« Forward Non-relational Finitary Static Analysis, Part II »

Patrick Cousot

Jerome C. Hunsaker Visiting Professor
Massachusetts Institute of Technology
Department of Aeronautics and Astronautics

cousot@mit.edu
www.mit.edu/~cousot

Course 16.399: “Abstract interpretation”

http://web.mit.edu/afs/athena.mit.edu/course/16/16.399/www/


THESE

présentée à

Université Scientifique et Médicale de Grenoble
Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR ES SCIENCES MATHEMATIQUES

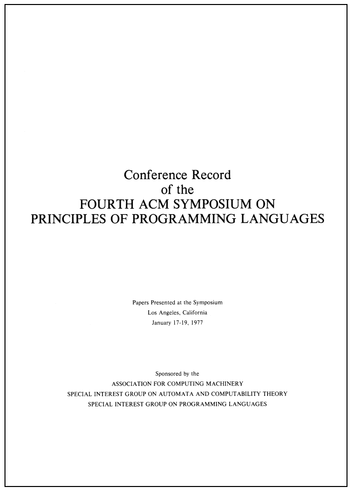
par

Patrick COUSOT

METHODES ITERATIVES DE CONSTRUCTION
ET D'APPROXIMATION DE PCINTS FIXES
D'OPERATEURS MONOTONES SUR UN TREILLIS,
ANALYSE SEMANTIQUE DES PROGRAMMES.

Thèse soutenue le 21 mars 1978 devant la Commission d'Examen :

Président : L. BOLLIET
Examinateurs : C. BENZAKEN
Ph. JORRAND
B. LORHO
C. PAIR
F. ROBERT
M. SINTZOFF


Conference Record
of the
FOURTH ACM SYMPOSIUM ON
PRINCIPLES OF PROGRAMMING LANGUAGES

Papers Presented at the Symposium
Los Angeles, California
January 17-19, 1977

Sponsored by the
ASSOCIATION FOR COMPUTING MACHINERY
SPECIAL INTEREST GROUP ON AUTOMATA AND COMPUTABILITY THEORY
SPECIAL INTEREST GROUP ON PROGRAMMING LANGUAGES

A few definitions from previous lectures...

– From lecture 8, recall the *forward/bottom-up collecting semantics of arithmetic expressions*:

$$\mathrm{Faexp} \in \mathrm{Aexp} \mapsto \wp(\mathrm{Env}[\![P]\!]) \xrightarrow{\sqcup} \wp(\mathbb{I}_\Omega),$$

$$\mathrm{Faexp}[\![A]\!]R \stackrel{\mathrm{def}}{=} \{v \mid \exists \rho \in R : \rho \vdash A \Rrightarrow v\} \tag{1}$$

– the *collecting semantics of boolean expressions*:

$$\mathrm{Cbexp} \in \mathrm{Bexp} \mapsto \wp(\mathbb{R}) \xrightarrow{\sqcup} \wp(\mathbb{R}),$$

$$\mathrm{Cbexp}[\![B]\!]R \stackrel{\mathrm{def}}{=} \{\rho \in R \mid \rho \vdash B \Rrightarrow \mathtt{tt}\} \tag{2}$$

– From lecture 14, the *forward reachability collecting semantics of commands*:

$$\mathrm{Rcom} \in \mathrm{Com} \mapsto \wp(\mathrm{Env}[\![P]\!]) \xrightarrow{\sqcup} (\mathrm{in}_P[\![C]\!] \mapsto \wp(\mathrm{Env}[\![P]\!]))$$

$$\mathrm{Rcom}[\![C]\!]R\ell \stackrel{\mathrm{def}}{=} \{\rho \mid \exists \rho' \in R : \langle\langle \mathrm{at}_P[\![C]\!],\ \rho'\rangle,\ \langle \ell,\ \rho\rangle\rangle \in \tau^\star[\![C]\!]\}$$

– From lecture 16, the *generic abstraction of value properties*:

$$\langle \wp(\mathbb{I}_\Omega),\ \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L,\ \sqsubseteq\rangle \tag{3}$$

– the *abstraction of environment properties*:

$$\langle \wp(\mathbb{V} \mapsto \mathbb{I}_\Omega),\ \subseteq\rangle \underset{\dot\alpha}{\overset{\dot\gamma}{\leftrightarrows}} \langle \mathbb{V} \mapsto L,\ \dot\sqsubseteq\rangle \tag{4}$$

where

$$\dot\alpha(R) \stackrel{\mathrm{def}}{=} \lambda \mathtt{X} \in \mathbb{V} \cdot \alpha(\{\rho(\mathtt{X}) \mid \rho \in R\}), \tag{5}$$

$$\dot\gamma(r) \stackrel{\mathrm{def}}{=} \{\rho \mid \forall \mathtt{X} \in \mathbb{V} : \rho(\mathtt{X}) \in \gamma(r(\mathtt{X}))\} \tag{6}$$

– the functional *abstraction of monotonic predicate transformers*:

$$\langle \wp(\mathbb{V} \mapsto \mathbb{I}_\Omega) \xrightarrow{\mathrm{m}} \wp(\mathbb{I}_\Omega),\ \dot\subseteq\rangle \underset{\alpha^\triangleright}{\overset{\gamma^\triangleright}{\leftrightarrows}} \langle (\mathbb{V} \mapsto L) \xrightarrow{\mathrm{m}} L,\ \dot\sqsubseteq\rangle \tag{7}$$

where

$$\alpha^{\triangleright}(\Phi) \stackrel{\text{def}}{=} \alpha \circ \Phi \circ \dot{\gamma}, \qquad (8)$$
$$\gamma^{\triangleright}(\varphi) \stackrel{\text{def}}{=} \gamma \circ \varphi \circ \dot{\alpha}$$

– the *nonrelational abstraction of the forward reachability collecting semantics of commands*:

$$\langle \wp(\mathbb{R}) \xrightarrow{\sqcup} (\mathrm{in}_P[\![C]\!] \mapsto \wp(\mathbb{R})),\ \ddot{\subseteq}\rangle \underset{\alpha[\![C]\!]}{\overset{\gamma[\![C]\!]}{\leftrightarrows}} \langle (\mathbb{V} \mapsto L) \xrightarrow{\mathrm{m}} (\mathrm{in}_P[\![C]\!] \mapsto (\mathbb{V} \mapsto L)),\ \stackrel{\therefore}{\sqsubseteq}\rangle$$

where

$$\alpha[\![C]\!]\varphi \stackrel{\text{def}}{=} \lambda r \cdot \lambda \ell \cdot \dot{\alpha}(\varphi(\dot{\gamma}(r))(\ell))$$
$$\gamma[\![C]\!]\psi \stackrel{\text{def}}{=} \lambda R \cdot \lambda \ell \cdot \dot{\gamma}(\psi(\dot{\alpha}(R))(\ell))$$

– From lecture 16, various generic nonrelational abstract semantics, starting from the *abstract semantics of arithmetic expressions*:

$$\mathrm{Faexp}^{\triangleright}[\![A]\!] \mathrel{\dot{\sqsupseteq}} \alpha^{\triangleright}(\mathrm{Faexp}[\![A]\!]) \qquad (9)$$

– the *generic nonrelational abstract semantics of boolean expressions*:

$$\mathrm{Abexp} \in \mathrm{Bexp} \mapsto (\mathbb{V} \mapsto L) \xrightarrow{\mathrm{m}} (\mathbb{V} \mapsto L) \qquad (10)$$
$$\mathrm{Abexp}[\![B]\!] \mathrel{\ddot{\sqsupseteq}} \ddot{\alpha}(\mathrm{Cbexp}[\![B]\!]) \qquad (11)$$

and for the empty set:

$$\mathrm{Abexp}[\![B]\!]\lambda \mathsf{Y} \cdot \bot \stackrel{\text{def}}{=} \lambda \mathsf{Y} \cdot \bot \quad \text{if } \gamma(\bot) = \emptyset \qquad (12)$$

– the *generic non-relational forward reachability abstract semantics of commands*:

$$\mathrm{Acom}[\![C]\!] \mathrel{\stackrel{\therefore}{\sqsupseteq}} \alpha[\![C]\!](\mathrm{Rcom}[\![C]\!]) \qquad (13)$$
$$(14)$$

– From lecture 16, *basic abstract operations*

$$\mathrm{n}^{\triangleright} = \alpha(\{\underline{\mathrm{n}}\}) \qquad (15)$$
$$?^{\triangleright} \sqsupseteq \alpha(\mathbb{I}) \qquad (16)$$

– From lecture 16, the *initialization and simple sign abstraction*:

$$\begin{array}{rl}\alpha(P) \stackrel{\text{def}}{=} & (\!|\ P \subseteq \{\Omega_{\mathtt{a}}\} \mathrel{?} \mathtt{BOT} \\ & \ |\ P \subseteq [\mathtt{min\_int}, -1] \cup \{\Omega_{\mathtt{a}}\} \mathrel{?} \mathtt{NEG} \\ & \ |\ P \subseteq \{0, \Omega_{\mathtt{a}}\} \mathrel{?} \mathtt{ZERO} \\ & \ |\ P \subseteq [1, \mathtt{max\_int}] \cup \{\Omega_{\mathtt{a}}\} \mathrel{?} \mathtt{POS} \\ & \ |\ P \subseteq \mathbb{I} \cup \{\Omega_{\mathtt{a}}\} \mathrel{?} \mathtt{INI} \\ & \ |\ P \subseteq \{\Omega_{\mathtt{i}}, \Omega_{\mathtt{a}}\} \mathrel{?} \mathtt{ERR} \\ & \ \lozenge\ \mathtt{TOP}\,|\!) \end{array} \qquad (17)$$

and *concretization*:

$$\begin{array}{rlrl}\gamma(\mathtt{BOT}) \stackrel{\text{def}}{=} & \{\Omega_{\mathtt{a}}\} & \gamma(\mathtt{INI}) \stackrel{\text{def}}{=} & \mathbb{I} \cup \{\Omega_{\mathtt{a}}\}, \\ \gamma(\mathtt{NEG}) \stackrel{\text{def}}{=} & [\mathtt{min\_int}, -1] \cup \{\Omega_{\mathtt{a}}\} & \gamma(\mathtt{ERR}) \stackrel{\text{def}}{=} & \{\Omega_{\mathtt{i}}, \Omega_{\mathtt{a}}\} \\ \gamma(\mathtt{ZERO}) \stackrel{\text{def}}{=} & \{0, \Omega_{\mathtt{a}}\} & \gamma(\mathtt{TOP}) \stackrel{\text{def}}{=} & \mathbb{I}_{\Omega} \\ \gamma(\mathtt{POS}) \stackrel{\text{def}}{=} & [1, \mathtt{max\_int}] \cup \{\Omega_{\mathtt{a}}\} & & \end{array} \qquad (18)$$

Generic backward/bottom-up static analysis of arithmetic expressions

Motivating example

– The forward/top-down static analysis of arithmetic expressions brings no information on the values of the variables appearing in the arithmetic expressions when the expected result of such expressions is known, e.g. in tests

– Example (initialization and simple sign):


```
% cd Initialization-Simple-Sign
% ./a.out ../Examples/example13.sil
{ y:ERR; r:ERR }
0:
  y := ?;
1:
  if (y = 0) then
    2:
      r := 0
    3:
  else {((y < 0) | (0 < y))}
    4:
      r := 0
    5:
  fi
6:
{ y:INI; r:ZERO }
```

```
% cd Initialization-Simple-Sign
% ./a.out ../Examples/example14.sil
{ y:ERR; r:ERR }
0:
  y := ?;
1:
  if (y = 0) then
    2:
      r := y
    3:
  else {((y < 0) | (0 < y))}
    4:
      r := 0
    5:
  fi
6:
{ y:INI; r:INI }
```


– Example 13 (left) where `r = 0` at point `6:` shows that in example 14 (right), `r` is not known to be 0 at line `3:` whence that `y` is not known to be 0 at line `2:` whence that `(y = 0)` brings no abstract information on `y` at line `1:`.

– More generally, any information on the possible result of an arithmetic expression should bring information on the values of the variables involved in the arithmetic expressions for its result to satisfy the knwon information

Backward/bottom-up collecting semantics of arithmetic expressions

– The *backward/top-down collecting semantics* $\mathrm{Baexp}[\![A]\!](R)P$ of an arithmetic expression A defines the subset of possible environments R such that the arithmetic expression may evaluate, without producing a runtime error, to a value belonging to given set P

$$\mathrm{Baexp} \;\in\; \mathrm{Aexp} \mapsto \wp(\mathbb{R}) \overset{\sqcup}{\longmapsto} \wp(\mathbb{I}_\Omega) \overset{\sqcup}{\longmapsto} \wp(\mathbb{R}),$$

$$\mathrm{Baexp}[\![A]\!](R)P \stackrel{\text{def}}{=} \{\rho \in R \mid \exists i \in P \cap \mathbb{I} : \rho \vdash A \Rrightarrow i\} \quad (19)$$

Operational semantics of arithmetic expressions (recall)

Let us recall that:

$\rho \vdash \mathrm{n} \Rrightarrow \underline{\mathrm{n}}$	decimal numbers;	(20)
$\rho \vdash \mathtt{X} \Rrightarrow \rho(\underline{\mathtt{X}})$	variables;	(21)
$\dfrac{i \in \mathbb{I}}{\rho \vdash ? \Rrightarrow i}$	random;	(22)
$\dfrac{\rho \vdash A \Rrightarrow v}{\rho \vdash \mathrm{u}\, A \Rrightarrow \underline{\mathrm{u}}\, v}$	unary arithmetic operations;	(23)
$\dfrac{\rho \vdash A_1 \Rrightarrow v_1 \quad \rho \vdash A_2 \Rrightarrow v_2}{\rho \vdash A_1 \,\mathrm{b}\, A_2 \Rrightarrow v_1 \,\underline{\mathrm{b}}\, v_2}$	binary arithmetic operations.	(24)

Backward/bottom-up collecting semantics of arithmetic expressions is a lower closure operator

THEOREM. $\forall P \in \wp(\mathbb{R}) : \lambda R \cdot \mathrm{Baexp}[\![A]\!](R)P$ is a lower closure operator ■

PROOF.
1. Monotone. If $R_1 \subseteq R_2$ then $\{\rho \in R_1 \mid \exists i \in P \cap \mathbb{I} : \rho \vdash A \Rrightarrow i\} \subseteq \{\rho \in R_2 \mid \exists i \in P \cap \mathbb{I} : \rho \vdash A \Rrightarrow i\}$ whence $\mathrm{Baexp}[\![A]\!]((R_1))P \sqsubseteq \mathrm{Baexp}[\![A]\!]((R_2))P$
2. Reductive. $\{\rho \in R \mid \exists i \in P \cap \mathbb{I} : \rho \vdash A \Rrightarrow i\} \subseteq R$ and so $\mathrm{Baexp}[\![A]\!]((R))P \subseteq R$
3. Idempotent. $\{\rho \in \{\rho' \in R \mid \exists i \in P \cap \mathbb{I} : \rho' \vdash A \Rrightarrow i\} \mid \exists j \in P \cap \mathbb{I} : \rho \vdash A \Rrightarrow j\} = \{\rho \in R \mid \exists j \in P \cap \mathbb{I} : \rho \vdash A \Rrightarrow j\}$ and so $\mathrm{Baexp}[\![A]\!]((\mathrm{Baexp}[\![A]\!]((R))P))P = \mathrm{Baexp}[\![A]\!]((R))P$

□

Structural definition of the backward/bottom-up collecting semantics of arithmetic expressions

$$\mathrm{Baexp}[\![\mathrm{n}]\!](R)P = (\!|\, \underline{\mathrm{n}} \in P \cap \mathbb{I} \;?\; R : \emptyset \,|\!) \quad (25)$$

$$\mathrm{Baexp}[\![\mathtt{X}]\!](R)P = \{\rho \in R \mid \rho(\mathtt{X}) \in P \cap \mathbb{I}\} \quad (26)$$

$$\mathrm{Baexp}[\![?]\!](R)P = (\!|\, P \cap \mathbb{I} = \emptyset \;?\; \emptyset : R \,|\!) \quad (27)$$

$$\mathrm{Baexp}[\![\mathrm{u}A']\!](R)P = \mathrm{Baexp}[\![A']\!](R)(\underline{\mathrm{u}}_{\mathcal{C}}^{\triangleleft}(\mathrm{Faexp}[\![A']\!]R, P)) \quad (28)$$

$$\text{where } \underline{\mathrm{u}}_{\mathcal{C}}^{\triangleleft}(Q, P) \stackrel{\text{def}}{=} \{v \in Q \mid \underline{\mathrm{u}}\, v \in P \cap \mathbb{I}\}$$

$$\mathrm{Baexp}[\![A_1 \,\mathrm{b}\, A_2]\!](R)P = \quad (29)$$

$$\text{let } \langle P_1,\ P_2\rangle = \underline{\mathrm{b}}_{\mathcal{C}}^{\triangleleft}(\mathrm{Faexp}[\![A_1]\!]R, \mathrm{Faexp}[\![A_2]\!]R, P) \text{ in } \mathrm{Baexp}[\![A_1]\!](R)P_1 \cap \mathrm{Baexp}[\![A_2]\!](R)P_2$$

$$\text{where } \underline{\mathrm{b}}_{\mathcal{C}}^{\triangleleft}(P_1, P_2, P) \stackrel{\text{def}}{=} \{\langle v_1,\ v_2\rangle \in P_1 \times P_2 \mid v_1 \,\mathrm{b}\, v_2 \in P \cap \mathbb{I}\}$$

Structural definition of the backward/bottom-up collecting semantics of arithmetic expressions

Before providing the proof, let us recall a few definitions from lectures 5 and 8:

$$
\begin{array}{rcll}
\underline{\mathrm{u}}\,\Omega_e & \overset{\text{def}}{=} & \Omega_e; & \\
\underline{\mathrm{u}}\,i & \overset{\text{def}}{=} & \mathrm{u}\,i, & \text{if } \mathrm{u}\,i \in \mathbb{I}; \qquad (30)\\
\underline{\mathrm{u}}\,i & \overset{\text{def}}{=} & \Omega_{\mathrm{a}}, & \text{if } \mathrm{u}\,i \notin \mathbb{I}\,.\\
\Omega_e\,\underline{\mathrm{b}}\,v & \overset{\text{def}}{=} & \Omega_e; & \\
i\,\underline{\mathrm{b}}\,\Omega_e & \overset{\text{def}}{=} & \Omega_e; & \\
i_1\,\underline{\mathrm{b}}\,i_2 & \overset{\text{def}}{=} & i_1\,\mathrm{b}\,i_2, & \text{if } \mathrm{b} \in \{+,-,*\} \wedge i_1\,\mathrm{b}\,i_2 \in \mathbb{I}; \qquad (31)\\
i_1\,\underline{\mathrm{b}}\,i_2 & \overset{\text{def}}{=} & i_1\,\mathrm{b}\,i_2, & \text{if } \mathrm{b} \in \{/,\mathtt{mod}\} \wedge i_1 \in \mathbb{I}\cap\mathbb{N} \wedge i_2 \in \mathbb{I}\cap\mathbb{N}_+ \wedge i_1\,\mathrm{b}\,i_2 \in \mathbb{I};\\
i_1\,\underline{\mathrm{b}}\,i_2 & \overset{\text{def}}{=} & \Omega_{\mathrm{a}}, & \text{if } i_1\,\mathrm{b}\,i_2 \notin \mathbb{I} \vee (\mathrm{b} \in \{/,\mathtt{mod}\} \wedge (i_1 \notin \mathbb{I}\cap\mathbb{N} \vee i_2 \notin \mathbb{I}\cap\mathbb{N}_+))\,.
\end{array}
$$

Proof.

