Cours MOdélisation, Vérification et EXpérimentations
Exercices (avec les corrections)
Analyse des programmes (I)
par Dominique Méry
23 mai 2025

Domaine des signes

Exercice 1

On définit une abstraction pour les entiers relatifs $\alpha \in (\mathcal{P}(\mathbb{Z}), \subseteq) \longrightarrow (\mathbb{S}igns, \sqsubseteq)$ *:*

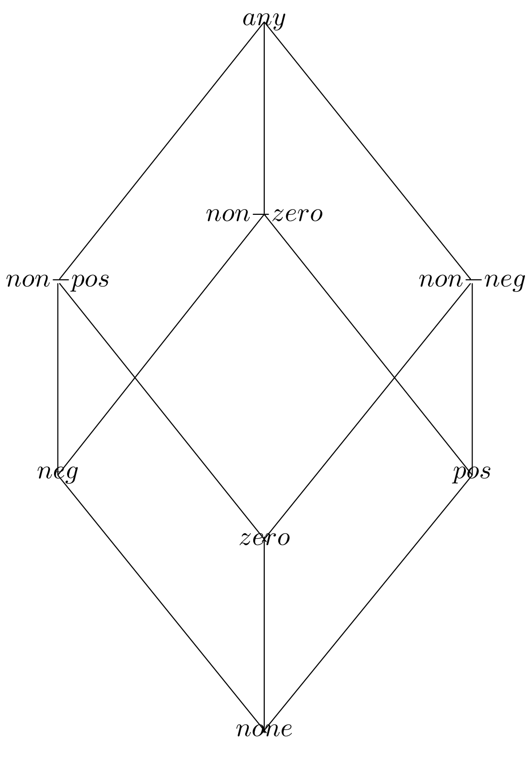
— *Si* $z < 0$*, alors* $\alpha(\{z\}) = neg$*.*

— *Si* $z > 0$*, alors* $\alpha(\{z\}) = pos$*.*

— *Si* $z = 0$*, alors* $\alpha(\{z\}) = zero$*.*

— *Si* $A \subseteq \mathbb{Z}$*, alors* $\alpha(A) = \bigsqcup\{\alpha(a) | a \in A\}$ *où* $\bigsqcup$ *désigne la borne supérieure dans* $\mathbb{S}igns$*.*

— *On notera* γ *l'opérateur de concrétisation associé à* α *et définit par la relation caractérisant les connexions de Galois.*
Cette paire (α, γ) *est une connexion de Galois, si elle satisfait* $\forall x_1 \in \mathcal{P}(\mathbb{Z}), x_2 \in \mathbb{S}igns$: $\alpha(x_1) \sqsubseteq x_2 \Leftrightarrow x_1 \subseteq \gamma(x_2)$



(α, γ) *est une connexion de Galois.*
On rappelle que les opérations arithmétiques sont étendues aux parties des ensembles comme suit. Si $A, B \in \mathcal{P}(\mathbb{Z})$*, alors* $A \dot{+} B = \{a+b | a \in A \wedge b \in B\}$*. La définition d'un opérateur abstrait* $+_a$ *sur* $\mathbb{S}igns$ *est donnée par :*
$x, y \in \mathbb{S}igns : x +_a y = \alpha(\gamma(x) \dot{+} \gamma(y))$

Question 1.1 *Pour construire les éléments de l'abstraction, on va devoir définir des extensions des opérations arithmétiques et logiques sur lénsemble des parties de* $\mathbb{Z}$ *:* $(\mathcal{P}(\mathbb{Z}), \subseteq)$ *:*

— $A, B \in \mathcal{P}(\mathbb{Z}) : A \dot{+} B = \{a+b | a \in A \wedge b \in B\}$

— $x, y \in \mathbb{S}igns : x +_a y = \alpha(\gamma(x) \dot{+} \gamma(y))$

Par exemple, on peut montrer que :

