One-sided $\top\top$-closed relations: The What and Why

Lars Birkedal Kristian Støvring Jacob Thamsborg
IT University of Copenhagen
{birkedal,kss,thamsborg}@itu.dk

Abstract

Syntactic $\top\top$-closed relations were introduced by Pitts [10] as an alternative to admissible relations. In the same issue of that same journal, Abadi [1] gave a semantic analogue of $\top\top$-closed relations and a proof that being $\top\top$-closed is a strictly stronger property than being admissible. We propose a notion of *one-sided* $\top\top$-closed relations, also in a semantic setting. Similarly to *ibid.*, we provide a characterization of these relations and show that one-sided $\top\top$-closed relations are one-sided admissible but not necessarily the other way round. Moreover, we argue that such one-sided relations are more useful than regular two-sided relations for proving program equivalences with semantic, relational techniques.

1. The What of One-sided $\top\top$-closed Relations

1.1 Preliminaries and Definition of $\top\top$-closure

A *partial order* is a set A with a binary relation $\sqsubseteq$ that is reflexive, transitive and antisymmetric. Is is *complete* if any chain $a_0 \sqsubseteq a_1 \sqsubseteq a_2 \sqsubseteq \cdots$ of elements of A has a least upper bound $\sqcup_m a_m$. And it is *pointed* if it has a least element $\perp$. We use the term *domain* for a complete, pointed partial order. We denote by *2* the two-element domain $\{\perp, \top\}$ with $\perp \sqsubseteq \top$.

A map $f : D \to E$ from one domain D to another E is *monotone* if for any two $d_0, d_1 \in D$ we have that $d_0 \sqsubseteq_D d_1$ implies $f(d_0) \sqsubseteq_E f(d_1)$. It is *continuous* if it is monotone and for any chain $d_0 \sqsubseteq_D d_1 \sqsubseteq_D d_2 \sqsubseteq_D \cdots$ of D we have $f(\sqcup_m d_m) = \sqcup_m f(d_m)$. And it is *strict* if $f(\perp_D) = \perp_E$. The set of all strict, continuous functions from D to E is denoted $D \multimap E$.

A subset $D' \subseteq D$ of a domain D is *complete* if for any chain $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \cdots$ in D' we have $\sqcup_m d_m \in D'$ too. And D' is *admissible* if it is complete and also has $\perp \in D'$. We say that $D' \subseteq D$ is *downwards closed* if $d \in D_0$ and $e \in D$ with $e \sqsubseteq d$ implies $e \in D'$ too. *Upwards closed* is the other way round.

We now have sufficient machinery to introduce the one-sided $\top\top$-closure of a relation between two domains:

Definition 1. *Let D and E be domains with $R \subseteq D \times E$ some relation between them. We define the* one-sided $\top$-closure *of R as*

$$R^\top = \{(k, l) \in (D \multimap 2) \times (E \multimap 2) \mid \forall (d, e) \in R.\, k(d) = \top \implies l(e) = \top\}$$

and the one-sided $\top\top$-closure *of R to be*

$$R^{\top\top} = \{(d, e) \in D \times E \mid \forall (k, l) \in R^\top.\, k(d) = \top \implies l(e) = \top\}.$$

We say that R is one-sided $\top\top$-closed *if we have $R = R^{\top\top}$.*

The definitions have termination of the left hand side implying termination of the right hand side, but not the other way round. Hence we call the relations one-sided. Replacing implication with biimplication yields the $\top\top$-closure defined by Abadi [1]. The idea here is to have termination of the left hand side approximate termination of the right hand side instead of going for equivalence between the two – we motivate this in the upcoming Section 2.

The closure is extensional in nature. One intuition is to disallow direct observation of values of the domains D and E: we cannot inspect these, *per se*, instead we must apply continuations, i.e., strict, continuous maps into 2, and ponder their termination behavior. Here where we think of $\top$ and $\perp$ as termination respectively non-termination. Under this interpretation we translate the intensional relation R into the extensional relation $R^\top$ of continuations with proper termination behavior on R. And then we might as well extend R to $R^{\top\top}$ because these are the same up to indirect observation by application of the continuations of $R^\top$.

