AUTOMATIC SYNTHESIS OF OPTIMAL INVARIANT ASSERTIONS : MATHEMATICAL FOUNDATIONS

Patrick COUSOT Radhia COUSOT

University of GRENOBLE
FRANCE

DEDUCTIVE SEMANTICS OF PROGRAMS

Program :

$\{P_1(x,y,\bar{x},\bar{y})\}$ <u>while</u> $x \geq y$ <u>do</u>

$\{P_2(x,y,\bar{x},\bar{y})\}$ $x := x - y$;

$\{P_3(x,y,\bar{x},\bar{y})\}$

<u>od</u>;

$\{P_4(x,y,\bar{x},\bar{y})\}$

A system of forward equations can be associated with the program by application of the rules defining the semantics of the elementary instructions :

$$
\begin{cases}
P_1(x,y,\bar{x},\bar{y}) = \{(x=\bar{x}) \text{ \underline{and} } (y=\bar{y})\} \\
P_2(x,y,\bar{x},\bar{y}) = \{P_1(x,y,\bar{x},\bar{y}) \text{ \underline{or} } P_3(x,y,\bar{x},\bar{y})\} \text{ \underline{and} } (x \geq y) \\
P_3(x,y,\bar{x},\bar{y}) = \{\exists v : P_2(v,y,\bar{x},\bar{y}) \text{ \underline{and} } x = v - y\} \\
\qquad\qquad\quad = P_2(x+y, y, \bar{x}, \bar{y}) \\
P_4(x,y,\bar{x},\bar{y}) = \{P_1(x,y,\bar{x},\bar{y}) \text{ \underline{or} } P_3(x,y,\bar{x},\bar{y})\} \text{ \underline{and} } (x<y)
\end{cases}
$$

system of the form :

$$P = F(P)$$

where

$$P = (P_1, P_2, P_3, P_4)$$

- The system of equations $P = F(P)$ has several solutions.
- An optimal solution P^{opt} exists. This SET P^{opt} OF OPTIMAL INVARIANT ASSERTIONS has the following properties:

- Solution to the system of equations:

$$P^{opt} = F(P^{opt})$$

- P^{opt} implies any other set of invariants:

$$\text{if} \quad P = F(P) \quad \underline{\text{then}} \quad P^{opt} \Rightarrow P$$

- P^{opt} is unique.
- Let $S_h(\bar{X})$ be the set of states of the variables X at point h of the program during any execution of the program starting with input values $\bar{X}$. ($S_h(\bar{X})$ is defined by the operational semantics of the language). Then P_h^{opt} exactly characterizes S_h:

$$S_h(\bar{X}) = \{X : P_h^{opt}(X, \bar{X})\}$$

- The theorem of TARSKI shows that

$$P^{opt} = \underline{\text{AND}} \{P : F(P) \Rightarrow P\}$$

(this formula is not constructive)

PROOF OF TOTAL CORRECTNESS

1. Specification

Input Specification $\phi(\bar{x})$ $\longrightarrow$ $\boxed{\text{program}}$ $\longrightarrow$ Output specification $\psi(x,\bar{x})$

2. operational proof:

$\forall\, \bar{x} : \phi(\bar{x})$, for some haltpoint h the set of final states S_h must not be empty $S_h(\bar{x}) = \{y\}$ (therefore the program terminate) and the final state y of the variables must satisfy the output specification ($\phi(y,\bar{x})$ must be true and therefore the program is partially correct).

3. equivalent logical proof:

$$\forall \bar{x} : \phi(\bar{x}),\ \exists h, \exists y :\ \underbrace{P_h^{opt}(y,\bar{x})}_{\text{termination}}\ \text{and}\ \underbrace{\phi(y,\bar{x})}_{\text{correctness}}$$

PROOF OF TOTAL CORRECTNESS (EXAMPLE)

Program :

```
{1}
{2}  while x >= y do
        x := x - y;
{3}
{4}  od;
```

Optimal invariants :

$$P_4^{opt} = \{\exists j \geq 0 : (\forall k \in [1, j], \bar{x} \geq k\bar{y}) \text{ and } \bar{x} < (j+1)\bar{y} \text{ and } x = \bar{x} - j\bar{y} \text{ and } y = \bar{y}\}$$

