Complete Functional Synthesis

Ruzica Piskac

∼ joint work with Viktor Kuncak, Mikaël Mayer and Philippe Suter ∼

Schloss Dagstuhl, April 2010

Motivation: Solving Integer Constraints

```
val totsec = .... (defined here)
....
val (hours, minutes, seconds) =
  choose((h: Int, m: Int, s: Int) =>
    h * 3600 + m * 60 + s == totsec &&
    0 =< m && m < 60 &&
    0 =< s && s < 60)
```

Motivation: Solving Integer Constraints

```
val (hours, minutes, seconds) = {
  val loc1 = totsec div 3600
  val num2 = totsec + ((−3600) ∗ loc1)
  val loc2 = min(num2 div 60, 59)
  val loc3 = totsec +
    (−3600 ∗ loc1) + (−60 ∗ loc2)
  (loc1, loc2, loc3)
}
```

Motivation: Solving Integer and Set Constraints

```
val bigSet = .... (defined here)
val maxDiff = .... (defined here)
....
val (setA, setB) =
  choose((a: Set[O], b: Set[O]) =>
   (−maxDiff =< a.size − b.size &&
    a.size − b.size =< maxDiff &&
    a union b == bigSet &&
    a intersect b == empty
))
```

Motivation: Solving Integer and Set Constraints

```
val i1 = bigSet.size − maxDiff
val i2 = bigSet.size + maxDiff
val i3 = (i1 / 2).ceiling
val i4 = (i2 / 2).floor
assert( i3 =< i4 )
val h101 = i4
val h011 = bigSet.size − i4
val setA = take(h101, bigSet)
val setB = take(h011, bigSet −− setA)
```

Synthesis Procedure

Definition

We denote an invocation of a synthesis procedure by $[\![\vec{x}, \vec{a}, F(\vec{x}, \vec{a}), \Gamma]\!] = (\text{pre}(\vec{a}), \Gamma_N)$, where:

- $\vec{a}$ is a vector of input parameters
- $\vec{x}$ is a vector of output parameters
- F is a formula defining constraints
- Γ is a code followed by “choose” command
- $\text{pre}(\vec{a})$ is a formula such that $\exists \vec{x}.F(\vec{x}, \vec{a}) \Leftrightarrow \text{pre}(\vec{a})$
- Γ_N is a newly generated code which effectively computes values of the output variables

Synthesis Procedure

1. emit a non-feasibility warning if the formula $\neg$pre is satisfiable, reporting the counterexample for which the synthesis problem has no solutions;
2. emit a non-uniqueness warning if the formula

   $$F \wedge F[\vec{x} := \vec{y}] \wedge \vec{x} \neq \vec{y}$$

   is satisfiable, reporting the values of all free variables as a counterexample showing that there are at least two solutions;
3. as the compiled code, emit the code that behaves as
   assert(pre)
   computation of a witness

Synthesis for Multiple Variables

$[\![(x_1, \ldots, x_{n-1}, x_n), \vec{a}, F(\vec{x}, \vec{a}), \Gamma]\!] = (\mathrm{pre}_2(\vec{a}), \Gamma :: \Gamma_2 :: \Gamma_1)$,
where:

- $(\mathrm{pre}_1(x_1, \ldots, x_{n-1}, \vec{a}), \Gamma_1) =$
  $[\![x_n, (x_1, \ldots, x_{n-1}) :: \vec{a}, F(\vec{x}, \vec{a}), ()]\!]$
- $(\mathrm{pre}_2(\vec{a}), \Gamma_2) = [\![(x_1, \ldots, x_{n-1}), \vec{a}, \mathrm{pre}_1(x_1, \ldots, x_{n-1}, \vec{a}), ()]\!]$

Synthesis for Disjunctions

$$[\![\vec{x}, \vec{a}, D_1 \vee \ldots \vee D_n, \Gamma]\!] =$$

$$\left(\bigvee_{i=1}^{n} \mathrm{pre}_i(\vec{a}), \Gamma :: \begin{cases} \quad\textbf{if } (\mathrm{pre}_1(\vec{a})) & \Gamma_1 \\ \textbf{else if } (\mathrm{pre}_2(\vec{a})) & \Gamma_2 \\ \ldots & \\ \textbf{else if } (\mathrm{pre}_n(\vec{a})) & \Gamma_n \\ \textbf{else} & \\ \quad \textbf{throw new } \text{Exception(“No solution”)} & \end{cases} \right),$$

where:

