A Relational Abstraction for Functions
B. Jeannet\textsuperscript{1}, D. Gopan\textsuperscript{2}, and T. Reps\textsuperscript{2}

\textsuperscript{1} IRISA; Bertrand.Jeannet@irisa.fr
\textsuperscript{2} Comp. Sci. Dept., Univ. of Wisconsin; \{gopan, reps\}@cs.wisc.edu

Overview. This paper concerns the abstraction of sets of functions for use in abstract interpretation. The paper gives an overview of abstract domains for functions and formalizes a new family of relational abstract domains that allows sets of functions to be abstracted more precisely than with known approaches, while being still machine-representable.

A major strength of Abstract Interpretation is the ability create complex abstract domains from simpler ones \cite{1}. In particular, (i) Galois connections can be composed, which allows a complex abstraction to be described as the composition of simpler ones, offering the ability to identify clearly the different kind of approximations that take place; (ii) given two abstractions for sets of elements, \( \varphi(D_i), i = 1, 2 \), there exist techniques for abstracting functions of signature \( D_1 \rightarrow D_2 \) \cite{2}.

The starting point for the paper is the abstraction method defined in \cite{5}, which presents a family of abstract domains that are useful when it is desired to connect storage elements (e.g., elements of arrays and lists) with numeric quantities. This paper reformulates that abstraction in a more general way—as a general method of abstracting a set of functions—which allows the basic idea from \cite{5} to be applied more widely. Moreover, when the new formulation is compared with previously known ways of abstracting a set of functions, it yields more precise abstractions. We are just beginning to explore instantiations of the method that go beyond the ones used in \cite{5}.

We formalize a generic abstract-interpretation combinator, which abstract sets of functions of signature \( D_1 \rightarrow D_2 \) in a relational way, assuming the existence of abstractions \( A_1 \) and \( A_2[n] \) for \( \varphi(D_1) \) and \( \varphi((D_2)^n) \), respectively (where \( A_1 \) is of finite cardinality \( n \)). The obtained abstract domain is precisely \( A_2[n] \). In contrast to \( A_1, A_2[n] \) may be a complex lattice (relational, infinite, and of infinite height), like the lattice of octagons \cite{4} or convex polyhedra \cite{3}. This relational function-abstraction is more precise than the classical approach described in the literature \cite{2}, because of its ability to represent relationships between the images of different elements mapped by a set of functions.

In terms of precision, the relational function-abstraction \( A_2[n] \) lies in-between the classical function-abstraction \( A_1 \rightarrow A_2 \) (\( \approx (A_2)^n \)) and its disjunctive completion. The important point is that \( A_2[n] \) is still finitely representable in the same circumstances as \( A_1 \rightarrow A_2 \) (i.e., when \( A_1 \rightarrow A_2 \) is finitely representable).

Classical abstractions of functions and relations. Fig. 1 depicts the classical methods for abstracting functions of signature \( D_1 \rightarrow D_2 \) and relations \( R \subseteq D_1 \times D_2 \), that can be built from two Galois connections \( \varphi(D_1) \xrightarrow{\gamma_1} A_1 \) and \( \varphi(D_2) \xrightarrow{\gamma_2} A_2 \). These abstractions are described in \cite{2} and we use the same notation. Some of the abstractions are illustrated on Fig. 3, where \( D_1 = U \),
$A_1 = (U^2)^\top$ (a flat lattice completed by $\top$ and $\bot$), $D_2 = \mathbb{B}$, and $A_2 = \varphi(D_2)$ (no abstraction of the codomain in the example).

In summary, $\varphi$ abstracts a set of functions $F \subseteq D_1 \rightarrow D_2$ by a single function $F^\varphi : D_1 \rightarrow \varphi(D_2)$. One can then abstract the equivalent transformer $F^\varphi : \varphi(D_1) \xrightarrow{\varphi} \varphi(D_2)$ with a function $F^\varphi : A_1 \xrightarrow{\alpha} A_2$ by abstracting both the domain and the codomain: $\alpha_\varphi(F^\varphi) = \alpha_2 \circ F^\varphi \circ \gamma_1$, $\gamma_\varphi(F^\varphi) = \gamma_2 \circ F^\varphi \circ \alpha_1$. One can also abstract separately the domain (abstraction $\delta$) and the codomain (pointwise abstraction $\pi$) to obtain two intermediate abstractions.

![Diagram](image)

(a) for sets of functions
(b) for relations

$L_1 \simeq L_2$ means lattice isomorphism
$L_1 \rightarrow L_2$ means that $L_1 \succeq L_2$ ($L_2$ abstracts $L_1$)
$L_1 \longrightarrow L_2$ means that $L_1$ is the disjunctive completion of $L_2$.