1.2 Characterization and Comparison to Admissibility

Definition 2. *A relation $R \subseteq D \times E$ between two domains D and E is called a* Bohr *relation if, for each $e \in E$, we have*

$$R(-, e) = \{d \in D \mid (d, e) \in R\}$$

admissible and downwards closed, and, for each $d \in D$, we have

$$R(d, -) = \{e \in E \mid (d, e) \in R\}$$

upwards closed.

Bohr relations were designed with the proof of Theorem 4 in mind. But the overall layout should be familiar to anyone building logical relations over domains: admissibility in the left coordinate is present, e.g., in the 'auxiliary' relation of the adequacy proof in Section 5 of [9]. And the relation used to prove adequacy in [13] has properties strikingly similar to the above, but for the fact that the upwards closure in the right coordinate is of a syntactical nature, cf. Lemma 4.2 of *ibid.*

There is an unsurprising closure operator associated to Bohr relations:

Proposition 3. *For any relation $R \subseteq D \times E$ between domains D and E we have that*

$$\overline{R} = \bigcap_{R \subseteq S \subseteq D \times E,\ S \text{ Bohr relation}} S$$

is a Bohr relation, furthermore it is least such that contain R.

The proof is straightforward as the defining properties of Bohr relations are preserved under intersection. $D \times E$ is itself a Bohr relation so the intersection is over a nonempty set of sets.

And now for the main result and an immediate corollary:

Theorem 4. *For any relation $R \subseteq D \times E$ between domains D and E we have that $R^{\top\top}$ is a Bohr relation, indeed we have $\overline{R} = R^{\top\top}$.*

Proof. It is not hard to verify that $R^{\top\top}$ is a Bohr relation and that it contains R as a subset. Hence we have $\overline{R} \subseteq R^{\top\top}$. To prove $R^{\top\top} \subseteq \overline{R}$ we pick arbitrary $(d, e) \in R^{\top\top}$ and aim to prove that $(d, e) \in \overline{R}$. Consider the two strict, continuous functions

$k \in D \multimap 2$ and $l \in E \multimap 2$ defined as follows:

$$k = \lambda f. \begin{cases} \bot, & \text{if } f \in \overline{R}(-, e) \\ \top, & \text{otherwise} \end{cases}, \quad l = \lambda f. \begin{cases} \bot, & \text{if } f \sqsubseteq e \\ \top, & \text{otherwise.} \end{cases}$$

Note that $k \in D \multimap 2$ holds because $\overline{R}(-, e)$ is admissible and downwards closed. Clearly we are done if $(k, l) \in R^{\top}$. So take $(d_0, e_0) \in R$ and assume that $k(d_0) = \top$. Then $(d_0, e) \notin \overline{R}$. But then $e_0 \sqsubseteq e$ cannot be true since we have $(d_0, e_0) \in R \subseteq \overline{R}$ and know that $\overline{R}(d_0, -)$ is upwards closed. □

Corollary 5. *R is one-sided $\top\top$-closed if and only if it is Bohr.*

We say that a relation $R \subseteq D \times E$ between domains D and E is *one-sided admissible* if $R(-, e)$ is admissible for each $e \in E$. If R is Bohr then it is one-sided admissible too by definition, but analogous to Proposition 7 of [1] we cannot turn the tables:

Proposition 6. *Not all one-sided admissible relations between domains are one-sided $\top\top$-closed.*

Proof. The relation $R = \{(\bot, \bot), (\top, \bot), (\bot, \top)\} \subseteq 2 \times 2$ is one-sided admissible but not Bohr as $R(\top, -) = \{\bot\}$ fails to be upwards closed. Hence, by Theorem 4, it is not one-sided $\top\top$-closed and we have our counterexample. □

2. The Why of One-sided $\top\top$-closed Relations

Having introduced one-sided $\top\top$-closed relations we now proceed to argue their usefulness. We aim for (but may fall a bit short of) a level of detail that suffices to state our point without committing to one particular technical setup.

