Underapproximating Predicate Transformers

David A. Schmidt*

Kansas State University, Manhattan, Kansas, USA
schmidt@cis.ksu.edu

Abstract. We study the underapproximation of the predicate transformers used to give semantics to the modalities in dynamic and temporal logic. Because predicate transformers operate on state sets, we define appropriate powerdomains for sound approximation. We study four such domains — two are based on "set inclusion" approximation, and two are based on "quantification" approximation — and we apply the domains to synthesize the most precise, underapproximating $\widetilde{pre}$ and pre transformers, in the latter case, introducing a $focus$ operation. We also show why the expected abstractions of $post$ and $\widetilde{post}$ are unsound, and we use the powerdomains to guide us to correct, sound underapproximations.

1 Introduction

When we prove a property, ϕ, of a program, P, we typically employ an abstraction on P's and ϕ's concrete domain, C, so that we *over*approximate P to $P^\sharp$ and *under*approximate ϕ to $\phi^\flat$, where $P^\sharp$ and $\phi^\flat$ are stated within an abstract domain, A. If we show $P^\sharp$ has property $\phi^\flat$, then we conclude P has ϕ as well.

This approach quickly becomes complicated: Although C might be a set, A is usually partially ordered. For example, when C is Int and A is $Sign$, we have orderings like $isPositive \sqsubseteq_{Sign} isNotNegative$, because $\gamma(isPositive) \subseteq \gamma(isNotNegative)$, where $\gamma : Sign \to \mathcal{P}(Int)$ concretizes signs. Even when A is a set, e.g., a set of state partitions, computing least- and greatest fixed points of state-transition functions and recursively defined assertions requires a powerset of the state partitions, partially ordered by subset inclusion [25].

Next, a logical property, ϕ, is interpreted as a set, $[\![\phi]\!] \in \mathcal{P}(C)$. When the property is abstracted to $\phi^\flat$, which is itself a set, $[\![\phi]\!]^A \in \mathcal{P}(A)$, A's ordering affects $\mathcal{P}(A)$'s, and denotational semantics indicates there are a variety of powerdomains that one might use [18, 24] to establish soundness, i.e., $[\![\phi]\!] \supseteq \gamma^*[\![\phi]\!]^A$.

The situation becomes more complex when program P's concrete transition function is nondeterministic, $f : C \to \mathcal{P}(C)$, meaning its abstraction should be $f^\sharp : A \to \mathcal{P}(A)$. What powerdomain should be used for $f^\sharp$'s codomain? Is it the same one as that used to define $[\![\phi]\!]^A$?

Yet another complication is that properties, ϕ, can be expressed by the predicate transformers, $\widetilde{pre}$, pre, $post$, and $\widetilde{post}$. The four predicate transformers behave differently with respect to a given $f^\sharp : A \to \mathcal{P}(A)$. Fortunately, for an

* Supported by NSF ITR-0086154 and ITR-0326577.

K. Yi (Ed.): SAS 2006, LNCS 4134, pp. 127–143, 2006.

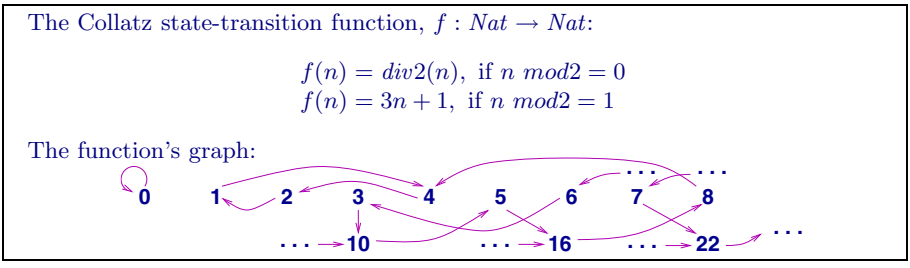


Fig. 1. Collatz program and its state-transition graph

overapproximating $f^\sharp$, $\widetilde{pre}_{f^\sharp}[\![\phi]\!]^A$ underapproximates $\widetilde{pre}_f[\![\phi]\!]$, meaning we can soundly calculate abstract preconditions like those in ACTL [3, 7].

But $pre_{f^\sharp}$ is *not* well behaved for $f^\sharp$, and the situations for *post* and $\widetilde{post}$ are even less clear.

This paper's primary contribution is its systematic study of the powerdomains and Galois connections necessary for sound *under*approximation of all four of the classic predicate transformers. The transformers operate on state sets, and we will require four powerdomains for sound approximation: two are based on "set inclusion" approximation, and two are based on "quantification" approximation. The first two are applied to abstract a logic; the latter two are applied to abstract state-transition functions. Our study of *pre*'s abstraction exposes its fundamental incompleteness, which is repaired by means of a *focus*sed abstraction. We also see why the expected abstractions of $post_f$ and $\widetilde{post}_f$ are *unsound*, and we use the powerdomains to define correct, sound underapproximations (which must be expressed in terms of $pre_{f^{-1}}$ and $\widetilde{pre}_{f^{-1}}$, respectively).

The guiding principle throughout our investigation is that property sets, $[\![\phi]\!]^A$, are *downwards-closed subsets* of A. We tailor the abstractions of the four predicate transformers so that their answers are always downwards-closed sets, and in two cases, this requires that the abstract transition function, $f^\sharp : A \to \mathcal{P}(A)$, used by the predicate transformer must calculate answer sets that are *upwards closed*. We select the appropriate Galois connection with the appropriate powerdomain to abstract f to the appropriate $f^\sharp$.

2 Background

Say that a program's semantics is defined by (the least fixpoint of) a state-transition function, $f : C \to C$. Figure 1 shows a coding of the Collatz function and its state-transition semantics, drawn as a graph. When $f \subseteq C \times C$ is a non-functional state-transition relation, we model it by $f : C \to \mathcal{P}(C)$, and we use this format hereon.

For calculating postconditions, we lift f to $f^* : \mathcal{P}(C) \to \mathcal{P}(C)$ in the usual way: $f^*(S) = \cup_{c \in S} f(c)$. For example, for $odd = \{2n + 1 \mid n \geq 0\}$, the strongest f-postcondition from Figure 1 is $f^*(odd) = \{4, 10, 16, 22, \cdots\}$.

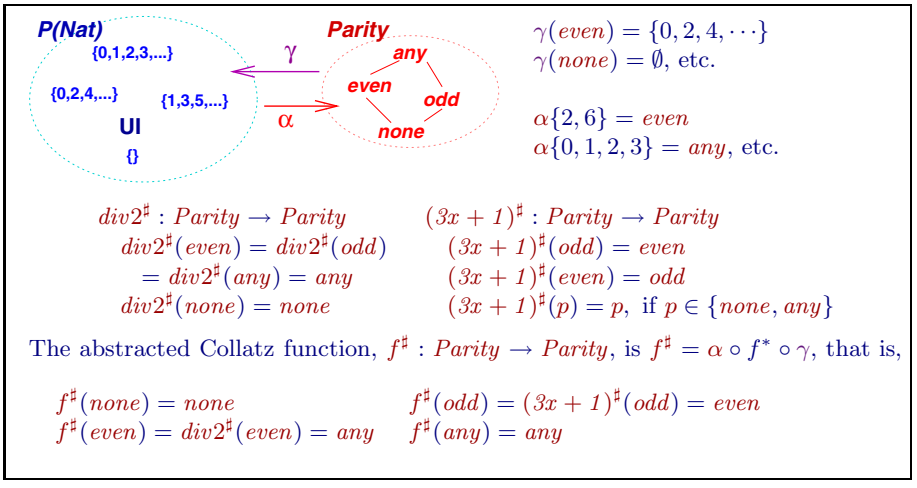


Fig. 2. Parity abstraction of natural numbers and the Collatz function

If a program's state space is "too large," we might abstract it. The abstraction might be a state partitioning [2, 25], but more generally it is a complete lattice, $(A, \sqsubseteq)$, such that there is a Galois connection of the form $(\mathcal{P}(C), \subseteq)\langle\alpha, \gamma\rangle(A, \sqsubseteq)$:[1] Figure 2 abstracts the concrete domain *Nat* in Figure 1 to the complete lattice of parities, *Parity*, which is applied to abstracting the Collatz function.

Each set, $S \subseteq C$, is abstracted by $\alpha(S) \in A$, and each $a \in A$ models the set $\gamma(a) \subseteq C$. The Galois connection *overapproximates* C, because for all $S \subseteq C$, $S \subseteq \gamma(\alpha(S))$.