— $\mathrm{Baexp}[\![\mathtt{n}]\!](R)P$

$\overset{\text{def}}{=}$ $\{\rho \in R \mid \exists j \in P \cap \mathbb{I} : \rho \vdash \mathtt{n} \Rightarrow j\}$ ⦃by (19)⦄

$=$ $\{\rho \in R \mid \underline{\mathtt{n}} \in P \cap \mathbb{I}\}$ ⦃by (20)⦄

$=$ $(\!|\, \underline{\mathtt{n}} \in P \cap \mathbb{I} \;?\; R \;\vdots\; \emptyset \,|\!)$

— $\mathrm{Baexp}[\![\mathtt{X}]\!](R)P$

$\overset{\text{def}}{=}$ $\{\rho \in R \mid \exists j \in P \cap \mathbb{I} : \rho \vdash \mathtt{X} \Rightarrow j\}$ ⦃by (19)⦄

$=$ $\{\rho \in R \mid \rho(\mathtt{X}) \in P \cap \mathbb{I}\}$ ⦃by (21)⦄

— $\mathrm{Baexp}[\![?]\!](R)P$

$\overset{\text{def}}{=}$ $\{\rho \in R \mid \exists j \in P \cap \mathbb{I} : \rho \vdash ? \Rightarrow j\}$ ⦃by (19)⦄

$=$ $\{\rho \in R \mid \exists j \in P \cap \mathbb{I}\}$ ⦃by (22)⦄

$=$ $(\!|\, P \cap \mathbb{I} = \emptyset \;?\; \emptyset \;\vdots\; R \,|\!)$

— $\mathrm{Baexp}[\![\mathrm{u}A']\!](R)P$

$\overset{\text{def}}{=}$ $\{\rho \in R \mid \exists j \in P \cap \mathbb{I} : \rho \vdash \mathrm{u}A' \Rightarrow j\}$ ⦃by (19)⦄

$=$ $\{\rho \in R \mid \exists v : \rho \vdash A' \Rightarrow v \wedge \underline{\mathrm{u}}\,v \in P \cap \mathbb{I}\}$ ⦃by (23)⦄

$=$ $\{\rho \in R \mid \exists i' \in \{v \mid \exists \rho' \in R : \rho' \vdash A' \Rightarrow v\} : \rho \vdash A' \Rightarrow i' \wedge \underline{\mathrm{u}}\,i' \in P \cap \mathbb{I}\}$ ⦃set theory⦄

$=$ $\{\rho \in R \mid \exists i' \in \mathrm{Faexp}[\![A']\!]R : \rho \vdash A' \Rightarrow i' \wedge \underline{\mathrm{u}}\,i' \in P \cap \mathbb{I}\}$ ⦃by def. (1) of $\mathrm{Faexp}[\![A']\!]R \overset{\text{def}}{=} \{v \mid \exists \rho' \in R : \rho' \vdash A' \Rightarrow v\}$⦄

$=$ $\{\rho \in R \mid \exists i' \in \{j \in \mathrm{Faexp}[\![A']\!]R \mid \underline{\mathrm{u}}\,j \in P \cap \mathbb{I}\} : \rho \vdash A' \Rightarrow i'\}$ ⦃set theory⦄

$=$ $\{\rho \in R \mid \exists i' \in \{j \in \mathrm{Faexp}[\![A']\!]R \mid \underline{\mathrm{u}}\,j \in P \cap \mathbb{I}\} \cap \mathbb{I} : \rho \vdash A' \Rightarrow i'\}$ ⦃by (30)⦄

$=$ $\{\rho \in R \mid \exists i' \in \cap\mathbb{I} : \underline{\mathrm{u}}^{\triangleleft}_{c}(\mathrm{Faexp}[\![A']\!]R, P)\rho \vdash A' \Rightarrow i'\}$ ⦃by letting $\underline{\mathrm{u}}^{\triangleleft}_{c}(Q, P) \overset{\text{def}}{=} \{j \in Q \mid \underline{\mathrm{u}}\,j \in P \cap \mathbb{I}\}$⦄

$=$ $\mathrm{Baexp}[\![A']\!](R)(\underline{\mathrm{u}}^{\triangleleft}_{c}(\mathrm{Faexp}[\![A]\!]R, P))$ ⦃by (19)⦄

— $\mathrm{Baexp}[\![A_1 \,\mathrm{b}\, A_2]\!](R)P$

$\overset{\text{def}}{=}$ $\{\rho \in R \mid \exists j \in P \cap \mathbb{I} : \rho \vdash A_1 \,\mathrm{b}\, A_2 \Rightarrow j\}$ ⦃by (19)⦄

$=$ $\{\rho \in R \mid \exists v_1, v_2 : \rho \vdash A_1 \Rightarrow v_1 \wedge \rho \vdash A_2 \Rightarrow v_2 \wedge v_1 \,\underline{\mathrm{b}}\, v_2 \in P \cap \mathbb{I}\}$ ⦃by (24)⦄

$=$ $\{\rho \in R \mid \exists v_1 \in \mathbb{I} : \exists v_2 \in \mathbb{I} : \rho \vdash A_1 \Rightarrow v_1 \wedge \rho \vdash A_2 \Rightarrow v_2 \wedge v_1 \,\underline{\mathrm{b}}\, v_2 \in P \cap \mathbb{I}\}$ ⦃by (31)⦄

$=$ $\{\rho \in R \mid \exists v_1 \in \{v_1' \mid \exists \rho' \in R : \rho \vdash A_1 \Rightarrow v_1'\} \cap \mathbb{I} : v_2 \in \{v_2' \mid \exists \rho' \in R : \rho \vdash A_1 \Rightarrow v_2'\} \cap \mathbb{I} : \rho \vdash A_1 \Rightarrow v_1 \wedge \rho \vdash A_2 \Rightarrow v_2 \wedge v_1 \,\underline{\mathrm{b}}\, v_2 \in P \cap \mathbb{I}\}$ ⦃set theory⦄

$=$ $\{\rho \in R \mid \exists v_1 \in \mathrm{Faexp}[\![A_1]\!]R \cap \mathbb{I} : v_2 \in \mathrm{Faexp}[\![A_2]\!]R \cap \mathbb{I} : \rho \vdash A_1 \Rightarrow v_1 \wedge \rho \vdash A_2 \Rightarrow v_2 \wedge v_1 \,\underline{\mathrm{b}}\, v_2 \in P \cap \mathbb{I}\}$ ⦃by (1)⦄

$=$ $\{\rho \in R \mid \exists \langle v_1,\, v_2\rangle \in \{\langle v_1',\, v_2'\rangle \in \mathrm{Faexp}[\![A_1]\!]R \times \mathrm{Faexp}[\![A_2]\!]R \mid v_1 \,\underline{\mathrm{b}}\, v_2 \in P \cap \mathbb{I}\} \cap \mathbb{I} \times \mathbb{I} : \rho \vdash A_1 \Rightarrow v_1 \wedge \rho \vdash A_2 \Rightarrow v_2\}$ ⦃set theory⦄

$=$ $\{\rho \in R \mid \exists \langle v_1,\, v_2\rangle \in \underline{\mathrm{b}}^{\triangleleft}_{c}(\mathrm{Faexp}[\![A_1]\!]R, \mathrm{Faexp}[\![A_2]\!]R, P) \cap \mathbb{I} \times \mathbb{I} : \rho \vdash A_1 \Rightarrow v_1 \wedge \rho \vdash A_2 \Rightarrow v_2\}$ ⦃by letting $\underline{\mathrm{b}}^{\triangleleft}_{c}(P_1, P_2, P) \overset{\text{def}}{=} \{\langle v_1',\, v_2'\rangle \in P_1 \times P_2 \mid v_1 \,\underline{\mathrm{b}}\, v_2 \in P \cap \mathbb{I}\}$⦄

$=$ let $\langle P_1,\, P_2\rangle = \underline{\mathrm{b}}^{\triangleleft}_{c}(\mathrm{Faexp}[\![A_1]\!]R, \mathrm{Faexp}[\![A_2]\!]R, P)$ in $\{\rho \in R \mid \exists v_1 \in P_1 \cap \mathbb{I} : \rho \vdash A_1 \Rightarrow v_1\} \cap \{\rho \in R \mid \exists v_2 \in P_2 \cap \mathbb{I} : \rho \vdash A_2 \Rightarrow v_2\}$ ⦃set theory⦄

$$= \text{let } \langle P_1,\ P_2\rangle = \underline{\mathrm{b}}^{\triangleleft}_{\mathcal{C}}(\mathrm{Faexp}[\![A_1]\!]R, \mathrm{Faexp}[\![A_2]\!]R, P) \text{ in}$$
$$\quad \mathrm{Baexp}[\![A_1]\!](R)P_1 \cap \mathrm{Baexp}[\![A_2]\!](R)P_2$$

$\square$

Observe that by their definitions, $\underline{\mathrm{u}}^{\triangleleft}_{\mathcal{C}}$ and $\underline{\mathrm{b}}^{\triangleleft}_{\mathcal{C}}$ are both monotonic in all their arguments.

Generic backward/bottom-up non-relational abstraction of arithmetic expressions

Given an approximation (3) of value properties, we approximate environment properties by the nonrelational abstraction (4) and apply the following functional abstraction

$$\langle \wp(\mathbb{R}) \stackrel{\mathrm{m}}{\longmapsto} \wp(\mathbb{I}_\Omega) \stackrel{\mathrm{m}}{\longmapsto} \wp(\mathbb{R}),\ \ddot{\subseteq}\rangle \xrightleftharpoons[\alpha^{\triangleleft}]{\gamma^{\triangleleft}} \langle(\mathbb{V} \mapsto L) \stackrel{\mathrm{m}}{\longmapsto} L \stackrel{\mathrm{m}}{\longmapsto} (\mathbb{V} \mapsto L),\ \ddot{\sqsubseteq}\rangle$$

where

$$\begin{aligned}
\Phi \mathrel{\ddot{\subseteq}} \Psi &\stackrel{\mathrm{def}}{=} \forall R \in \wp(\mathbb{R}) : \forall P \in \wp(\mathbb{I}_\Omega) : \Phi(R)P \subseteq \Psi(R)P,\\
\varphi \mathrel{\ddot{\sqsubseteq}} \psi &\stackrel{\mathrm{def}}{=} \forall r \in \mathbb{V} \mapsto L : \forall p \in L : \varphi(r)p \mathrel{\dot{\sqsubseteq}} \psi(r)p,\\
\alpha^{\triangleleft}(\Phi) &\stackrel{\mathrm{def}}{=} \lambda r \in \mathbb{V} \mapsto L \cdot \lambda p \in L \cdot \dot{\alpha}(\Phi(\dot{\gamma}(r))\gamma(p)), \qquad (32)\\
\gamma^{\triangleleft}(\varphi) &\stackrel{\mathrm{def}}{=} \lambda R \in \wp(\mathbb{R}) \cdot \lambda P \in \wp(\mathbb{I}_\Omega) \cdot \dot{\gamma}(\varphi(\dot{\alpha}(R))\alpha(P))\ .
\end{aligned}$$

Generic backward/bottom-up non-relational abstract semantics of arithmetic expressions

We now design the backward/bottom-up abstract semantics of arithmetic expressions

$$\mathrm{Baexp}^{\triangleleft} \in \mathrm{Aexp} \mapsto (\mathbb{V} \mapsto L) \stackrel{\mathrm{m}}{\longmapsto} L \stackrel{\mathrm{m}}{\longmapsto} (\mathbb{V} \mapsto L)\ .$$

The objective is to get an overapproximation of the backward collecting semantics (19) such that

$$\mathrm{Baexp}^{\triangleleft}[\![A]\!] \mathrel{\ddot{\sqsupseteq}} \alpha^{\triangleleft}(\mathrm{Baexp}[\![A]\!])\ . \qquad (33)$$

Structural definition of the generic backward/bottom-up non-relational abstract semantics of arithmetic expressions

$$\begin{aligned}
\mathrm{Baexp}^{\triangleleft}[\![A]\!](\lambda \mathrm{Y} \cdot \bot)p &\stackrel{\mathrm{def}}{=} \lambda \mathrm{Y} \cdot \bot \qquad \text{if } \gamma(\bot) = \emptyset & (34)\\
\mathrm{Baexp}^{\triangleleft}[\![\mathrm{n}]\!](r)p &\stackrel{\mathrm{def}}{=} (\!|\ \mathrm{n}^{\triangleleft}(p)\ ?\ r\ \text{⦂}\ \lambda \mathrm{Y} \cdot \bot |\!)\\
\mathrm{Baexp}^{\triangleleft}[\![\mathrm{X}]\!](r)p &\stackrel{\mathrm{def}}{=} r[\mathrm{X} := r(\mathrm{X}) \sqcap p \sqcap ?^{\triangleright}] & (35)\\
\mathrm{Baexp}^{\triangleleft}[\![?]\!](r)p &\stackrel{\mathrm{def}}{=} (\!|\ ?^{\triangleleft}(p)\ ?\ r\ \text{⦂}\ \lambda \mathrm{Y} \cdot \bot |\!)\\
\mathrm{Baexp}^{\triangleleft}[\![\mathrm{u}\ A']\!](r)p &\stackrel{\mathrm{def}}{=} \mathrm{Baexp}^{\triangleleft}[\![A']\!](r)(\mathrm{u}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A']\!]r, p))\\
\mathrm{Baexp}^{\triangleleft}[\![A_1\ \mathrm{b}\ A_2]\!](r)p &\stackrel{\mathrm{def}}{=} \text{let } \langle p_1,\ p_2\rangle = \mathrm{b}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r, p) \text{ in}\\
&\qquad \mathrm{Baexp}^{\triangleleft}[\![A_1]\!](r)p_1 \mathrel{\dot{\sqcap}} \mathrm{Baexp}^{\triangleleft}[\![A_2]\!](r)p_2
\end{aligned}$$

Parameterized by the following backward abstract operations on L:

$$
\begin{array}{rcll}
\mathrm{n}^{\triangleleft}(p) & \stackrel{\text{def}}{=} & (\underline{\mathrm{n}} \in \gamma(p) \cap \mathbb{I}) & (36)\\
?^{\triangleleft}(p) & \stackrel{\text{def}}{=} & (\gamma(p) \cap \mathbb{I} \neq \emptyset) & (37)\\
\mathrm{u}^{\triangleleft}(q, p) & \sqsupseteq & \alpha(\{i \in \gamma(q) \mid \underline{\mathrm{u}}\, i \in \gamma(p) \cap \mathbb{I}\}) & (38)\\
 & = & \alpha \circ \mathrm{b}^{\triangleleft}(\gamma(q), \gamma(p)) & \\
\mathrm{b}^{\triangleleft}(q_1, q_2, p) & \sqsupseteq^2 & \alpha^2(\{\langle i_1,\ i_2\rangle \in \gamma^2(\langle q_1,\ q_2\rangle) \mid i_1 \,\underline{\mathrm{b}}\, i_2 \in \gamma(p) \cap \mathbb{I}\}) & (39)\\
 & = & \alpha^2(\mathrm{u}^{\triangleleft}(\gamma(q_1), \gamma(q_2), \gamma(p))) &
\end{array}
$$

Calculational design of the generic backward/bottom-up non-relational abstract semantics of arithmetic expressions

PROOF. We derive $\mathrm{Baexp}^{\triangleleft}[\![A]\!]$ by calculus, as follows

$$
\begin{array}{ll}
 & \alpha^{\triangleleft}(\mathrm{Baexp}[\![A]\!])\\
= & \qquad \wr \text{def. (32) of } \alpha^{\triangleleft} \int\\
 & \lambda r \in \mathbb{V} \mapsto L \cdot \lambda p \in L \cdot \dot{\alpha}(\mathrm{Baexp}[\![A]\!](\dot{\gamma}(r))\gamma(p))\\
= & \qquad \wr \text{def. (19) of } \mathrm{Baexp}[\![A]\!] \int\\
 & \lambda r \in \mathbb{V} \mapsto L \cdot \lambda p \in L \cdot \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists i \in \gamma(p) \cap \mathbb{I} : \rho \vdash A \Rrightarrow i\})\ .
\end{array}
$$

If r is the infimum $\lambda \mathrm{Y} \cdot \bot$ where the infimum $\bot$ of L is such that $\gamma(\bot) = \emptyset$, then $\dot{\gamma}(r) = \emptyset$ whence

$$\alpha^{\triangleleft}(\mathrm{Baexp}[\![A]\!])(\lambda \mathrm{Y} \cdot \bot)p$$

$$
\begin{array}{ll}
= & \qquad \wr \text{def. (6) of } \dot{\gamma} \int\\
 & \dot{\alpha}(\emptyset)\\
= & \qquad \wr \text{def. (5) of } \dot{\alpha} \int\\
 & \lambda \mathrm{Y} \cdot \bot\ .
\end{array}
$$

Given any $r \in \mathbb{V} \mapsto L$, $r \neq \lambda \mathrm{Y} \cdot \bot$ or $\gamma(\bot) \neq \emptyset$ and $p \in L$, we proceed by structural induction on the arithmetic expression A.