Proof of non-termination when $\bar{x} \geq 0$, $\bar{y} = 0$

$$(\forall (\bar{x}, \bar{y}) : \bar{x} \geq 0 \text{ and } \bar{y} = 0), \exists h, \exists (x, y) : P_h^{opt}(x, y, \bar{x}, \bar{y})$$

$$P_4^{opt}(x, y, \bar{x}, \bar{y}) = \{\exists j \geq 0 : \text{--------} \text{ and } \bar{x} < 0 \text{ and } x = \bar{x} \text{ and } y = \bar{y}\}$$
$$= \underline{false}$$

Input condition $\phi(\bar{x}, \bar{y})$ guaranteeing the termination :

$$\phi(\bar{x}, \bar{y}) = \exists (x, y) : P_4^{opt}(x, y, \bar{x}, \bar{y})$$
$$= \{\exists j \geq 0 : (\forall k \in [1, j]\ \bar{x} \geq k\bar{y}) \text{ and } \bar{x} < (j+1)\bar{y}\}$$
$$= \{(0 < \bar{y}) \text{ or } (\bar{x} < \bar{y} \leq 0)\}$$

(Alternative to FLOYD's method).

CONSTRUCTIVE DEFINITION OF THE SET OF OPTIMAL INVARIANT ASSERTIONS.

by the iterative method of successive approximations:

$$p^0 = \underline{false}$$
$$p^1 = F(p^0)$$
$$\vdots$$
$$p^{i+1} = F(p^i)$$
$$\vdots$$
$$p^{opt} = \lim_{i \to \infty} F^i(p^0)$$

the problem of computing p^{opt} is undecidable
$\Rightarrow$ the sequence of approximations is usually infinite

Any chaotic iteration method can be used: one can arbitrarily determine at each step which are the components of the system of equations

$$\begin{cases} P_1 = f_1(P_1, \ldots, P_n) \\ \vdots \\ P_n = f_n(P_1, \ldots, P_n) \end{cases}$$

which will evolve and in what order (as long as no component is forgotten indefinitely).

SYMBOLIC EXECUTION consists in solving the semantic equations by chaotic iterations

Program :

$\{P_1\}$ **while** $x \geq y$ **do**
$\{P_2\}$ $x := x - y$;
$\{P_3\}$
$\{P_4\}$ **od** ;

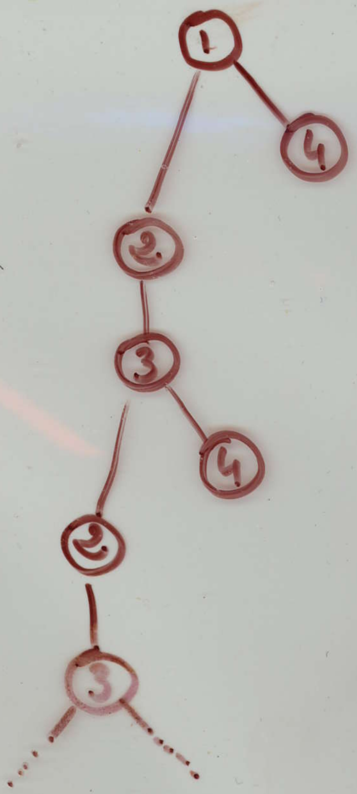
Equations :

$$\begin{cases} P_1 = cte \\ P_2 = f'(P_1, P_3) = f(P_3) \\ P_3 = g(P_2) \\ P_4 = h'(P_1, P_3) = h(P_3) \end{cases}$$

Symbolic execution

implicitly : $P_1^0 = P_2^0 = P_3^0 = P_4^0 = \underline{false}$

symbolic execution tree :



$P_1^1 = cte$

$P_4^1 = h(P_3^0) = h(\underline{false})$

$P_2^1 = f(P_3^0) = f(\underline{false})$

$P_3^1 = g(P_2^1) = g(f(\underline{false}))$

$P_4^2 = h(P_3^1) = h(g(f(\underline{false})))$

$P_2^2 = f(P_3^1) = f(g(f(\underline{false})))$

$P_3^2 = g(P_2^2) = g(f(g(f(\underline{false}))))$

$= (\bar{x} \geq \bar{y}$ **and** $x = \bar{x} - \bar{y}$ **and** $y = \bar{y})$

or

$(\bar{x} \geq \bar{y}$ **and** $x \geq 2\bar{y}$ **and** $x = \bar{x} - 2\bar{y}$ **and** $y = \bar{y})$

PROBLEM :
Passage to the limit

SYNTHESIS OF OPTIMAL INVARIANT ASSERTIONS :

The use of difference equations.