— $pos +_a neg = \alpha(\gamma(pos) \dot{+} \gamma(neg)) = \alpha((1, +\infty) \dot{+} (-\infty, -1)) = \alpha((-\infty, +\infty))$

— $pos +_a zero = \alpha(\gamma(pos) \dot{+} \gamma(zero)) = \alpha((1, +\infty) \dot{+} (0)) = \alpha((1, +\infty)) = pos$

Construire une table des opérations abstraites pour $+_a$*.*

◇— Solution de l'exercice 1

— $pos +_a neg = \alpha(\gamma(pos) \dot{+} \gamma(neg)) = \alpha((1, +\infty) \dot{+} (-\infty, -1)) = \alpha((-\infty, +\infty))$

— $pos +_a zero = \alpha(\gamma(pos) + \gamma(zero)) = \alpha((1, +\infty) + (0)) = \alpha((1, +\infty)) = pos$
— $non\!-\!pos +_a neg = \alpha(\gamma(non\!-\!pos) + \gamma(neg)) = \alpha((-\infty, 0) + (-\infty, -1)) = \alpha((-\infty, 0)) = non\!-\!pos$

Poursuivre en complétant la table.

Fin 1

Question 1.2 *Calculer les valeurs suivantes en justifiant le résultat :*

1. $\alpha(\{n | n \in \mathbb{Z} \wedge pair(|n|)\})$
2. $\alpha(\{16, 1, -1\})$
3. $\alpha(\{-91, -889\})$
4. $\gamma(pos)$
5. $\gamma(non\!-\!pos)$

◇– **Solution de la question 1.2**

1. $\alpha(\{n | n \in \mathbb{Z} \wedge pair(|n|)\}) = any$
2. $\alpha(\{16, 1, -1\}) = non\!-\!zero$
3. $\alpha(\{-91, -889\}) = neg$
4. $\gamma(pos) = (1, +\infty)$
5. $\gamma(non\!-\!pos) = (-\infty, 0)$

Fin 1.2

Question 1.3

Un état concret est noté cv et appartient à lénsemble $Var \longrightarrow \mathcal{P}(\mathbb{Z})$: si X est une variable de Var, alors $cv(X) \in \mathcal{P}(\mathbb{Z})$.
Un état abstrait est noté av et appartient à lénsemble $Var \longrightarrow \mathbb{S}igns$: si X est une variable de Var, alors $av(X) \in \mathbb{S}igns$.
La connexion de Galois (α, γ) s'étend naturellement en une connection de Galois :
(α_1, γ_1) entre $(Var \longrightarrow \mathcal{P}(\mathbb{Z}), \subseteq)$ et $(Var \longrightarrow \mathbb{S}igns, \sqsubseteq)$. En particulier, $\alpha_1(cv) = av$ et, pour tout X de Var, $av(X) = \alpha(cv(X))$; $\gamma_1(av) = cv$ et, pour tout X de Var, $cv(X) = \gamma(av(X))$.
Toute expression e peut être intepétée selon l'un des domaines choisi. On peut donc définir deux évaluations possibles de e :

— *domaine concret $States = Var \longrightarrow \mathcal{P}(\mathbb{Z})$: $[\![e]\!] \in (Var \longrightarrow \mathcal{P}(\mathbb{Z})) \longrightarrow \mathcal{P}(\mathbb{Z})$ et $[\![e]\!](cv)$ est donc léxpression e interprétée dans l'état cv avec les valeurs concrètes (dans ce cas, les valeurs sont des ensembles déntiers).*
— *domaine abstrait $AStates = Var \longrightarrow \mathbb{S}igns$: $[\![e]\!]_a \in (Var \longrightarrow \mathbb{S}igns) \longrightarrow \mathbb{S}igns$ et $[\![e]\!]_a(av)$ est donc léxpression e interprétée dans l'état av avec les valeurs abstraites (dans ce cas, les valeurs sont des valeurs de $\mathbb{S}igns$.*
— *La meilleure abstraction est simplement écrite sous la fome suivante : $[\![e]\!]_{best}(av) = \alpha \circ [\![e]\!] \circ \gamma_1(av)$.*

On peut donc réaliser une exécution d'un petit algorithme en utilisant les variables contenat des valeurs abstraites. On demande de remplir le tableau avec les valeurs abstraites obtenues par évaluation abstraite : $[\![e]\!]_{best}(av) = \alpha \circ [\![e]\!] \circ \gamma_1(av)$.