$f^* : \mathcal{P}(C) \to \mathcal{P}(C)$ is *soundly abstracted* by a monotone function, $f^\sharp : A \to A$, iff $\alpha \circ f^* \sqsubseteq_{\mathcal{P}(C)\to A} f^\sharp \circ \alpha$ iff $f^* \circ \gamma \sqsubseteq_{A\to\mathcal{P}(C)} \gamma \circ f^\sharp$ [7]. We work only with monotone functions. The most precise, sound, abstraction of f^* is $\alpha \circ f^* \circ \gamma$ — see Figure 2 for an example.

Complete lattice A possesses an "internal logic," where $\gamma(a)$ interprets the "assertion" $a \in A$, and for $c \in C$, write $c \models a$ iff $c \in \gamma(a)$. This makes $f^\sharp : A \to A$ a sound postcondition transformer for f: if $c \models a$, then $f(c) \models f^\sharp(a)$. Since γ preserves meets in A,[2] $\sqcap_A$ is "logical conjunction": $c \models a_1 \sqcap a_2$ iff $c \models a_1$ and $c \models a_2$. This logic forms the foundation for static analyses based on A.

There is no guarantee that γ preserves joins; see lattice *Sign* in Figure 3 and consider $0 \models neg \sqcup pos$, which holds even though $0 \not\models neg$ and $0 \not\models pos$. We can improve the situation by building the *disjunctive completion* [7] of A, which is

[1] A *Galois connection* between two complete lattices, P and Q, written $P\langle\alpha, \gamma\rangle Q$, is a pair of monontonic functions, $\alpha : P \to Q$ and $\gamma : Q \to P$, such that $id_{P\to P} \sqsubseteq \gamma \circ \alpha$ and $\alpha \circ \gamma \sqsubseteq id_{Q\to Q}$ [7, 13]. Note that γ's inverse, α, is uniquely defined as $\alpha(p) = \sqcap\{q \mid p \sqsubseteq_P \gamma(q)\}$ and α's inverse is $\gamma(q) = \sqcup\{p \mid \alpha(p) \sqsubseteq_Q q\}$.

[2] That is, for every $T \subseteq A$, $\cap_{a\in T}\gamma(a) = \gamma(\sqcap T)$, which is necessary and sufficient for γ to be the upper adjoint of a Galois connection.

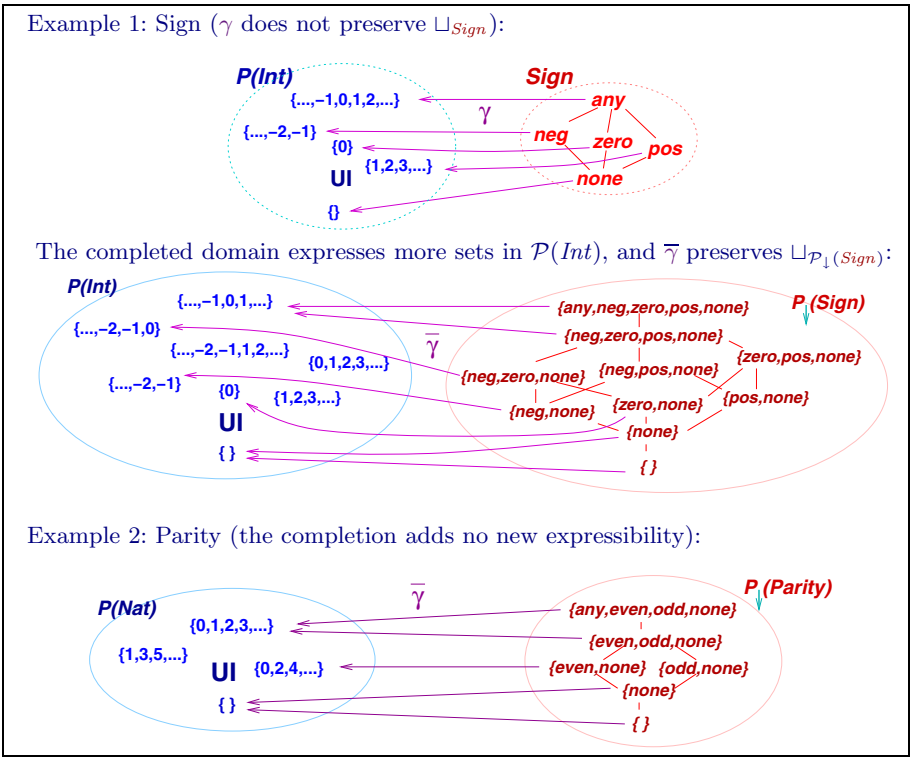


Fig. 3. Two examples of disjunctive completion

$(\mathcal{P}_\downarrow(A), \subseteq)$, that is, all downclosed subsets of A, ordered by subset inclusion.[3] Here is the resulting Galois connection:

$$(\mathcal{P}(C), \subseteq)\langle \overline{\alpha_o}, \overline{\gamma}\rangle(\mathcal{P}_\downarrow(A), \subseteq), \quad \text{where} \quad \begin{array}{l} \overline{\gamma}(T) = \gamma^*(T) = \cup_{a\in A}\gamma(a) \\ \overline{\alpha_o}(S) = \cap\{T \mid S \subseteq \overline{\gamma}(T)\} = \downarrow\{\alpha\{c\} \mid c \in S\} \end{array}$$

See Figure 3. The downclosed sets ensure monotonicity of key functions, like injection, $\{\!|\cdot|\!\} : A \to \mathcal{P}_\downarrow(A)$ (defined $\{\!|a|\!\} = \downarrow\{a\}$ so that $a \sqsubseteq a'$ implies $\{\!|a|\!\} \subseteq \{\!|a'|\!\}$), without changing $\overline{\gamma}$'s image: $\overline{\gamma}(\downarrow S) = \overline{\gamma}(S)$. Because $\overline{\gamma} : \mathcal{P}_\downarrow(A) \to \mathcal{P}(C)$ preserves *both* joins and meets, we have this useful internal logic for $\mathcal{P}_\downarrow(A)$:

$$\phi ::= a \mid \phi_1 \sqcap \phi_2 \mid \phi_1 \sqcup \phi_2$$

$$\begin{array}{c} c \models a \text{ iff } c \in \gamma(a) \\ c \models \phi_1 \sqcap \phi_2 \text{ iff } c \models \phi_1 \text{ and } c \models \phi_2 \\ c \models \phi_1 \sqcup \phi_2 \text{ iff } c \models \phi_1 \text{ or } c \models \phi_2 \end{array}$$

The Galois connection is overapproximating, and we can define a sound abstraction of $f : C \to \mathcal{P}(C)$ in the form, $f^\sharp : A \to \mathcal{P}_\downarrow(A)$; the most precise such abstraction is $f^\sharp_{best} = \overline{\alpha_o} \circ f^* \circ \gamma$. (E.g., in Figure 3, Example 1,

[3] $\mathcal{P}_\downarrow(A) = \{\downarrow T \mid T \subseteq A\}$, where $\downarrow T = \{a \in A \mid \text{there exists } a' \in T,\ a \sqsubseteq_A a'\}$.

$$\mathcal{L} \ni \phi ::= a \mid [f]\phi \mid \langle f\rangle\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$$

$$[\![\cdot]\!] : \mathcal{L} \to \mathcal{P}(C)$$

$$[\![a]\!] = \gamma(a) \qquad [\![\phi_1 \wedge \phi_2]\!] = [\![\phi_1]\!] \cap [\![\phi_2]\!]$$
$$[\![[f]\phi]\!] = \widetilde{pre}_f[\![\phi]\!] \qquad [\![\phi_1 \vee \phi_2]\!] = [\![\phi_1]\!] \cup [\![\phi_2]\!]$$
$$[\![\langle f\rangle\phi]\!] = pre_f[\![\phi]\!]$$

Fig. 4. Precondition logic

$succ^\sharp_{best}(neg) = \{neg, zero, none\}$ (successor), and in Example 2, $div2^\sharp_{best}(even) = \{even, odd, none\}$.)