1 —— When $A = \mathrm{n} \in \mathrm{Nat}$ is a number, we have

$$
\begin{array}{lll}
 & \alpha^{\triangleleft}(\mathrm{Baexp}[\![\mathrm{n}]\!])(r)p & \\
= & \dot{\alpha}(\mathrm{Baexp}[\![\mathrm{n}]\!](\dot{\gamma}(r))\gamma(p)) & \wr \text{def. (32) of } \alpha^{\triangleleft} \int\\
= & \dot{\alpha}((\!| \underline{\mathrm{n}} \in \gamma(p) \cap \mathbb{I} \mathrel{?} \dot{\gamma}(r) \mathrel{\vdots} \emptyset |\!)) & \wr \text{def. (25) of } \mathrm{Baexp}[\![\mathrm{n}]\!] \int\\
= & \qquad \wr \text{def. conditional } (\!| \ldots \mathrel{?} \ldots \mathrel{\vdots} \ldots |\!) \int & \\
 & (\!| \underline{\mathrm{n}} \in \gamma(p) \cap \mathbb{I} \mathrel{?} \dot{\alpha}(\dot{\gamma}(r)) \mathrel{\vdots} \dot{\alpha}(\emptyset) |\!) & \\
\dot{\sqsubseteq} & \qquad \wr \dot{\alpha} \circ \dot{\gamma} \text{ is reductive and def. (5) of } \dot{\alpha} \int &
\end{array}
$$

$$
\begin{array}{ll}
 & (\!| \underline{\mathrm{n}} \in \gamma(p) \cap \mathbb{I} \mathrel{?} r \mathrel{\vdots} \lambda \mathrm{Y} \cdot \bot |\!)\\
= & \qquad \wr \text{by defining } \mathrm{n}^{\triangleleft}(p) \stackrel{\text{def}}{=} (\underline{\mathrm{n}} \in \gamma(p) \cap \mathbb{I}) \int\\
 & (\!| \mathrm{n}^{\triangleleft}(p) \mathrel{?} r \mathrel{\vdots} \lambda \mathrm{Y} \cdot \bot |\!)\\
= & \qquad \wr \text{by defining } \mathrm{Baexp}^{\triangleleft}[\![\mathrm{n}]\!](r)p \stackrel{\text{def}}{=} (\!| \mathrm{n}^{\triangleleft}(p) \mathrel{?} r \mathrel{\vdots} \lambda \mathrm{Y} \cdot \bot |\!) \int\\
 & \mathrm{Baexp}^{\triangleleft}[\![\mathrm{n}]\!](r)p\ .
\end{array}
$$

2 —— When $A = \mathrm{X} \in \mathbb{V}$ is a variable, we have

$$
\begin{array}{lll}
 & \alpha^{\triangleleft}(\mathrm{Baexp}[\![\mathrm{X}]\!])(r)p & \\
= & \dot{\alpha}(\mathrm{Baexp}[\![\mathrm{X}]\!](\dot{\gamma}(r))\gamma(p)) & \wr \text{def. (32) of } \alpha^{\triangleleft} \int\\
= & \qquad \wr \text{def. (26) of } \mathrm{Baexp}[\![\mathrm{X}]\!] \int & \\
 & \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \rho(\mathrm{X}) \in \gamma(p) \cap \mathbb{I}\}) & \\
\dot{\sqsubseteq} & \qquad \wr [\gamma \circ \alpha \text{ is extensive and } \dot{\alpha} \text{ is monotone } \int & \\
 & \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \rho(\mathrm{X}) \in \gamma(p) \cap \gamma \circ \alpha(\mathbb{I})\}) &
\end{array}
$$

$=$ $\wr$def. (6) of $\dot{\gamma}$$\int$

$\dot{\alpha}(\{\rho \mid \forall \mathtt{Y} \neq \mathtt{X} : \rho(\mathtt{Y}) \in \gamma(r(\mathtt{Y})) \wedge \rho(\mathtt{X}) \in \gamma(r(\mathtt{X})) \cap \gamma(p) \cap \gamma \circ \alpha(\mathbb{I})\})$

$=$ $\wr$$\gamma$ is a complete meet morphism$\int$

$\dot{\alpha}(\{\rho \mid \forall \mathtt{Y} \neq \mathtt{X} : \rho(\mathtt{Y}) \in \gamma(r(\mathtt{Y})) \wedge \rho(\mathtt{X}) \in \gamma(r(\mathtt{X}) \sqcap p \sqcap \alpha(\mathbb{I}))\})$

$=$ $\wr$def. environment assignment$\int$

$\dot{\alpha}(\{\rho \mid \forall \mathtt{Y} \neq \mathtt{X} : \rho(\mathtt{Y}) \in \gamma(r[\mathtt{X} := r(\mathtt{X}) \sqcap p \sqcap \alpha(\mathbb{I})](\mathtt{Y})) \wedge \rho(\mathtt{X}) \in \gamma(r[\mathtt{X} := r(\mathtt{X}) \sqcap p \sqcap \alpha(\mathbb{I})](\mathtt{X})\})$

$=$ $\wr$def. (6) of $\dot{\gamma}$$\int$

$\dot{\alpha}(\{\rho \mid \rho \in \dot{\gamma}(r[\mathtt{X} := r(\mathtt{X}) \sqcap p \sqcap \alpha(\mathbb{I})])\})$

$=$ $\wr$set notation$\int$

$\dot{\alpha}(\dot{\gamma}(r[\mathtt{X} := r(\mathtt{X}) \sqcap p \sqcap \alpha(\mathbb{I})]))$

$\dot{\sqsubseteq}$ $\wr$$\dot{\alpha} \circ \dot{\gamma}$ is reductive $\int$

$r[\mathtt{X} := r(\mathtt{X}) \sqcap p \sqcap \alpha(\mathbb{I})]$

$\dot{\sqsubseteq}$ $\wr$def. (16) of $?^{\triangleright}$$\int$

$r[\mathtt{X} := r(\mathtt{X}) \sqcap p \sqcap ?^{\triangleright}]$

$=$ $\wr$by defining $\mathrm{Baexp}^{\triangleleft}[\![\mathtt{X}]\!](r)p \stackrel{\text{def}}{=} r[\mathtt{X} := r(\mathtt{X}) \sqcap p \sqcap ?^{\triangleright}]$$\int$

$\mathrm{Baexp}^{\triangleleft}[\![\mathtt{X}]\!](r)p$.

3 —— When $A = ?$ is random, we have

$\alpha^{\triangleleft}(\mathrm{Baexp}[\![?]\!])(r)p$

$=$ $\dot{\alpha}(\mathrm{Baexp}[\![?]\!](\dot{\gamma}(r))\gamma(p))$ $\wr$def. (32) of $\alpha^{\triangleleft}$$\int$

$=$ $\wr$def. (27) of $\mathrm{Baexp}[\![?]\!]$$\int$

$=$ $\dot{\alpha}((\!| \gamma(p) \cap \mathbb{I} = \emptyset \; ? \; \emptyset \mathbin{:} \dot{\gamma}(r) |\!))$

$=$ $\wr$def. conditional $(\!| \ldots ? \ldots \mathbin{:} \ldots |\!)$$\int$

$(\!| \gamma(p) \cap \mathbb{I} = \emptyset \; ? \; \dot{\alpha}(\emptyset) \mathbin{:} \dot{\alpha}(\dot{\gamma}(r)) |\!)$

$\dot{\sqsubseteq}$ $\wr$def. (5) of $\dot{\alpha}$ and $\dot{\alpha} \circ \dot{\gamma}$ reductive $\int$

$(\!| \gamma(p) \cap \mathbb{I} = \emptyset \; ? \; \lambda \mathtt{Y} \cdot \bot \mathbin{:} r |\!)$

$=$ $\wr$negation$\int$

$(\!| \gamma(p) \cap \mathbb{I} \neq \emptyset \; ? \; r \mathbin{:} \lambda \mathtt{Y} \cdot \bot |\!)$

$=$ $\wr$by defining $?^{\triangleleft}(p) \stackrel{\text{def}}{=} (\gamma(p) \cap \mathbb{I} \neq \emptyset)$$\int$

$(\!| ?^{\triangleleft}(p) \; ? \; r \mathbin{:} \lambda \mathtt{Y} \cdot \bot |\!)$

$=$ $\wr$by defining $\mathrm{Baexp}^{\triangleleft}[\![?]\!] \stackrel{\text{def}}{=} (\!| ?^{\triangleleft}(p) \; ? \; r \mathbin{:} \lambda \mathtt{Y} \cdot \bot |\!)$$\int$

$\mathrm{Baexp}^{\triangleleft}[\![?]\!](r)p$.

4 —— When $A = \mathrm{u}\, A'$ is a unary operation, we have

$\alpha^{\triangleleft}(\mathrm{Baexp}[\![\mathrm{u}\, A']\!])(r)p$

$=$ $\dot{\alpha}(\mathrm{Baexp}[\![\mathrm{u}\, A']\!](\dot{\gamma}(r))(\gamma(p)))$ $\wr$def. (32) of $\alpha^{\triangleleft}$$\int$

$=$ $\wr$def. (28) of $\mathrm{Baexp}[\![\mathrm{u}\, A']\!]$$\int$

$\dot{\alpha}(\mathrm{Baexp}[\![A']\!](\dot{\gamma}(r))(\underline{\mathrm{u}}_c^{\triangleleft}(\mathrm{Faexp}[\![A']\!](\dot{\gamma}(r)), \gamma(p))))$

$\dot{\sqsubseteq}$ $\wr$$\gamma \circ \alpha$ is extensive and monotony$\int$

$\dot{\alpha}(\mathrm{Baexp}[\![A']\!](\dot{\gamma}(r))(\gamma \circ \alpha \circ \underline{\mathrm{u}}_c^{\triangleleft}(\gamma \circ \alpha \circ \mathrm{Faexp}[\![A']\!](\dot{\gamma}(r)), \gamma(p))))$

$=$ $\wr$def. (8) of $\alpha^{\triangleright}(\Phi) \stackrel{\text{def}}{=} \alpha \circ \Phi \circ \dot{\gamma}$$\int$

$\dot{\alpha}(\mathrm{Baexp}[\![A']\!](\dot{\gamma}(r))(\gamma \circ \alpha \circ \underline{\mathrm{u}}_c^{\triangleleft}(\gamma \circ \alpha^{\triangleright}(\mathrm{Faexp}[\![A']\!])(r), \gamma(p))))$

$\dot{\sqsubseteq}$ $\wr$by (9) so that $\mathrm{Faexp}^{\triangleright}[\![A]\!] \mathrel{\dot{\sqsupseteq}} \alpha^{\triangleright}(\mathrm{Faexp}[\![A]\!])$ and monotony$\int$

$\dot{\alpha}(\mathrm{Baexp}[\![A']\!](\dot{\gamma}(r))(\gamma \circ \alpha \circ \underline{\mathrm{u}}_c^{\triangleleft}(\gamma \circ \mathrm{Faexp}^{\triangleright}[\![A']\!](r), \gamma(p))))$

$\dot{\sqsubseteq}$ $\wr$by def. (28) of $\mathrm{u}^{\triangleleft}(Q, P) \stackrel{\text{def}}{=} \{v \in Q \mid \underline{\mathrm{u}}\, v \in P \cap \mathbb{I}\}$ so that $\mathrm{u}^{\triangleleft}(Q, P) \sqsupseteq \alpha \circ \underline{\mathrm{u}}_c^{\triangleleft}(\gamma(Q), \gamma(P))$$\int$

$\dot{\alpha}(\mathrm{Baexp}[\![A']\!](\dot{\gamma}(r))(\gamma \circ \mathrm{u}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A']\!](r), p)))$

$=$ $\mathrm{Baexp}[\![A']\!](\dot{\gamma}(r))(\mathrm{u}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A']\!](r), p))$ $\wr$def. (32) of $\alpha^{\triangleleft}(\Phi)(r)p \stackrel{\text{def}}{=} \dot{\alpha}(\Phi(\dot{\gamma}(r))\gamma(p))$$\int$

$\mathrm{Baexp}^{\triangleleft}[\![A']\!](r)(\mathrm{u}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A']\!]r, p))$

$=$ $\wr$defining $\mathrm{Baexp}^{\triangleleft}[\![\mathrm{u}\, A']\!](r)p \stackrel{\text{def}}{=} \mathrm{Baexp}^{\triangleleft}[\![A']\!](r)(\mathrm{u}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A']\!]r, p))$$\int$

$\mathrm{Baexp}^{\triangleleft}[\![\mathrm{u}\ A']\!](r)p$.

5 —— When $A = A_1\ \mathrm{b}\ A_2$ is a binary operation, we have

$\alpha^{\triangleleft}(\mathrm{Baexp}[\![A_1\ \mathrm{b}\ A_2]\!])(r)p$

$= \dot{\alpha}(\mathrm{Baexp}[\![A_1\ \mathrm{b}\ A_2]\!](\dot{\gamma}(r))(\gamma(p)))$ $\quad \wr$def. (32) of $\alpha^{\triangleleft}\wr$

$=$ $\quad \wr$def. (29) of $\mathrm{Baexp}[\![A_1\ \mathrm{b}\ A_2]\!]\wr$

$=$ let $\langle p_1,\ p_2\rangle = \underline{\mathrm{b}}_c^{\triangleleft}(\mathrm{Faexp}[\![A_1]\!](\dot{\gamma}(r)), \mathrm{Faexp}[\![A_2]\!](\dot{\gamma}(r)), (\gamma(p)))$ in
$\quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!]((\dot{\gamma}(r)))p_1 \cap \mathrm{Baexp}[\![A_2]\!]((\dot{\gamma}(r)))p_2)$

$\dot{\sqsubseteq}$ $\quad \wr\gamma \circ \alpha$ is extensive and by monotony$\wr$

let $\langle p_1,\ p_2\rangle = \underline{\mathrm{b}}_c^{\triangleleft}(\gamma \circ \alpha^{\triangleright}(\mathrm{Faexp}[\![A_1]\!])(r), \gamma \circ \alpha \circ \mathrm{Faexp}[\![A_2]\!](\dot{\gamma}(r)), (\gamma(p)))$ in
$\quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!]((\dot{\gamma}(r)))\gamma \circ \alpha(p_1) \cap \mathrm{Baexp}[\![A_2]\!]((\dot{\gamma}(r)))\gamma \circ \alpha(p_2))$

$=$ $\quad \wr$def. (8) of $\alpha^{\triangleright}(\phi) \stackrel{\mathrm{def}}{=} \alpha \circ \phi \circ \dot{\gamma}\wr$

let $\langle p_1,\ p_2\rangle = \underline{\mathrm{b}}_c^{\triangleleft}(\gamma \circ \alpha^{\triangleright}(\mathrm{Faexp}[\![A_1]\!])(r), \gamma \circ \alpha^{\triangleright}(\mathrm{Faexp}[\![A_2]\!])(r), (\gamma(p)))$ in
$\quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!]((\dot{\gamma}(r)))\gamma \circ \alpha(p_1) \cap \mathrm{Baexp}[\![A_2]\!]((\dot{\gamma}(r)))\gamma \circ \alpha(p_2))$

$\dot{\sqsubseteq}$ $\quad \wr$by (9) so that $\mathrm{Faexp}^{\triangleright}[\![A]\!] \ \dot{\sqsupseteq}\ \alpha^{\triangleright}(\mathrm{Faexp}[\![A]\!])$ and monotony$\wr$

let $\langle p_1,\ p_2\rangle = \underline{\mathrm{b}}_c^{\triangleleft}(\gamma \circ \mathrm{Faexp}^{\triangleright}[\![A_1]\!](r), \gamma \circ \mathrm{Faexp}^{\triangleright}[\![A_2]\!](r), (\gamma(p)))$ in
$\quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!]((\dot{\gamma}(r)))\gamma \circ \alpha(p_1) \cap \mathrm{Baexp}[\![A_2]\!]((\dot{\gamma}(r)))\gamma \circ \alpha(p_2))$

$=$ $\quad \wr$def. $\alpha^2(\langle x,\ y\rangle) \stackrel{\mathrm{def}}{=} \langle \alpha(x),\ \alpha(y)\rangle\wr$

let $\langle p_1,\ p_2\rangle = \alpha^2 \circ \underline{\mathrm{b}}_c^{\triangleleft}(\gamma \circ \mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \gamma \circ \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r, (\gamma(p)))$ in
$\quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!]((\dot{\gamma}(r)))\gamma(p_1) \cap \mathrm{Baexp}[\![A_2]\!]((\dot{\gamma}(r)))\gamma(p_2))$

$\dot{\sqsubseteq}$ $\quad \wr$by def. (29) of $\underline{\mathrm{b}}_c^{\triangleleft}(p_1, p_2, p) \stackrel{\mathrm{def}}{=} \{\langle v_1,\ v_2\rangle \in p_1 \times p_2 \mid v_1\ \mathrm{b}\ v_2 \in p \cap \mathbb{I}\}$ so that $\mathrm{b}^{\triangleleft}(p_1, p_2, p) \sqsupseteq \alpha^2 \circ \underline{\mathrm{u}}_c^{\triangleleft}(\gamma(p_1), \gamma(p_2, \gamma(p)))$ and monotony$\wr$

let $\langle p_1,\ p_2\rangle = \mathrm{b}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r, p$ in
$\quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!](\dot{\gamma}(r))(\gamma(p_1)) \cap \mathrm{Baexp}[\![A_2]\!](\dot{\gamma}(r))(\gamma(p_2)))$

$\dot{\sqsubseteq}$ $\quad \wr\alpha^{\triangleleft}$ is monotone and so $\alpha^{\triangleleft}(x \cap y) \ \dot{\sqsubseteq}\ \alpha^{\triangleleft}(x) \ \dot{\sqcap}\ \alpha^{\triangleleft}(y)\wr$

let $\langle p_1,\ p_2\rangle = \mathrm{b}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r, p)$ in
$\quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!](\dot{\gamma}(r))(\gamma(p_1))) \ \dot{\sqcap}\ \dot{\alpha}(\mathrm{Baexp}[\![A_2]\!](\dot{\gamma}(r))(\gamma(p_2)))$

$=$ $\quad \wr$def. (32) of $\alpha^{\triangleleft}(\phi)(r)p \stackrel{\mathrm{def}}{=} \dot{\alpha}(\phi(\dot{\gamma}(r))\gamma(p))\wr$

let $\langle p_1,\ p_2\rangle = \mathrm{b}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r, p)$ in
$\quad \alpha^{\triangleleft}(\mathrm{Baexp}[\![A_1]\!])(r)p_1 \ \dot{\sqcap}\ \alpha^{\triangleleft}(\mathrm{Baexp}[\![A_2]\!])(r)p_2$

$\dot{\sqsubseteq}$ $\quad \wr$by (33) so that $\mathrm{Baexp}^{\triangleleft}[\![A]\!] \ \dot{\sqsupseteq}\ \alpha^{\triangleleft}(\mathrm{Baexp}[\![A]\!])$ and monotony of $\dot{\sqcap}\wr$

let $\langle p_1,\ p_2\rangle = \mathrm{b}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r, p)$ in
$\quad \mathrm{Baexp}^{\triangleleft}[\![A_1]\!](r)p_1 \ \dot{\sqcap}\ \mathrm{Baexp}^{\triangleleft}[\![A_2]\!])(r)p_2$

$=$ $\quad \wr$defining $\mathrm{Baexp}^{\triangleleft}[\![A_1\ \mathrm{b}\ A_2]\!](r)p \stackrel{\mathrm{def}}{=}$
$\quad$ let $\langle p_1,\ p_2\rangle = \mathrm{b}^{\triangleleft}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r, p)$ in
$\quad\quad \mathrm{Baexp}^{\triangleleft}[\![A_1]\!](r)p_1 \ \dot{\sqcap}\ \mathrm{Baexp}^{\triangleleft}[\![A_2]\!](r)p_2\wr$

$\mathrm{Baexp}^{\triangleleft}[\![A_1\ \mathrm{b}\ A_2]\!](r)p$.