$\ell_0[X := 1]$;
$\ell_1[Y := 5]$;
$\ell_2[X := X+1]$;
$\ell_3[Y := Y-1]$;
$\ell_4[X := Y+X]$;
$\ell_{final}[skip]$;

ℓ	X	Y
ℓ_0		
ℓ_1		
ℓ_2		
ℓ_3		
ℓ_4		
ℓ_{final}		

◇– **Solution de la question 1.3**

ℓ	X	Y
ℓ_0	any	any
ℓ_1	pos	any
ℓ_2	pos	pos
ℓ_3	pos	pos
ℓ_4	pos	$non{-}neg$
ℓ_{final}	any	$non{-}neg$

Fin 1.3

Question 1.4 *L'évaluation d'une expression e a été faite en utilisant la meilleure approximation par rapport à la connection de Galois choisie ($[\![e]\!]_{best}(av) = \alpha\circ[\![e]\!]\circ\gamma_1(av)$). Cette évaluation revient à ramener le calcul abstrait sous la forme d'un calcul concret sur les expressions. Cela veut dire que cela est complexe dans la mesure où cela reste dans le domaine concret. Une autre voie est d'utiliser une approximation correcte pour définir une sémantique abstraite des expressions notée $[\![e]\!]_a$ et telle que, pour tout état abstrait av, $[\![e]\!]_{best}(av) \sqsubseteq [\![e]\!]_a(av)$.*
$av \in Var \longrightarrow \mathbb{S}igns$:
— $[\![const]\!]_a(v) = \alpha(\{c\})$
— $[\![x]\!]_a(v) = v(x)$
— $[\![e_1{+}e_2]\!]_a(v) = [\![e_1]\!]_a(v)\oplus[\![e_2]\!]_a(v)$
— $[\![e_1{\times}e_2]\!]_a(v) = [\![e_1]\!]_a(v)\otimes[\![e_2]\!]_a(v)$
Dans le cas d'une instruction de la forme $\ell[X := E]$, on décide d'affecter à X la valeur abstaite obtenue par l'interprétation $[\![E]\!]_a$ en $av[\![E]\!]_a(av)$. Par exemple, si $E = Y{+}X{+}6$, alors $[\![Y{+}X{+}6]\!]_a(av) = [\![Y]\!]_a(av){+}_a[\![X]\!]_a(av){+}_a[\![6]\!]_a(av)$. Ainsi, si on considère l'affectation $\ell[Y := Y{-}1]$ avec $av(Y) = pos$, on obtient :
— $[\![Y{-}1]\!]_a(av) = [\![Y]\!]_a(av)\oplus[\![-1]\!](av)_a = pos\oplus neg = any$
— $[\![Y{-}1]\!]_{best}(av) = \alpha_1\circ[\![Y{-}1]\!]\circ\gamma_1(av) == \alpha_1([\![Y{-}1]\!](\gamma_1(av))) = \alpha_1([\![Y{-}1]\!](\{Y \mapsto (1,+\infty)\}) = \alpha_1((1{+}\infty){+}(-1)) = \alpha_1((0,+\infty)) = non{-}neg$
Appliquer l'analyse sur léxemple :

$\ell_0[X := 1]$;
$\ell_1[Y := 5]$;
$\ell_2[X := X{+}1]$;
$\ell_3[Y := Y{-}1]$;
$\ell_4[X := Y{+}X]$;
$\ell_{final}[skip]$;