3 Preconditions

For state-transition function, $f : C \to \mathcal{P}(C)$, S's postcondition is $f^*(S)$, but when f is nondeterministic, there are two useful preconditions:

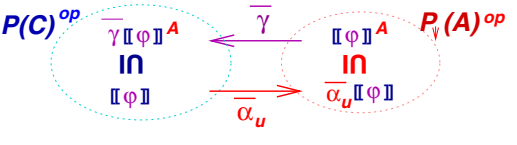
$$\widetilde{pre}_f(S) = \{c \mid \text{for all } c' \in C,\ c' \in f(c) \text{ implies } c' \in S\} = \{c \mid f(c) \subseteq S\}$$
$$pre_f(S) = \{c \mid \text{there exists } c' \in f(c),\ c' \in S\} = \{c \mid f(c) \cap S \neq \emptyset\}.$$

The first computes those states whose f-image lies entirely in S (where the f-image might be empty); the second defines those states whose f-image has at least one state in S. We study the two preconditions in their standard logical representations; Figure 4 gives the syntax and interpretation of the logic. We write $c \models \phi$ iff $c \in [\![\phi]\!]$, e.g., both $12 \models [div2]even$ and $12 \models \langle div2\rangle even$.

It is important to note that the logic in Figure 4 is *not* an internal logic of $\mathcal{P}_\downarrow(A)$ — we have no guarantee that $\overline{\gamma}$ preserves either $\widetilde{pre}_{f^\sharp}$ or $pre_{f^\sharp}$.[4] To check $c \models \phi$ within $\mathcal{P}_\downarrow(A)$, we must abstract each $[\![\phi]\!] \in \mathcal{P}(C)$ to a *sound* $[\![\phi]\!]^A \in \mathcal{P}_\downarrow(A)$: that is, for all $\phi \in \mathcal{L}$, $a \in A$, we require

$$a \in [\![\phi]\!]^A \text{ implies } c \in [\![\phi]\!], \text{ for all } c \in \gamma(a)$$

which is equivalent to requiring that $\overline{\gamma}[\![\phi]\!]^A \subseteq [\![\phi]\!]$. We can insert the latter requirement into the following adjunction situation:



Since $\overline{\gamma} : \mathcal{P}_\downarrow(A) \to \mathcal{P}(C)$ preserves joins as well as meets, we realize the adjunction as the Galois connection, $\mathcal{P}(C)^{op}\langle\overline{\alpha_u}, \overline{\gamma}\rangle\mathcal{P}_\downarrow(A)^{op}$:[5] where $\overline{\gamma}(T) = \cup_{a\in A}\gamma(a)$, as before, and

[4] Giacobazzi, Ranzato, and their colleagues have intensively studied this problem, which is connected to the *backwards completeness* of f [15, 17, 25, 26].

[5] Where $(P, \sqsubseteq_P)^{op}$ is $(P, \sqsupseteq_P)$.

$$\overline{\alpha_u}(S) = \cup\{T \mid S \supseteq \overline{\gamma}(T)\} = \{a \mid \gamma(a) \subseteq S\}.$$

This is an *underapproximating* Galois connection, because $S \supseteq \overline{\gamma}(\overline{\alpha_u}(S))$. We can use it to define this most precise abstraction of $[\![\phi]\!] \in \mathcal{P}(C)$:

$$[\![\phi]\!]^A = \overline{\alpha_u}[\![\phi]\!].$$

But such a definition is not finitely computable, and we desire an inductive definition of $[\![\cdot]\!]^A$. For each logical connective, op_k, interpreted by $g_k : \mathcal{P}(C)^{arity(k)} \to \mathcal{P}(C)$ in the form,

$$[\![op_k(\phi_i)_{i<arity(k)}]\!] = g_k([\![\phi_i]\!])_{i<arity(k)}$$

its most precise, inductively defined underapproximation is

$$[\![op_k(\phi_i)_{i<arity(k)}]\!]^A = g_{k^\flat_{best}}([\![\phi_i]\!]^A)_{i<arity(k)}, \text{ where } g_{k^\flat_{best}} = \overline{\alpha_u} \circ g_k \circ \overline{\gamma}^{arity(k)}$$

Since $g_{k^\flat_{best}}$ as stated is not finitely computable, we search for a sound approximation that is. For example, for logical disjunction we settle for

$$[\![\phi_1 \vee \phi_2]\!]^A = [\![\phi_1]\!]^A \cup [\![\phi_2]\!]^A$$

as a sound underapproximation of

$$\cup^\flat_{best}([\![\phi_1]\!]^A, [\![\phi_2]\!]^A), \text{ where } \cup^\flat_{best} = \overline{\alpha_u} \circ \cup_{\mathcal{P}(C)} \circ (\overline{\gamma} \times \overline{\gamma}).$$

Note that $[\![\phi_1 \vee \phi_2]\!]^A \neq \overline{\alpha_u}[\![\phi_1 \vee \phi_2]\!]$: For example, $any \in \overline{\alpha_u}[\![even \vee odd]\!]$ but $any \notin [\![even \vee odd]\!]^A$, where $[\![even]\!]^A = \overline{\alpha_u}(\gamma(even)) = \{even, none\}$ (and similarly for $[\![odd]\!]^A$).

3.1 Abstracting $\widetilde{pre}_f$

We apply the above-stated techniques to $\widetilde{pre}_f(S) = \{c \mid f(c) \subseteq S\}$ and its logical depiction,

$$[\![[f]\phi]\!] = \widetilde{pre}_f[\![\phi]\!].$$

Using the Galois connections at our disposal, we define $(\widetilde{pre}_f)^\flat_{best} = \overline{\alpha_u} \circ \widetilde{pre}_f \circ \overline{\gamma}$ and compute:

$$\begin{aligned}[\![[f]\phi]\!]^A &= (\widetilde{pre}_f)^\flat_{best}[\![\phi]\!]^A \\ &= \{a \mid \gamma(a) \subseteq \widetilde{pre}_f(\overline{\gamma}[\![\phi]\!]^A)\} \\ &= \{a \mid f^*[\gamma(a)] \subseteq \overline{\gamma}[\![\phi]\!]^A\}.\end{aligned}$$

The definition is not finitely computable, so we propose $\widetilde{pre}_{f^\sharp}$ as a sound underapproximation — since $f^\sharp : A \to \mathcal{P}_\downarrow(A)$ overapproximates f's transitions, $f^\sharp$'s preimages will correspond to supersets of f's preimages. This gives the standard result [8]:

Proposition 1. *If* $f^\sharp : A \to \mathcal{P}_\downarrow(A)$ *is overapproximating sound (that is,* $\alpha \circ f \sqsubseteq_{\mathcal{P}(C)\to A} f^\sharp \circ \alpha$*), then* $\widetilde{pre}_{f^\sharp}$ *is underapproximating sound:* $\overline{\alpha_u}(\widetilde{pre}_f(S)) \supseteq \widetilde{pre}_{f^\sharp}(\overline{\alpha_u}(S))$.

We also have this pleasing result, which shows that the preimage of the best overapproximation equals the best underapproximation of the preimage:

Theorem 2. $\widetilde{pre}_{f^\sharp_{best}} = (\widetilde{pre}_f)^\flat_{best}$, *where* $f^\sharp_{best} = \overline{\alpha_o} \circ f^* \circ \gamma$.

Proof. First, $(\widetilde{pre}_f)^\flat_{best}(T) = \{a \mid f^*(\gamma(a)) \subseteq \overline{\gamma}(T)\}$, and next, $\widetilde{pre}_{f^\sharp_{best}}(T) = \{a \mid \overline{\alpha_o} \circ f^* \circ \gamma(a) \subseteq T\}$. Assume $f^*(\gamma(a)) \subseteq \overline{\gamma}(T)$; then $\overline{\alpha_o} \circ f^* \circ \gamma(a) \subseteq \overline{\alpha_o} \circ \overline{\gamma}(T)$. Since $\overline{\alpha_o}(\overline{\gamma}(T)) \subseteq T$, we are finished. □

Function $f^\sharp_{best} : A \to \mathcal{P}_\downarrow(A)$ has been intensively studied:

$$f^\sharp_{best}(a) \;=\; (\overline{\alpha_o} \circ f^* \circ \gamma)(a) \;=\; \downarrow\{\alpha\{c'\} \mid c \in \gamma(a), c' \in f(c)\}.$$

Cleaveland, Iyer, and Yankelevich [4] and Dams [9] showed that $f^\sharp_{best}$ proves the most $[f]$-properties in the logic in Figure 4.

