Università degli Studi di Udine
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Informatica

Corso di Interpretazione Astratta

Udine, 21 – 25 Sep. 1998

Patrick Cousot

Département de Mathématique et Informatique
École Normale Supérieure
45 rue d'Ulm
75230 Paris cedex 05
Francia

Testo d'Esame[1]

Whichever programming language is considered, program analyzers have to handle boolean tests over arithmetic expressions. Most proposed solutions are naïve whence imprecise. The purpose of the exam is to study a non-trivial solution.

When proofs of theorems are long and repetitive, only a few typical cases should be considered.

Syntax and semantics of arithmetic expressions

We consider the language "Aexp" of arithmetic expressions with the following abstract syntax:

Int	:	natural numbers in decimal denotation	
Var	:	variables/identifiers	
$A \in \text{Aexp}$	:	arithmetic expressions	
Aexp	::=	Int	
		$\mid \mathsf{Var}$	
		$\mid \mathtt{?}$	random
		$\mid \mathtt{+}\ \text{Aexp} \mid \mathtt{-}\ \text{Aexp}$	unary operators

[1] Questions about the exam text should to be sent by electronic mail to `cousot@dmi.ens.fr`. The answers are due by december 1, 1998. They should be sent by email (in PostScript form) or postal mail.

$$
\begin{array}{ll}
\mid \text{Aexp} + \text{Aexp} \mid \text{Aexp} - \text{Aexp} & \text{binary operators} \\
\mid \text{Aexp} * \text{Aexp} \mid \text{Aexp} \,/\, \text{Aexp} & \\
\mid \text{Aexp} \ \texttt{mod} \ \text{Aexp} &
\end{array}
$$

The semantics of arithmetic expressions is defined using the following values and sets of values:

- Ω(initialization)(or Ω(i) for short): uninitialized value;
- Ω(arithmetic)(or Ω(a) for short): arithmetic error value (overflow, underflow, division by zero,...);
- $z \in \mathbb{Z}$: integers;
- $\mathbb{Z}_\Omega \stackrel{\text{def}}{=} \mathbb{Z} \cup \{\Omega(\text{initialization}), \Omega(\text{arithmetic})\}$;
- $\texttt{max\_int} > 0$, greatest machine integer;
- $\texttt{min\_int} = -\texttt{max\_int} - 1$ smallest machine integer;
- $\mathbb{V} \stackrel{\text{def}}{=} [\texttt{min\_int}, \texttt{max\_int}]$: bounded machine integers;
- $\mathbb{V}_\Omega \stackrel{\text{def}}{=} \mathbb{V} \cup \{\Omega(\text{initialization}), \Omega(\text{arithmetic})\}$: arithmetic variable values;
- $\rho \in \mathsf{Env} \stackrel{\text{def}}{=} \mathsf{Var} \mapsto \mathbb{Z}_\Omega$: environments recording values $\rho(\texttt{X})$ of free variables $\texttt{X} \in \mathsf{Var}$.

Informally the semantics of arithmetic expressions is that implemented on computers, with the following pecularities:

- The value v of a natural number in Int is that of the usual decimal denotation except that it is Ω(arithmetic)when this value is not included between $\texttt{min\_int}$ and $\texttt{max\_int}$ in this decimal denotation;
- The random operator `?` always returns an integer included between the bounds $\texttt{min\_int}$ and $\texttt{max\_int}$;
- Unary and binary operators are evaluated from left to right;
- If an argument is erroneous (Ω(initialization) or Ω(arithmetic)), then so is the result;
- The value of an arithmetic operator for non-erroneous arguments is that of the corresponding mathematical arithmetic operator except that it is Ω(arithmetic) if the mathematical result is not included within the bounds $\texttt{min\_int}$ and $\texttt{max\_int}$;
- The value of a division or modulo is Ω(arithmetic) when the first argument is non-erroneous and strictly negative or the second argument is non-erroneous and non positive;

The big-step operational semantics of arithmetic expression defines judgements:

$$\rho \vdash A \Rightarrow z$$

such that the evaluation of arithmetic expression $A \in \text{Aexp}$ in environment $\rho \in \mathsf{Env}$ may return the value $z \in \mathbb{V}_\Omega$ according to the informal evaluation rules stated above. We say "may" because of the presence of the nondeterministic choice operator.