ℓ	X	Y
ℓ_0		
ℓ_1		
ℓ_2		
ℓ_3		
ℓ_4		
ℓ_{final}		

◇— **Solution de la question 1.4**

ℓ	X	Y
ℓ_0	any	any
ℓ_1	pos	any
ℓ_2	pos	pos
ℓ_3	pos	pos
ℓ_4	pos	any
ℓ_{final}	any	any

Fin 1.4

Exercice 2

```
#include <stdio.h>
#include <stdlib.h>
#define N 624
main()
{
  /* D\'eclarations des variables */
  int i,s,r;
```

$\ell_0[i = 1;]$
$\ell_1[s = 0;]$
$\ell_2[r = -1;]$
WHILE $\ell_3[s <= N]$
  $\ell_4[s+ = i];$
  $\ell_5[i+ = 2];$
  $\ell_6[r{+}{+};];$
END-WHILE
$\ell_{final}[skip]$

Question 2.1 *Produire le graphe de flôt d'analyse.*

Question 2.2 *Ecrire le système d'équations définissant la sémantique collectrice.*

Question 2.3 *Ecrire une analyse abstraite dans le cadre du domaine des signes.*

Exercice 3 *Soit la connexion de Galois suivante :*

$$\alpha \in \mathcal{P}(\mathbb{Z}) \longrightarrow \mathbb{S}igns : \begin{cases} z & \alpha(z) \\ z < 0 & neg \\ z > 0 & pos \\ z = 0 & zero \end{cases}$$

Question 3.1 *Soient* $f \in \mathcal{P}(\mathbb{Z}) \longrightarrow \mathcal{P}(\mathbb{Z})$ *where* $f(X) = \{0\} \cup \{x+2 | x \in \mathbb{Z} \wedge x \in X\}$ *and* $g = \alpha \circ f \circ \gamma$.

1. *Compute the sequence* f^0, f^1, f^2, f^i.
2. *Compute the sequence* g^0, g^1, g^2, g^i.
3. *Compute* $\mu.g$.
4. *What is the link between* $\mu.f$ *and* $\mu.g$.

Question 3.2 *Soient* $f \in \mathcal{P}(\mathbb{Z}) \longrightarrow \mathcal{P}(\mathbb{Z})$ *where* $f(X) = \{1\} \cup \{x+2 | x \in \mathbb{Z} \wedge x \in X\}$ *and* $g = \alpha \circ f \circ \gamma$.

1. *Compute the sequence* f^0, f^1, f^2, f^i.
2. *Compute the sequence* g^0, g^1, g^2, g^i.
3. *Compute* $\mu.g$.
4. *What is the link between* $\mu.f$ *and* $\mu.g$.

Cours MOdélisation, Vérification et EXpérimentations
Exercices (avec les corrections)
Analyse des programmes (II)
par Dominique Méry
23 mai 2025

Domaine des intervalles

Exercice 4

On définit le domaine des intervalles sur $\mathbb{Z}$*, noté* $\mathbb{I}(\mathbb{Z})$ *comme suit :* $\mathbb{I}(\mathbb{Z}) = \{\perp\} \cup \{[l,u] | l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{\infty\}, l \leq u\}$
On définit une relation $\sqsubseteq$ *sur les intervalles comme suit :*
$[l_1, u_1] \sqsubseteq [l_2, u_2]$ *si, et seulement si,* $l2 \leq l1$ *et* $u_1 \leq u_2$*.*

Question 4.1 *Montrer que* $(\mathbb{I}(\mathbb{Z}), \sqsubseteq)$ *est une structure partiellement ordonnée.*

Question 4.2 *Montrer que*

1. $[l_1, u_1] \sqcup [l_2, u_2] = [min(l_1, l_2), max(u_1, u_2)]$
2. *et que* $[l_1, u_1] \sqcap [l_2, u_2] = \begin{cases} [max(l_1, l_2), min(u_1, u_2)] \\ \perp, si\ max(l_1, l_2) > min(u_1, u_2) \end{cases}$