3.2 Abstracting pre_f

Recall that $pre_f(S) = \{c \mid f(c) \cap S \neq \emptyset\}$. The concrete semantics,

$$[\![\langle f \rangle \phi]\!] = pre_f[\![\phi]\!]$$

defines those states that have a successor state in $[\![\phi]\!]$. We must underapproximate this set, and we define $(pre_f)^\flat_{best} = \overline{\alpha_u} \circ pre_f \circ \overline{\gamma}$. This gives us

$$\begin{aligned}[\![\langle f \rangle \phi]\!]^A &= (pre_f)^\flat_{best}[\![\phi]\!]^A \\ &= \{a \mid \text{for every } c \in \gamma(a), f(c) \cap \overline{\gamma}(T) \neq \emptyset\}.\end{aligned}$$

We search for an approximation of $(pre_f)^\flat_{best}$ expressed in the form, pre_g. Clearly, $pre_{f^\sharp}$, for $f^\sharp : A \to \mathcal{P}_\downarrow(A)$, is unsound, because $f^\sharp(a)$ overestimates a's successors.[6] To underapproximate $f : C \to \mathcal{P}(C)$, we might try $f^\flat_u(a) = (\overline{\alpha_u} \circ f^* \circ \gamma)(a) = \{a' \mid \gamma(a') \subseteq f^*[\gamma(a)]\}$. This looks reasonable, but the consequences are surprising:

Proposition 3. *For* $g : A \to \mathcal{P}_\downarrow(A)$*, for* $T \in \mathcal{P}_\downarrow(T)$*,* $pre_g(T)$ *is an* upwards-closed *set and is not necessarily downwards closed.*

Proof. We first show, if $T \neq \emptyset$, then $pre_g(T) = \{a \mid g(a) \neq \emptyset\}$: For $a \in A$, let $g(a) \neq \emptyset$. Then $\bot_A \in g(a)$, because the set is downwards closed. Since T is downwards closed and nonempty, $\bot_A \in T$ as well. This set is upclosed (because g is monotonic) but need not be downclosed (e.g., when $g(\bot_A) = \emptyset$, where $\gamma(\bot_A) = \emptyset$). When $T = \emptyset$, $pre_g(T) = \emptyset$, which is upclosed. □

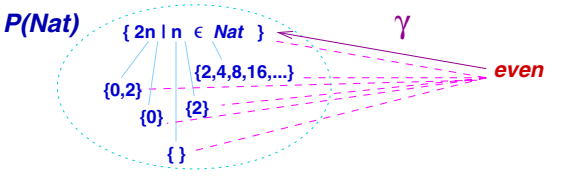
[6] For example, $div2^\sharp_{best}(even) = \downarrow\{even, odd\}$, hence $even \in pre_{div2^\sharp_{best}} \downarrow\{even\}$, yet $6 \in \gamma(even)$ and $div2(6) = \{3\}$.

The result goes against our intuition that propositions are interpreted as downclosed subsets of A. To make $pre_g(T)$ into a downclosed set, it is necessary that $a \sqsubseteq_A a'$ implies $g(a) \supseteq g(a')$, that is, g's codomain must be partially ordered by $\supseteq$. In such a codomain, we must ensure that set injection is monotonic, that is, $a_0 \sqsubseteq_A a_1$ implies $\{\!|a_0|\!\} \supseteq \{\!|a_1|\!\}$, which forces $\{\!|a|\!\} = \uparrow\{a\}$.

For these reasons, we define underapproximating transition functions of arity, $f^\flat : A \to \mathcal{P}_\uparrow(A)$, where $(\mathcal{P}_\uparrow(A), \supseteq)$ is all upclosed subsets of A, ordered by superset inclusion.[7] The following section provides some intuition.

3.3 Interpreting Downclosed and Upclosed Sets

When we use an overapproximating Galois connection, like $\mathcal{P}(Nat)\langle\alpha, \gamma\rangle Parity$, to analyze a program and we compute that the program's output is $even$, we are asserting, "$\forall even$" — all the program's concrete outputs are even-valued. The upper adjoint, $\gamma : Parity \to \mathcal{P}(Nat)$, selects the largest set modelled by $even$,



but the program's output set might be any $S \subseteq Nat$ such that $S \subseteq \gamma(even)$.

This reading applies also to the Galois connection, $\mathcal{P}(Nat)\langle\overline{\alpha_o}, \overline{\gamma}\rangle\mathcal{P}_\downarrow(Parity)$, where a downclosed set like $\{even, odd, none\}$ asserts $\forall\{even, odd, none\} \equiv \forall(even \vee odd \vee none) \equiv \forall(even \vee odd)$ — *all outputs are even- or odd-valued.* The program's output might be any $S \subseteq Nat$ such that $S \subseteq \overline{\gamma}\{even, odd, none\}$.

What is the dual of an overapproximating "universal assertion"? In the previous section, we tried using the Galois connection, $\mathcal{P}(Nat)^{op}\langle\overline{\alpha_u}, \overline{\gamma}\rangle\mathcal{P}_\downarrow(Parity)^{op}$, to underapproximate a program's outputs, but the results were disappointing[8] and dubious (cf. Proposition 3).

The desired dual is an "existential assertion": If an overapproximating $even \in Parity$ asserts "$\forall even$," then an underapproximating $even$ should assert "$\exists even$" — *there exists an even number in the program's outputs.* Now, a function like $3x + 1 : Nat \to \mathcal{P}(Nat)$ can be underapproximated such that $(3x + 1)^\flat(odd) = \{even\}$ — there exists an even number in the function's output.

This idea extends to compound "existential assertions": an *upclosed* set like $\{even, odd, any\}$ asserts $\exists\{even, odd, any\} \equiv \exists even \wedge \exists odd \wedge \exists any \equiv \exists even \wedge \exists odd$ — there exist both even- and odd-valued numbers in the output set.

But there is a problem: How do we concretize an underapproximating set like $\{even\}$ into $\mathcal{P}(Nat)^{op}$? There is no minimal set that contains an even number:

[7] $\mathcal{P}_\uparrow(A) = \{\uparrow T \mid T \subseteq A\}$, where $\uparrow T = \{a \in A \mid \text{ there exists } a' \in T, a' \sqsubseteq_A a\}$.

[8] For example, $3x + 1 : Nat \to \mathcal{P}(Nat)$ is approximated by $(3x + 1)^\flat_{best} = \overline{\alpha}_u \circ (3x + 1)^* \circ \gamma$. Then, $(3x + 1)^\flat_{best}(odd) = \overline{\alpha}_u((3x + 1)^*\{1, 3, 5, \cdots\} = \overline{\alpha}_u\{4, 10, 16, 22, \cdots\} = \{none\}$ (!)

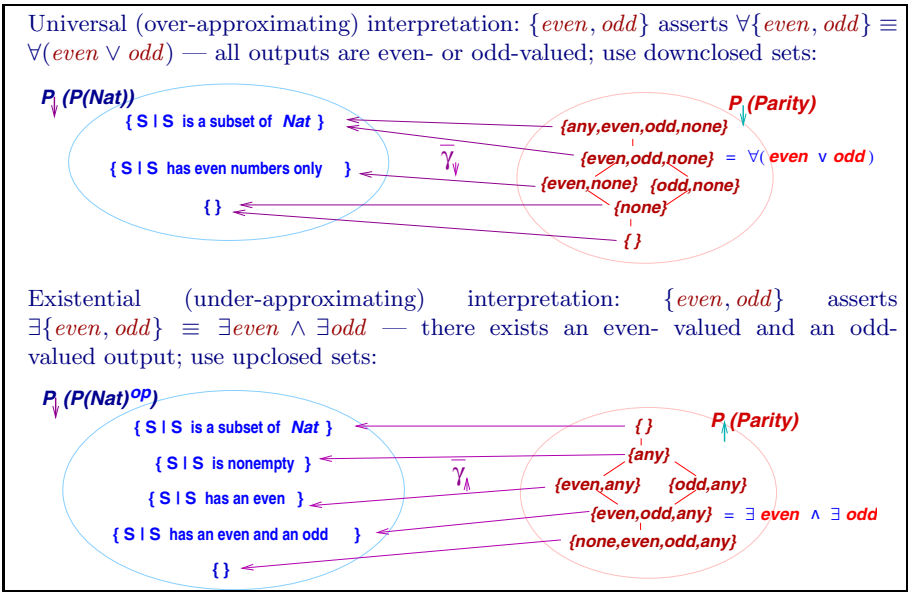
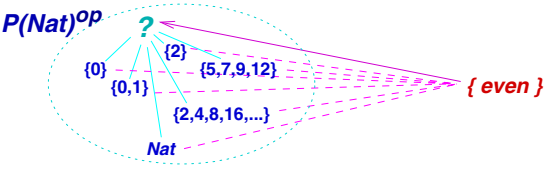


Fig. 5. Powersets for the *Parity* abstraction



Indeed, $\{even\}$'s concretization is not a single set — it must be a *set of sets*:

$$\gamma'\{even\} = \{S \in \mathcal{P}(Nat) \mid S \cap \gamma(even) \neq \emptyset\}.$$

3.4 Upper and Lower Powerset Constructions

To interpret downclosed sets ("universal assertions") and upclosed sets ("existential assertions") we use concrete domains that are *sets of sets*. Figure 5 displays the universal and existential interpretations of sets of parities.