□

Implementation of the structural definition of the generic backward/bottom-up non-relational abstract semantics of arithmetic expressions

```
(* baexp.mli *)
open Abstract_Syntax
open Avalues
open Aenv
(* backward evaluation of arithmetic operations *)
val b_aexp : aexp -> Aenv.t -> Avalues.t -> Aenv.t
```

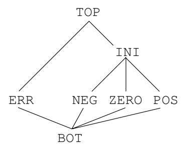
```
(* baexp.ml *)
open Abstract_Syntax
(* Backward abstract interpretation of arithmetic operations *)
let rec b_aexp' a r p =
  match a with
  | (NAT i) -> if (Avalues.b_NAT i p) then r else (Aenv.bot ())
  | (VAR v) ->
      (Aenv.set r v (Avalues.meet (Avalues.meet (Aenv.get r v) p) (Avalues.f_RANDOM ()
  | RANDOM -> if (Avalues.b_RANDOM p) then r else (Aenv.bot ())
  | (UMINUS a1) -> (b_aexp' a1 r (Avalues.b_UMINUS (Aaexp.a_aexp a1 r) p))
  | (UPLUS a1)  -> (b_aexp' a1 r (Avalues.b_UPLUS (Aaexp.a_aexp a1 r) p))
  | (PLUS (a1, a2)) ->
    let (p1,p2) = (Avalues.b_PLUS (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r) p)
      in (Aenv.meet (b_aexp' a1 r p1) (b_aexp' a2 r p2))
  | (MINUS (a1, a2))  ->
    let (p1,p2) = (Avalues.b_MINUS (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r) p)
      in (Aenv.meet (b_aexp' a1 r p1) (b_aexp' a2 r p2))
  | (TIMES (a1, a2))  ->
    let (p1,p2) = (Avalues.b_TIMES (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r) p)
```



```
      in (Aenv.meet (b_aexp' a1 r p1) (b_aexp' a2 r p2))
  | (DIV (a1, a2))  ->
    let (p1,p2) = (Avalues.b_DIV (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r) p)
      in (Aenv.meet (b_aexp' a1 r p1) (b_aexp' a2 r p2))
  | (MOD (a1, a2))  ->
    let (p1,p2) = (Avalues.b_MOD (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r) p)
      in (Aenv.meet (b_aexp' a1 r p1) (b_aexp' a2 r p2))
let b_aexp a r p =
 if (Aenv.is_bot r) & (Avalues.isbotempty ()) then (Aenv.bot ()) else b_aexp' a r p
```


Primitive backward/bottom-up non-relational abstract operations for initialization and simple sign analysis

We must now define the primitive operations $\mathtt{n}^{\triangleleft}$, $\mathtt{?}^{\triangleleft}$, $\mathtt{u}^{\triangleleft}$, $\mathtt{b}^{\triangleleft}$ for the abstract lattice



$$\gamma(\mathtt{BOT}) \stackrel{\text{def}}{=} \{\Omega_{\mathtt{a}}\} \qquad \gamma(\mathtt{INI}) \stackrel{\text{def}}{=} \mathbb{I} \cup \{\Omega_{\mathtt{a}}\},$$

$$\gamma(\mathtt{NEG}) \stackrel{\text{def}}{=} [\mathtt{min\_int}, -1] \cup \{\Omega_{\mathtt{a}}\} \qquad \gamma(\mathtt{ERR}) \stackrel{\text{def}}{=} \{\Omega_{\mathtt{i}}, \Omega_{\mathtt{a}}\}$$

$$\gamma(\mathtt{ZERO}) \stackrel{\text{def}}{=} \{0, \Omega_{\mathtt{a}}\} \qquad \gamma(\mathtt{TOP}) \stackrel{\text{def}}{=} \mathbb{I}_{\Omega}$$

$$\gamma(\mathtt{POS}) \stackrel{\text{def}}{=} [1, \mathtt{max\_int}] \cup \{\Omega_{\mathtt{a}}\}$$

Implementation of the primitive backward/ bottom-up non-relational abstract arithmetic operations for initialization and simple sign analysis

1 —— In the abstract interpretation (35) of variables, we have

$$\mathtt{?}^{\triangleright} = \mathtt{INI}$$

by definition (17) of α.

2 —— From the definition (36) of $\mathrm{n}^{\triangleleft}$ and (18) of γ, we directly get by case analysis

$\mathrm{n}^{\triangleleft}(p)$	p: BOT	NEG	ZERO	POS	INI	ERR	TOP
$\underline{\mathrm{n}} \in [\mathtt{min\_int}, -1]$	ff	tt	ff	ff	tt	ff	tt
$\underline{\mathrm{n}} = 0$	ff	ff	tt	ff	tt	ff	tt
$\underline{\mathrm{n}} \in [1, \mathtt{max\_int}]$	ff	ff	ff	tt	tt	ff	tt
$\underline{\mathrm{n}} < \mathtt{min\_int} \vee \underline{\mathrm{n}} > \mathtt{max\_int}$	ff	ff	ff	ff	ff	ff	ff

3 —— From the definition (37) of $?^{\triangleleft}$ and (18) of γ, we directly get by case analysis

p	BOT	NEG	ZERO	POS	INI	ERR	TOP
$?^{\triangleleft}(p)$	ff	tt	tt	tt	tt	ff	tt

4 —— For the backward unary arithmetic operations (38), we have

p	BOT	NEG	ZERO	POS	INI	ERR	TOP
$+^{\triangleleft}(q,p)$	BOT	$q \sqcap \mathtt{NEG}$	$q \sqcap \mathtt{ZERO}$	$q \sqcap \mathtt{POS}$	$q \sqcap \mathtt{INI}$	BOT	$q \sqcap \mathtt{INI}$
$-^{\triangleleft}(q,p)$	BOT	$q \sqcap \mathtt{POS}$	$q \sqcap \mathtt{ZERO}$	$q \sqcap \mathtt{NEG}$	$q \sqcap \mathtt{INI}$	BOT	$q \sqcap \mathtt{INI}$

PROOF. Let us consider a few typical cases.

5 —— If $p = \mathtt{BOT}$ or $p = \mathtt{ERR}$ then by (18),

$$\underline{\mathrm{u}}\, i \in \gamma(p) \cap \mathbb{I} \subseteq \{\Omega_{\mathrm{i}}, \Omega_{\mathrm{a}}\} \cap [\mathtt{min\_int}, \mathtt{max\_int}] = \emptyset$$

is false so that $\mathrm{u}^{\triangleleft}(q,p) = \alpha(\emptyset) = \mathtt{BOT}$.

6 —— If $p = \mathtt{POS}$ then by (18), $\underline{-}\, i \in \gamma(p) \cap \mathbb{I} = [1, \mathtt{max\_int}]$ if and only if ≀by def. (30) of $\underline{-}$∫ $i \in [\mathtt{min\_int}+1, -1]$ so that $-^{\triangleleft}(q,p) = \alpha(\gamma(q) \cap [\mathtt{min\_int}+1, -1]) \subseteq \alpha(\gamma(q) \cap \gamma(\mathtt{NEG}))$ by (18). But γ preserves meets whence this is equal to $\alpha(\gamma(q \sqcap \mathtt{NEG})) \sqsubseteq q \sqcap \mathtt{NEG}$ since $\alpha \circ \gamma$ is reductive.

7 —— If $p = \mathtt{INI}$ or $p = \mathtt{TOP}$ then by (18), $\underline{-}\, i \in \gamma(p) \cap \mathbb{I} = [\mathtt{min\_int}, \mathtt{max\_int}]$ if and only if ≀by def. (30) of $\underline{-}$∫ $i \in [\mathtt{min\_int}+1, \mathtt{max\_int}]$ so that $-^{\triangleleft}(q,p) = \alpha(\gamma(q) \cap [\mathtt{min\_int}+1, \mathtt{max\_int}]) \subseteq \alpha(\gamma(q) \cap \gamma(\mathtt{INI}))$ by (18). But γ preserves meets whence this is equal to $\alpha(\gamma(q \sqcap \mathtt{INI})) \sqsubseteq q \sqcap \mathtt{INI}$ since $\alpha \circ \gamma$ is reductive. □

8 —— For the backward binary arithmetic operations (39), we have

$$
\begin{aligned}
&/^{\triangleleft}(q_1, q_2, p) \stackrel{\mathrm{def}}{=} \mathrm{mod}^{\triangleleft}(q_1, q_2, p) \stackrel{\mathrm{def}}{=} \\
&\quad (\!|\, q_1 \in \{\mathtt{BOT}, \mathtt{NEG}, \mathtt{ERR}\} \vee q_2 \in \{\mathtt{BOT}, \mathtt{NEG}, \mathtt{ZERO}, \mathtt{ERR}\} \vee p \in \{\mathtt{BOT}, \mathtt{NEG}, \mathtt{ERR}\} \;?\; \\
&\qquad \langle \mathtt{BOT},\ \mathtt{BOT}\rangle \\
&\qquad ¿\; (\!|\, p = \mathtt{POS} \;?\; \mathit{smash}(\langle q_1 \sqcap \mathtt{POS},\ q_2 \sqcap \mathtt{POS}\rangle) \;¿\; \langle q_1 \sqcap \mathtt{INI},\ q_2 \sqcap \mathtt{POS}\rangle \,|\!)\, |\!) \\
&\mathit{smash}(\langle x,\ y\rangle) \stackrel{\mathrm{def}}{=} (\!|\, x = \mathtt{BOT} \vee y = \mathtt{BOT} \;?\; \langle \mathtt{BOT},\ \mathtt{BOT}\rangle \;¿\; \langle x,\ y\rangle \,|\!)\ .
\end{aligned}
$$

PROOF. If $\mathrm{b} \in \{/, \mathrm{mod}\}$ and $q_1 \in \{\mathtt{BOT}, \mathtt{NEG}, \mathtt{ERR}\}$ or $q_2 \in \{\mathtt{BOT}, \mathtt{NEG}, \mathtt{ZERO}, \mathtt{ERR}\}$ then $i_1 \in \gamma(q_1) \subseteq [\mathtt{min\_int}, -1] \cup \{\Omega_{\mathrm{i}}, \Omega_{\mathrm{a}}\}$ or $i_2 \in \gamma(q_2) \subseteq [\mathtt{min\_int}, 0] \cup \{\Omega_{\mathrm{i}}, \Omega_{\mathrm{a}}\}$ in which case $i_1 \,\underline{\mathrm{b}}\, i_2 \notin \mathbb{I}$ by (31). If follows that $\mathrm{b}^{\triangleleft}(q_1, q_2, p) = \alpha^2(\emptyset) = \langle \mathtt{BOT},\ \mathtt{BOT}\rangle$ by componentwise definition of α^2 and (17).

If $p \in \{\mathtt{BOT}, \mathtt{NEG}, \mathtt{ERR}\}$ then $i_1 \,\underline{\mathrm{b}}\, i_2 \notin \gamma(p) \cap \mathbb{I} \subseteq [\mathtt{min\_int}, -1]$ in contradiction with (31) showing that $i_1 \,\underline{\mathrm{b}}\, i_2$ is not negative. Again $\mathrm{b}^{\triangleleft}(q_1, q_2, p) = \alpha^2(\emptyset) = \langle \mathtt{BOT},\ \mathtt{BOT} \rangle$ by componentwise definition of α^2 and (17).
Otherwise to have $i_1 \,\underline{\mathrm{b}}\, i_2 \in \mathbb{I}$, we must have $i_1 \in [0, \mathtt{max\_int}]$ and $i_2 \in [1, \mathtt{max\_int}]$ whence necessarily $i_1 \in \gamma(\mathtt{INI})$ and $i_2 \in \gamma(\mathtt{POS})$ so that $\alpha^2(\gamma^2(\langle q_1 \sqcap \mathtt{INI},\ q_2 \sqcap \mathtt{POS} \rangle)) \sqsubseteq^2 \langle q_1 \sqcap \mathtt{INI},\ q_2 \sqcap \mathtt{POS} \rangle \stackrel{\text{def}}{=} \mathrm{b}^{\triangleleft}(q_1, q_2, p)$. Moreover the quotient is strictly positive only if the dividend is non zero. □

9 —— With the same reasoning, for addition $+^{\triangleleft}$, we have

$$+^{\triangleleft}(q_1, q_2, p) = \langle \mathtt{BOT},\ \mathtt{BOT} \rangle \quad \text{if } q_1 \in \{\mathtt{BOT}, \mathtt{ERR}\} \vee q_2 \in \{\mathtt{BOT}, \mathtt{ERR}\} \vee p \in \{\mathtt{BOT}, \mathtt{ERR}\}$$

$$+^{\triangleleft}(q_1, q_2, p) = \langle q_1 \sqcap \mathtt{INI},\ q_2 \sqcap \mathtt{INI} \rangle \quad \text{if } p \in \{\mathtt{INI}, \mathtt{TOP}\}\ .$$

Otherwise

$+^{\triangleleft}(q_1, q_2, \mathtt{NEG})$		q_2			
		NEG	ZERO	POS	INI, TOP
q_1	NEG	⟨NEG, NEG⟩	⟨NEG, ZERO⟩	⟨NEG, POS⟩	⟨NEG, INI⟩
	ZERO	⟨ZERO, NEG⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨ZERO, NEG⟩
	POS	⟨POS, NEG⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨POS, NEG⟩
	INI, TOP	⟨INI, NEG⟩	⟨NEG, ZERO⟩	⟨NEG, POS⟩	⟨INI, INI⟩

$+^{\triangleleft}(q_1, q_2, \mathtt{ZERO})$		q_2			
		NEG	ZERO	POS	INI, TOP
q_1	NEG	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨NEG, POS⟩	⟨NEG, POS⟩
	ZERO	⟨BOT, BOT⟩	⟨ZERO, ZERO⟩	⟨BOT, BOT⟩	⟨ZERO, ZERO⟩
	POS	⟨POS, NEG⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨POS, NEG⟩
	INI, TOP	⟨POS, NEG⟩	⟨ZERO, ZERO⟩	⟨NEG, POS⟩	⟨INI, INI⟩

$+^{\triangleleft}(q_1, q_2, \mathtt{POS})$		q_2			
		NEG	ZERO	POS	INI, TOP
q_1	NEG	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨NEG, POS⟩	⟨NEG, POS⟩
	ZERO	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨ZERO, POS⟩	⟨ZERO, POS⟩
	POS	⟨POS, NEG⟩	⟨POS, ZERO⟩	⟨POS, POS⟩	⟨POS, INI⟩
	INI, TOP	⟨POS, NEG⟩	⟨POS, ZERO⟩	⟨INI, POS⟩	⟨INI, INI⟩

10 —— The backward ternary substraction operation $-^{\triangleleft}$ is defined as

$$-^{\triangleleft}(q_1, q_2, p) \stackrel{\text{def}}{=} \text{let } (r_1, r_2) = -^{\triangleleft}(q_1, -^{\triangleright}(q_2), p) \text{ in } (r_1, -^{\triangleright}(r_2))\ .$$

11 —— The handling of the backward ternary multiplication operation $*^{\triangleleft}$ is similar

$*^{\triangleleft}(q_1, q_2, \mathtt{NEG})$		q_2			
		NEG	ZERO	POS	INI, TOP
q_1	NEG	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨NEG, POS⟩	⟨NEG, POS⟩
	ZERO	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩
	POS	⟨POS, NEG⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨POS, NEG⟩
	INI, TOP	⟨POS, NEG⟩	⟨BOT, BOT⟩	⟨NEG, POS⟩	⟨INI, INI⟩

$*^{\triangleleft}(q_1, q_2, \texttt{ZERO})$		q_2: NEG	q_2: ZERO	q_2: POS	q_2: INI, TOP
q_1	NEG	⟨BOT, BOT⟩	⟨NEG, ZERO⟩	⟨BOT, BOT⟩	⟨NEG, ZERO⟩
	ZERO	⟨ZERO, NEG⟩	⟨ZERO, ZERO⟩	⟨ZERO, POS⟩	⟨ZERO, INI⟩
	POS	⟨BOT, BOT⟩	⟨POS, ZERO⟩	⟨BOT, BOT⟩	⟨POS, ZERO⟩
	INI, TOP	⟨ZERO, NEG⟩	⟨INI, ZERO⟩	⟨ZERO, POS⟩	⟨INI, INI⟩

