TRABAJO SEMESTRAL

Luis Pinochet

Federico Olmedo

PROGRAMA DE EJEMPLO

1. x := 1; b:=1
2. while (b==1) {b:= 0.5<0> + 0.5 <1>; x:=2x}
3. while(x>0){x:=x-1}

$$\mathbf{ert}[C](\mathbf{0}) = \mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})))$$

TRANSFORMADA

$$\mathbf{ert}[C](\mathbf{0}) = \mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})))$$

TRANSFORMADA

ert$[C]$(**0**) = **ert**$[C_1]$(**ert**$[C_2]$(**ert**$[C_3]$(**0**)))

- **ert**$[C_3]$(**0**) >= **1** + [x > 0] * 2x
- **ert**$[C_2]$(**ert**$[C_3]$(**0**)) >= **ert**$[C_2]$(**1** + [x > 0]*2x) >= 1 + [b!=1]*(**1** + [x >0]*2x)

  + [b==1]*(**7** + [x > 0]***inf**)

TRANSFORMADA

$\mathbf{ert}[C](\mathbf{0}) = \mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})))$

- $\mathbf{ert}[C_3](\mathbf{0}) >= \mathbf{1} + [x > 0] * 2x$
- $\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})) >= \mathbf{ert}[C_2](\mathbf{1} + [x > 0]*2x) >= 1 + [\, b!=1]*(\mathbf{1} + [x >0]*2x)$

  $+ [b==1]*(\, \mathbf{7} + [x > 0]*\mathbf{inf})$

- $\mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0}))) >= \mathbf{ert}[C_1](1 + [\, b!=1]*(\mathbf{1} + [x >0]*2x) + [b==1]*(\mathbf{7} + [x>0]*\mathbf{inf}))$

TRANSFORMADA

ert$[C]$(**0**) = **ert**$[C_1]$(**ert**$[C_2]$(**ert**$[C_3]$(**0**)))

- **ert**$[C_3]$(**0**) >= **1** + [x > 0] * 2x
- **ert**$[C_2]$(**ert**$[C_3]$(**0**)) >= **ert**$[C_2]$(**1** + [x > 0]*2x) >= 1 + [b!=1]*(**1** + [x >0]*2x)

  + [b==1]*(**7** + [x > 0]***inf**)
- **ert**$[C_1]$(**ert**$[C_2]$(**ert**$[C_3]$(**0**))) >= **ert**$[C_1]$(1 + [b!=1]*(**1** + [x >0]*2x) + [b==1]*(**7** + [x>0]***inf**))

  = **8 + inf**

TRANSFORMADA

ert$[C]$(**0**) = **ert**$[C_1]$(**ert**$[C_2]$(**ert**$[C_3]$(**0**)))

- **ert**$[C_3]$(**0**) >= **1** + [x > 0] * 2x
- **ert**$[C_2]$(**ert**$[C_3]$(**0**)) >= **ert**$[C_2]$(**1** + [x > 0]*2x) >= 1 + [b!=1]*(**1** + [x >0]*2x)

  + [b==1]*(**7** + [x > 0]***inf**)
- **ert**$[C_1]$(**ert**$[C_2]$(**ert**$[C_3]$(**0**))) >= **ert**$[C_1]$(1 + [b!=1]*(**1** + [x >0]*2x) + [b==1]*(**7** + [x>0]***inf**))

  = 8 + inf = inf

OBJETIVO GENERAL

El objetivo general de esta memoria es desarrollar una herramienta que calcule de manera automática el tiempo de ejecución de programas probabilísticos, usando la técnica presentada por Kaminski et al[6]. . En particular, la herramienta va a tomar un programa donde cada bucle está anotado con su respectivo *invariante*, y va a devolver una cota superior del tiempo de ejecución del programa. En caso de que el programa no contenga bucles se espera calcular con exactitud el tiempo estimado de ejecución del mismo.

OBJETIVO GENERAL

El objetivo general de esta memoria es desarrollar una herramienta que calcule de manera automática el tiempo de ejecución de programas probabilísticos, usando la técnica presentada por Kaminski et al[6]. . En particular, la herramienta va a tomar un programa donde cada bucle está anotado con su respectivo *invariante*, y va a devolver una cota superior del tiempo de ejecución del programa. En caso de que el programa no contenga bucles se espera calcular con exactitud el tiempo estimado de ejecución del mismo.