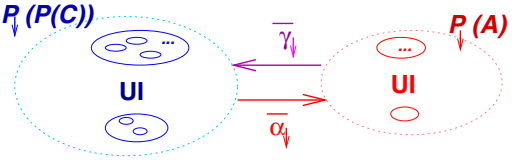
The universal interpretation is developed as follows: For Galois connection, $\mathcal{P}(C)\langle\alpha,\gamma\rangle A$, define $\rho_\downarrow \subseteq \mathcal{P}(C) \times \mathcal{P}_\downarrow(A)$ as

$$S\,\rho_\downarrow\,T \text{ iff for all } c \in S, \text{ there exists } a \in A \text{ such that } c \in \gamma(a).$$

This is the lower ("Hoare") powerdomain ordering, used in denotational semantics [24]. Note that $S\,\rho_\downarrow\,T$ iff $S \subseteq \overline{\gamma}(T)$. Next, define this Galois connection:

$$\mathcal{P}_\downarrow(\mathcal{P}(C))\langle\overline{\alpha_\downarrow},\overline{\gamma_\downarrow}\rangle\mathcal{P}_\downarrow(A) \text{ where } \begin{array}{l}\overline{\gamma_\downarrow}(T) = \{S \mid S\,\rho_\downarrow\,T\} \\ \overline{\alpha_\downarrow}(\overline{S}) = \cap\{T \mid \text{for all } S \in \overline{S}, S\,\rho_\downarrow\,T\}\end{array}$$

$\overline{\gamma_{\downarrow}}(T)$ concretizes T to all the sets covered by T — It is an *over*approximation of an *over*approximation:



Because $S \, \rho_{\downarrow} \, T$ iff $S \subseteq \overline{\gamma}(T)$, no new expressibility is gained by using the new Galois connection over $\mathcal{P}(C)\langle\overline{\alpha_o}, \overline{\gamma}\rangle\mathcal{P}_{\downarrow}(A)$: for all $f : C \to \mathcal{P}(C)$, $f^{\sharp}_{best} : A \to \mathcal{P}_{\downarrow}(A)$ is $\overline{\alpha_{\downarrow}} \circ (\{\!|\cdot|\!\} \circ f)^* \circ \gamma = \overline{\alpha_o} \circ f^* \circ \gamma$ [30,31]. But we might argue nonetheless that this Galois connection "truly defines" the sound overapproximation of f.

On the other hand, the existential interpretation is truly new; it uses the *Smyth-powerdomain ordering* from denotational semantics [24]: Define $\rho_{\uparrow} \subseteq \mathcal{P}(C) \times \mathcal{P}_{\uparrow}(A)$ as

$$S \, \rho_{\uparrow} \, T \text{ iff for all } a \in T, \text{ there exists } c \in S \text{ such that } c \in \gamma(a).$$

That is, every $a \in T$ is a witness to some $c \in S$. Note that $S \, \rho_{\uparrow} \, T$ iff for all $a \in T$, $\gamma(a) \cap S \neq \emptyset$. Next, define this Galois connection:

$$\mathcal{P}_{\downarrow}(\mathcal{P}(C)^{op})\langle\overline{\alpha_{\uparrow}}, \overline{\gamma_{\uparrow}}\rangle\mathcal{P}_{\uparrow}(A) \text{ where } \begin{array}{l}\overline{\gamma_{\uparrow}}(T) = \{S \mid S \, \rho_{\uparrow} \, T\} \\ \overline{\alpha_{\uparrow}}(\overline{S}) = \cup\{T \mid \text{for all } S \in \overline{S}, S \, \rho_{\uparrow} \, T\}\end{array}$$

$\overline{\gamma_{\uparrow}}(T)$ concretizes T to all sets that T "witnesses" — It is an *over*approximation of an *under*approximation:

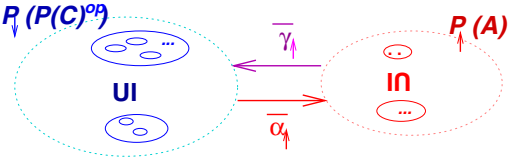


Figure 6 summarizes the Galois connections developed so far.

3.5 Properties of $pre_{f^{\flat}}$

We underapproximate $f : C \to \mathcal{P}(C)$ by a sound $f^{\flat} : A \to \mathcal{P}_{\uparrow}(A)$. We define $f^{\flat}_{best} : A \to \mathcal{P}_{\uparrow}(A)$ as

$$\begin{array}{rl} f^{\flat}_{best}(a) & = (\overline{\alpha_{\uparrow}} \circ (\{\!|\cdot|\!\} \circ f)^* \circ \gamma)(a) \\ & = \uparrow\{\alpha(S) \mid \text{for all } c \in \gamma(a),\ f(c) \cap S \neq \emptyset\} \\ & = \{a' \mid \text{for all } c \in \gamma(a),\ f(c) \cap \gamma(a') \neq \emptyset\} \end{array}$$

We have that $pre_{f^{\flat}_{best}}(T)$ is downclosed and also that

Proposition 4. *$pre_{f^{\flat}_{best}}$ is* sound*: $pre_{f^{\flat}_{best}}(T) \subseteq (\overline{\alpha_u} \circ pre_f \circ \overline{\gamma})(T)$.*

Figure 7 shows the abstracted precondition logic. Cleaveland, Iyer, and Yankelevich [4], Dams, et al. [10], and Schmidt [30] showed that $pre_{f^{\flat}_{best}}$ proves the most sound $\langle f \rangle$-properties in the logic of Figure 4.

	overapproximation	**underapproximation**
set inclusion	$\mathcal{P}(C)\langle\overline{\alpha_o},\overline{\gamma}\rangle\mathcal{P}_\downarrow(A)$ where $\overline{\gamma}(T) = \cup_{a\in A}\gamma(a)$ $\overline{\alpha_o}(S) = \downarrow\{\alpha\{c\} \mid c \in S\}$	$\mathcal{P}(C)^{op}\langle\overline{\alpha_u},\overline{\gamma}\rangle\mathcal{P}_\downarrow(A)^{op}$ where $\overline{\gamma}(T) = \cup_{a\in A}\gamma(a)$ $\overline{\alpha_u}(S) = \{a \mid \gamma(a) \subseteq S\}$
quantification	$\mathcal{P}_\downarrow(\mathcal{P}(C))\langle\overline{\alpha_\downarrow},\overline{\gamma_\downarrow}\rangle\mathcal{P}_\downarrow(A)$ where $\overline{\gamma_\downarrow}(T) = \{S \mid S\,\rho_\downarrow\,T\}$ $\overline{\alpha_\downarrow}(S) = \cap\{T \mid$ for all $S \in \overline{S}$, $S\,\rho_\downarrow\,T\}$	$\mathcal{P}_\downarrow(\mathcal{P}(C)^{op})\langle\overline{\alpha_\uparrow},\overline{\gamma_\uparrow}\rangle\mathcal{P}_\uparrow(A)$ where $\overline{\gamma_\uparrow}(T) = \{S \mid S\,\rho_\uparrow\,T\}$ $\overline{\alpha_\uparrow}(S) = \cup\{T \mid$ for all $S \in \overline{S}$, $S\,\rho_\uparrow\,T\}$

where $\rho_\downarrow \subseteq \mathcal{P}(C) \times \mathcal{P}_\downarrow(A)$ is defined
$S\,\rho_\downarrow\,T$ iff for all $c \in S$, there exists $a \in A$ such that $c \in \gamma(a)$
and $\rho_\uparrow \subseteq \mathcal{P}(C) \times \mathcal{P}_\uparrow(A)$ is defined
$S\,\rho_\uparrow\,T$ iff for all $a \in T$, there exists $c \in S$ such that $c \in \gamma(a)$.