$*^{\triangleleft}(q_1, q_2, \texttt{POS})$		q_2: NEG	q_2: ZERO	q_2: POS	q_2: INI, TOP
q_1	NEG	⟨NEG, NEG⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨NEG, NEG⟩
	ZERO	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩
	POS	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨POS, POS⟩	⟨POS, POS⟩
	INI, TOP	⟨NEG, NEG⟩	⟨BOT, BOT⟩	⟨POS, POS⟩	⟨INI, INI⟩

Implementation of the primitive backward/bottom-up non-relational abstract arithmetic operations for initialization and simple sign analysis

```
(* avalues.mli *)
(* abstraction of sets of machine integers by initialization *)
(* and simple sign                                            *)
type t
...
(* backward abstract interpretation of arithmetic expressions *)
val b_NAT    : string -> t -> bool
val b_RANDOM : t -> bool
val b_UMINUS : t -> t -> t
val b_UPLUS  : t -> t -> t
```



```
val b_PLUS   : t -> t -> t -> t * t
val b_MINUS  : t -> t -> t -> t * t
val b_TIMES  : t -> t -> t -> t * t
val b_DIV    : t -> t -> t -> t * t
val b_MOD    : t -> t -> t -> t * t
...
```



```
(* avalues.ml *)
open Values
(* abstraction of sets of machine integers by initialization *)
(* and  simple sign *)
(* complete lattice *)
type t = BOT | NEG | ZERO | POS | INI | ERR | TOP
(* \gamma(BOT)  = {_O_(a)}                                    *)
(* \gamma(NEG)  = [min_int,-1] U {_O_(a)}                     *)
(* \gamma(POS)  = [1,max_int] U {_O_(a)}                      *)
(* \gamma(ZERO) = {0, _O_(a)}                                 *)
(* \gamma(INI)  = [min_int,max_int] U {_O_(a)}                *)
(* \gamma(ERR)  = {_O_(i)}                                    *)
(* \gamma(TOP)  = [min_int,max_int] U {_O_(a),_O_(i)}         *)
...
(* backward abstract interpretation of arithmetic expressions *)
exception Error_f_NAT of string
let remove_zeros i =
  let l = (String.length i) in
  if l = 0 then raise (Error_f_NAT "empty integer")
```

```
  else if l = 1 then i
  else if (String.get i 0) = '0' then String.sub i 1  (l - 1)
  else i
let b_NAT i p =
  let i' = (remove_zeros i) in
  if i' = "0" then
    match p with
    | BOT  -> false
    | NEG  -> false
    | ZERO -> true
    | POS  -> false
    | INI  -> true
    | ERR  -> false
    | TOP  -> true
  else
    match p with
    | BOT  -> false
    | NEG  -> false
    | ZERO -> false
    | POS  -> true
```



```
    | INI  -> true
    | ERR  -> false
    | TOP  -> true
let b_RANDOM p =
  match p with
  | BOT -> false
  | ERR -> false
  | _   -> true
let b_UMINUS q p =
  match p with
  | BOT  -> BOT
  | NEG  -> meet q POS
  | ZERO -> meet q ZERO
  | POS  -> meet q NEG
  | INI  -> meet q INI
  | ERR  -> BOT
  | TOP  -> meet q INI
let b_UPLUS q p =
  match p with
  | BOT -> BOT
```



```
  | ERR -> BOT
  | TOP -> meet q INI
  | _   -> meet q p
exception Error_b_PLUS of string
let nat_of_lat' u =
  match u with
  | NEG  -> 0
  | ZERO -> 1
  | POS  -> 2
  | INI  -> 3
  | TOP  -> 4
  | _    -> raise (Error_b_PLUS "impossible selection")
let select' t u v = t.(nat_of_lat' u).(nat_of_lat' v)
let b_PLUS_NEG_table =
(*              NEG          ZERO         POS         INI          TOP      *)
(*NEG*)[|[| (NEG,NEG) ;  (NEG,ZERO) ; (NEG,POS) ; (NEG,INI) ;  (NEG,INI)  |];
(*ZERO*) [| (ZERO,NEG) ; (BOT,BOT) ;  (BOT,BOT) ; (ZERO,NEG) ; (ZERO,NEG) |];
(*POS*)  [| (POS,NEG) ;  (BOT,BOT) ;  (BOT,BOT) ; (POS,NEG) ;  (POS,NEG)  |];
(*INI*)  [| (INI,NEG) ;  (NEG,ZERO) ; (NEG,POS) ; (INI,INI) ;  (INI,INI)  |];
(*TOP*)  [| (INI,NEG) ;  (NEG,ZERO) ; (NEG,POS) ; (INI,INI) ;  (INI,INI)  |]|]
```



```
let b_PLUS_ZERO_table =
(*NEG*)[|[| (BOT,BOT) ; (BOT,BOT) ;   (NEG,POS) ; (NEG,POS) ;   (NEG,POS)   |];
(*ZERO*) [| (BOT,BOT) ; (ZERO,ZERO) ; (BOT,BOT) ; (ZERO,ZERO) ; (ZERO,ZERO) |];
(*POS*)  [| (POS,NEG) ; (BOT,BOT) ;   (BOT,BOT) ; (POS,NEG) ;   (POS,NEG)   |];
(*INI*)  [| (POS,NEG) ; (ZERO,ZERO) ; (NEG,POS) ; (INI,INI) ;   (INI,INI)   |];
(*TOP*)  [| (POS,NEG) ; (ZERO,ZERO) ; (NEG,POS) ; (INI,INI) ;   (INI,INI)   |]|]
let b_PLUS_POS_table =
(*NEG*)[|[| (BOT,BOT) ; (BOT,BOT) ;  (NEG,POS) ;  (NEG,POS) ;  (NEG,POS)  |];
(*ZERO*) [| (BOT,BOT) ; (BOT,BOT) ;  (ZERO,POS) ; (ZERO,POS) ; (ZERO,POS) |];
(*POS*)  [| (POS,NEG) ; (POS,ZERO) ; (POS,INI) ;  (POS,INI) ;  (POS,INI)  |];
(*INI*)  [| (POS,NEG) ; (POS,ZERO) ; (INI,POS) ;  (INI,INI) ;  (INI,INI)  |];
(*TOP*)  [| (POS,NEG) ; (POS,ZERO) ; (INI,POS) ;  (INI,INI) ;  (INI,INI)  |]|]
let b_PLUS q1 q2 p =
  if (q1=BOT)||(q1=ERR)||(q2=BOT)||(q2=ERR)||(p=BOT)||(p=ERR) then
    (BOT,BOT)
  else if (p=INI)||(p=TOP) then
    ((meet q1 INI),(meet q2 INI))
  else if p = NEG then  select' b_PLUS_NEG_table q1 q2
  else if p = ZERO then select' b_PLUS_ZERO_table q1 q2
  else if p = POS then  select' b_PLUS_POS_table q1 q2
```

```
  else raise (Error_b_PLUS "impossible case")
let b_MINUS q1 q2 p =
  let r1,r2 = b_PLUS q1 (f_UMINUS q2) p in
    r1,(f_UMINUS  r2)
let b_TIMES u v r = print_string "b_TIMES not yet implemented\n";
                    u,v (*  b_TIMES not yet implemented *)
let smash x y = if (x=BOT)||(y=BOT) then (BOT,BOT) else (x,y)
let b_DIV q1 q2 p =
  if (q1=BOT)||(q1=NEG)||(q1=ERR)||
     (q2=BOT)||(q2=NEG)||(q2=ZERO)||(q2=ERR)||
(p=BOT)||(p=NEG)||(p=ERR) then
(BOT,BOT)
  else if p = POS then
     (smash (meet q1 POS) (meet q2 POS))
  else
     (smash (meet q1 INI) (meet q2 POS))
let b_MOD = b_DIV
...
```


Improving the non-relational analysis of boolean expressions using the backward analysis of its arithmetic subexpressions

Revisiting the non-relational abstract interpretation of boolean expressions

The abstract interpretation of boolean expressions can be revised as follows, using the backward abstract interpretation of arithmetic expressions:

$$\begin{aligned}
\mathrm{Abexp}[\![B]\!]\lambda \mathrm{Y}\cdot\bot &\stackrel{\mathrm{def}}{=} \lambda \mathrm{Y}\cdot\bot \qquad \text{if } \gamma(\bot)=\emptyset \qquad (40)\\
\mathrm{Abexp}[\![\mathtt{true}]\!]r &\stackrel{\mathrm{def}}{=} r\\
\mathrm{Abexp}[\![\mathtt{false}]\!]r &\stackrel{\mathrm{def}}{=} \lambda \mathrm{Y}\cdot\bot\\
\mathrm{Abexp}[\![A_1 \ \mathtt{c}\ A_2]\!]r &\stackrel{\mathrm{def}}{=}\\
&\quad \text{let } \langle p_1,\ p_2\rangle = \check{\mathtt{c}}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in}\\
&\qquad \mathrm{Baexp}^{\triangleleft}[\![A_1]\!](r)p_1 \ \dot{\sqcap}\ \mathrm{Baexp}^{\triangleleft}[\![A_2]\!](r)p_2
\end{aligned}$$

$$\begin{aligned}
\mathrm{Abexp}[\![B_1 \ \&\ B_2]\!]r &\stackrel{\mathrm{def}}{=} \mathrm{Abexp}[\![B_1]\!]r \ \dot{\sqcap}\ \mathrm{Abexp}[\![B_2]\!]r\\
\mathrm{Abexp}[\![B_1 \mid B_2]\!]r &\stackrel{\mathrm{def}}{=} \mathrm{Abexp}[\![B_1]\!]r \ \dot{\sqcup}\ \mathrm{Abexp}[\![B_2]\!]r
\end{aligned}$$

parameterized by the following abstract comparison operations $\check{\mathtt{c}}$, $\mathtt{c} \in \{<, =\}$ on L

$$\begin{aligned}
\check{\mathtt{c}}(p_1, p_2) \sqsupseteq^2 \alpha^2(\{\langle i_1,\ i_2\rangle \mid\ & i_1 \in \gamma(p_1) \cap \mathbb{I}\\
& \wedge i_2 \in \gamma(p_2) \cap \mathbb{I} \wedge i_1 \ \underline{\mathtt{c}}\ i_2 = \mathtt{tt}\})
\end{aligned}$$

Calculational design of the revisited non-relational abstract interpretation of boolean expressions

PROOF. All cases have already be handled, except when $B = A_1 \mathrel{\mathsf{c}} A_2$ is an arithmetic comparison. Let us recall that

$$\frac{\rho \vdash A_1 \Mapsto v_1,\ \rho \vdash A_2 \Mapsto v_2}{\rho \vdash A_1 \mathrel{\mathsf{c}} A_2 \Mapsto v_1 \mathrel{\underline{\mathsf{c}}} v_2}, \qquad (41)$$

and

$$\begin{aligned} \Omega_e \mathrel{\underline{\mathsf{c}}} v &\stackrel{\text{def}}{=} \Omega_e, \\ i \mathrel{\underline{\mathsf{c}}} \Omega_e &\stackrel{\text{def}}{=} \Omega_e, \\ i_1 \mathrel{\underline{\mathsf{c}}} i_2 &\stackrel{\text{def}}{=} i_1 \mathrel{\mathsf{c}} i_2 \,. \end{aligned} \qquad (42)$$

We have

$$\begin{aligned}
& \ddot{\alpha}(\mathrm{Cbexp}[\![A_1 \mathrel{\mathsf{c}} A_2]\!])r \\
= \ & \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \rho \vdash A_1 \mathrel{\mathsf{c}} A_2 \Mapsto \mathtt{tt}\}) \\
= \ & \quad \wr \text{def. (41) of } \rho \vdash A_1 \mathrel{\mathsf{c}} A_2 \Mapsto b \int \\
= \ & \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists v_1, v_2 \in \mathbb{I}_\Omega : \rho \vdash A_1 \Mapsto v_1 \wedge \rho \vdash A_2 \Mapsto v_2 \wedge v_1 \mathrel{\underline{\mathsf{c}}} v_2 = \mathtt{tt}\}) \\
= \ & \quad \wr \text{set theory and } \gamma \circ \alpha \text{ is extensive } \int \\
= \ & \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists v_1 \in \gamma(\alpha(\{v \mid \exists \rho \in \dot{\gamma}(r) : \rho \vdash A_1 \Mapsto v\})) : \\
& \qquad \exists v_2 \in \gamma(\alpha(\{v \mid \exists \rho \in \dot{\gamma}(r) : \rho \vdash A_2 \Mapsto v\})) : \\
& \qquad\quad \rho \vdash A_1 \Mapsto v_1 \wedge \rho \vdash A_2 \Mapsto v_2 \wedge v_1 \mathrel{\underline{\mathsf{c}}} v_2 = \mathtt{tt}\}) \\
= \ & \quad \wr \text{set theory and (9)} \int \\
= \ & \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists v_1 \in \gamma(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r) : \exists v_2 \in \gamma(\mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) : \\
& \quad \rho \vdash A_1 \Mapsto v_1 \wedge \rho \vdash A_2 \Mapsto v_2 \wedge v_1 \mathrel{\underline{\mathsf{c}}} v_2 = \mathtt{tt}\}) \\
= \ & \quad \wr \text{let notation} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \langle \mathrm{Faexp}^{\triangleright}[\![A_1]\!]r,\ \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r\rangle \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists v_1 \in \gamma(p_1) : \exists v_2 \in \gamma(p_2) : \\
& \qquad \rho \vdash A_1 \Mapsto v_1 \wedge \rho \vdash A_2 \Mapsto v_2 \wedge v_1 \mathrel{\underline{\mathsf{c}}} v_2 = \mathtt{tt}\})
\end{aligned}$$

$$\begin{aligned}
= \ & \quad \wr \text{def. (42) of } \underline{\mathsf{c}} \text{ implying } v_1, v_2 \not\in \mathbb{E} = \{\Omega_{\mathrm{i}}, \Omega_{\mathrm{a}}\} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \langle \mathrm{Faexp}^{\triangleright}[\![A_1]\!]r,\ \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r\rangle \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists i_1 \in \gamma(p_1) \cap \mathbb{I} : \exists i_2 \in \gamma(p_2) \cap \mathbb{I} : \\
& \qquad \rho \vdash A_1 \Mapsto i_1 \wedge \rho \vdash A_2 \Mapsto i_2 \wedge i_1 \mathrel{\underline{\mathsf{c}}} i_2 = \mathtt{tt}\}) \\
= \ & \quad \wr \text{set theory} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \langle \mathrm{Faexp}^{\triangleright}[\![A_1]\!]r,\ \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r\rangle \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists \langle i_1,\ i_2\rangle \in \{\langle i_1',\ i_2'\rangle \mid i_1' \in \gamma(p_1) \cap \mathbb{I} \wedge i_2' \in \gamma(p_2) \cap \mathbb{I} \wedge i_1' \mathrel{\underline{\mathsf{c}}} i_2' = \mathtt{tt}\} : \\
& \qquad \rho \vdash A_1 \Mapsto i_1 \wedge \rho \vdash A_2 \Mapsto i_2 \wedge\}) \\
\dot{\sqsubseteq} \ & \quad \wr \gamma^2 \circ \alpha^2 \text{ extensive and } \dot{\alpha} \text{ monotone (4)} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \langle \mathrm{Faexp}^{\triangleright}[\![A_1]\!]r,\ \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r\rangle \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists \langle i_1,\ i_2\rangle \in \gamma^2(\alpha^2(\{\langle i_1',\ i_2'\rangle \mid i_1' \in \gamma(p_1) \cap \mathbb{I} \wedge i_2' \in \gamma(p_2) \cap \mathbb{I} \wedge \\
& \qquad i_1' \mathrel{\underline{\mathsf{c}}} i_2' = \mathtt{tt}\})) : \\
& \qquad \rho \vdash A_1 \Mapsto i_1 \wedge \rho \vdash A_2 \Mapsto i_2 \wedge\}) \\
\dot{\sqsubseteq} \ & \quad \wr \text{defining } \check{\mathsf{c}} \text{ such that:} \\
& \quad \check{\mathsf{c}}(p_1, p_2) \sqsupseteq^2 \alpha^2(\{\langle i_1',\ i_2'\rangle \mid i_1' \in \gamma(p_1) \cap \mathbb{I} \wedge i_2' \in \gamma(p_2) \cap \mathbb{I} \wedge i_1' \mathrel{\underline{\mathsf{c}}} i_2' = \mathtt{tt}\}), \\
& \quad \gamma^2 \text{ and } \dot{\alpha} \text{ monotone (4)} \int
\end{aligned}$$

$$\begin{aligned}
& \text{let } \langle p_1,\ p_2\rangle = \langle \mathrm{Faexp}^{\triangleright}[\![A_1]\!]r,\ \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r\rangle \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists \langle i_1,\ i_2\rangle \in \gamma^2(\check{\mathsf{c}}(p_1, p_2)) : \rho \vdash A_1 \Mapsto i_1 \wedge \rho \vdash A_2 \Mapsto i_2\}) \\
= \ & \quad \wr \text{let notation} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \check{\mathsf{c}}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists \langle i_1,\ i_2\rangle \in \gamma^2(\langle p_1,\ p_2\rangle)) : \quad \rho \vdash A_1 \Mapsto i_1 \wedge \rho \vdash A_2 \Mapsto i_2\}) \\
= \ & \quad \wr \text{set theory} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \check{\mathsf{c}}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists i_1 \in \gamma(p_1) : \rho \vdash A_1 \Mapsto i_1\} \cap \{\rho \in \dot{\gamma}(r) \mid \exists i_2 \in \gamma(p_2) : \rho \vdash A_2 \Mapsto i_2\}) \\
\dot{\sqsubseteq} \ & \quad \wr \dot{\alpha} \text{ monotone (4)} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \check{\mathsf{c}}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in} \\
& \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists i_1 \in \gamma(p_1) : \rho \vdash A_1 \Mapsto i_1\}) \ \dot{\sqcap} \\
& \qquad \dot{\alpha}(\{\rho \in \dot{\gamma}(r) \mid \exists i_2 \in \gamma(p_2) : \rho \vdash A_2 \Mapsto i_2\}) \\
= \ & \quad \wr \text{def. (19) of Baexp} \int \\
& \text{let } \langle p_1,\ p_2\rangle = \check{\mathsf{c}}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in} \\
& \quad \dot{\alpha}(\mathrm{Baexp}[\![A_1]\!](\dot{\gamma}(r))\gamma(p_1)) \ \dot{\sqcap} \ \dot{\alpha}(\mathrm{Baexp}[\![A_2]\!](\dot{\gamma}(r))\gamma(p_2)) \\
= \ & \quad \wr \text{def. (32) of } \alpha^{\triangleleft} \int
\end{aligned}$$

$$
\begin{array}{ll}
 & \text{let } \langle p_1,\ p_2\rangle = \check{c}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in} \\
 & \quad \alpha^{\triangleleft}(\mathrm{Baexp}[\![A_1]\!])(r)p_1 \ \dot{\sqcap}\ \alpha^{\triangleleft}(\mathrm{Baexp}[\![A_2]\!])(r)p_2 \\
\dot{\sqsubseteq} & \qquad \wr \text{def. (33) of } \mathrm{Baexp}^{\triangleleft} \text{ and } \dot{\sqcap} \text{ monotone} \wr \\
 & \text{let } \langle p_1,\ p_2\rangle = \check{c}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in} \\
 & \quad \mathrm{Baexp}^{\triangleleft}[\![A_1]\!](r)p_1 \ \dot{\sqcap}\ \mathrm{Baexp}^{\triangleleft}[\![A_2]\!](r)p_2 \\
= & \qquad \wr \text{by defining } \mathrm{Abexp}[\![A_1 \mathrel{c} A_2]\!]r \stackrel{\text{def}}{=} \\
 & \qquad\qquad \text{let } \langle p_1,\ p_2\rangle = \check{c}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \text{ in} \\
 & \qquad\qquad\quad \mathrm{Baexp}^{\triangleleft}[\![A_1]\!](r)p_1 \ \dot{\sqcap}\ \mathrm{Baexp}^{\triangleleft}[\![A_2]\!](r)p_2 \wr \\
 & \mathrm{Abexp}[\![A_1 \mathrel{c} A_2]\!]r\ .
\end{array}
$$