Program :

```
{P1}  while x ≥ y do
{P2}
{P3}      x := x - y;
{P4}  od;
```

Equations :

$$P_2 = f(P_1) \text{ \underline{or} } g(P_2)$$

where $f(P_1) = x = \bar{x} \text{ \underline{and} } y = \bar{y} \text{ \underline{and} } \bar{x} \geq \bar{y}$

$g(P_2) = P_2(x+y, y, \bar{x}, \bar{y}) \text{ \underline{and} } x \geq y$

Resolution :

$$\begin{tikzcd} P_1 \arrow[d, "f"] \\ P_2 \arrow[loop below, "g"] \end{tikzcd} \qquad P_2 = \underset{i=0}{\overset{\infty}{\text{OR}}}\; g^i(f(P_1))$$

Difference equations :

$$g^0(f(P_1)) = x = \bar{x} \text{ \underline{and} } y = \bar{y} \text{ \underline{and} } \bar{x} \geq \bar{y}$$

$$= x = x_0 \text{ \underline{and} } y = y_0 \text{ \underline{and} } c_0$$

$$g^i(f(P_1)) = \boxed{x = x_i \text{ \underline{and} } y = y_i \text{ \underline{and} } c_i}$$

$$g(g^i(f(P_1))) = x = x_i - y_i \text{ \underline{and} } y = y_i \text{ \underline{and} } (c_i \text{ \underline{and} } x_i \geq y_i)$$

$$g^{i+1}(f(P_1)) = x = x_{i+1} \text{ \underline{and} } y = y_{i+1} \text{ \underline{and} } c_{i+1}$$

Resolution :

$$y_0 = \bar{y}, \quad y_{i+1} = y_i \implies y_i = \bar{y}$$

$$x_0 = \bar{x}, \quad x_{i+1} = x_i - \bar{y} \implies x_i = \bar{x} - i\bar{y}$$

$$\left.\begin{array}{l} c_0 = \bar{x} \geq \bar{y} \\ c_{i+1} = c_i \text{ \underline{and} } \bar{x} \geq (i+1)\bar{y} \end{array}\right\} \quad c_i = \underset{k=1}{\overset{\max(i,1)}{\text{AND}}} (\bar{x} \leq k\bar{y})$$

Solution :

$$P_2^{opt} = \left\{ \underset{i=1}{\overset{\infty}{\text{OR}}} \left(\underset{k=1}{\overset{i}{\text{AND}}} (\bar{x} \leq k\bar{y}) \text{ \underline{and} } x = \bar{x} - i\bar{y} \text{ \underline{and} } y = \bar{y} \right) \right\}$$

NOTION OF APPROXIMATE INVARIANTS

Program :

{1}
{2} **while** $x \geq y$ **do**
{3} $x := x - y$;
{4} **od** ;

Example of approximate invariants :

$$\begin{cases} P_1 = (x = \bar{x}) \text{ and } (y = \bar{y}) \\ P_2 = (x \geq y) \text{ and } (y = \bar{y}) \\ P_3 = (x \geq 0) \text{ and } (y = \bar{y}) \\ P_4 = (x < y) \text{ and } (y = \bar{y}) \text{ and } \{x = \bar{x} \text{ or } x \geq 0\} \end{cases}$$

System of implications :

$$\begin{cases} P_1 \Leftarrow (x = \bar{x}) \text{ and } (y = \bar{y}) \\ P_2 \Leftarrow (P_1 \text{ or } P_3) \text{ and } x \geq y \\ P_3 \Leftarrow P_2(x+y, y, \bar{x}, \bar{y}) \\ P_4 \Leftarrow (P_1 \text{ or } P_3) \text{ and } (x < y) \end{cases}$$

$\boxed{P \Leftarrow F(P)}$

Partial Correctness :

Input condition : $\phi(\bar{x}, \bar{y}) = (\bar{x} \geq 0)$

Output specification : $\psi(x, y, \bar{x}, \bar{y}) = (y > x \geq 0)$

Proof of partial correctness :

$$P_4(x, y, \bar{x}, \bar{y}) \Longrightarrow \psi(x, y, \bar{x}, \bar{y})$$

WHY CAN WE USE A SET P OF APPROXIMATE INVARIANTS (such that $P \Leftarrow F(P)$) FOR PARTIAL CORRECTNESS PROOFS ?