Question 4.3 *Montrer que la structure ordonnée est un treillis complet.*

On définit deux fonctions :
$\alpha(X) = \begin{cases} [min(X), max(X)] \\ \perp, si\ X = \varnothing \end{cases}$
$\gamma([l, u]) = [l..u]$ *et* $\gamma(\perp) = \varnothing$

Question 4.4 *Montrer que* (α, γ) *est une connexion de Galois.*

On définit des opérations sur les intervalles comme suit :

1. $i_1 \oplus i_2 = [l_1 + l_2, u_1 + u_2]$
2. $i_1 \ominus i_2 = [l_1 - u_2, u_1 - l_2]$
3. $i_1 \otimes i_2 = [min(l_1 \cdot l_2, l_1 \cdot u_2, u_1 \cdot l_2, u_1 \cdot u_2, max(l_1 \cdot l_2, l_1 \cdot u_2, u_1 \cdot l_2, u_1 \cdot u_2)]$
4. $i_1 \oslash i_2 = [min(l_1/l_2, l_1/u_2, u_1/l_2, u_1/u_2, max(l_1/l_2, l_1/u_2, u_1/l_2, u_1/u_2)]$

Question 4.5 *Appliquer la technique de calcul abstrait sur cet exemple.*

$\ell_0[I := 1];$
WHILE $\ell_1[I \leq 100]$
$\ell_2[I := I+1];$
END-WHILE
$\ell_{final}[skip]$

$$\begin{tikzcd}
\ell_0 \arrow[d, "[\![I:=1]\!]"] & \\
\ell_1 \arrow[r, "[\![\textbf{assert } \neg(I \leq 100)]\!]"] \arrow[d, bend left, "[\![\textbf{assert } I \leq 100]\!]"] & \ell_{final} \\
\ell_2 \arrow[u, bend left, "[\![I:=I-1]\!]"] &
\end{tikzcd}$$

⋄— Solution de la question 4.5

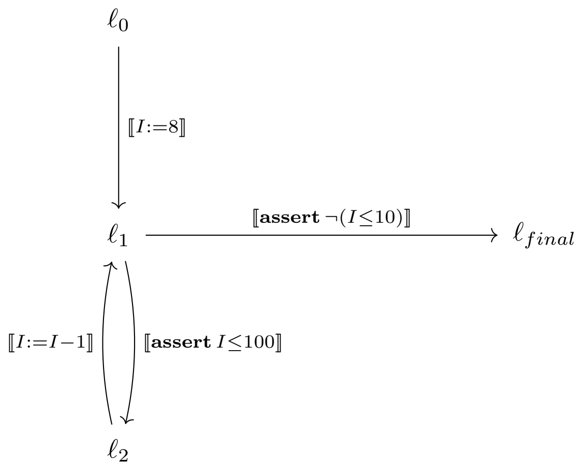
On peut utiliser l'analyseur Interproc http ://pop-art.inrialpes.fr/interproc/ et on peut utiliser différentes analyses notamment l'analyse dite box qui correspond aux intervalles.

```
Result

Annotated program after forward analysis
var I : int;
begin
  /* (L2 C5) top */
  I = 1; /* (L3 C6) [|I-1>=0; -I+101>=0|] */
  while I <= 100 do
    /* (L4 C20) [|I-1>=0; -I+100>=0|] */
    I = I + 1; /* (L5 C9)
                   [|I-2>=0; -I+101>=0|] */
  done; /* (L6 C5) [|I-101=0|] */
end

Source

var I:int;
begin
I = 1;
while  (I <= 100) do
 I = I+1;
done;
end
```

Fin 4.5

Question 4.6 *Appliquer la technique de calcul abstrait sur cet exemple.*

$\ell_0[I := 8]$;
WHILE $\ell_1[I \leq 10]$
$\ell_2[I := I+1]$;
END-WHILE
$\ell_{final}[skip]$