Fig. 6. Summary of Galois connections derived from $\mathcal{P}(C)\langle\alpha,\gamma\rangle A$

$$[\![\cdot]\!] : \mathcal{L} \to \mathcal{P}_\downarrow(A)$$

$$[\![a]\!]^A = \overline{\alpha_u}(\gamma(a))$$
$$[\![[f]\phi]\!]^A = \widetilde{pre}_{f^\sharp_{best}}[\![\phi]\!]^A \qquad [\![\phi_1 \wedge \phi_2]\!]^A = [\![\phi_1]\!]^A \cap [\![\phi_2]\!]^A$$
$$[\![\langle f\rangle\phi]\!]^A = pre_{f^\flat_{best}}[\![\phi]\!]^A \qquad [\![\phi_1 \vee \phi_2]\!]^A = [\![\phi_1]\!]^A \cup [\![\phi_2]\!]^A$$

Fig. 7. The abstracted precondition logic

3.6 Incompleteness and *Focus*

Although $f^\flat_{best}$ is the most precise (maximal) sound underapproximation, there is no guarantee that $pre_{f^\flat_{best}}$ equals $(pre_f)^\flat_{best} = \overline{\alpha_u} \circ pre_f \circ \overline{\gamma}$.

Here is a counterexample: Consider the *Parity* abstract domain and the assertion, $\langle div2\rangle(even \vee odd)$. This assertion holds for all $c \in \gamma(even)$, and indeed, for the downclosed set, $T_0 = \{even, odd, none\}$, we have that $even \in (pre_{div2})^\flat_{best}(T_0)$. But $div2^\flat_{best}(even) = \{any\}$, and $any \notin T_0$, implying that $even \notin pre_{div2^\flat_{best}}(T_0)$.

The underlying issue is the well-known incompleteness of disjunction in approximation [8]; here, $any \notin T_0$, even though $\gamma(any) \subseteq \overline{\gamma}(T_0)$. The standard repair is a *focus* operation, as used in the TVLA system [28], and in disjunctive transition systems [11, 14, 20], and in tree automata [12], to "split" values like *any* into more-precise cases that "cover" all of $\gamma(any)$. For the example, $T_1 = \{even, odd\}$ is a *focus set* that covers *any* because $\gamma(any) \subseteq \overline{\gamma}(T_1)$. Since both $even \in T_0$ and $odd \in T_0$, we conclude *any* "belongs" to T_0 as well.

The domain-theoretic connection is clear: A downclosed set, like $T_0 = \{even, odd, none\}$ should be read as the quantified disjunction, $\forall(even \vee odd \vee none)$, and a *focus* operation helps validate the disjunction.

Definition 5. *For $a \in A$, define $focus(a) = \{U \subseteq A \mid \gamma(a) \subseteq \overline{\gamma}(U)\}$, and define $pre^{focus}_{f^\flat}(T) = \{a \mid$ there exist $a' \in f^\flat(a)$ and $U \in focus(a')$ such that $U \subseteq T\}$.*

Evidently, $pre^{focus}_{f^\flat}(T) = \{a \mid \text{exists } a' \in f^\flat(a),\ T \in focus(a')\}$. Definition 5 yields the expressivity and completeness results immediately below, but of course the selection of a specific focus set is a critical pragmatic decision.[9]

Proposition 6. *For all $T \in \mathcal{P}_\downarrow(A)$, $pre_{f^\flat_{best}}(T) \subseteq pre^{focus}_{f^\flat_{best}}(T) \subseteq (pre_f)^\flat_{best}(T)$.*

Proof. The first inclusion follows by choosing $\{a'\} \in focus(a')$. For the second inclusion, assume there exists $a' \in f^\flat_{best}(a)$ such that $\gamma(a') \subseteq \overline{\gamma}(T)$. Since $a' \in f^\flat_{best}(a)$, this implies for all $c \in \gamma(a)$, $f(c) \cap \gamma(a') \neq \emptyset$. Since $\gamma(a') \subseteq \overline{\gamma}(T)$, we have the result. □

Theorem 7. *If $\gamma : A \to \mathcal{P}(C)$ preserves joins, then $pre^{focus}_{f^\flat_{best}}(T) = (pre_f)^\flat_{best}(T)$.*

Proof. We have $\subseteq$; to show $\supseteq$, assume that some $a_0 \in (pre_f)^\flat_{best}(T)$, that is, for all $c \in \gamma(a_0)$, $f(c) \cap \overline{\gamma}(T) \neq \emptyset$. We must show that there exists $a' \in f^\flat_{best}(a_0)$ such that $T \in focus(a')$.

Define $T_{a_0} = \{a_c \in T \mid \text{exists } c \in \gamma(a_0), f(c) \cap \gamma(a_c) \neq \emptyset\}$, and define $a' = \sqcup T_{a_0}$. Immediately, we can conclude that $a' \in f^\flat_{best}(a_0)$. Now we must show $T \in focus(a')$, that is, $\gamma(a') \subseteq \overline{\gamma}(T)$.

For each $a_c \in T_{a_0}$, we have that $\gamma(a_c) \subseteq \overline{\gamma}(T)$, hence $(\sqcup_{a_c \in T_{a_0}} \gamma(a_c)) \subseteq \overline{\gamma}(T)$. Since γ preserves joins, we have that $\gamma(a') = \gamma(\sqcup T_{a_0}) \subseteq \overline{\gamma}(T)$. □

Partition domains [25] are the standard example where γ preserves joins: given state set C, partition P, and $\delta : P \to \mathcal{P}(C)$ that maps each partition to its members, the generated partition domain is $(\mathcal{P}(P), \subseteq)$, where $\gamma = \delta^*$.

If γ preserves joins, then we know that the first inclusion in Proposition 6 can be proper (e.g., $T_0 = \{even, odd, none\}$); if γ fails to preserve joins, there can be a T that makes the first inclusion an equality and the second one proper, because there is some $c \in \overline{\gamma}(T)$ that cannot be "isolated" by a focus set [16].

4 Postconditions

Earlier, we noted that $f^* : \mathcal{P}(C) \to \mathcal{P}(C)$, for $f : C \to \mathcal{P}(C)$, defines f's postcondition transformer and $f^\sharp : A \to A$ is its sound overapproximation. For example,

[9] Focus sets are also known as *must hyper transitions* [32], and there is a dual notion of *may hyper transitions*, which prove useful when $\gamma : A \to C$ is *not* the upper adjoint of a Galois connection [33].

$$[\![\cdot]\!] : \mathcal{L} \to \mathcal{P}(C)$$

$$[\![a]\!] = \gamma(a) \qquad [\![\phi_1 \wedge \phi_2]\!] = [\![\phi_1]\!] \cap [\![\phi_2]\!]$$
$$[\![\overline{[f]}\phi]\!] = \widetilde{post}_f [\![\phi]\!] \qquad [\![\phi_1 \vee \phi_2]\!] = [\![\phi_1]\!] \cup [\![\phi_2]\!]$$
$$[\![\overline{\langle f \rangle}\phi]\!] = post_f [\![\phi]\!]$$

Fig. 8. The postcondition logic

Available Expressions:

$AE(p) =_{gfp} \bigcap_{p' \in \overline{pred}\ p} ((AE(p') \cap notModified(p')) \cup Gen(p'))$

$isAvail(e) = \nu Z.\ \overline{[p]}((Z \wedge \neg isModified(e)) \vee isGen(e))$

Live Variables:

$LV(p) =_{lfp}\ Used(p) \cup (notModified(p) \cap (\bigcup_{p' \in succ\ p} LV(p')))$

$isLive(x) = \mu Z.\ isUsed(x) \vee (\neg isModified(x) \wedge (\langle p \rangle Z))$

Very Busy Expressions:

$VBE(p) =_{gfp}\ Used(p) \cup (notModified(p) \cap (\bigcap_{p' \in succ\ p'}\ VBE(p')))$

$isVBE(e) = \nu Z.\ isUsed(e) \vee (\neg isModified(e) \wedge [p]Z)$

Reaching Definitions:

$RD(p) =_{lfp} \bigcup_{p' \in \overline{pred}\ p} ((RD(p') \cap notModified(p')) \cup Defined(p'))$

$isReaching(d) = \mu Z.\ \overline{\langle p \rangle}((Z \wedge \neg isModified(d)) \vee isDefined(d))$

Fig. 9. Data-flow analyses and their encodings in logical form [29]

$succ^*\{1,3,5,\cdots\} = \{2,4,6,\cdots\}$ and $succ^\sharp_{best}(odd) = even$, where $succ^\sharp_{best} = \alpha \circ succ^* \circ \gamma$ is the strongest postcondition transformer for Galois connection, $\mathcal{P}(Nat)\langle\alpha,\gamma\rangle Parity$. Similarly, from $f : C \to \mathcal{P}(C)$ and $\mathcal{P}(C)\langle\overline{\alpha_o},\overline{\gamma}\rangle\mathcal{P}_\downarrow(A)$, we define $f^\sharp_{best} : A \to \mathcal{P}_\downarrow(A)$ as $f^\sharp_{best} = \overline{\alpha_o} \circ f^* \circ \gamma$.