- **Proof of Partial Correctness** :

$$\{\forall \bar{x} : \phi(\bar{x}),\ \forall h,\ \forall y : P_h^{opt}(y,\bar{x}) \Rightarrow \phi(y,\bar{x})\}$$

but

$$P^{opt} = \underline{AND}\{P : F(P) \Rightarrow P\}$$

hence the partial correctness condition is :

$$\{\forall \bar{x} : \phi(\bar{x}),\ \forall h,\ \forall y : \underline{AND}\{P_h(y,\bar{x}) : F(P) \Rightarrow P\} \Rightarrow \phi(y,\bar{x})\}$$

$$\equiv \{\forall x : \phi(\bar{x}),\ \forall h,\ \forall y,\ \exists P : P \Leftarrow F(P) \text{ and } P_h(y,\bar{x}) \Rightarrow \phi(y,\bar{x})\}$$

- **Proof of termination** :

$$\{\forall \bar{x} : \phi(\bar{x}),\ \exists h,\ \exists y : P_h^{opt}(y,\bar{x})\}$$

$$\{\forall P : F(P) \Rightarrow P,\ \forall \bar{x} : \phi(\bar{x}),\ \exists h,\ \exists y : P_h(y,\bar{x})\}$$

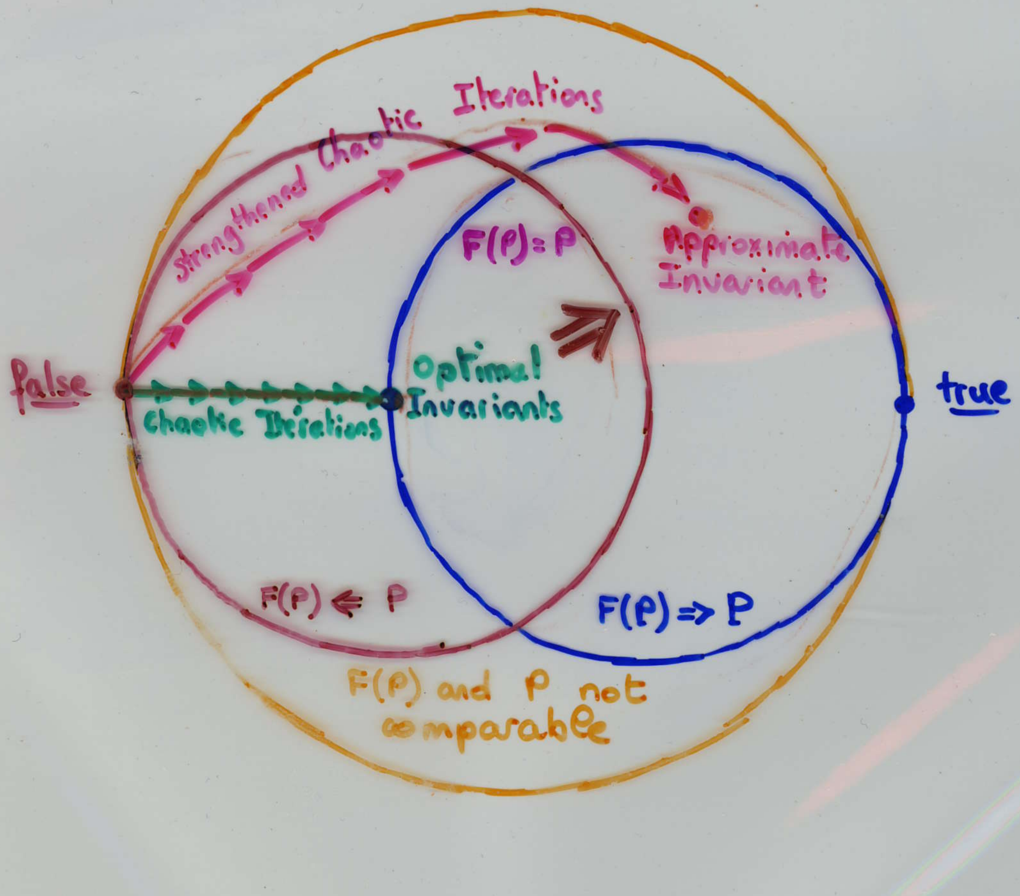
not utilizable in practice since the definition of the optimal invariants in term of the approximate invariants is not constructive.