⋄— Solution de la question 4.6

```
Result

Annotated program after forward analysis
var I : int;
begin
  /* (L2 C5) top */
```

```
  I = 8; /* (L3 C6) [|I-8>=0; -I+11>=0|] */
  while I <= 10 do
    /* (L4 C19) [|I-8>=0; -I+10>=0|] */
    I = I + 1; /* (L5 C9)
                 [|I-9>=0; -I+11>=0|] */
  done; /* (L6 C5) [|I-11=0|] */
end
```

```
Source

var I:int;
begin
I = 8;
while  (I <= 10) do
 I = I+1;
done;
end
```

Fin 4.6

Exercice 5

Soit l'algorithme suivant :

$\ell_0[Q := 0]$;
$\ell_1[R := X]$;
IF $\ell_5[Y > 0]$
  WHILE $\ell_2[R \geq Y]$
    $\ell_3[Q := Q{+}1]$;
    $\ell_4[R := R{-}Y]$
  ENDWHILE
ELSE
  $\ell_6[skip]$
ENDIF

Question 5.1 *Produire un graphe de flôt de contrôle de cetv algorithme.*

Question 5.2 *Analyser l'algorithme avec le doamien abstrait des intervalles.*

⋄– **Solution de la question 5.2**

```
Result

Annotated program after forward analysis
var Q : int, R : int, X : int, Y : int;
begin
  /* (L3 C5) top */
  Q = 0; /* (L4 C4) [|Q=0|] */
  R = Y; /* (L5 C4) [|-R+Y=0; Q=0|] */
  if Y > 0 then
     /* (L6 C15)
        [|-Q-R+Y>=0; -Q+1>=0; Q>=0; Q+R-1>=0|] */
     while R >= Y do
       /* (L7 C20) [|-R+Y=0; Q=0; R-1>=0|] */
       Q = Q + 1; /* (L8 C13)
```

```
                     [|-R+Y=0; Q-1=0; R-1>=0|] */
       R = R - Y; /* (L9 C15)
                     [|R=0; Q-1=0; Y-1>=0|] */
     done; /* (L10 C7)
              [|-Q+1>=0; -R+Y-1>=0; Q>=0; Q+R-1>=0|] */
  else
    /* (L11 C4) [|-R+Y=0; Q=0; -R>=0|] */
    skip; /* (L12 C7) [|-R+Y=0; Q=0; -R>=0|] */
  endif; /* (L13 C6)
            [|-Q-R+Y>=0; -Q+1>=0; Q>=0|] */
end

Source

/* type your program here ! */
var Q:int,R:int,X:int,Y:int;
begin
Q=0;
R=Y;
if (Y > 0) then
  while (R >=  Y) do
     Q = Q+1;
     R = R - Y;
  done;
else
  skip;
endif;
end
```

Fin 5.2

Exercice 6

```
#include <stdio.h>
#include <stdlib.h>
#define N 624
main()
{
  /* D\'eclarations des variables */
  int i,s,r;
```

Question 6.1 *Produire le graphe de flôt d'analyse.*

Question 6.2 *Ecrire le système d'équations définissant la sémantique collectrice.*

Question 6.3 *Ecrire une analyse abstraite dans le cadre du domaine des intervalles.*

◇– **Solution de la question 6.3**

precondition : ...
postcondition : ...