$\square$

Implementation of the revisited non-relational abstract interpretation of boolean expressions

```
(* abexp.mli *)
open Abstract_Syntax
(* abstract interpretation of boolean operations *)
val a_bexp : bexp -> Aenv.t -> Aenv.t
```



```
(* abexp.ml *)
open Abstract_Syntax
(* abstract interpretation of boolean operations *)
let rec a_bexp' b r =
  match b with
  | TRUE          -> r
  | FALSE         -> (Aenv.bot ())
  | EQ (a1, a2)   ->
    let (p1,p2) = (Avalues.a_EQ (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r))
     in (Aenv.meet (Baexp.b_aexp a1 r p1) (Baexp.b_aexp a2 r p2))
  | LT (a1, a2)   ->
    let (p1,p2) = (Avalues.a_LT (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r))
     in (Aenv.meet (Baexp.b_aexp a1 r p1) (Baexp.b_aexp a2 r p2))
  | AND (b1, b2) -> (Aenv.meet (a_bexp' b1 r) (a_bexp' b2 r))
  | OR (b1, b2)  -> (Aenv.join (a_bexp' b1 r) (a_bexp' b2 r))
let a_bexp b r =
  if (Aenv.is_bot r) & (Avalues.isbotempty ()) then (Aenv.bot ())
  else a_bexp' b r

```


Abstract arithmetic comparison operations for the initialization and simple sign analysis

– Generic abstract boolean equality.

The calculational design of the abstract equality operation $\check{=}$ does not depend upon the specific choice of L.

$$p_1 \mathrel{\check{=}} p_2 \stackrel{\text{def}}{=} \text{let } p = p_1 \sqcap p_2 \sqcap ?^{\triangleright} \text{ in } \langle p,\ p\rangle\ .$$

Proof.

$$
\begin{array}{l}
\alpha^2(\{\langle i_1,\ i_2\rangle \mid i_1 \in \gamma(p_1) \cap \mathbb{I} \wedge i_2 \in \gamma(p_2) \cap \mathbb{I} \wedge i_1 \mathrel{\underline{=}} i_2 = \mathtt{tt}\}) \\
\qquad \wr \text{def. (42) of } \underline{=} \wr \\
\alpha^2(\{\langle i,\ i\rangle \mid i \in \gamma(p_1) \cap \gamma(p_2) \cap \mathbb{I}\})
\end{array}
$$

$\sqsubseteq^2 \qquad \wr\,\gamma \circ \alpha$ is extensive and α^2 is monotone$\,\int$

$\quad \alpha^2(\{\langle i,\ i\rangle \mid i \in \gamma(p_1) \cap \gamma(p_2) \cap \gamma(\alpha(\mathbb{I}))\})$

$\qquad \wr\,\gamma$ preserves meets$\,\int$

$\quad \alpha^2(\{\langle i,\ i\rangle \mid i \in \gamma(p_1 \sqcap p_2 \sqcap \alpha(\mathbb{I}))\})$

$\qquad \wr\,$componentwise def. of $\gamma^2\,\int$

$\quad \alpha^2(\gamma^2(\langle p_1 \sqcap p_2 \sqcap \alpha(\mathbb{I}),\ p_1 \sqcap p_2 \sqcap \alpha(\mathbb{I})\rangle))$

$\sqsubseteq^2$

$\qquad \wr\,\alpha^2 \circ \gamma^2$ is reductive and let notation$\,\int$

$\quad$ let $p = p_1 \sqcap p_2 \sqcap \alpha(\mathbb{I})$ in $\langle p,\ p\rangle$

$\sqsubseteq^2 \qquad \wr\,$def. (16) of $?^{\triangleright}\,\int$

$\quad$ let $p = p_1 \sqcap p_2 \sqcap ?^{\triangleright}$ in $\langle p,\ p\rangle$

$= \qquad \wr\,$by defining $\check{=} \stackrel{\text{def}}{=}$ let $p = p_1 \sqcap p_2 \sqcap ?^{\triangleright}$ in $\langle p,\ p\rangle\,\int$

$\quad \check{=}$.

□

– Initialization and simple sign abstarct arithmetic comparison operations.

The abstract strict comparison

$$\check{\lesssim}(p_1, p_2)\ \sqsupseteq^2\ \alpha^2(\{\langle i_1,\ i_2\rangle \mid i_1 \in \gamma(p_1) \cap \mathbb{I} \wedge i_2 \in \gamma(p_2) \cap \mathbb{I} \wedge i_1 \underline{\leq} i_2 = \mathtt{tt}\}) \quad (43)$$

for initialization and simple sign analysis is as follows

	$\check{\lesssim}(p_1, p_2)$	p_2: BOT, ERR	NEG	ZERO	POS	INI, TOP
p_1	BOT, ERR	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩
	NEG	⟨BOT, BOT⟩	⟨NEG, NEG⟩	⟨NEG, ZERO⟩	⟨NEG, POS⟩	⟨NEG, INI⟩
	ZERO	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨ZERO, POS⟩	⟨ZERO, POS⟩
	POS	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨BOT, BOT⟩	⟨POS, POS⟩	⟨POS, POS⟩
	INI, TOP	⟨BOT, BOT⟩	⟨NEG, NEG⟩	⟨NEG, ZERO⟩	⟨INI, POS⟩	⟨INI, INI⟩

Proof. Let us consider a few typical cases.

– If $p_i \in \{\mathtt{BOT}, \mathtt{ERR}\}$ where $i = 1$ or $i = 2$ then $\gamma(p_i) \subseteq \mathbb{E} = \{\Omega_{\mathrm{i}}, \Omega_{\mathrm{a}}\}$ so that $\gamma(p_i) \cap \mathbb{I} = \emptyset$ and we get:

$\quad \alpha^2(\{\langle i_1,\ i_2\rangle \mid i_1 \in \gamma(p_1) \cap \mathbb{I} \wedge i_2 \in \gamma(p_2) \cap \mathbb{I} \wedge i_1 \underline{\leq} i_2 = \mathtt{tt}\})$

$= \quad \alpha^2(\emptyset)$

$= \qquad \wr\,$componentwise definition of α^2 and (17) of $\alpha\,\int$

$\quad \langle\mathtt{BOT},\ \mathtt{BOT}\rangle$

$\stackrel{\text{def}}{=} \qquad \wr\,$def. (43) of $\check{\lesssim}\,\int$

$\quad \check{\lesssim}(\mathtt{BOT}, \mathtt{BOT})$;

– For $\langle\mathtt{POS},\ \mathtt{ZERO}\rangle$, we have

$\quad \alpha^2(\{\langle i_1,\ i_2\rangle \mid i_1 \in \gamma(\mathtt{POS}) \cap \mathbb{I} \wedge i_2 \in \gamma(\mathtt{ZERO}) \cap \mathbb{I} \wedge i_1 \underline{\leq} i_2 = \mathtt{tt}\})$

$= \qquad \wr\,$def. (18) of γ and (42) of $\underline{\leq}\,\int$

$= \quad \alpha^2(\{\langle i_1,\ 0\rangle \mid i_1 \in [1, \mathtt{max\_int}] \wedge i_1 \leq 0\})$

$= \qquad \wr\,$set theory$\,\int$

$\quad \alpha^2(\emptyset)$

$= \qquad \wr\,$componentwise definition of α^2 and (17) of $\alpha\,\int$

$\quad \langle\mathtt{BOT},\ \mathtt{BOT}\rangle$

$\stackrel{\text{def}}{=} \qquad \wr\,$def. (43) of $\check{\lesssim}\,\int$

$\quad \check{\lesssim}(\mathtt{POS}, \mathtt{ZERO})$.

– For $\langle\mathtt{TOP},\ \mathtt{TOP}\rangle$, we have

$\quad \alpha^2(\{\langle i_1,\ i_2\rangle \mid i_1 \in \gamma(\mathtt{TOP}) \cap \mathbb{I} \wedge i_2 \in \gamma(\mathtt{TOP}) \cap \mathbb{I} \wedge i_1 \underline{\leq} i_2 = \mathtt{tt}\})$

$= \qquad \wr\,$def. (18) of γ and (42) of $\underline{\leq}\,\int$

s $\quad \alpha^2(\{\langle i_1,\ i_2\rangle \mid i_1 \in \mathbb{I} \wedge i_2 \in \mathbb{I} \wedge i_1 \leq i_2\})$

$= \qquad \wr\,$componentwise definition of $\alpha^2\,\int$

$\quad \langle\alpha(\mathbb{I}),\ \alpha(\mathbb{I})\rangle$

$$
\begin{aligned}
= \quad & \wr \text{def. (17) of } \alpha \int \\
& \langle \mathtt{INI},\ \mathtt{INI} \rangle \\
\stackrel{\text{def}}{=} \quad & \wr \text{def. (43) of } \breve{\leq} \int \\
& \breve{<}(\mathtt{TOP}, \mathtt{TOP})\ .
\end{aligned}
$$

□

Implementation of the abstract arithmetic comparison operations for the initialization and simple sign analysis

```
(* avalues.mli *)
(* abstraction of sets of machine integers by initialization *)
(* and simple sign                                           f*)
type t
...
(* abstract interpretation of boolean expressions *)
val a_EQ     : t -> t -> t * t
val a_LT     : t -> t -> t * t
```



```
(* avalues.ml *)
open Values
(* abstraction of sets of machine integers by initialization *)
(* and  simple sign *)
(* complete lattice *)
type t = BOT | NEG | ZERO | POS | INI | ERR | TOP
(* \gamma(BOT)  = {_O_(a)}                                  *)
(* \gamma(NEG)  = [min_int,-1] U {_O_(a)}                   *)
(* \gamma(POS)  = [1,max_int] U {_O_(a)}                    *)
(* \gamma(ZERO) = {0, _O_(a)}                               *)
(* \gamma(INI)  = [min_int,max_int] U {_O_(a)}              *)
(* \gamma(ERR)  = {_O_(i)}                                  *)
(* \gamma(TOP)  = [min_int,max_int] U {_O_(a),_O_(i)}       *)
(* infimum *)
let bot () = BOT
(* bottom is emptyset? *)
let isbotempty () = false (* \gamma(BOT) = {_O_(a)} <> \emptyset *)
...
(* boolean expressions *)
```



```
let a_EQ p1 p2 =
  let p = (meet p1 (meet p2 (f_RANDOM ()))) in
    (p,p)
let a_LT_table =[|
(*     <      BOT       NEG       ZERO       POS        INI        ERR       TOP     *)
(*BOT*) [|(BOT,BOT);(BOT,BOT);(BOT,BOT) ;(BOT,BOT) ;(BOT,BOT) ;(BOT,BOT);(BOT,BOT) |];
(*NEG*) [|(BOT,BOT);(NEG,NEG);(NEG,ZERO);(NEG,POS) ;(NEG,INI) ;(BOT,BOT);(NEG,INI) |];
(*ZERO*)[|(BOT,BOT);(BOT,BOT);(BOT,BOT) ;(ZERO,POS);(ZERO,POS);(BOT,BOT);(ZERO,POS)|];
(*POS*) [|(BOT,BOT);(BOT,BOT);(BOT,BOT) ;(POS,POS) ;(POS,POS) ;(BOT,BOT);(POS,POS) |];
(*INI*) [|(BOT,BOT);(NEG,NEG);(NEG,ZERO);(INI,POS) ;(INI,INI) ;(BOT,BOT);(INI,INI) |];
(*ERR*) [|(BOT,BOT);(BOT,BOT);(BOT,BOT) ;(BOT,BOT) ;(BOT,BOT) ;(BOT,BOT);(BOT,BOT) |];
(*TOP*) [|(BOT,BOT);(NEG,NEG);(NEG,ZERO);(INI,POS) ;(INI,INI) ;(BOT,BOT);(INI,INI) |]
|]
let a_LT   u v = select a_LT_table u v
```

Back to the motivating example

Local decreasing iterations


```
% cd Initialization-Simple-Sign-FB
% ./a.out ../Examples/example13.sil
{ y:ERR; r:ERR }
0:
  y := ?;
1:
  if (y = 0) then
    2:
      r := 0
    3:
  else {((y < 0) | (0 < y))}
    4:
      r := 0
    5:
  fi
6:
{ y:INI; r:ZERO }
```

```
% cd Initialization-Simple-Sign-FB
% ./a.out ../Examples/example14.sil
{ y:ERR; r:ERR }
0:
  y := ?;
1:
  if (y = 0) then
    2:
      r := y
    3:
  else {((y < 0) | (0 < y))}
    4:
      r := 0
    5:
  fi
6:
{ y:INI; r:ZERO }
```


Motivating example

```
% cd Initialization-Simple-Sign-FB
% ./a.out ../Examples/example15.sil
0: { x:ERR; y:ERR; z:ERR; r:ERR }
  x := 0; y := ?; z := ?;
  if ((x = y) & (y = z)) then
    4:
      r := z
    5:
  else {(((x < y) | (y < x)) | ((y < z) | (z < y)))}
    6:
      r := 0
    7:
  fi
8: { x:ZERO; y:INI; z:INI; r:INI }
```

- In this example, the test `(x = y)` yields the information that `y = 0`. Independently, the test `(y = z)` brings no new information, on `y` and `z`. The conjunction is

  `{ x:ZERO; y:ZERO; z:INI; r:TOP }`
- The same analyis, starting from this valid information yields `z = 0`.
- More generally, the analysis of the tests should be iterated until no new information can be brought in.
- The idea is formalized by noticing that the analysis of tests is a lower closure operator.

A theorem on the composition of lower closure operators and Galois connections

THEOREM. If $\langle P, \leq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle Q, \sqsubseteq\rangle$ and ρ is a lower closure operator on P (monotone, idempotent and reductive) then $\alpha \circ \rho \circ \gamma$ is a lower closure operator on Q. ■

PROOF. – Since $\alpha \circ \rho \circ \gamma$ is the composition of monotone operators, it is monotone

- Since ρ is reductive i.e. $\forall x \in P : \rho(x) \leq x$, we have in particular $\forall y \in Q : \rho(\gamma(y)) \leq \gamma(y)$ whence by monotony $\forall y \in Q : \alpha(\rho(\gamma(y))) \sqsubseteq \alpha(\gamma(y)) \sqsubseteq y$ since $\alpha \circ \gamma$ is reductive in a Galois connection. By transitivity, $forally \in Q : \alpha \circ \rho \circ \gamma(y) \sqsubseteq y$ proving $\alpha \circ \rho \circ \gamma$ to be reductive.

- In particular, for $y = \alpha \circ \rho \circ \gamma(z)$, we have $\forall z \in Q : \alpha \circ \rho \circ \gamma(\alpha \circ \rho \circ \gamma(z) \sqsubseteq \alpha \circ \rho \circ \gamma(z)$ hence $\alpha \circ \rho \circ \gamma \circ \alpha \circ \rho \circ \gamma \mathrel{\dot\sqsubseteq} \alpha \circ \rho \circ \gamma$
- Moreover

$$\begin{array}{ll}
\rho \circ \gamma \mathrel{\dot\leq} \gamma \circ \alpha \circ \rho \circ \gamma & \wr \gamma \circ \alpha \text{ is extensive} \wr \\
\rho \circ \rho \circ \gamma \mathrel{\dot\leq} \rho \circ \gamma \circ \alpha \circ \rho \circ \gamma & \wr \rho \text{ is monotone} \wr \\
\rho \circ \gamma \mathrel{\dot\leq} \rho \circ \gamma \circ \alpha \circ \rho \circ \gamma & \wr \rho \text{ is idempotent} \wr \\
\alpha \circ \rho \circ \gamma \mathrel{\dot\sqsubseteq} \alpha \circ \rho \circ \gamma \circ \alpha \circ \rho \circ \gamma & \wr \alpha \text{ monotone} \wr
\end{array}$$

- By antisymmetry, we conclude that $\alpha \circ \rho \circ \gamma = \alpha \circ \rho \circ \gamma \circ \alpha \circ \rho \circ \gamma$ □

Abstraction of lower closure operators

- This theorem shows that whenever we have a lower closure operator in the concrete (e.g. the analysis of boolean expressions) then its abstraction $\alpha \circ \rho \circ \gamma$ is also a lower closure operator in the abstract
- Therefore, the abstract interpretation $\overline{\rho}$ of the operator $\alpha \circ \rho \circ \gamma \mathrel{\dot\sqsubseteq} \overline{\rho}$ can always be chosen to be a lower closure operator
- In this case, if $\overline{\rho}$ is not the best abstraction of ρ, that is $\alpha \circ \rho \circ \gamma \mathrel{\dot\sqsubset} \overline{\rho}$, $\overline{\rho}$ can be improved by *local decreasing iterations* [1].