- $(\mathrm{pre}_i(\vec{a}), \Gamma_i) = [\![\vec{x}, \vec{a}, D_i, ()]\!]$

Synthesis for Linear Integer Arithmetic

Pre-processing

Elimination of negations and divisibility constraints:

- $[\![\vec{x}, \vec{a}, (c|t) \wedge F, \Gamma]\!] =$
  $[\![\vec{x} :: k, \vec{a}, t = c * k \wedge F, \Gamma]\!]$
- $[\![\vec{x}, \vec{a}, \neg(c|t) \wedge F, \Gamma]\!] =$
  $[\![\vec{x} :: (k, r), \vec{a}, t = c * k + r \wedge 0 < r < c \wedge F, \Gamma]\!]$
- negations from equalities and inequalities are eliminated using:
  - $\neg(t_1 \geq t_2)$ is equivalent to $t_2 \geq t_1 + 1$
  - $\neg(t_1 = t_2)$ is equivalent to $(t_1 \geq t_2 + 1) \vee (t_2 \geq t_1 + 1)$

From now on: formula F is a conjunction of equalites and inequlities!

Solving Equality Constraints

$$[\![\vec{x}, \vec{a}, E \wedge F, \Gamma]\!] = (\mathrm{pre}_1(\vec{a}) \wedge \mathrm{pre}_2(\vec{a}),$$

$$\Gamma :: \Gamma_1 :: \left\{ \begin{array}{l} \textbf{val}\ x_1 = w_1 + \lambda_1 * s_{11} + \ldots + \lambda_{n-1} * s_{1(n-1)} \\ \quad \ldots \\ \textbf{val}\ x_n = w_n + \lambda_1 * s_{n1} + \ldots + \lambda_{n-1} * s_{n(n-1)} \end{array} \right\}),$$

where

- $(\mathrm{pre}_1(\vec{a}), \vec{\lambda}, \{\vec{s}_1, \ldots, \vec{s}_{n-1}\}, \vec{w}) = \mathrm{eqSyn}(\vec{x}, E)$
- $F' = F[\vec{x} \mapsto \vec{w} + \lambda_1 * \vec{s}_1 + \ldots + \lambda_{n-1} * \vec{s}_{n-1}]$
- $(\mathrm{pre}_2(\vec{a}), \Gamma_1) = [\![\vec{\lambda}, \vec{a}, F', ()]\!]$

Solving Equality Constraints - Example

$[\![(x, y, z), (a, b), 2a - b + 3x + 4y + 8z = 0 \wedge 5x + 4z \leq y - b, \Gamma]\!] = ???$

Solving Equality Constraints - Example

$[\![(x, y, z), (a, b), 2a - b + 3x + 4y + 8z = 0 \wedge 5x + 4z \leq y - b, \Gamma]\!] =???$

$$\text{eqSyn}((x, y, z), 2a - b + 3x + 4y + 8z = 0) =$$

$$(1|2a - b, (\lambda_1, \lambda_2), \left\{ \begin{pmatrix} 4 \\ -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix} \right\}, \begin{pmatrix} 2a - b \\ b - 2a \\ 0 \end{pmatrix})$$

Solving Equality Constraints - Example

$\llbracket (x, y, z), (a, b), 2a - b + 3x + 4y + 8z = 0 \wedge 5x + 4z \leq y - b, \Gamma \rrbracket = ???$

$\mathrm{eqSyn}((x, y, z), 2a - b + 3x + 4y + 8z = 0) =$

$$(1 | 2a - b, (\lambda_1, \lambda_2), \left\{ \begin{pmatrix} 4 \\ -3 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix} \right\}, \begin{pmatrix} 2a - b \\ b - 2a \\ 0 \end{pmatrix})$$

$F' = 7a - 3b + 13\lambda_1 \leq 4\lambda_2$

Solving Equality Constraints - Example

$[\![(x,y,z),(a,b),2a-b+3x+4y+8z=0 \wedge 5x+4z \leq y-b,\Gamma]\!] = ???$

$\text{eqSyn}((x,y,z),2a-b+3x+4y+8z=0) =$

$$\left(1 | 2a-b,(\lambda_1,\lambda_2),\left\{\begin{pmatrix}4\\-3\\0\end{pmatrix},\begin{pmatrix}0\\2\\-1\end{pmatrix}\right\},\begin{pmatrix}2a-b\\b-2a\\0\end{pmatrix}\right)$$