Since $f : C \to \mathcal{P}(C)$ denotes a nondeterministic transition relation, there are two variants of logical postcondition:

$$post_f(S) = \{d \mid \text{there exists } c \in S, d \in f(c)\} = f^*(S)$$
$$\widetilde{post}_f(S) = \{d \mid \text{for all } c \in C,\ d \in f(c) \text{ implies } c \in S\}.$$

$d \in post_f(S)$ means that *one* of d's immediate f-predecessors belongs to S; $d \in \widetilde{post}_f(S)$ means that *all* of d's immediate f-predecessors belong to S. These transformers have a natural place in a logic; see Figure 8.

Steffen [34] showed how to use the $\overline{[f]}$- and $\overline{\langle f \rangle}$-modalities to define forwards data-flow analyses, and Schmidt [29] applied Steffen's ideas, as displayed in Figure 9, to write mu-calculus formulas [19] that define the naive but standard forwards and backwards data-flow analyses on annotated control-flow graphs, where $p \in ProgramPoint$.

For the purposes of program validation and code improvement, the abstractions of the two *post*-modalities must be *underapproximating*.[10] Clearly, underapproximating the logical interpretation of the postcondition transformers is different from overapproximating a transition function's postcondition, and the following proposition indicates how careful we must be:

Proposition 8. *Let* $f : D \to \mathcal{P}_\delta(D)$, *where* $\delta \in \{\downarrow, \uparrow\}$. *Let* $\tilde{\downarrow} = \uparrow$ *and* $\tilde{\uparrow} = \downarrow$.

Then, for all $S \in \mathcal{P}(D)$,
- $\widetilde{pre}_f(S) \in \mathcal{P}_\delta(D)$
- $pre_f(S) \in \mathcal{P}_{\tilde{\delta}}(D)$
- $post_f(S) \in \mathcal{P}_\delta(D)$
- $\widetilde{post}_f(S) \in \mathcal{P}_{\tilde{\delta}}(D)$.

Proof. Recall that $f : D \to \mathcal{P}_\delta(D)$. When reasoning about f, we use the notation, $\leq_\delta$, to denote $\sqsubseteq_D$, when $\delta = \downarrow$, and $\sqsupseteq_D$, when $\delta = \uparrow$. We have that f is monotonic iff $c \leq_\delta d$ implies $f(c) \subseteq f(d)$. Here are the four proofs:

$\widetilde{pre}_f(S)\{c \mid f(c) \subseteq S\}$: If $f(c) \subseteq S$ and $d \leq_\delta c$, then $f(d) \subseteq f(c)$, by f's monotonicity.

$pre_f S = \{c \mid f(c) \cap S \neq \emptyset\}$: If $f(c) \cap S \neq \emptyset$ and $c \leq_\delta d$ (that is, $d \leq_{\tilde{\delta}} c$), then $f(c) \subseteq f(d)$, implying $f(d) \cap S \neq \emptyset$.

$post_f(S) = \{d \mid \text{exists } c \in S, d \in f(c)\}$: If there exists some $c \in S$ such that $d \in f(c)$, and then $d' \leq_\delta d$, then $d' \in f(c)$, because f's codomain is $\mathcal{P}_\delta(D)$.

$\widetilde{post}_f S = \{d \mid \text{for all } c \in D, d \in f(c) \text{ implies } c \in S\}$: Say that $d \leq_\delta d'$ (that is, $d' \leq_{\tilde{\delta}} d$) and $d \in \widetilde{post}_f S$. For $c' \in D$, say that $d' \in f(c')$ — we must show that $c' \in S$, as well. Since $d \leq_\delta d'$, this means $d \in f(c')$, because f's codomain is $\mathcal{P}_\delta(D)$. This places $c' \in S$. □

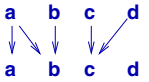
The proposition confirms why $\widetilde{pre}_{f^\sharp}$ and $pre_{f^\flat}$ correctly underapproximated $\widetilde{pre}_f$ and pre_f – the abstract transformers generated downclosed sets as answers. The proposition also makes clear that $post_{f^\flat}$ and $\widetilde{post}_{f^\sharp}$ are *unacceptable* as underapproximations, because they generate *upclosed* sets as answers:

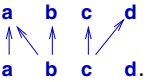
$$\text{for } f^\flat : A \to \mathcal{P}_\uparrow(A), \quad post_{f^\flat} : \mathcal{P}_\downarrow(A) \to \mathcal{P}_\uparrow(A)$$
$$\text{for } f^\sharp : A \to \mathcal{P}_\downarrow(A), \quad \widetilde{post}_{f^\sharp} : \mathcal{P}_\downarrow(A) \to \mathcal{P}_\uparrow(A).$$

Unfortunately, starting from $\gamma : A \to \mathcal{P}(C)$ and $f : C \to \mathcal{P}(C)$, there is no nontrivial overapproximating $f^\sharp : A \to \mathcal{P}_\uparrow(A)$ (because, for all $f^\sharp(a) \neq \emptyset$, upclosure implies that $\top_A \in f^\sharp(a)$, implying that $\overline{\gamma}(f^\sharp(a)) = C$). A similar problem arises in the search for a nontrivial underapproximating $f^\flat : A \to \mathcal{P}_\downarrow(A)$.[11] There is a repair, however. If we draw

[10] For performing data-flow analysis, one usually abstracts a program, f, to its control-flow graph, $f^\sharp_{cfg}$. A *naive* application of the four analyses in Figure 9 to $f^\sharp_{cfg}$ gives *underapproximating* calculations of available expressions and very-busy expressions and *overapproximating* calculations of reaching definitions and live variables (but see [8] for clarification). The set-complements of the latter two calculations — "not-reaching" and "not-live," respectively — are used in practice.

[11] In contrast, both $post_{f^\sharp}$ and $\widetilde{post}_{f^\flat}$ are well defined *overapproximations* of the two postcondition transformers!

$f : C \to \mathcal{P}(C)$ as 

, then $f^{-1} : C \to \mathcal{P}(C)$ is 

That is, $f^{-1}(c) = \{d \mid c \in f(d)\}$.

Proposition 9. *[21]:* $(f^{-1})^{-1} = f$, $post_f = pre_{f^{-1}}$, *and* $\widetilde{post}_f = \widetilde{pre}_{f^{-1}}$.

Proposition 10. *For* $f : A \to \mathcal{P}_\delta(A)$, $\delta \in \{\downarrow, \uparrow\}$, $f^{-1} : A \to \mathcal{P}_{\tilde{\delta}}(A)$ *is well defined and monotonic.*

Proof. $f^{-1}(a) = \{a' \mid a \in f(a')\}$. We use $\leq_\delta$ to denote $\sqsubseteq_D$, when $\delta = \downarrow$, and to denote $\sqsupseteq_D$, when $\delta = \uparrow$. First, note that $f : A \to \mathcal{P}_\delta(A)$ is monotonic iff $c \leq_\delta d$ implies $f(c) \subseteq f(d)$.

f^{-1}'s image are $\tilde{\delta}$-closed sets: Say that $a' \in f^{-1}(a)$, that is, $a \in f(a')$ and say that $a' \leq_\delta b'$. We must show $a \in f(b')$ — this follows from $f(a') \subseteq f(b')$.

f^{-1} is monotonic: Assume $a \leq_\delta b$; we must show $f^{-1}(a) \sqsubseteq_{\mathcal{P}_{\tilde{\delta}}(A)} f^{-1}(b)$. First, we show that $f^{-1}(b) \subseteq f^{-1}(a)$: Assume $x \in f^{-1}(b)$, that is, $b \in f(x)$. Then $\delta b \subseteq f(x)$, because $f(x)$ is a δ-closed set. This implies $a \in f(x)$ as well, that is, $x \in f^{-1}(a)$. The monotonicity of f^{-1} follows, because $\mathcal{P}_{\tilde{\delta}}(A)$ uses the inverse ordering used by $\mathcal{P}_\delta(A)$. □