One popular thing to do in denotational semantics is to carve out semantic types as relations on a so-called *universal* domain that loosely corresponds to the set of closed, syntactic values. Solving a recursive domain equation as prescribed by Smyth and Plotkin [12] is arguably the foremost way of building a universal domain. We need to know that this yields not only a domain D with a certain structure but also a sequence of projections $(\pi_m)_{m \in \omega}$ on that domain. These are idempotent members of $D \multimap D$ with the property that for any $d \in D$ we have

$$\bot = \pi_0(d) \sqsubseteq \pi_1(d) \sqsubseteq \pi_2(d) \sqsubseteq \cdots \sqsubseteq \sqcup_m \pi_m(d) = d.$$

The projections provide a useful 'hold' on each element of D as the least upper bound of its projections; this is explored extensively by Pitts [9].

Consider now the interpretation of types as relations on D. If we have recursion on terms or types we run into trouble if we place no restrictions on these relations. For the interpretation of recursive terms, i.e., fixed points of functions, to belong to the proper types we need the relations to respect the denotational construction of fixed points. The standard fix dating at least back to Reynolds [11] is to demand *admissibility* of the relations: $R \subseteq D \times D$ must contain $(\bot, \bot)$ and if $(d_m, e_m) \in R$ holds for all $m \in \omega$ we must have $(\sqcup_m d_m, \sqcup_m e_m) \in R$ too whenever $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \cdots$ and $e_0 \sqsubseteq e_1 \sqsubseteq e_2 \sqsubseteq \cdots$ are chains of D.

If we have recursive types too, we need further restrictions. A standard requirement present in, e.g., [2] and [4] is to enforce *uniformity*: $R \subseteq D \times D$ must have the property that for any $(d, e) \in R$ and any $m \in \omega$ we have $(\pi_m(d), \pi_m(e)) \in R$ too. To understand the need for uniformity, we remark that for an admissible and uniform relation $R \subseteq D \times D$ we have $(d, e) \in R$ iff $\forall m \in \omega.\ (\pi_m(d), \pi_m(e)) \in R$. This provides an inductive-like approach to building relations; the details vary quite from one presentation to the other.

So far, we have covered only well-known territory. But in recent work by the authors, two additional demands have been made on the interpretation of types. First, we think of the relations not so much as the meaning of types, but rather as an approximation to typed operational equivalence. In other words, we would like for two syntactic values of some type to be operationally equivalent if their interpretations are related in the interpretation of that type. This is not new, but it does impose further restrictions on the relations: in [6] we disallow relations (corresponding to value types) that relate bottom to non-bottom elements or the converse. In [5] we disallow bottom in relations (corresponding to value types) in general as befits a model of a call-by-value language – but we simultaneously rephrase uniformity to mean that for for any $(d, e) \in R$ and any $m \in \omega$ we have $(\pi_m(d), \pi_m(e)) \in R \cup \{\bot, \bot\}$. These demands are not new, they are present also in, e.g., work by Crary and Harper [8] and by Bohr and Birkedal [7], if somewhat convoluted in the latter case.

The second additional demand we add is to support relational parametric reasoning by introducing universal types, and this is where the chain comes off. The conceptual relations that one 'plugs in' to do relational parametric reasoning often fail to meet the technical requirements accumulated above, uniformity in particular. Taking some closure that entails admissibility and uniformity is not always possible and so we are prevented from using the full power of relational parametric reasoning. We sketch one such example: It is not unreasonable to relate (interpretations of) finite lists to their lengths. But one possible encoding of lists has the property that applying a projection π_n to a list with n less than or equal to the length of the list returns bottom. On the other hand, lengths of lists (integers) are preserved by all projections except π_0. So we can have no relation that contains, say, the pair of some list of length 3 and the integer 3 whilst at the same time being uniform and meeting the requirements introduced in the previous paragraph.