Question 1 *Formally define the operational semantics of arithmetic expressions in rule-based form, by induction on their syntax.*

Syntax and semantics of boolean expressions

We consider the language "Bexp" of boolean expressions with the following abstract syntax:

$$
\begin{array}{lcll}
\mathrm{Bexp} & : & \text{boolean expressions;} & \\
\mathrm{Bexp} & ::= & \texttt{true} \mid \texttt{false} & \\
 & & \mid \mathrm{Aexp}\ \texttt{<}\ \mathrm{Aexp} \mid \mathrm{Aexp}\ \texttt{<=}\ \mathrm{Aexp} & \text{arithmetic} \\
 & & \mid \mathrm{Aexp}\ \texttt{=}\ \mathrm{Aexp} \mid \mathrm{Aexp}\ \texttt{<>}\ \mathrm{Aexp} & \text{comparisons} \\
 & & \mid \mathrm{Aexp}\ \texttt{>}\ \mathrm{Aexp} \mid \mathrm{Aexp}\ \texttt{>=}\ \mathrm{Aexp} & \\
 & & \mid \mathrm{Bexp}\ \texttt{|}\ \mathrm{Bexp} & \text{disjunction} \\
 & & \mid \mathrm{Bexp}\ \texttt{\&}\ \mathrm{Bexp} & \text{conjunction} \\
 & & \mid \neg\ \mathrm{Bexp} & \text{negation}
\end{array}
$$

The semantics of boolean expressions is defined using the following values and sets of values:

- $\mathbb{B} \stackrel{\text{def}}{=} \{\mathrm{tt}, \mathrm{ff}\}$: booleans;
- $b \in \mathbb{B}_\Omega \stackrel{\text{def}}{=} \mathbb{B} \cup \{\Omega(\text{initialization}), \Omega(\text{arithmetic})\}$: boolean values.

The big-step operational semantics defines judgements:

$$\rho \vdash B \Rightarrow b$$

such that the evaluation of boolean expressions $B \in \mathrm{Bexp}$ in environment $\rho \in \mathsf{Env}$ may yield $b \in \mathbb{B}_\Omega$ which is the logical interpretation except that uninitialization and arithmetic errors are propagated.

Question 2 *Formally define the operational semantics of boolean expressions in rule-based form, by induction on their syntax.*

Transformation of boolean expressions

To simplify the handling of boolean expressions (in particular of negation), we will transform boolean expressions into an equivalent normalized form, with the following abstract syntax:

$$
\begin{array}{lcll}
\mathrm{NBexp} & : & \text{normalized boolean expressions;} & \\
\mathrm{NBexp} & ::= & \texttt{true} \mid \texttt{false} & \\
 & & \mid \mathrm{Aexp}\ \texttt{<}\ \mathrm{Aexp} \mid \mathrm{Aexp}\ \texttt{=}\ \mathrm{Aexp} & \text{arithmetic comparisons} \\
 & & \mid \mathrm{NBexp}\ \texttt{|}\ \mathrm{NBexp} & \text{disjunction} \\
 & & \mid \mathrm{NBexp}\ \texttt{\&}\ \mathrm{NBexp} & \text{conjunction}
\end{array}
$$

Question 3 *Specify the transformation $T \in Bexp \mapsto NBexp$ of a boolean expression $B \in Bexp$ in normal form $T(B) \in NBexp$ by induction on its argument B.*

Question 4 *Prove that the operational semantics of any boolean expression $B \in Bexp$ is equivalent to that of its normal form $T(B)$.*

A program analyzer can use the boolean expression $T(B)$ on the true branch of a test for condition $B \in \mathrm{Bexp}$ and $T(\neg B)$ on the false branch of this same test. Hence in the following we have only to consider the true branch of normalized boolean expression tests $B \in \mathrm{NBexp}$.

Collecting semantics

The forward collecting semantics of arithmetic expressions is:

$$\begin{array}{rcl} \mathsf{Fco}[\![\bullet]\!] & \in & \mathrm{Aexp} \mapsto \wp(\mathsf{Var} \mapsto \mathbb{V}_\Omega) \mapsto \wp(\mathbb{V}_\Omega) \\ \mathsf{Fco}[\![A]\!]R & \stackrel{\text{def}}{=} & \{z \mid \exists \rho \in R : \rho \vdash A \Rightarrow z\} \end{array}$$

The backward collecting semantics of arithmetic expressions is:

$$\begin{array}{rcl} \mathsf{Bco}[\![\bullet]\!] & \in & \mathrm{Aexp} \mapsto \wp(\mathsf{Var} \mapsto \mathbb{V}_\Omega) \mapsto \wp(\mathbb{V}_\Omega) \mapsto \wp(\mathsf{Var} \mapsto \mathbb{V}_\Omega) \\ \mathsf{Bco}[\![A]\!](R)V & \stackrel{\text{def}}{=} & \{\rho \in R \mid \exists z \in V \cap \mathbb{V} : \rho \vdash A \Rightarrow z\} \end{array}$$

(Observe that $z \in \mathbb{V}$ so that the result should not be erroneous.)