$\ell_0[i = 1;]$
$\ell_1[s = 0;]$
$\ell_2[r = -1;]$
while $\ell_3[s <= N]$ **do**
$\quad \ell_4[s+ = i]; \ell_5[i+ = 2]; \ell_6[r++;];$
;
$\ell_{final}[skip]$

Algorithme 1: $PROGRAM1$ annotée

```
Annotated program after forward analysis
var I : int, S : int, R : int, N : int;
begin
  /* (L2 C5) top */
  I = 1; /* (L3 C4) [|I-1=0|] */
  S = 0; /* (L4 C4) [|I-1=0; S=0|] *  R = -1; /* (L5 C5) [|I-1>=0; R+1>=0; S>=0|] *
  while S <= N do
    /* (L6 C16)
       [|I-1>=0; N>=0; R+1>=0; S>=0|] */
    S = S + I; /* (L7 C6)
                 [|I-1>=0; N>=0; R+1>=0; S-1>=0|] */
    I = I + 2; /* (L8 C6)
                 [|I-3>=0; N>=0; R+1>=0; S-1>=0|] */
    R = R + 1; /* (L9 C6)
                 [|I-3>=0; N>=0; R>=0; S-1>=0|] */
  done; /* (L10 C5) [|I-1>=0; R+1>=0; S>=0|] */
end

Source

var I:int,S:int,R:int,N:int;
begin
I=1;
S=0;
R=-1;
while (S<= N) do
S=S+I;
I=I+2;
R=R+1;
done;
end
```

Fin 6.3

Cours MOdélisation, Vérification et EXpérimentations
Exercices (avec les corrections)
Abstraction, approximation et calcul
par Dominique Méry
23 mai 2025

Domaine des intervalles

Exercice 7

On définit le domaine des intervalles sur $\mathbb{Z}$, noté $\mathbb{I}(\mathbb{Z})$ comme suit : $\mathbb{I}(\mathbb{Z}) = \{\bot\} \cup \{[l,u] | l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{\infty\}, l \leq u\}$
On définit une relation $\sqsubseteq$ sur les intervalles comme suit :
$[l_1, u_1] \sqsubseteq [l_2, u_2]$ *si, et seulement si,* $l2 \leq l1$ *et* $u_1 \leq u_2$.

On peut montrer que $(\mathbb{I}(\mathbb{Z}), \sqsubseteq)$ est une structure partiellement ordonnée.

Question 7.1 *Montrer que*

1. $[l_1, u_1] \sqcup [l_2, u_2] = [min(l_1, l_2), max(u_1, u_2)]$
2. *et que* $[l_1, u_1] \sqcap [l_2, u_2] = \begin{cases} [max(l_1, l_2), min(u_1, u_2)] \\ \bot, si\ max(l_1, l_2) > min(u_1, u_2) \end{cases}$

On peut montrer que $(\mathbb{I}(\mathbb{Z}), \sqsubseteq)$ est un treillis complet.

On définit deux fonctions :
$\alpha(X) = \begin{cases} [min(X), max(X)] \\ \bot, si\ X = \varnothing \end{cases}$
$\gamma([l,u]) = [l..u]$ *et* $\gamma(\bot) = \varnothing$

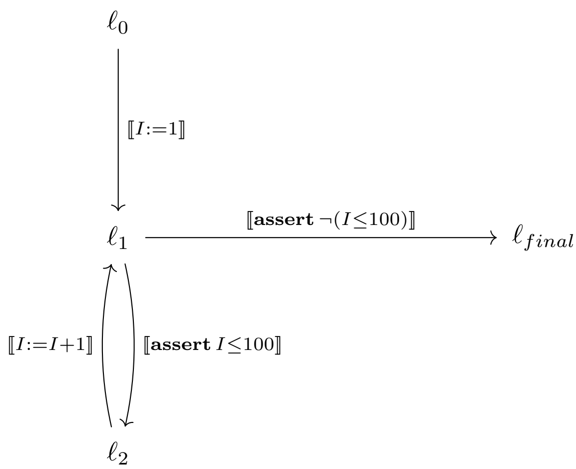
Question 7.2 *Montrer que (α, γ) est une connexion de Galois cést-à-dire que*
Pour toute partie X de $\mathbb{Z}$, pour toutes les valeurs m et M de $\mathbb{Z}$, $\alpha(X) \sqsubseteq [m, M]$ si, et seulement si, $X \subseteq \gamma([m, M])$