PROGRAMA DE EJEMPLO

1. x := 1; b:=1
2. while (b==1) {b:= 0.5<0> + 0.5 <1>; x:=2x}
3. while(x>0){x:=x-1}

$\mathbf{ert}[C](\mathbf{0}) = \mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})))$

PROGRAMA DE EJEMPLO

1. x := 1; b:=1
2. while (b==1) {b:= 0.5<0> + 0.5 <1>; x:=2x}
3. while(x>0){x:=x-1}

$\mathbf{ert}[C](\mathbf{0}) = \mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})))$

TRANSFORMADA

ert$[C]$(**0**) = **ert**$[C_1]$(**ert**$[C_2]$(**ert**$[C_3]$(**0**)))

- **ert**$[C_3]$(**0**) >= **1** + [x > 0] * 2x
- **ert**$[C_2]$(**ert**$[C_3]$(**0**)) >= **ert**$[C_2]$(**1** + [x > 0]*2x) >= 1 + [b!=1]*(**1** + [x >0]*2x)

  + [b==1]*(**7** + [x > 0]***inf**)
- **ert**$[C_1]$(**ert**$[C_2]$(**ert**$[C_3]$(**0**))) >= **ert**$[C_1]$(1 + [b!=1]*(**1** + [x >0]*2x) + [b==1]*(**7** + [x>0]***inf**))
  = 8 + inf = inf

TRANSFORMADA

$$\mathbf{ert}[C](\mathbf{0}) = \mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})))$$

- $\mathbf{ert}[C_3](\mathbf{0}) >= \boxed{\mathbf{1} + [x > 0] * 2x}$
- $\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0})) >= \mathbf{ert}[C_2](\mathbf{1} + [x > 0]*2x) >= \boxed{\begin{array}{l} 1 + [\, b!=1]*(\mathbf{1} + [x > 0]*2x) \\ \quad + [b==1]*(\,\mathbf{7} + [x > 0]*\mathbf{inf}) \end{array}}$

- $\mathbf{ert}[C_1](\mathbf{ert}[C_2](\mathbf{ert}[C_3](\mathbf{0}))) >= \mathbf{ert}[C_1](1 + [\, b!=1]*(\mathbf{1} + [x > 0]*2x) + [b==1]*(\mathbf{7} + [x>0]*\mathbf{inf}))$

  $= \mathbf{8 + inf} = \mathbf{inf}$

OBJETIVO ESP 1

Generación del conjunto de restricciones. Dado un programa con las anotaciones de *invariantes* respectivas y un tiempo de ejecución candidato f, se busca generar el conjunto de restricciones que debe satisfacer f para que sea efectivamente una cota superior del tiempo de ejecución del programa.

INPUTS

- Candidato :

INPUTS

- Candidato : f = **15*x + b**

INPUTS

- Candidato : f = **15*x + b**
- Anotaciones :

INPUTS

- Candidato : **f = 15*x + b**
- Anotaciones :
  - C_3 => **1** + [x > 0] * 2x
  - C_2 => **1** + [b!=1]*(**1** + [x >0]*2x) + [b==1]*(**7** + [x>0]***inf**))

GENERACIÓN DE RESTRICCIONES

TRANSFORMADA

ert[C](**0**) = **ert**[C_1](**ert**[C_2](**ert**[C_3](**0**)))

- **ert**[C_3](**0**) >= **1** + [x > 0] * 2x
- **ert**[C_2](**ert**[C_3](**0**)) >= **ert**[C_2](**1** + [x > 0]*2x) >= 1 + [b!=1]*(**1** + [x >0]*2x)

  + [b==1]*(**7** + [x > 0]***inf**)

- **ert**[C_1](**ert**[C_2](**ert**[C_3](**0**))) >= **ert**[C_1](1 + [b!=1]*(**1** + [x >0]*2x) + [b==1]*(**7** + [x>0]***inf**))

  = **8 + inf = inf**

TEOREMAS

Definition 3 (Upper invariants). *Let* $f \in \mathbb{T}$, $C \in \mathsf{pProgs}$ *and* $\xi \in \mathsf{DExp}$. *We say that* $I \in \mathbb{T}$ *is an* upper invariant *of loop* $\texttt{while}\,(\xi)\,\{C\}$ *with respect to* f *iff*

$$\mathbf{1} + [\![\xi\colon \mathsf{false}]\!] \cdot f + [\![\xi\colon \mathsf{true}]\!] \cdot \mathsf{ert}\,[C]\,(I) \quad \preceq I$$

or, equivalently, iff $F_f^{\langle \xi, C\rangle}(I) \preceq I$, *where* $F_f^{\langle \xi, C\rangle}$ *is the characteristic functional.*

The presence of an upper invariant of a loop readily establishes an upper bound of the loop's run–time.