SYNTHESIS OF APPROXIMATE ASSERTIONS

- By successive strengthened approximations :
  - $P^0 = \underline{\text{false}}$
  - P^{i+1} : $\left.\begin{array}{r} P^i \Rightarrow P^{i+1} \\ F(P^i) \Rightarrow P^{i+1} \end{array}\right\}$ whenever $\underline{\text{not}}\left(F(P^i) \Rightarrow P^i\right)$

$\Rightarrow$ $P = \lim_{i \to \infty} P^i$ is such that $P \Leftarrow F(P)$

-



SYNTHESIS OF APPROXIMATE ASSERTIONS (EXAMPLE)

Program :

```
{1}
     while x >= y do
{2}
        x := x - y;
{3}
     od;
{4}
```

Equations :

$$P_2 = (\bar{x} \geq \bar{y} \text{ and } x = \bar{x} \text{ and } y = \bar{y}) \text{ or } (x \geq y \text{ and } P_2(x+y, y, \bar{x}, \bar{y}))$$

Strengthened chaotic iteration sequence :

$P_2^0 = \text{false}$

$P_2^1 = \bar{x} \geq \bar{y} \text{ and } x = \bar{x} \text{ and } y = \bar{y}$

$$P_2^2 = (\bar{x} \geq \bar{y} \text{ and } x = \bar{x} \text{ and } y = \bar{y}) \text{ or } ((\forall k \in [1,2], \bar{x} \geq k\bar{y}) \text{ and } x = \bar{x} - \bar{y} \text{ and } y = \bar{y})$$

$$F(P_2^2) = (\bar{x} \geq \bar{y} \text{ and } x = \bar{x} \text{ and } y = \bar{y}) \text{ or } ((\forall k \in [1,2], \bar{x} \geq k\bar{y}) \text{ and } x = \bar{x} - \bar{y} \text{ and } y = \bar{y}) \text{ or } ((\forall k \in [1,3], \bar{x} \geq k\bar{y}) \text{ and } x = \bar{x} - 2\bar{y} \text{ and } y = \bar{y})$$

since $\text{not}\,(F(P_2^2) \Rightarrow P_2^2)$, P_2^2 is strengthened :

$P_2^3 = \{\exists j \in [0,2] : x = \bar{x} - j\bar{y} \text{ and } y = \bar{y}\}$

notice that $P_2^2 \Rightarrow P_2^3$ and $F(P_2^2) \Rightarrow P_2^3$

$F(P_2^3) = \{\exists j \in [0,3] : \bar{x} \geq (j+1)\bar{y} \text{ and } x = \bar{x} - j\bar{y} \text{ and } y = \bar{y}\}$

since $\text{not}\,(F(P_2^3) \Rightarrow P_2^3)$, P_2^3 is strengthened :

$P_2^4 = \{\exists j \geq 0 : x = \bar{x} - j\bar{y} \text{ and } y = \bar{y}\}$

notice that $P_2^3 \Rightarrow P_2^4$ and $F(P_2^3) \Rightarrow P_2^4$

$F(P_2^4) = \{\exists j \geq 0 : \bar{x} \geq (j+1)\bar{y} \text{ and } x = \bar{x} - j\bar{y} \text{ and } y = \bar{y}\}$

since $F(P_2^4) \Rightarrow P_2^4$, P_2^4 is an approximate invariant !