References

[1] Philippe Granger. "Improving the Results of Static Analyses Programs by Local Decreasing Iteration". FSTTCS 1992: 68–79.

Local decreasing iterations

THEOREM. If $\langle M, \preceq\rangle$ is poset, $f \in M \mapsto M$ is monotone and idempotent, $\langle M, \preceq\rangle \xleftrightarrow[\alpha]{\gamma} \langle L, \sqsubseteq\rangle$ is a Galois connection, $\langle L, \sqsubseteq, \sqcap\rangle$ is a dual dcpo, $g \in L \mapsto L$ is monotone and reductive and $\alpha \circ f \circ \gamma \mathrel{\dot\sqsubseteq} g$ then the lower closure operator $g^\star \stackrel{\text{def}}{=} \lambda x \cdot \mathbf{gfp}_x^{\sqsubseteq}\, g$ is a better abstract interpretation of f than g

$$\alpha \circ f \circ \gamma \mathrel{\dot\sqsubseteq} g^\star \mathrel{\dot\sqsubseteq} g\ .$$

■

PROOF. For all $x \in L$, $g(x) \sqsubseteq x$, so that by monotony the sequence $g^0(x) \stackrel{\text{def}}{=} x$, $g^{\delta+1}(x) \stackrel{\text{def}}{=} g(g^\delta(x))$ for all successor ordinals $\delta+1$ and $g^\lambda \stackrel{\text{def}}{=} \bigsqcap_{\delta<\lambda} g^\delta(x)$ for all limit ordinals λ is a well-defined decreasing chain in the dual dcpo $\langle L, \sqsubseteq, \sqcap\rangle$ whence ultimately stationary. It converges to g^ϵ where ϵ is the order of g, which is the greatest fixpoint $g^\epsilon = \mathbf{gfp}_x^{\sqsubseteq}\, g$ of g which is $\sqsubseteq$-less than x. It follows that $g^\star \stackrel{\text{def}}{=} \mathbf{gfp}_P^{\sqsubseteq}\, xg$ is the greatest lower closure operator $\dot\sqsubseteq$-less than g. In particular $g^\star \mathrel{\dot\sqsubseteq} g$.

We have $\alpha \circ f \circ \gamma(x) \sqsubseteq g^1(x) = g(x) \sqsubseteq x = g^0(x)$. If $\alpha \circ f \circ \gamma(x) \sqsubseteq g^\delta(x)$ then

$\quad \alpha \circ f \circ \gamma(x)$

$= \quad \wr f$ idempotent$\wr$

$\quad \alpha \circ f \circ \gamma \circ \alpha \circ f \circ \gamma(x)$

$\sqsubseteq \quad \wr \alpha \circ f \circ \gamma(x) \sqsubseteq g^\delta(x)$ by induction hypothesis, γ, f and α are monotone$\wr$

$\quad \alpha \circ f \circ \gamma(g^\delta(x))$

$\sqsubseteq \quad \wr \alpha \circ f \circ \gamma \mathrel{\dot\sqsubseteq} g$ hypothesis$\wr$

$\quad g(g^\delta(x))$

$= \quad \wr$def. $g^{\delta+1}(x)\wr$

$\quad g^{\delta+1}(x)$.

If $\alpha \circ f \circ \gamma(x) \sqsubseteq g^\delta(x)$ for all $\delta < \lambda$ and λ is a limit ordinal then by definition of lubs and g^λ, we have $\alpha \circ f \circ \gamma(x) \sqsubseteq \bigsqcap_{\delta<\lambda} g^\delta(x) = g^\lambda$. By transfinite induction, $\alpha \circ f \circ \gamma(x) \sqsubseteq g^\epsilon(x) = g^\star(x)$. □

Note: when $\langle L, \sqsubseteq\rangle$ does not satisfy the decreasing chain condition then a narrowing must be used to enseure convergence.

The forward/bottom-up collecting semantics of boolean expressions is a lower closure operator

Recall the *collecting semantics* $\text{Cbexp}[\![B]\!]R$ of a boolean expression B from course 8:

$$\text{Cbexp}[\![B]\!]R \stackrel{\text{def}}{=} \{\rho \in R \mid \rho \vdash B \Rrightarrow \mathtt{tt}\}\ . \qquad (44)$$

THEOREM. $\text{Cbexp}[\![B]\!]$ is a lower closure operator. ■

Proof. – $\mathrm{Cbexp}[\![B]\!]$ is monotone: if $R_1 \subseteq R_2$ then $\{\rho \in R_1 \mid \rho \vdash B \Rightarrow \mathtt{tt}\} \subseteq \{\rho \in R_2 \mid \rho \vdash B \Rightarrow \mathtt{tt}\}$ and so $\mathrm{Cbexp}[\![B]\!]R_1 \subseteq \mathrm{Cbexp}[\![B]\!]R_2$

– $\mathrm{Cbexp}[\![B]\!]$ is reductive: $\mathrm{Cbexp}[\![B]\!]R = \{\rho \in R \mid \rho \vdash B \Rightarrow \mathtt{tt}\} \subseteq R$

– $\mathrm{Cbexp}[\![B]\!]$ is idempotent: $\mathrm{Cbexp}[\![B]\!](\mathrm{Cbexp}[\![B]\!]R) = \{\rho' \in \{\rho \in R \mid \rho \vdash B \Rightarrow \mathtt{tt}\} \mid \rho' \vdash B \Rightarrow \mathtt{tt}\} = \{\rho' \in R \mid \rho' \vdash B \Rightarrow \mathtt{tt}\} = \mathrm{Cbexp}[\![B]\!]R$

$\square$

The forward/top-down nonrelational abstract semantics of arithmetic expressions is monotone

Recall the forward/top-down nonrelational abstract semantics of arithmetic expressions

$$
\begin{aligned}
\mathrm{Faexp}^{\triangleright}[\![A]\!](\lambda \mathtt{Y} \cdot \bot) &\stackrel{\text{def}}{=} \bot \qquad \text{if } \gamma(\bot) = \emptyset \\
\mathrm{Faexp}^{\triangleright}[\![\mathtt{n}]\!]r &\stackrel{\text{def}}{=} \mathtt{n}^{\triangleright} \\
\mathrm{Faexp}^{\triangleright}[\![\mathtt{X}]\!]r &\stackrel{\text{def}}{=} r(\mathtt{X}) \\
\mathrm{Faexp}^{\triangleright}[\![?]\!]r &\stackrel{\text{def}}{=} ?^{\triangleright} \\
\mathrm{Faexp}^{\triangleright}[\![\mathtt{u}\ A']\!]r &\stackrel{\text{def}}{=} \mathtt{u}^{\triangleright}(\mathrm{Faexp}^{\triangleright}[\![A']\!]r) \\
\mathrm{Faexp}^{\triangleright}[\![A_1\ \mathtt{b}\ A_2]\!]r &\stackrel{\text{def}}{=} \mathtt{b}^{\triangleright}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r)
\end{aligned} \tag{45}
$$

parameterized by the following forward abstract operations

$$\mathtt{n}^{\triangleright} = \alpha(\{\underline{\mathtt{n}}\}) \tag{46}$$

$$?^{\triangleright} \sqsupseteq \alpha(\mathbb{I}) \tag{47}$$

$$\mathtt{u}^{\triangleright}(p) \sqsupseteq \alpha(\{\underline{\mathtt{u}}\, v \mid v \in \gamma(p)\}) \tag{48}$$

$$\mathtt{b}^{\triangleright}(p_1, p_2) \sqsupseteq \alpha(\{v_1\ \underline{\mathtt{b}}\ v_2 \mid v_1 \in \gamma(p_1) \wedge v_2 \in \gamma(p_2)\}) \tag{49}$$

Theorem. If $\mathtt{u}^{\triangleright} \in L \stackrel{\mathrm{m}}{\longmapsto} L$ and $\mathtt{b}^{\triangleright} \in L \times L \stackrel{\mathrm{m}}{\longmapsto} L$ then $\mathrm{Faexp}^{\triangleright}[\![A]\!]$ is monotone. ■

Proof. – In the case $\dot\bot \mathrel{\dot\sqsubseteq} r$, we have $\mathrm{Faexp}^{\triangleright}[\![A]\!]\dot\bot \stackrel{\text{def}}{=} \bot \sqsubseteq \mathrm{Faexp}^{\triangleright}[\![A]\!]r$

(a) If $r_1 \mathrel{\dot\sqsubseteq} r_2$ then

(b) If $r_1 \mathrel{\dot\sqsubseteq} r_2$ then by reflexivity $\mathrm{Faexp}^{\triangleright}[\![\mathtt{n}]\!]r_1 \stackrel{\text{def}}{=} \mathtt{n}^{\triangleright} \sqsubseteq \mathtt{n}^{\triangleright} \stackrel{\text{def}}{=} \mathrm{Faexp}^{\triangleright}[\![\mathtt{n}]\!]r_2$

(c) If $r_1 \mathrel{\dot\sqsubseteq} r_2$ then by pointwise def. of $\dot\sqsubseteq$, we have $\mathrm{Faexp}^{\triangleright}[\![\mathtt{X}]\!]r_1 \stackrel{\text{def}}{=} r_1(\mathtt{X}) \sqsubseteq r_2(\mathtt{X}) \stackrel{\text{def}}{=} \mathrm{Faexp}^{\triangleright}[\![\mathtt{X}]\!]r_2$

(d) If $r_1 \mathrel{\dot\sqsubseteq} r_2$ then by reflexivity $\mathrm{Faexp}^{\triangleright}[\![?]\!]r_1 \stackrel{\text{def}}{=} ?^{\triangleright} \sqsubseteq ?^{\triangleright} \stackrel{\text{def}}{=} \mathrm{Faexp}^{\triangleright}[\![?]\!]r_2$

(e) By induction hypothesis, $\mathrm{Faexp}^{\triangleright}[\![\mathtt{u}\ A']\!]$ is monotonic ans so is $\mathtt{u}^{\triangleright}$ by hypothesis whence is their composition $\mathtt{u}^{\triangleright} \circ \mathrm{Faexp}^{\triangleright}[\![A']\!] \stackrel{\text{def}}{=} \mathrm{Faexp}^{\triangleright}[\![\mathtt{u}\ A']\!]$

(f) By induction hypothesis, $(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]$ and $(\mathrm{Faexp}^{\triangleright}[\![A_2]\!]$ are monotonic and so is $\mathtt{b}^{\triangleright}$ by hypothesis and so is their composiition $\lambda r \cdot \mathtt{b}^{\triangleright}(\mathrm{Faexp}^{\triangleright}[\![A_1]\!]r, \mathrm{Faexp}^{\triangleright}[\![A_2]\!]r) \stackrel{\text{def}}{=} \lambda r \cdot \mathrm{Faexp}^{\triangleright}[\![A_1\ \mathtt{b}\ A_2]\!]r$

$\square$

The forward/top-down nonrelational abstract semantics of boolean expressions is monotone and reductive

We have defined

$$\text{Abexp}[\![B]\!] \mathrel{\ddot{\sqsupseteq}} \ddot{\alpha}(\text{Cbexp}[\![B]\!]) \qquad (50)$$

such that

$$\begin{array}{rcll}
\text{Abexp}[\![B]\!]\dot{\bot} & \stackrel{\text{def}}{=} & \dot{\bot} & \text{if } \gamma(\bot) = \emptyset \\
\text{Abexp}[\![\mathtt{true}]\!]r & \stackrel{\text{def}}{=} & r & \\
\text{Abexp}[\![\mathtt{false}]\!]r & \stackrel{\text{def}}{=} & \dot{\bot} & \\
\text{Abexp}[\![A_1 \mathrel{\mathsf{c}} A_2]\!]r & \stackrel{\text{def}}{=} & \check{\mathsf{c}}(\text{Faexp}^{\triangleright}[\![A_1]\!]r, \text{Faexp}^{\triangleright}[\![A_2]\!]r) & \\
\text{Abexp}[\![B_1 \mathbin{\&} B_2]\!]r & \stackrel{\text{def}}{=} & \text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqcap}} \text{Abexp}[\![B_2]\!]r & \\
\text{Abexp}[\![B_1 \mid B_2]\!]r & \stackrel{\text{def}}{=} & \text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqcup}} \text{Abexp}[\![B_2]\!]r &
\end{array} \qquad (51)$$

parameterized by the following abstract comparison operations $\check{\mathsf{c}}$, $\mathsf{c} \in \{<, =\}$ on L

$$\check{\mathsf{c}}(p_1, p_2)r \stackrel{\text{def}}{=} (\!|\, \breve{B}(p_1, p_2) \;?\; r \;¿\; \dot{\bot} \,|\!) \qquad (52)$$

$$\breve{B}(p_1, p_2) \Longrightarrow \exists v_1 \in \gamma(p_1) : \exists v_2 \in \gamma(p_2) \cap \mathbb{I} : v_1 \mathrel{\underline{\mathsf{c}}} v_2 = \mathtt{tt}$$

THEOREM. $\text{Abexp}[\![B]\!]$ is reductive and if $\breve{B} \in \langle L, \sqsubseteq\rangle \times \langle L, \sqsubseteq\rangle \stackrel{\text{m}}{\longmapsto} \langle \mathbb{B}, \Longrightarrow\rangle$ is monotonic then $\text{Abexp}[\![B]\!]$ is monotonic. ■

PROOF. – $\text{Abexp}[\![B]\!]$ is reductive. The proof is by structural induction on B.

(a) For the infimum $\dot{\bot} \mathrel{\dot{\sqsupseteq}} \dot{\bot} \stackrel{\text{def}}{=} \text{Abexp}[\![B]\!]\dot{\bot}$

(b) By reflexivity $r \mathrel{\dot{\sqsupseteq}} r \stackrel{\text{def}}{=} \text{Abexp}[\![\mathtt{true}]\!]r$

(c) For the infimum $r \mathrel{\dot{\sqsupseteq}} \dot{\bot} \stackrel{\text{def}}{=} \text{Abexp}[\![\mathtt{false}]\!]r$

(d) If $\breve{B}(p_1, p_2)$ holds then $r \mathrel{\dot{\sqsupseteq}} r \stackrel{\text{def}}{=} \check{\mathsf{c}}(p_1, p_2)$. Otherwise $\neg(\breve{B}(p_1, p_2))$ holds and so $r \mathrel{\dot{\sqsupseteq}} \dot{\bot} \stackrel{\text{def}}{=} \check{\mathsf{c}}(p_1, p_2)$. So $\check{\mathsf{c}}(p_1, p_2)$ is reductive for all p_1 and p_2 and so in particular $\text{Abexp}[\![A_1 \mathrel{\mathsf{c}} A_2]\!]r \stackrel{\text{def}}{=} \check{\mathsf{c}}(\text{Faexp}^{\triangleright}[\![A_1]\!]r, \text{Faexp}^{\triangleright}[\![A_2]\!]r)$ is reductive

(e) By induction hypothesis, $\text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqsubseteq}} r$ and $\text{Abexp}[\![B_2]\!]r \mathrel{\dot{\sqsubseteq}} r$ so $\text{Abexp}[\![B_1 \mathbin{\&} B_2]\!]r \stackrel{\text{def}}{=} \text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqcap}} \text{Abexp}[\![B_2]\!]r \mathrel{\dot{\sqsubseteq}} r \mathrel{\dot{\sqcap}} r = r$ by def. glb

(f) Similarly, $\text{Abexp}[\![B_1 \mid B_2]\!]r \stackrel{\text{def}}{=} \text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqcup}} \text{Abexp}[\![B_2]\!]r \mathrel{\dot{\sqsubseteq}} r \mathrel{\dot{\sqcup}} r = r$ by ind. hyp. and def. lub.

– $\text{Abexp}[\![B]\!]$ is monotone. The proof is by structural induction on B.