4.1 Abstracting $post_f$ and $\widetilde{post}_f$

With Propositions 8, 9, and 10 in hand, we can define sound underapproximations for the two postcondition transformers. For $post_f$, we have

$$[\![\overline{\langle f \rangle}\phi]\!] \;=\; post_f[\![\phi]\!] \;=\; pre_{f^{-1}}[\![\phi]\!]$$

where $f^{-1} : C \to \mathcal{P}(C)$. The inductively defined underapproximation is

$$[\![\overline{\langle f \rangle}\phi]\!]A = (\overline{\alpha_u} \circ pre_{f^{-1}} \circ \overline{\gamma})[\![\phi]\!]^A.$$

By Proposition 4, this is soundly underapproximated by

$$[\![\overline{\langle f \rangle}\phi]\!]^A = pre_{(f^{-1})^\flat_{best}}[\![\phi]\!]^A,$$
$$\text{where } (f^{-1})^\flat_{best} : A \to \mathcal{P}_\uparrow(A) \text{ is } (f^{-1})^\flat_{best} = \overline{\alpha_\uparrow} \circ (\{\!|\cdot|\!\} \circ f^{-1})^* \circ \gamma.$$

The same development applied to $\widetilde{post}_f$ yields

$$[\![\overline{[f]}\phi]\!] \;=\; \widetilde{post}_f[\![\phi]\!] \;=\; \widetilde{pre}_{f^{-1}}[\![\phi]\!].$$

By Theorem 2, the most precise underapproximation is

$$[\![\overline{\langle f \rangle}\phi]\!]^A = (\overline{\alpha_u} \circ \widetilde{pre}_{f^{-1}} \circ \overline{\gamma})[\![\phi]\!]^A = \widetilde{pre}_{(f^{-1})^\sharp_{best}}[\![\phi]\!]^A,$$
$$\text{where } (f^{-1})^\sharp_{best} : A \to \mathcal{P}_\downarrow(A) \text{ is } (f^{-1})^\sharp_{best} = \overline{\alpha_o} \circ (f^{-1})^* \circ \gamma.$$

This approach of computing postconditions as preconditions of inverted state-transition relations is implemented in Steffen's fixpoint analysis machine [35].

5 Related Work

Abstraction of predicate transformers begin in Cousot's thesis [5]; details were spelled out in a subsequent series of papers by Cousot and Cousot [6–8] and applied by Bourdoncle to *abstract debugging* [1], which was generalized by Massé [22, 23]. Loiseaux, et al. [21] formalized underapproximation of $\widetilde{pre}$.

Cleaveland, Iyer, and Yankelevich [4], Dams [9], and Dams's colleagues [10] were the first to study underapproximations of *pre*. Studies of precision of such approximations were undertaken by Giacobazzi, Ranzato, and Scozzari [17], who developed completeness properties, and by Ranzato and Tapparo [25–27], who studied completeness of *pre* for state-partition abstract domains. The incompleteness of *pre* has been addressed by Larsen and Xinxin [20], Dams and Namjoshi [11, 12], and Shoham and Grumberg [32]. Steffen [34, 35] was the first to connect data-flow analysis to forwards-backwards temporal-logic modalities, and this connection provides the application area for the results in this paper.

Acknowledgments

Allen Emerson and Kedar Namjoshi let me present early thoughts on this work at VMCAI'06, and Kedar asked several key questions. Michael Huth and Dennis Dams provided valuable advice within earlier collaborations. The referees gave many useful comments.

References

1. F. Bourdoncle. Abstract debugging of higher-order imperative languages. In *Proc. ACM Conf. PLDI*, pages 46–55, 2003.
2. E.M. Clarke, O. Grumberg, and D.E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, 1994.
3. E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 2000.
4. R. Cleaveland, P. Iyer, and D. Yankelevich. Optimality in abstractions of model checking. In *Proc. SAS'95*, LNCS 983. Springer, 1995.
5. P. Cousot. *Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique de programmes*. PhD thesis, University of Grenoble, 1978.
6. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs. In *Proc. 4th ACM Symp. POPL*, pages 238–252, 1977.
7. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proc. 6th ACM Symp. POPL*, pages 269–282, 1979.
8. P. Cousot and R. Cousot. Temporal abstract interpretation. In *Proc. 27th ACM Symp. on Principles of Programming Languages*, pages 12–25. ACM Press, 2000.
9. D. Dams. *Abstract interpretation and partition refinement for model checking*. PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 1996.
10. D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Trans. Prog. Lang. Systems*, 19:253–291, 1997.
11. D. Dams and K. Namjoshi. The existence of finite abstractions for branching time model checking. In *Proc. IEEE Symp. LICS'04*, pages 335–344, 2004.

12. D. Dams and K. Namjoshi. Automata as abstractions. In *Proc. VMCAI'05*, LNCS 3385, pages 216–232. Springer-Verlag, 2005.
13. B.A. Davey and H.A Priestly. *Introduction to Lattices and Order, 2d ed.* Cambridge Univ. Press, 2002.
14. H. Fecher and M. Huth. Complete abstractions through extensions of disjunctive modal transition systems. Technical Report 0604, Institut für Informatik und Praktische Mathematik der Christian-Albrechts-Universitaet zu Kiel, 2005.
15. R. Giacobazzi and E. Quintarelli. Incompleteness, counterexamples, and refinements in abstract model checking. In *Static Analysis Symposium*, LNCS 2126, pages 356–373. Springer Verlag, 2001.
16. R. Giacobazzi and F. Ranzato. The reduced relative power operation on abstract domains. *Theoretical Comp. Sci.*, 216:159–211, 1999.
17. R. Giacobazzi, F. Ranzato, and F. Scozzari. Making abstract interpretations complete. *J. ACM*, 47:361–416, 2000.
18. R. Heckmann. *Power domain constructions*. PhD thesis, Univ. Saarbrücken, 1990.
19. K. Larsen. Proof systems for Hennessy-Milner logic with recursion. In *CAAP88*, LNCS 299. Springer-Verlag, 1988.
20. K.G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *LICS'90*, 1990.
21. C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for verification of concurrent systems. *Formal Methods in System Design*, 6:1–36, 1995.
22. D. Massé. Combining backward and forward analyses of temporal properties. In *Proc. PADO'01*, LNCS 2053, pages 155–172. Springer, 2001.
23. D. Massé. Property checking driven abstract interpretation-based static analysis. In *Proc. VMCAI'03*, LNCS 2575, pages 56–69. Springer, 2003.
24. G. Plotkin. Domains. Lecture notes, Univ. Pisa/Edinburgh, 1983.
25. F. Ranzato and F. Tapparo. Strong preservation as completeness in abstract interpretation. In *Proc. ESOP*, LNCS 2986, pages 18–32. Springer, 2004.
26. F. Ranzato and F. Tapparo. An abstract interpretation-based refinement algorithm for strong preservation. In *TACAS'05*, LNCS 3440, pages 140–156. Springer, 2005.
27. F. Ranzato and F. Tapparo. Strong preservation of temporal fixpoint-based operators by abstract interpretation. In *Proc. Conf. VMCAI'06*, LNCS 3855, pages 332–347. Springer Verlag, 2006.
28. M. Sagiv, T. Reps, and R. Wilhelm. Parametric shape analysis via 3-valued logic. *ACM TOPLAS*, 24:217–298, 2002.
29. D.A. Schmidt. Data-flow analysis is model checking of abstract interpretations. In *Proc. 25th ACM Symp. on Principles of Prog. Languages*. ACM Press, 1998.
30. D.A. Schmidt. Closed and logical relations for over- and under-approximation of powersets. In *Proc. SAS'04*, LNCS 3148, pages 22–37. Springer, 2004.
31. D.A. Schmidt. A calculus of logical relations for over- and underapproximating static analyses. *Science of Computer Programming*, in press.
32. S. Shoham and O. Grumberg. Monotonic abstraction refinement for CTL. In *TACAS'04*. Springer LNCS, 2004.
33. S. Shoham and O. Grumberg. 3-valued abstraction: More precision at less cost. In *LICS'06*, 2006.
34. B. Steffen. Generating data-flow analysis algorithms for modal specifications. *Science of Computer Programming*, 21:115–139, 1993.
35. B. Steffen, A. Classen, M. Klein, J. Knoop, and T. Margaria. The fixpoint analysis machine. In *Proc. CONCUR'95*, LNCS 962, pages 72–87. Springer, 1995.