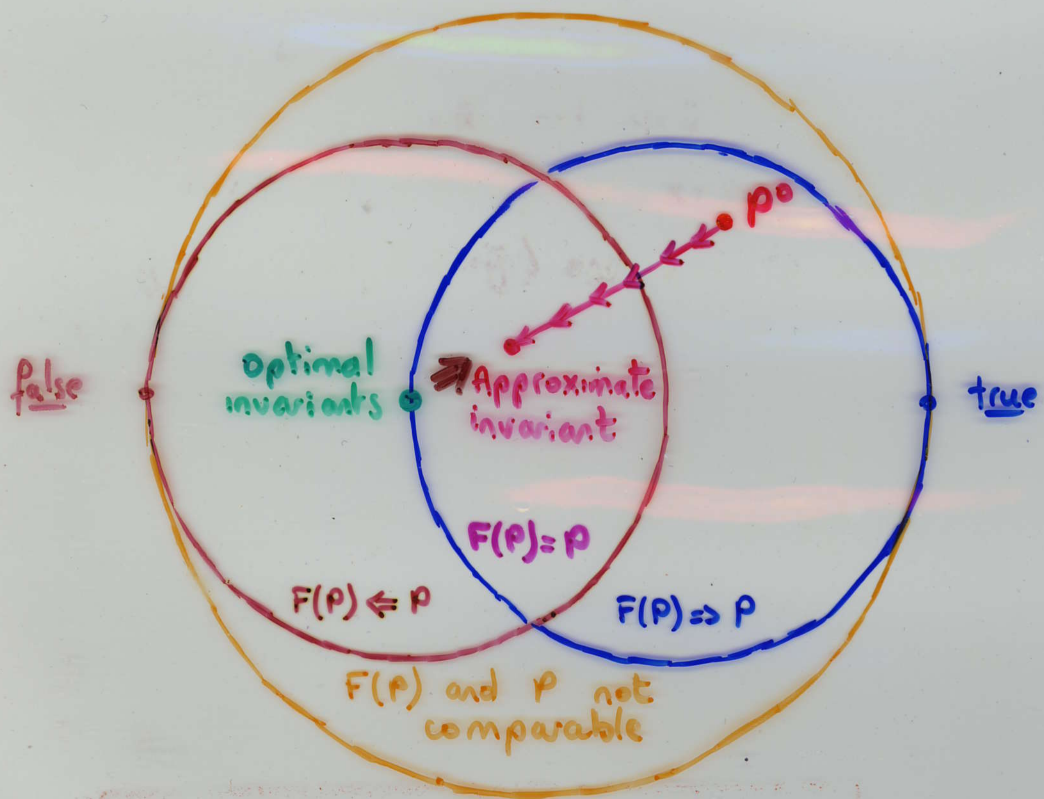
SYNTHESIS OF APPROXIMATE ASSERTIONS
(CONTINUED)

- By successive weakened approximations :

  - P^0 such that $F(P^0) \Rightarrow P^0$
  - P^{i+1} : $F(P^i) \Rightarrow P^{i+1} \Rightarrow P^i$

$\Rightarrow P = \lim_{i \to \infty} P^i$ is such that $\begin{cases} F(P) \Rightarrow P \\ P \Rightarrow P^0 \end{cases}$

-



SYNTHESIS OF APPROXIMATE ASSERTIONS (EXAMPLE)

Program :

{1} <u>while</u> $x \geq y$ <u>do</u>
{2}
    $x := x - y$;
{3}
  <u>od</u>;
{4}

Equations :

$$\begin{cases} P_1 = x = \bar{x} \text{ and } y = \bar{y} \\ P_2 = (P_1 \text{ or } P_3) \text{ and } x \geq y \\ P_3 = P_2(x+y, y, \bar{x}, \bar{y}) \\ P_4 = (P_1 \text{ or } P_3) \text{ and } x < y \end{cases}$$

Weakened chaotic iteration sequence :

$$P_2^0 = (\exists j \geq 0 : x = \bar{x} - j\bar{y} \text{ and } y = \bar{y})$$

$$F(P_2^0) = (\exists j \geq 0 : \bar{x} \geq (j+1)\bar{y} \text{ and } x = \bar{x} - j\bar{y} \text{ and } y = \bar{y})$$

such that $F(P_2^0) \Rightarrow P_2^0$, choosing $P_2^1 = F(P_2^0)$

$$\boxed{P_2^1 = (\exists j \geq 0 : \bar{x} \geq (j+1)\bar{y} \text{ and } x = \bar{x} - j\bar{y} \text{ and } y = \bar{y})}$$

$$F(P_2^1) = (\exists j \geq 0 : (j = 0 \text{ or } \bar{x} \geq j\bar{y}) \text{ and } \bar{x} \geq (j+1)\bar{y} \text{ and } x = \bar{x} - j\bar{y} \text{ and } y = \bar{y})$$

weakening :

$$P_2^2 = (\exists j \geq 0 : \bar{x} \geq (j+1)\bar{y} \text{ and } x = \bar{x} - j\bar{y} \text{ and } y = \bar{y})$$

$$= P_2^1$$

stop.

CONCLUSION

- The synthesis of invariant assertions consists in computing the optimal (total correctness) or approximate (partial correctness) solution to a system of equations defining the semantics of the program

- Mathematicians have studied the resolution of equations during centuries. However they have not been interested in solving logical equations
  - A new research area is opened
  - Analogy with mathematics and numerical analysis can give ideas to find methods for solving these equations.
  - The techniques of Artificial Intelligence will be necessary. (as opposed to numerical software)