The collecting semantics of boolean expressions is:

$$\begin{array}{rcl} \mathsf{Col}[\![\bullet]\!] & \in & \mathrm{NBexp} \mapsto \wp(\mathsf{Var} \mapsto \mathbb{V}_\Omega) \mapsto \wp(\mathsf{Var} \mapsto \mathbb{V}_\Omega) \\ \mathsf{Col}[\![B]\!]R & \stackrel{\text{def}}{=} & \{\rho \in R \mid \rho \vdash B \Rightarrow \mathtt{tt}\} \end{array}$$

Non relational abstraction

We consider an abstraction of sets of values by a Galois connection:

$$\langle \wp(\mathbb{V}_\Omega),\, \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L,\, \sqsubseteq\rangle \tag{1}$$

which, for the moment is left unspecified, and where $\langle L,\, \sqsubseteq,\, \bot,\, \top,\, \sqcup,\, \sqcap\rangle$ is a complete lattice. The corresponding non relational abstraction of sets of pairs of values is:

$$\langle \wp(\mathbb{V}_\Omega \times \mathbb{V}_\Omega),\, \subseteq\rangle \underset{\alpha^2}{\overset{\gamma^2}{\leftrightarrows}} \langle L \times L,\, \sqsubseteq^2\rangle \tag{2}$$

where $\sqsubseteq^2$ is the pairwise ordering for $\sqsubseteq$ and γ^2 is defined by:

$$\gamma^2(\langle X,\, Y\rangle) \stackrel{\text{def}}{=} \{\langle x,\, y\rangle \mid x \in \gamma(X) \wedge y \in \gamma(Y)\}$$

Question 5 *Give an explicit definition of (the unique) α^2 and prove that (2) holds.*

The corresponding non relational abstraction of sets of environments is:

$$\langle \wp(\mathsf{Var} \mapsto \mathbb{V}_\Omega),\, \subseteq\rangle \underset{\dot\alpha}{\overset{\dot\gamma}{\leftrightarrows}} \langle \mathsf{Env}^\sharp,\, \dot\sqsubseteq\rangle \tag{3}$$

where $\dot\sqsubseteq$ is the pointwise version of $\sqsubseteq$, the abstract environments are:

$$\mathsf{Env}^\sharp \stackrel{\text{def}}{=} \{\rho \in \mathsf{Var} \mapsto L \mid (\exists \mathtt{x} \in \mathsf{Var} : \rho(\mathtt{x}) = \bot) \Rightarrow (\forall \mathtt{x} \in \mathsf{Var} : \rho(\mathtt{x}) = \bot)\}$$

and $\dot\gamma$ is defined by:

$$\dot\gamma(R) \stackrel{\text{def}}{=} \{\rho \in \mathsf{Var} \mapsto \mathbb{V}_\Omega \mid \forall x \in \mathsf{Var} : \rho(x) \in \gamma(R(x))\}$$

Question 6 *Give an explicit definition of (the unique) $\dot\alpha$ and prove that (3) holds.*

Forward abstract interpretation of arithmetic operations

In the following, we assume that for $i \in \mathbb{N}$:

$$\begin{aligned} i^\sharp &\sqsupseteq \alpha(\{i\}) && (4) \\ \texttt{?}^\sharp &\sqsupseteq \alpha(\mathbb{V}) && (5) \end{aligned}$$

and for each unary arithmetic operator u, we are given the forward abstract unary operator $u^\sharp$ such that:

$$u^\sharp X \quad \sqsupseteq \quad \alpha(\{u\, x \mid x \in \gamma(X)\}) \tag{6}$$

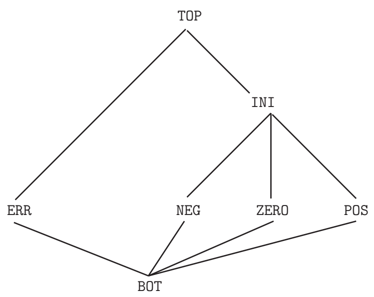
and for each binary arithmetic operator b, we are given the forward abstract binary operator $b^\sharp$ such that:

$$X\, b^\sharp\, Y \quad \sqsupseteq \quad \alpha(\{x\, b\, y \mid x \in \gamma(X) \wedge y \in \gamma(Y)\}) \tag{7}$$

where u and b are the semantic operators on $\mathbb{V}_\Omega$ defining the semantics of the arithmetic operators.