(a) if $\dot{\bot} \mathrel{\dot{\sqsubseteq}} r$ then $\text{Abexp}[\![B]\!]\dot{\bot} \stackrel{\text{def}}{=} \dot{\bot} \mathrel{\dot{\sqsubseteq}} \text{Abexp}[\![B]\!]r$

(b) $\text{Abexp}[\![\mathtt{true}]\!]$ is the identity, which is monotone

(c) $\text{Abexp}[\![\mathtt{false}]\!]$ is a constant function, which is monotone

(d) If $r_1 \mathrel{\dot{\sqsubseteq}} r_2$ then $\text{Faexp}^{\triangleright}[\![A_i]\!]r_1 \sqsubseteq \text{Faexp}^{\triangleright}[\![A_i]\!]r_2$ since $\text{Faexp}^{\triangleright}[\![A_i]\!]r$, $i = 1, 2$ has been shown to be monotone. It follows by monotony of $\breve{B}$ that if $r_1 \mathrel{\dot{\sqsubseteq}} r_2$ then $\breve{B}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_1, \text{Faexp}^{\triangleright}[\![A_2]\!]r_1) \Longrightarrow \breve{B}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_2, \text{Faexp}^{\triangleright}[\![A_2]\!]r_2)$ and so, by cases:

· If $\breve{B}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_1, \text{Faexp}^{\triangleright}[\![A_2]\!]r_1) = \mathtt{tt}$ then $\breve{B}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_2, \text{Faexp}^{\triangleright}[\![A_2]\!]r_2) = \mathtt{tt}$ in which case $\text{Abexp}[\![A_1 \mathrel{\mathsf{c}} A_2]\!]r_1 \stackrel{\text{def}}{=} \check{\mathsf{c}}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_1, \text{Faexp}^{\triangleright}[\![A_2]\!]r_1) = r_1 \mathrel{\dot{\sqsubseteq}} r_2 = \check{\mathsf{c}}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_2, \text{Faexp}^{\triangleright}[\![A_2]\!]r_2) \stackrel{\text{def}}{=} \text{Abexp}[\![A_1 \mathrel{\mathsf{c}} A_2]\!]r_2$

· Otherwise, $\breve{B}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_1, \text{Faexp}^{\triangleright}[\![A_2]\!]r_1) = \mathtt{ff}$ and so $\text{Abexp}[\![A_1 \mathrel{\mathsf{c}} A_2]\!]r_1 \stackrel{\text{def}}{=} \check{\mathsf{c}}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_1, \text{Faexp}^{\triangleright}[\![A_2]\!]r_1) = \dot{\bot} \mathrel{\dot{\sqsubseteq}} r_2 = \check{\mathsf{c}}(\text{Faexp}^{\triangleright}[\![A_1]\!]r_2, \text{Faexp}^{\triangleright}[\![A_2]\!]r_2) \stackrel{\text{def}}{=} \text{Abexp}[\![A_1 \mathrel{\mathsf{c}} A_2]\!]r_2$

(e,f) By induction hypothesis, $\text{Abexp}[\![B_1]\!]$ and $\text{Abexp}[\![B_2]\!]$ are monotone, as well as $\dot{\sqcap}$ and $\dot{\sqcup}$ and so are their composition $\text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqcap}} \text{Abexp}[\![B_2]\!]r \stackrel{\text{def}}{=} \text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqcap}} \text{Abexp}[\![B_2]\!]r$ and $\text{Abexp}[\![B_1]\!]r \mathrel{\dot{\sqcup}} \text{Abexp}[\![B_2]\!]r \stackrel{\text{def}}{=} \text{Abexp}[\![B_1 \mid B_2]\!]r$

□

Reductive iterations for boolean expressions

Reductive iteration has a direct application to the analysis of boolean expressions. The abstract interpretation $\text{Abexp}[\![B]\!]$ (12) of boolean expressions B can always be replaced by its reductive iteration $\text{Abexp}[\![B]\!]^\star$ which is sound (11) and always more precise. By the local decreasing iterations Th. (page 85), we have

$$\ddot{\alpha}(\text{Cbexp}[\![B]\!]) \mathrel{\ddot{\sqsubseteq}} \text{Abexp}[\![B]\!]^\star \mathrel{\ddot{\sqsubseteq}} \text{Abexp}[\![B]\!] \ .$$

PROOF. – $\text{Cbexp}[\![B]\!]$ is a lower closure operator

– $\text{Abexp}[\![B]\!]$ is monotone and reductive (but not idempotent as shown for the motivating example)

– So $\text{Abexp}[\![B]\!]^\star = \lambda x \cdot \mathbf{gfp}_x^{\dot{\sqsubseteq}} \text{Abexp}[\![B]\!]x$ is a better abstraction of $\text{Cbexp}[\![B]\!]$ than $\text{Abexp}[\![B]\!]$ □

Implementation of the reductive iterations for abstract interrpetation of boolean expressions

```
(* fixpoint.mli *)
open Aenv
(* lfp x c f : iterative computation of the c-least fixpoint of f *)
(* c-greater than or equal to the prefixpoint x (f(x) >= x)       *)
val lfp : t -> (t -> t -> bool) -> (t -> t) -> t
(* gfp x c f : iterative computation of the c-greateast fixpoint  *)
(* of f c-less than or equal to the postfixpoint x (f(x) <= x)    *)
val gfp : t -> (t -> t -> bool) -> (t -> t) -> t
(* fixpoint.ml *)
open Aenv
(* lfp x c f : iterative computation of the c-least fixpoint of f *)
(* c-greater than or equal to the prefixpoint x (f(x) >= x)       *)
let rec lfp x c f =
```



```
  let x' = (f x) in
    if (c x' x) then x'
      else lfp x' c f
(* gfp x c f : iterative computation of the c-greatest fixpoint of *)
(* f, c-less than or equal to the postfixpoint x (f(x) <= x)       *)
let gfp x c f =
  let c_1 a b = c b a in
    lfp x c_1 f
(* abexp.ml *)
open Abstract_Syntax
open Fixpoint
(* abstract interpretation of boolean operations with iterative *)
(* reduction                                                    *)
let rec a_bexp' b r =
 match b with
 | TRUE          -> r
 | FALSE         -> (Aenv.bot ())
 | EQ (a1, a2)   ->
```



```
   let (p1,p2) = (Avalues.a_EQ (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r))
    in (Aenv.meet (Baexp.b_aexp a1 r p1) (Baexp.b_aexp a2 r p2))
 | LT (a1, a2)   ->
   let (p1,p2) = (Avalues.a_LT (Aaexp.a_aexp a1 r) (Aaexp.a_aexp a2 r))
    in (Aenv.meet (Baexp.b_aexp a1 r p1) (Baexp.b_aexp a2 r p2))
 | AND (b1, b2) -> (Aenv.meet (a_bexp' b1 r) (a_bexp' b2 r))
 | OR (b1, b2)  -> (Aenv.join (a_bexp' b1 r) (a_bexp' b2 r))
let a_bexp b r =
  if (Aenv.is_bot r) & (Avalues.isbotempty ()) then (Aenv.bot ())
  else gfp r Aenv.leq (a_bexp' b)
```

Motivating example ... revisited

```
% cd Initialization-Simple-Sign-FB-LDI-B
% ./a.out ../Examples/example15.sil
0: { x:ERR; y:ERR; z:ERR; r:ERR }
  x := 0; y := ?; z := ?;
  if ((x = y) & (y = z)) then
    4:
      r := z
    5:
  else {(((x < y) | (y < x)) | ((y < z) | (z < y)))}
    6:
      r := 0
    7:
  fi
8: { x:ZERO; y:INI; z:INI; r:ZERO }
```


Improving the non-relational analysis of assignments using the backward analysis of its righthandside arithmetic expression

Motivating example

```
% cd Initialization-Simple-Sign-FB-LDI-B
% ./a.out ../Examples/example18.sil
{ x:ERR; y:ERR }
0:
  x := ?;
1:
  y := (1 / x)
2:
{ x:INI; y:INI }
```


Although the program execution is blocked at line 1: when x ≤ 0 (the division requires its second argument to be strictly positive), this fact is not taken into account ar line 2: since the abstract assignment

$$\mathrm{Acom}[\![\mathtt{X} := A]\!]R(\mathrm{after}_P[\![\mathtt{X} := A]\!]) = R[\mathtt{X} := \mathrm{Faexp}[\![A]\!](R) \sqcap ?^{\triangleright}]$$

does not restricts the values of variables (other than X) in environment R to those for which the expression A is well-defined. (its values belonging to $\mathbb{I}$, which excludes those errors for which the execution stops).

Revisiting the forward/bottom-up nonrelational abstract interpretation of assignments using the backward analysis of arithmetic expressions

By restricting the values of the variables to those for which the expression A is well-defined, (its values belonging to $\mathbb{I}$, which excludes those errors for which the execution stops), we get

THEOREM.

$$\text{Rcom}[\![\mathtt{X} := A]\!]R(\text{after}_P[\![\mathtt{X} := A]\!]) = \text{Rcom}[\![\mathtt{X} := A]\!](\text{Baexp}[\![A]\!](R)\mathbb{I})(\text{after}_P[\![\mathtt{X} := A]\!])$$

■

PROOF.

$$\begin{aligned}
& \text{Rcom}[\![\mathtt{X} := A]\!]R(\text{after}_P[\![\mathtt{X} := A]\!]) \\
= \; & \{\rho[\mathtt{X} := i] \mid \rho \in R \wedge i \in (\text{Faexp}[\![A]\!]\{\rho\}) \cap \mathbb{I}\} && \wr\text{def. Rcom}\wr \\
= \; & \{\rho[\mathtt{X} := i] \mid \rho \in R \wedge \rho \vdash A \Rrightarrow i \wedge i \in \mathbb{I} \wedge i \in (\text{Faexp}[\![A]\!]\{\rho\}) \cap \mathbb{I}\} && \wr\text{def. (1) of } \text{Faexp}[\![A]\!]R \stackrel{\text{def}}{=} \{v \mid \exists \rho \in R : \rho \vdash A \Rrightarrow v\}\wr \\
= \; & \{\rho[\mathtt{X} := i] \mid \rho \in R \wedge \exists j \in \mathbb{I} : \rho \vdash A \Rrightarrow j \wedge i \in (\text{Faexp}[\![A]\!]\{\rho\}) \cap \mathbb{I}\} \\
= \; & \{\rho[\mathtt{X} := i] \mid \rho \in \{\rho' \in R \mid \exists j \in \mathbb{I} \cap \mathbb{I} : \rho \vdash A \Rrightarrow j\} \wedge i \in (\text{Faexp}[\![A]\!]\{\rho\}) \cap \mathbb{I}\} \\
= \; & \{\rho[\mathtt{X} := i] \mid \rho \in \text{Baexp}[\![A]\!](R)\mathbb{I} \wedge i \in (\text{Faexp}[\![A]\!]\{\rho\}) \cap \mathbb{I}\} && \wr\text{def. (19) of } \text{Baexp}[\![A]\!](R)P \stackrel{\text{def}}{=} \{\rho \in R \mid \exists i \in P \cap \mathbb{I} : \rho \vdash A \Rrightarrow i\}\wr \\
= \; & \text{Rcom}[\![\mathtt{X} := A]\!](\text{Baexp}[\![A]\!](R)\mathbb{I})(\text{after}_P[\![\mathtt{X} := A]\!]) && \wr\text{def. Rcom}\wr
\end{aligned}$$

□

In the abstract, we have:

THEOREM.

$$\alpha[\![\mathtt{X} := A]\!](\text{Rcom}[\![\mathtt{X} := A]\!])R(\text{after}_P[\![\mathtt{X} := A]\!]) \sqsubseteq \text{Acom}[\![\mathtt{X} := A]\!](\text{Baexp}^{\triangleleft}[\![A]\!](R)\,?^{\triangleright})(\text{after}_P[\![\mathtt{X} := A]\!])$$

■

PROOF. By (13), we have $\alpha[\![\mathtt{X} := A]\!](\text{Rcom}[\![\mathtt{X} := A]\!]) \stackrel{.}{\sqsubseteq} \text{Acom}[\![\mathtt{X} := A]\!]$ whence by def. (9) of $\alpha[\![\mathtt{X} := A]\!]\varphi \stackrel{\text{def}}{=} \lambda r \cdot \lambda \ell \cdot \dot{\alpha}(\varphi(\dot{\gamma}(r))(\ell))$, we have for all r and ℓ, $\dot{\alpha}(\text{Rcom}[\![\mathtt{X} := A]\!](\dot{\gamma}(r))(\ell)) \sqsubseteq \text{Acom}[\![\mathtt{X} := A]\!](r)\ell$. In particular for $r = \text{Baexp}^{\triangleleft}[\![A]\!](r')\,?^{\triangleright}$, we get $\forall r' : \forall \ell : \dot{\alpha}(\text{Rcom}[\![\mathtt{X} := A]\!](\dot{\gamma}(\text{Baexp}^{\triangleleft}[\![A]\!](r')\,?^{\triangleright}))(\ell)) \sqsubseteq \text{Acom}[\![\mathtt{X} := A]\!](r)\ell$.

By (33), we have $\alpha^{\triangleleft}(\text{Baexp}[\![A]\!]) \stackrel{.}{\sqsubseteq} \text{Baexp}^{\triangleleft}[\![A]\!]$ so that by def. (32) of $\alpha^{\triangleleft}(\Phi) \stackrel{\text{def}}{=} \lambda r \cdot \lambda p \cdot \dot{\alpha}(\Phi(\dot{\gamma}(r))\gamma(p))$, we have $\forall r : \forall p : \dot{\alpha}(\text{Baexp}[\![A]\!](\dot{\gamma}(r))\gamma(p)) \sqsubseteq \text{Baexp}^{\triangleleft}[\![A]\!](r)p$ whence for $p = ?^{\triangleright}$ such that $\gamma(?^{\triangleright}) \stackrel{\text{def}}{=} \mathbb{I}$ we get $\forall r : \dot{\alpha}(\text{Baexp}[\![A]\!](\dot{\gamma}(r))\mathbb{I}) \sqsubseteq \text{Baexp}^{\triangleleft}[\![A]\!](r)\,?^{\triangleright}$. It follows, by the Galois connection property (4), that $\forall r' : \text{Baexp}[\![A]\!](\dot{\gamma}(r'))\mathbb{I} \subseteq \dot{\gamma}(\text{Baexp}^{\triangleleft}[\![A]\!](r')\,?^{\triangleright})$.

It follows by monotony of $\dot{\alpha}$ and $\text{Rcom}[\![\mathtt{X} := A]\!]$ that $\dot{\alpha}(\text{Rcom}[\![\mathtt{X} := A]\!](\text{Baexp}[\![A]\!](\dot{\gamma}(r'))\mathbb{I})(\ell)) \sqsubseteq \dot{\alpha}(\text{Rcom}[\![\mathtt{X} := A]\!](\dot{\gamma}(\text{Baexp}^{\triangleleft}[\![A]\!](r')\,?^{\triangleright}))(\ell))$ and so we conclude

$$\begin{aligned}
& \alpha[\![\mathtt{X} := A]\!](\text{Rcom}[\![\mathtt{X} := A]\!])R(\text{after}_P[\![\mathtt{X} := A]\!]) \\
= \; & \dot{\alpha}(\text{Rcom}[\![\mathtt{X} := A]\!](\dot{\gamma}(R))(\text{after}_P[\![\mathtt{X} := A]\!])) && \wr\text{by def (9) of } \alpha[\![C]\!]\varphi \stackrel{\text{def}}{=} \lambda r \cdot \lambda \ell \cdot \dot{\alpha}(\varphi(\dot{\gamma}(r))(\ell))\wr \\
= \; & \dot{\alpha}(\text{Rcom}[\![\mathtt{X} := A]\!](\text{Baexp}[\![A]\!](\dot{\gamma}(R))\mathbb{I})(\text{after}_P[\![\mathtt{X} := A]\!])) && \wr\text{by previous theorem}\wr \\
\sqsubseteq \; & \dot{\alpha}(\text{Rcom}[\![\mathtt{X} := A]\!](\dot{\gamma}(\text{Baexp}^{\triangleleft}[\![A]\!](R)\,?^{\triangleright}))(\text{after}_P[\![\mathtt{X} := A]\!])) && \wr\text{as shown above}\wr \\
\sqsubseteq \; & \alpha[\![\mathtt{X} := A]\!](\text{Rcom}[\![\mathtt{X} := A]\!])(\text{Baexp}^{\triangleleft}[\![A]\!](R)\,?^{\triangleright})(\text{after}_P[\![\mathtt{X} := A]\!])) && \wr\text{def. } \alpha[\![\mathtt{X} := A]\!]\wr \\
\sqsubseteq \; & \text{Acom}[\![\mathtt{X} := A]\!](\text{Baexp}^{\triangleleft}[\![A]\!](R)\,?^{\triangleright})(\text{after}_P[\![\mathtt{X} := A]\!]) && \wr\text{by (13) so that } \alpha[\![C]\!](\text{Rcom}[\![C]\!]) \stackrel{.}{\sqsubseteq} \text{Acom}[\![C]\!]\wr
\end{aligned}$$

□

Implementation of the revisited forward/bottom-up nonrelational abstract interpretation of assignments using the backward analysis of arithmetic expressions

```
(* acom.ml *)
open Abstract_Syntax
open Labels
open Aenv
open Aaexp
open Abexp
open Fixpoint
open Baexp
(* forward abstract semantics of commands *)

```



```
exception Error of string
let rec acom c r l =
 match c with
 | (SKIP (l', l'')) ->
     if (l = l') then r
     else if (l = l'') then r
     else (raise (Error "SKIP incoherence"))
 | (ASSIGN (l',x,a,l'')) ->
     if (l = l') then r
     else if (l = l'') then
      f_ASSIGN x (a_aexp a) (b_aexp a r (Avalues.f_RANDOM ()))
     else (raise (Error "ASSIGN incoherence"))
 | (SEQ (l', s, l'')) ->
     (acomseq s r l)
 | (IF (l', b, nb, t, f, l'')) ->
     (if (l = l') then r
     else if (incom l t) then
      (acom t (a_bexp b r) l)
```



```
     else if (incom l f) then
      (acom f (a_bexp nb r) l)
     else if (l = l'') then
      (join (acom t (a_bexp b r) (after t))
            (acom f (a_bexp nb r) (after f)))
     else (raise (Error "IF incoherence")))
 | (WHILE (l', b, nb, c', l'')) ->
   let f x = join r (acom c' (a_bexp b x) (after c'))
   in let i = lfp (bot ()) leq f in
     (if (l = l') then i
      else if (incom l c') then (acom c' (a_bexp b i) l)
      else if (l = l'') then (a_bexp nb i)
      else (raise (Error "WHILE incoherence")))
and acomseq s r l = match s with
 | []   -> raise (Error "empty SEQ incoherence")
 | [c]  -> if (incom l c) then (acom c r l)
           else (raise (Error "SEQ incoherence"))
 | h::t -> if (incom l h) then (acom h r l)
```



```
           else (acomseq t (acom h r (after h)) l)

```

Motivating example ... revisited

```
% cd Initialization-Simple-Sign-FB-LDI-BA
% ./a.out ../Examples/example18.sil
{ x:ERR; y:ERR }
0:
  x := ?;
1:
  y := (1 / x)
2:

{ x:POS; y:INI }
```


Personal project: homework 2

– Change file`avalues.ml` of the non-relational initialization and simple sign abstract interpretors with backward analysis of expressions discussed in this lecture to implement the finitary analysis that you have chosen.

Bibliography

[2] P. Cousot. "The Calculational Design of a Generic Abstract Interpreter". In M. Broy and R. Steinbrüggen (eds.): *Calculational System Design*. NATO ASI Series F. Amsterdam: IOS Press, 1999.

[3] P. Cousot. "The Marktoberdorf'98 Generic Abstract Interpreter". `http://www.di.ens.fr/~cousot/Marktoberdorf98.shtml`.

THE END

My MIT web site is http://www.mit.edu/~cousot/

The course web site is http://web.mit.edu/afs/athena.mit.edu/course/16/16.399/www/.