Theorem 3 (Upper bounds from upper invariants). *Let* $f \in \mathbb{T}$, $C \in \mathsf{pProgs}$ *and* $\xi \in \mathsf{DExp}$. *If* $I \in \mathbb{T}$ *is an upper invariant of* $\texttt{while}\,(\xi)\,\{C\}$ *with respect to* f *then*

$$\mathsf{ert}\,[\texttt{while}\,(\xi)\,\{C\}]\,(f) \quad \preceq \quad I\ .$$

TEOREMAS

Definition 4 **(ω–invariants).** *Let $f \in \mathbb{T}$, $C \in$* pProgs *and $\xi \in$* DExp. *Moreover let $I_n \in \mathbb{T}$ be a run–time parametrized by $n \in \mathbb{N}$. We say that I_n is a* lower ω–invariant *of loop* `while` $(\xi)\ \{C\}$ *with respect to f iff*

$$F_f^{\langle \xi, C \rangle}(\mathbf{0}) \succeq I_0 \qquad \textit{and} \qquad F_f^{\langle \xi, C \rangle}(I_n) \succeq I_{n+1} \quad \textit{for all } n \geq 0 \ .$$

Dually, we say that I_n is an upper ω–invariant *iff*

$$F_f^{\langle \xi, C \rangle}(\mathbf{0}) \preceq I_0 \qquad \textit{and} \qquad F_f^{\langle \xi, C \rangle}(I_n) \preceq I_{n+1} \quad \textit{for all } n \geq 0 \ .$$

TEOREMAS

Definition 4 **(ω–inva[illegible]iants).** *Let* $f \in \mathbb{T}$, $C \in$ pProgs *[illegible]d* $\xi \in$ DExp. *Moreover let* $I_n \in \mathbb{T}$ *be a run–[illegible]metrized by* $n \in$ *[illegible]y that* I_n *is a* lower ω–invariant *of loop w[illegible]ith res[illegible]*

$$F_f^{\langle \xi, C \rangle}(\mathbf{0}) \succeq I_0 \qquad \text{an[illegible]} \; I_{n+1} \quad \text{for all } n \geq 0 \,.$$

Dually, we say that I[illegible]inva[illegible]

$$F_f^{\langle \xi, C \rangle}(\mathbf{0}) \preceq I\text{[illegible]} \qquad \text{and} \qquad F_f^{\langle \xi, C \rangle}(I_n) \preceq I_{n+1} \; \text{[illegible]or all } n \geq 0 \,.$$

GENERACIÓN DE RESTRICCIONES

- I por cada ciclo while usando el teorema 3

GENERACIÓN DE RESTRICCIONES

- 1 por cada ciclo while usando el teorema 3
- 1 para comparar el tiempo candidato **f** con el resultado calculado.

EN ESTE CASO

- **ert**[C_1](**ert**[C_2](**ert**[C_3](**0**))) **>= 8 + inf**

EN ESTE CASO

- **ert**$[C_1]$**(ert**$[C_2]$**(ert**$[C_3]$**(0))) >= 8 + inf**
- **f = 15** $*$ **x + b >= 8 + inf**

EN ESTE CASO

- **ert**$[C_1]$**(ert**$[C_2]$**(ert**$[C_3]$**(0))) >= 8 + inf**
- **f = 15** $*$ **x + b >= 8 + inf -> unsat**

OBJETICO ESP 2

Verificación del conjunto de restricciones. En este punto se busca verificar que el tiempo de ejecución f propuesto efectivamente satisfaga el conjunto de restricciones generados en el apartado anterior. Para ello se plantea el uso de *SMT solvers*.