Sign and initialization abstract interpretation of arithmetic operations

We now consider an application where (1) is chosen as explained below. The lattice L is defined by the following Hasse diagram:



with the following meaning:

$$\begin{aligned} \gamma(\texttt{BOT}) &\stackrel{\text{def}}{=} \{\Omega(\text{arithmetic})\} \\ \gamma(\texttt{POS}) &\stackrel{\text{def}}{=} [1, \texttt{max\_int}] \cup \{\Omega(\text{arithmetic})\} \\ \gamma(\texttt{ZERO}) &\stackrel{\text{def}}{=} \{0, \Omega(\text{arithmetic})\} \\ \gamma(\texttt{NEG}) &\stackrel{\text{def}}{=} [\texttt{min\_int}, -1] \cup \{\Omega(\text{arithmetic})\} && (8) \\ \gamma(\texttt{INI}) &\stackrel{\text{def}}{=} \mathbb{V} \cup \{\Omega(\text{arithmetic})\} \\ \gamma(\texttt{ERR}) &\stackrel{\text{def}}{=} \{\Omega(\text{initialization}), \Omega(\text{arithmetic})\} \\ \gamma(\texttt{TOP}) &\stackrel{\text{def}}{=} \mathbb{V}_\Omega \end{aligned}$$

Question 7 *Define the corresponding abstraction α such that (1) holds.*

Question 8 *What would be wrong if the definition (8) were changed so that:*

$$\gamma(\texttt{BOT}) \stackrel{\textit{def}}{=} \emptyset$$

Question 9 *Provide a definition of the constant $i^\sharp$, $\texttt{?}^\sharp$, unary $\texttt{-}^\sharp$ and binary $\texttt{+}^\sharp$, $\texttt{-}^\sharp$, $\texttt{*}^\sharp$, $\texttt{/}^\sharp$ and $\texttt{mod}^\sharp$ forward abstract arithmetic operators for the above sign and initialization abstraction defined by γ at (8) and prove that they respectively satisfy (4), (5), (6) and (7).*

Question 10 *Do you think that the following interpretation of the sign and initialization analysis would be interesting? why?*

$$\begin{aligned}
\gamma(\texttt{BOT}) &\stackrel{\textit{def}}{=} \emptyset \\
\gamma(\texttt{POS}) &\stackrel{\textit{def}}{=} [1, \texttt{max_int}] \\
\gamma(\texttt{ZERO}) &\stackrel{\textit{def}}{=} \{0\} \\
\gamma(\texttt{NEG}) &\stackrel{\textit{def}}{=} [\texttt{min_int}, -1] \\
\gamma(\texttt{INI}) &\stackrel{\textit{def}}{=} \mathbb{V} \\
\gamma(\texttt{ERR}) &\stackrel{\textit{def}}{=} \{\Omega(\text{initialization})\} \\
\gamma(\texttt{TOP}) &\stackrel{\textit{def}}{=} \mathbb{V}_\Omega
\end{aligned}$$

Question 11 *What do you think of the following other possible interpretation of the sign and initialization analysis:*

$$\begin{aligned}
\gamma(\texttt{BOT}) &\stackrel{\textit{def}}{=} \{\Omega(\text{arithmetic})\} \\
\gamma(\texttt{POS}) &\stackrel{\textit{def}}{=} [1, \texttt{max_int}] \cup \{\Omega(\text{arithmetic})\} \\
\gamma(\texttt{ZERO}) &\stackrel{\textit{def}}{=} \{0, \Omega(\text{arithmetic})\} \\
\gamma(\texttt{NEG}) &\stackrel{\textit{def}}{=} [\texttt{min_int}, -1] \cup \{\Omega(\text{arithmetic})\} \\
\gamma(\texttt{INI}) &\stackrel{\textit{def}}{=} \mathbb{V} \cup \{\Omega(\text{arithmetic})\} \\
\gamma(\texttt{ERR}) &\stackrel{\textit{def}}{=} \{\Omega(\text{initialization})\} \\
\gamma(\texttt{TOP}) &\stackrel{\textit{def}}{=} \mathbb{V}_\Omega
\end{aligned}$$

