability
	A **note** is saturated if no nonredundant rule **application** can be performed	node; applications	grammar

	a calculus of equality **sets** with applications as operations	of sets	grammar
	Refl, Sym and Trans, **Refl** are encoded properties	-	repetition
	whereas Inject computes **unification** (downward) closure	the unification	grammar
5	We write $\mathcal{A}(x)$ to get a representative for **equivalence class** of x	an/the equivalence class	grammar
	Clash represents **failure** to unify	a/the failure	grammar
	Figure 1. The Clash rule: $\mathbf{C}(t_1, \ldots, t_n) \approx \mathcal{D}(u_1, \ldots, u_n) \in E \quad \mathbf{C} \neq \mathbf{D}$	-; -; not unified display style for the same entities	notation
	$C \in \mathscr{F}F_{ctr}$	$\mathscr{F}_{ctr}$	typo
	All datatypes therefore ... and all codatatypes – as **cyclic**	ones	grammar
	α-disequivalent—e.g.	too long dash	typo
6	Hence, $\mathcal{A}(t^\tau)$ ... can take in models of E**. by appling**	E by; applying	grammar, punctuation
	Because of μ**-terms**, **there is** no infinitely expanding **terms so number** of times	Because of μ-terms what?; not stated where afterwards; terms**,** so; the number	readability, grammar, punctuation
	They capture **properties** of (co)datatypes described above	the properties	grammar
	Informally first rule of this phase ... implements **search**	Informally**,** the first rule; searching	grammar
	The second rule is needed to deal with **described** degenerate case	the described	grammar
	with **root node**	the root node (multiple occurences found throughout the paper)	grammar
	If Split is applied on some t^τ, **by** the previous step	at	grammar
7	Now we construct **family** of sets	a family	grammar
	Consider S_t^∞ ... because all chains of selectors **in input**	its/the input	grammar
	Depth of a tree branch is therefore	The depth	grammar
	First we define ***expansion*** ... with **free variable** x **is** substituted by **body** of t	the expansion; a free variable; unnecessary “is”; the body	grammar
	By definition of μ**-terms after**	μ-terms**,** after	punctuation

8	Finally a normal form … recursively normalizing **the other** subterms	other	grammar
	Now that we defined	Having defined/ Now that we have defined	grammar, readability
	$\mathcal{I}_{\mathscr{Y}}(\tau)$ denotes **domain** of type τ	the domain	grammar
	terms that are not **equalized**	equal/unified	readability
	We will base the process on **mapping** $\mathcal{A}$ from section	the mapping	grammar
9	The set is constructed … by exhaustively **appling**	applying	typo
	First point can be proven	The first	grammar
	Points 2 and 3 can be proven by induction. **First one needs to prove that**	To prove/for proving the first one, we need to prove …	grammar, readability
	Next one needs to prove	Next**,** one	grammar
10	Then because $\mathcal{A}$ is an equivalence class **that implies** all equalities … are satisfied	The whole sentence needs to be rephrased to mitigate the ambiguity of its interpretation. E.g.: "Then, taking into account, that $\mathcal{A}$ is an equivalence class, all qualities …"	readability
	by induction hypothesis **then** it's	an unnecessary word	readability
	by induction hypothesis then **it's**	it is	informal language
	We use the fact that because we consider a saturated node	As a saturated node is considered, …	readability
	If we let u'_i be i**th** argument	the ith	grammar
	We **can't** apply Split	cannot	informal language
	If $j \neq j'$ **we use that** Cong is exhausted	not unnecessary words	readability
	Cong is **exhausted so** for any	exhausted**,** so	punctuation
	$C_{j'}\left(s^1_{j'}(u), \ldots, s^n_{j'}(u)\right)$ **for unknown yet** j'	for yet unknown	grammar
	Conflict does not **apply so** …	apply**,** so	punctuation
	Because $\mathcal{I}_{\mathscr{F}}(t) =_\alpha \mathcal{A}^*(t)$ … and they are not equal by **lemma 4(3) we** have	lemma 4(3)**,** we	punctuation