It is this shortcoming we address with the one-sided $\top\top$-closed relations. Or, more to the point, we suggest relations that approximate contextual approximation rather than contextual equivalence: if the interpretations of two syntactic values of some type are related, then we (would like to) know that termination of the left hand side in some context implies termination of the right hand side in that same context – but not necessarily the other way round. The restrictions on our relations now move to the first coordinate, they must be one-sided admissible and one-sided uniform as is the case for, e.g., one-sided $\top\top$-closed relations. In line with [6], we also must outlaw pairs with a non-bottom first coordinate and bottom as the second coordinate, but the one-sided $\top\top$-closure fortunately never introduces such pairs (unless they are present in the original relation, that is). *Thus the overall idea is to remove the artificial 'synchronization' restriction imposed by (two-sided) uniformity and so be free to apply relational parametric reasoning at will.*

Going for contextual approximation instead of contextual equivalence seems standard in recent step-indexed models of recursive types. There is an analogy to one-sided uniformity here: we do, e.g., in [3], not require expressions to terminate in the same number of steps to be related. Rather we allot a number of steps for the left hand side to terminate, and if this happens then we require the right hand side to terminate in any number of steps. Requiring the expressions to march in step would, most likely, not invalidate the soundness of the reasoning but rather prove fewer (albeit stronger) equivalences.

We finally remark that the idea of approximating contextual approximation rather than contextual equivalence is present in the 4-tuples of [7] by Bohr and Birkedal. Their work had, as noted by Hongseok Yang, the ability to function with any kind of relation, whether admissible or not, uniform or not. Distilling this ability is the goal of the present work, a goal that we think has been met: 4-tuples are – roughly and in retrospect – just two one-sided $\top\top$-

closed relations grouped together to be able to argue both ways of contextual approximation in one go.

References

[1] Martín Abadi. top-top-closed relations and admissibility. *Mathematical Structures in Computer Science*, 10(3):313–320, 2000.

[2] Martín Abadi and Gordon D. Plotkin. A per model of polymorphism and recursive types. In *LICS*, pages 355–365. IEEE Computer Society, 1990.

[3] Amal J. Ahmed. Step-indexed syntactic logical relations for recursive and quantified types. In Peter Sestoft, editor, *ESOP*, volume 3924 of *Lecture Notes in Computer Science*, pages 69–83. Springer, 2006.

[4] Roberto M. Amadio. Recursion over realizability structures. *Inf. Comput.*, 91(1):55–85, 1991.

[5] Lars Birkedal, Kristian Støvring, and Jacob Thamsborg. Realizability semantics of parametric polymorphism, general references, and recursive types. In Luca de Alfaro, editor, *FOSSACS*, volume 5504 of *Lecture Notes in Computer Science*, pages 456–470. Springer, 2009.

[6] Lars Birkedal, Kristian Støvring, and Jacob Thamsborg. Relational parametricity for references and recursive types. In Andrew Kennedy and Amal Ahmed, editors, *TLDI*, pages 91–104. ACM, 2009.

[7] Nina Bohr and Lars Birkedal. Relational reasoning for recursive types and references. In Naoki Kobayashi, editor, *APLAS*, volume 4279 of *Lecture Notes in Computer Science*, pages 79–96. Springer, 2006.

[8] Karl Crary and Robert Harper. Syntactic logical relations for polymorphic and recursive types. *Electr. Notes Theor. Comput. Sci.*, 172:259–299, 2007.

[9] Andrew M. Pitts. Relational properties of domains. *Inf. Comput.*, 127(2):66–90, 1996.

[10] Andrew M. Pitts. Parametric polymorphism and operational equivalence. *Mathematical Structures in Computer Science*, 10(3):321–359, 2000.

[11] John C. Reynolds. Types, abstraction and parametric polymorphism. In *IFIP Congress*, pages 513–523, 1983.

[12] Michael B. Smyth and Gordon D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM J. Comput.*, 11(4):761–783, 1982.

[13] Thomas Streicher. *Domain-theoretic Foundations of Functional Programming*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2006.