Forward abstract interpretation of arithmetic expressions

Question 12 *Give an explicit definition of the forward abstract interpretation of arithmetic expressions, parameterized by the abstract operators (4), (5), (6) and (7), and such that:*

$$\begin{aligned}
\mathsf{Faexp}^\sharp[\![\bullet]\!] &\in \mathit{Aexp} \mapsto \mathsf{Env}^\sharp \mapsto L \\
\mathsf{Faexp}^\sharp[\![A]\!] &\mathrel{\dot{\sqsupseteq}} \alpha \circ \mathsf{Fco}[\![A]\!] \circ \dot{\gamma}
\end{aligned} \qquad (9)$$

Prove that (9) holds for your definition of $\mathsf{Faexp}^\sharp[\![\bullet]\!]$.

Backward abstract interpretation of arithmetic operations

In the following, we assume that for all $X \in L$:

$$i^\flat(R)X \quad \sqsupseteq \quad \dot{\alpha}(\{\rho \in \dot{\gamma}(R) \mid i \in \gamma(X) \cap \mathbb{V}\} \tag{10}$$

$$\texttt{?}^\flat(R)X \quad \sqsupseteq \quad \alpha(\{\rho \in \dot{\gamma}(R) \mid \gamma(X) \cap \mathbb{V} \neq \emptyset\} \tag{11}$$

and that for each unary arithmetic operator u, we are given the backward abstract binary operator $u^\flat$ such that:

$$u^\flat(Z, X) \quad \sqsupseteq \quad \alpha(\{x \mid x \in \gamma(X) \wedge u\, x \in \gamma(Z) \cap \mathbb{V}\}) \tag{12}$$

and for each binary arithmetic operator b, we are given the backward abstract ternary operator $b^\flat$ such that:

$$b^\flat(Z, X, Y) \quad \sqsupseteq^2 \quad \alpha^2(\{\langle x,\, y\rangle \mid x \in \gamma(X) \wedge y \in \gamma(Y) \wedge x\, b\, y \in \gamma(Z) \cap \mathbb{V}\}) \tag{13}$$

where u and b are the semantic operators on $\mathbb{V}_\Omega$ defining the semantics of the arithmetic operators.

Question 13 *Provide a definition of the unary* $i^\flat$, $\texttt{?}^\flat$, *binary* $\texttt{-}^\flat$ *and ternary* $\texttt{+}^\flat$, $\texttt{-}^\flat$, $\texttt{*}^\flat$, $\texttt{/}^\flat$ *and* $\texttt{mod}^\flat$ *backward abstract arithmetic operators for the sign and initialization abstraction defined above by* γ *at (8) and prove that they respectively satisfy (10), (11), (12) and (13).*

Backward abstract interpretation of arithmetic expressions

Question 14 *Give an explicit definition of the backward abstract interpretation of arithmetic expressions, parameterized by the backward abstract operators (10), (11), (12) and (13), and such that:*

$$\mathsf{Baexp}^\flat[\![\bullet]\!] \quad \in \quad NBexp \mapsto \mathsf{Env}^\sharp \mapsto L \mapsto \mathsf{Env}^\sharp$$

$$\mathsf{Baexp}^\flat[\![A]\!](R)X \quad \dot{\sqsupseteq} \quad \dot{\alpha} \circ \mathsf{Bco}[\![A]\!]\dot{\gamma}(R) \circ \gamma(X) \tag{14}$$

Prove that (14) holds for your definition of $\mathsf{Baexp}^\flat[\![\bullet]\!]$.

Abstract interpretation of boolean operations

For arithmetic equality and inequality, we consider ($X, Y \in L$):

$$=^\natural(X, Y) \quad \stackrel{\text{def}}{=} \quad \alpha^2(\{\langle x,\, x\rangle \mid \; x \in \gamma(X) \cap \gamma(Y) \cap \mathbb{V}\}) \tag{15}$$

$$<^\natural(X, Y) \quad \stackrel{\text{def}}{=} \quad \alpha^2(\{\langle x,\, y\rangle \mid \; x \in \gamma(X) \cap \mathbb{V} \wedge y \in \gamma(Y) \cap \mathbb{V} \wedge x < y\}) \tag{16}$$

Question 15 *Provide a definition of* $=^\natural$ *and* $<^\natural$ *for the sign and initialization abstraction defined above by* γ *at (8) and prove that they respectively satisfy (15) and (16).*