On définit des opérations sur les intervalles comme suit :

1. $i_1 \oplus i_2 = [l_1 + l_2, u_1 + u_2]$
2. $i_1 \ominus i_2 = [l_1 - u_2, u_1 - l_2]$
3. $i_1 \otimes i_2 = [min(l_1 \cdot l_2, l_1 \cdot u_2, u_1 \cdot l_2, u_1 \cdot u_2, max(l_1 \cdot l_2, l_1 \cdot u_2, u_1 \cdot l_2, u_1 \cdot u_2)]$
4. $i_1 \oslash i_2 = [min(l_1/l_2, l_1/u_2, u_1/l_2, u_1/u_2, max(l_1/l_2, l_1/u_2, u_1/l_2, u_1/u_2)]$

Question 7.3 *Appliquer la technique de calcul abstrait sur cet exemple.*

$\ell_0[I := 1]$;
WHILE $\ell_1[I \leq 100]$
$\ell_2[I := I+1]$;
END-WHILE
$\ell_{final}[skip]$



◇— **Solution de la question 7.3**

On peut utiliser l'analyseur Interproc http ://pop-art.inrialpes.fr/interproc/ et on peut utiliser différentes analyses notamment l'analyse dite box qui correspond aux intervalles.

```
Result

Annotated program after forward analysis
var I : int;
begin
  /* (L2 C5) top */
  I = 1; /* (L3 C6) [|I-1>=0; -I+101>=0|] */
  while I <= 100 do
    /* (L4 C20) [|I-1>=0; -I+100>=0|] */
    I = I + 1; /* (L5 C9)
                 [|I-2>=0; -I+101>=0|] */
  done; /* (L6 C5) [|I-101=0|] */
end

Source

var I:int;
begin
I = 1;
while  (I <= 100) do
 I = I+1;
done;
end
```

Fin 7.3

Exercice 8

```
#include <stdio.h>
#include <stdlib.h>
#define N 624
main()
{
  /* D\'eclarations des variables */
  int i,s,r;
```

precondition : ...
postcondition : ...

$\ell_0[i = 1;]$
$\ell_1[s = 0;]$
$\ell_2[r = -1;]$
while $\ell_3[s <= N]$ **do**
$\quad \ell_4[s+ = i]; \ell_5[i+ = 2]; \ell_6[r{+}{+};];$
;
$\ell_{final}[skip]$

Algorithme 2: $PROGRAM1$ annotée

Question 8.1 *Produire le graphe de flôt d'analyse.*

Question 8.2 *Ecrire le système d'équations définissant la sémantique collectrice.*

Question 8.3 *Ecrire une analyse abstraite dans le cadre du domaine des intervalles.*

◇— **Solution de la question 8.3**

```
Annotated program after forward analysis
var I : int, S : int, R : int, N : int;
begin
  /* (L2 C5) top */
  I = 1; /* (L3 C4) [|I-1=0|] */
  S = 0; /* (L4 C4) [|I-1=0; S=0|] */
  R = -1; /* (L5 C5) [|I-1>=0; R+1>=0; S>=0|] */
  while S <= N do
    /* (L6 C16)
       [|I-1>=0; N>=0; R+1>=0; S>=0|] */
    S = S + I; /* (L7 C6)
                  [|I-1>=0; N>=0; R+1>=0; S-1>=0|] */
    I = I + 2; /* (L8 C6)
                  [|I-3>=0; N>=0; R+1>=0; S-1>=0|] */
    R = R + 1; /* (L9 C6)
                  [|I-3>=0; N>=0; R>=0; S-1>=0|] */
  done; /* (L10 C5) [|I-1>=0; R+1>=0; S>=0|] */
end
```

```
Source

var I:int,S:int,R:int,N:int;
begin
I=1;
S=0;
R=-1;
while (S<= N) do
S=S+I;
I=I+2;
R=R+1;
done;
end
```

Fin 8.3