Fig. 1. Lattice of abstract domains for functions and relations.

Fig. 2 shows also that viewing a function $D_1 \rightarrow D_2$ as a Boolean function $D_1 \times D_2 \rightarrow \mathbb{B}$, as canonical abstraction [5] does, leads to a better precision.

Viewing a function as: $D_1 \times D_2 \rightarrow \mathbb{B}$, $D_1 \rightarrow D_2$, $\varphi(D_1 \times D_2)$

Abstracting sets of functions with:

![Abstracting Diagram](image)

Resulting abstract domains: $A_1 \times A_2 \rightarrow \varphi(\mathbb{B}) \succeq A_1 \rightarrow A_2 \succeq A_1 \times A_2$

Fig. 2. Different ways of coding (sets of) functions, and the resulting abstractions.

Relational function-abstraction. The principle of relational function-abstraction is illustrated in Fig. 3. In this figure, we abstract by different methods the set of functions $A_1$, and we observe in $A_2$, $A_3$ and $A_4$, which are images of $A_1$ by different $\gamma \circ \alpha$ mappings, the loss of information induced by these methods.

The point of our abstraction is to avoid the abstraction $\gamma$, which loses (in $D$) the relation $f(u_1) = f(u_2)$ that holds for any $f \in A_1$. Instead, we directly abstract the domain $U$ separately on each element of $A_1$, obtaining $B_1$. This is equivalent to the disjunctive completion of $\alpha_\delta \circ \alpha_\varphi$.

We then apply the original abstraction $\eta$ to obtain $C$. The point is that if $U^\varphi$ is finite, $C$ is a set of $[U^\varphi]$-dimensional vectors and can in turn be abstracted by any numerical lattices (octagons, polyhedra). The definition of $\eta$, and particularly
\( \gamma_n \), is somewhat subtle:

\[
\alpha_\eta(F) = \bigcup_{f \in F} \{ f^\# | \forall u^\#: f^\#(u^\#) \in f(u^\#) \}
\]

(1)

\[
\gamma_\eta(F^\#) = \{ f | \forall f^\# \in (U^\# \to \mathbb{R}) : \left( \forall u^\#, f^\#(u^\#) \in f(u^\#) \right) \Rightarrow f^\# \in F^\# \}
\]

(2)

Another formulation can be given by treating functions from (finite) \( U^\# \) as vectors:

\[
\alpha_\eta(F) = \bigcup_{(x_1, \ldots, x_n) \in F} X_1 \times \ldots \times X_n
\]

(3)

\[
\gamma_\eta(F^\#) = \{(X_1, \ldots, X_n) | X_1 \times \ldots \times X_n \subseteq F^\# \}
\]

(4)

A comparison of B2 with B1 shows what is lost by \( \eta \). Observe, however, that the relation \( f(u^1) = f(u^2) \) is preserved. A comparison of A3 with A4 shows what is gained with relational function-abstraction \( \Phi \) compared to the classical abstraction \( \phi \). For instance, we preserve in A3 the property \( f(u_2) + 2 \leq f(u_3) \leq f(u_2) + 3 \), unlike in A4.

Because \( \gamma_\eta \) is not trivial, our contribution can also be viewed as a way to give a meaning to a set \([U^\#]_n\)-dimensional vectors as a set of functions \( U \to \mathbb{R} \), where \( U \) may be infinite, using the concretization \( \gamma_{\delta, \epsilon} \circ \gamma_\eta \).

In full generality, if the codomain is also abstracted, assuming \( \alpha_2[n] \) abstracts \( \varphi(D^\#) \) with \( A_2[n] \), relational function-abstraction \( \alpha_\phi \) is defined as \( \alpha_2[n] \circ \alpha_\eta \circ \alpha_{\delta, \epsilon} \).

References

Fig. 3. Different abstractions of the concrete set of functions A1 and the loss of information induced by them (shown by concrete values A2, A3, and A4). Abstract value C (whose concretization is A3) is the abstract value obtained with relational function-abstraction $\Phi$, whereas abstract value E (whose concretization is A4) is the abstract value obtained by the abstraction $\phi$. 

$U, U^1$, and $\pi : U \rightarrow U^1 : U$

$u_1 \mapsto \{1\} \{2\}$
$u_2 \mapsto \{2\} \{3\}$
$u_3 \mapsto \{4\} \{5\}$
$u_4 \mapsto \{4\} \{5\}$

$U \rightarrow \phi(R)$

$u_1 \mapsto \{1\} \{2\}$
$u_2 \mapsto \{2\} \{3\}$
$u_3 \mapsto \{4\} \{5\}$
$u_4 \mapsto \{4\} \{5\}$

$U^1 \rightarrow \phi(R)$