Abstract interpretation of boolean expressions

In order to have an idea of the abstract interpretation of boolean expressions, let us consider a simple example for initialization and sign analysis. In the abstract environment:

$$\texttt{X} : \texttt{POS}, \texttt{Y} : \texttt{INI}$$

the test `X * Y < 0` implies that the value of `Y` should be negative so the result of the abstract interpretation of boolean expression `X * Y < 0` for the above abstract environment should be:

$$\texttt{X} : \texttt{POS}, \texttt{Y} : \texttt{NEG} \;.$$

Starting for the initial environment:

$$\texttt{X} : \texttt{POS}, \texttt{Y} : \texttt{INI}$$

the analyzer performs a forward abstract interpretation of `X * Y`:

$$*^{\sharp}(\texttt{POS}, \texttt{INI}) \quad = \quad \texttt{INI}$$

and of `0`:

$$\texttt{0}^{\sharp} \quad = \quad \texttt{ZERO} \;.$$

Now the abstract interpretation of the test is:

$$<^{\natural}(\texttt{INI}, \texttt{ZERO}) \quad = \quad \langle \texttt{NEG},\ \texttt{ZERO} \rangle$$

proving that `X * Y` should be negative (and `0` should be null, which is obvious). A backward asbtract interpretation of the arithmetic expressions `X * Y` (and `0`) propagates this information to the variables, as follows:

$$\begin{aligned} & *^{\flat}(\texttt{NEG}, \texttt{POS}, \texttt{INI}) \\ = \;& \langle \texttt{POS},\ \texttt{NEG} \rangle \end{aligned}$$

so that we get:

$$\texttt{X} : \texttt{POS}, \texttt{Y} : \texttt{NEG},$$

as required.

Question 16 *Provide a definition of the abstract interpretation* $\mathsf{bexp}^{\natural}[\![B]\!]$ *of normalized boolean expressions* $B \in$ *NBexp, parameterized by* $i^{\sharp}$, $\texttt{?}^{\sharp}$, *unary* $\texttt{-}^{\sharp}$, *binary* $\texttt{-}^{\sharp}$, $\texttt{+}^{\sharp}$, $*^{\sharp}$, $\texttt{/}^{\sharp}$, $\texttt{mod}^{\sharp}$, *binary* $\texttt{-}^{\flat}$, *ternary* $\texttt{-}^{\flat}$, $\texttt{+}^{\flat}$, $*^{\flat}$, $\texttt{/}^{\flat}$, $\texttt{mod}^{\flat}$, *logical* $=^{\natural}$ *and* $<^{\natural}$ *and satisfying (for all* $R \in \mathsf{Env}^{\sharp}$*):*

$$\mathsf{bexp}^{\natural}[\![B]\!](R) \quad \dot{\sqsupseteq} \quad \dot{\alpha} \circ \mathsf{Col}[\![B]\!] \circ \dot{\gamma}(R)$$

Local decreasing iterations

Question 17 *Prove that for all* $B \in$ *NBexp,* $\mathsf{bexp}^{\natural}[\![B]\!]$ *is reductive and monotone* [2]*.*

[2] If this is not the case then the precision of your $\mathsf{bexp}^{\natural}[\![B]\!]$ can be improved so that this is the case.

We define:

$$\mathsf{BEXP}^\natural[\![B]\!]R \quad = \quad \dot{\bigsqcap_{n>0}} \mathsf{bexp}^\natural[\![B]\!]^n(R)$$

Question 18 *Prove that* $\mathsf{BEXP}^\natural[\![B]\!]R$ *is a correct abstract interpretation of the normalized boolean expression* $B \in \mathit{NBexp}$ *in abstract environment* $R \in \mathsf{Env}^\sharp$*.*

Question 19 *Why is* $\mathsf{BEXP}^\natural[\![B]\!]R$ *always more precise than* $\mathsf{bexp}^\natural[\![B]\!](R)$*?*

Question 20 *Discuss the termination of the abstract interpretation algorithm based upon* $\mathsf{BEXP}^\natural[\![B]\!]R$*.*

Question 21 *Provide an example showing that* $\mathsf{BEXP}^\natural[\![B]\!]R$ *is a strictly more precise abstract interpretation algorithm of normalized boolean expressions* B *than* $\mathsf{bexp}^\natural[\![B]\!](R)$*.*

Question 22 *Prove that for all* $B \in \mathit{NBexp}$*,* $\mathsf{BEXP}^\natural[\![B]\!]R$ *is a lower closure operator (reductive, monotone and idempotent).*