FORMA RUNTIME

FORMA RUNTIME

- G(sigma) = **X**·**K**

FORMA RUNTIME

- G(**X**) = **X**·**K**, **X** variables y **K** >= 0

FORMA RUNTIME

- G(**X**) = **X**·**K**, **X** variables y (**K** >= 0)?
- G(**X**) = **X**·**K** **+** ∑ ∏ [de] $\boldsymbol{x_i k_i}$ (lo que tengo)

FORMA RUNTIME

- G(**X**) = **X**·**K**, **X** variables y **K** >= 0
- G(**X**) = **X**·**K** **+** ∑ ∏ [de] $\boldsymbol{x_i k_i}$ (lo que tengo)
- Para el caso que no haya ciclos no hay problemas.

FORMA RUNTIME

- G(**X**) = **X**·**K**, **X** variables y **K** >= 0
- G(**X**) = **X**·**K** + ∑ ∏ [de] $\boldsymbol{x_i k_i}$ (lo que tengo)
- Para el caso que no haya ciclos no hay problemas.
- Para el caso de los whiles, hay que investigar.

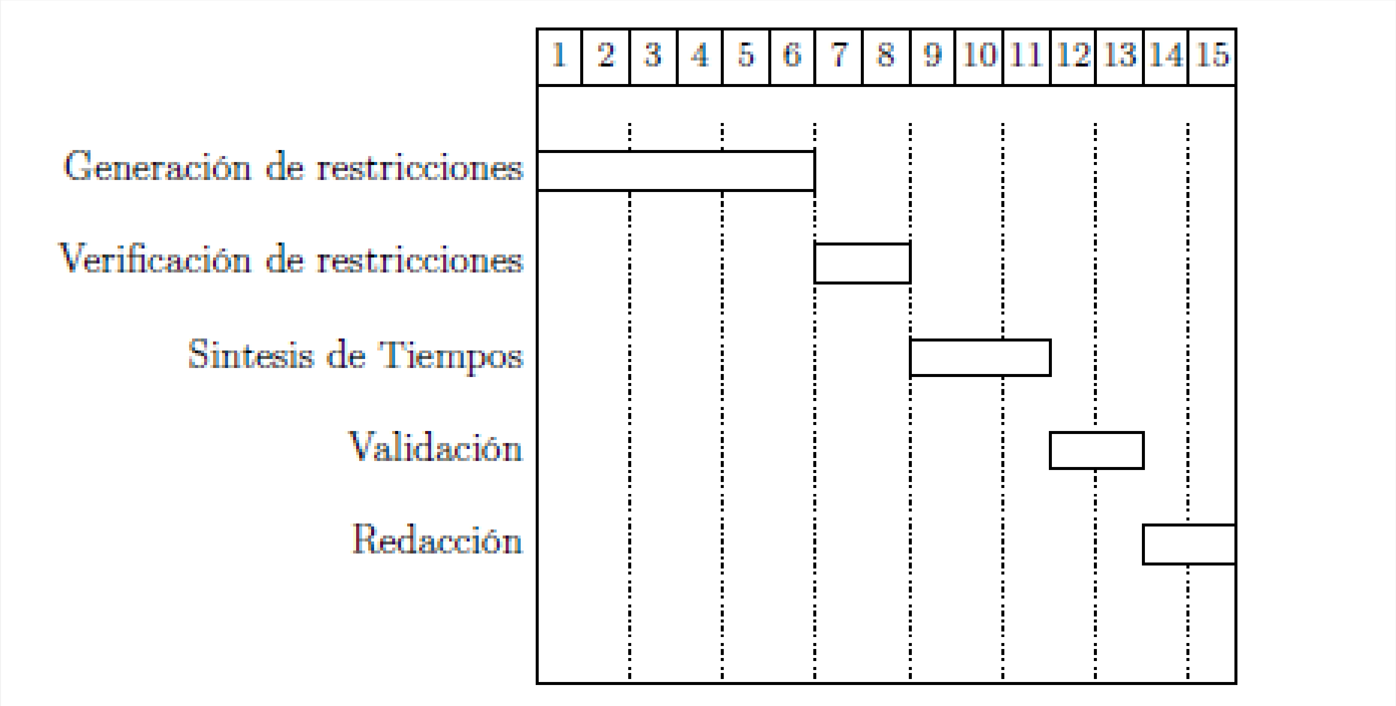
OBJETIVO ESP 3

Síntesis de los tiempos de ejecución. En este punto se busca dar el siguiente paso y poder sintetizar el menor tiempo de ejecución f que sea solución de las restricciones generadas en el primer apartado.

OBSERVACIONES

Busco en el espacio de funciones

CALENDARIO PROPUESTO



CALENDARIO PROPUESTO