$F' = 7a - 3b + 13\lambda_1 \leq 4\lambda_2$

$(\text{pre}_1(a,b),\Gamma_1) = [\![(\lambda_1,\lambda_2),(a,b),7a-3b+13\lambda_1 \leq 4\lambda_2,()]\!]$

Solving Equality Constraints - Example

$$[\![(x,y,z),(a,b),2a-b+3x+4y+8z=0 \wedge 5x+4z \leq y-b,\Gamma]\!] =$$

$$(\text{pre}_1(a,b),\Gamma :: \Gamma_1 :: \left\{\begin{array}{l} \textbf{val}\ x = 2*a-b+4*\lambda_1 \\ \textbf{val}\ y = b-2*a-b-3*\lambda_1+2*\lambda_2 \\ \textbf{val}\ z = -\lambda_2 \end{array}\right\})$$

The eqSyn Algorithm - Summary

Equality: $\sum_{i=1}^{m} \beta_i b_i + \sum_{j=1}^{n} \gamma_j y_j = 0$
Let $T = \sum_{i=1}^{m} \beta_i b_i$

1. obtain a linear set representation of the set
   $S_H = \{\vec{y} \mid \sum_{j=1}^{n} \gamma_j y_j = 0\}$
   i.e. compute $\vec{s}_1, \ldots, \vec{s}_{n-1}$ such that

$$S_H = \{\vec{y} \mid \exists \lambda_1, \ldots, \lambda_{n-1} \in \mathbb{Z}.\ \vec{y} = \sum_{i=1}^{n-1} \lambda_i \vec{s}_i\}$$

2. find one particular solution such that $T + \sum_{j=1}^{n} \gamma_j w_j = 0$
3. return as the solution $\vec{w} + \sum_{i=1}^{n-1} \lambda_i \vec{s}_i$

The eqSyn Algorithm

Equality: $\sum_{i=1}^{m} \beta_i b_i + \sum_{j=1}^{n} \gamma_j y_j = 0$
Let $T = \sum_{i=1}^{m} \beta_i b_i$
Let $d = \text{gcd}(\beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_n)$

- if $d > 1$: $\text{eqSyn}(\vec{y}, \sum_{i=1}^{m} \beta_i b_i + \sum_{j=1}^{n} \gamma_j y_j = 0) = \text{eqSyn}(\vec{y}, \sum_{i=1}^{m} \beta_i / d * b_i + \sum_{j=1}^{n} \gamma_j / d * y_j = 0)$

The eqSyn Algorithm

Equality: $\sum_{i=1}^{m} \beta_i b_i + \sum_{j=1}^{n} \gamma_j y_j = 0$
Let $T = \sum_{i=1}^{m} \beta_i b_i$
Let $d = \mathrm{gcd}(\beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_n)$

- if $d > 1$: $\mathrm{eqSyn}(\vec{y}, \sum_{i=1}^{m} \beta_i b_i + \sum_{j=1}^{n} \gamma_j y_j = 0) = \mathrm{eqSyn}(\vec{y}, \sum_{i=1}^{m} \beta_i / d * b_i + \sum_{j=1}^{n} \gamma_j / d * y_j = 0)$
- else: $\mathrm{eqSyn}(y_1, T + \gamma_1 y_1 = 0) = ((\gamma_1 | T), 0, \emptyset, \ -T / \gamma_1)$

The eqSyn Algorithm

Equality: $\sum_{i=1}^{m} \beta_i b_i + \sum_{j=1}^{n} \gamma_j y_j = 0$
Let $T = \sum_{i=1}^{m} \beta_i b_i$
Let $d = \text{gcd}(\beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_n)$

- if $d > 1$: $\text{eqSyn}(\vec{y}, \sum_{i=1}^{m} \beta_i b_i + \sum_{j=1}^{n} \gamma_j y_j = 0) = \text{eqSyn}(\vec{y}, \sum_{i=1}^{m} \beta_i / d * b_i + \sum_{j=1}^{n} \gamma_j / d * y_j = 0)$
- else: $\text{eqSyn}(y_1, T + \gamma_1 y_1 = 0) = ((\gamma_1 | T), 0, \emptyset, -T / \gamma_1)$
- $\text{eqSyn}(y_1, \ldots, y_n, T + \Sigma_{j=1}^{n} \gamma_j y_j = 0) = (\text{gcd}(\gamma_1, \ldots, \gamma_n) | T, \vec{\lambda}, S, \vec{w})$,
  where
  - vector $\vec{\lambda}$ = $n - 1$ fresh variables
  - $S = \text{linearSet}(\gamma_1, \ldots, \gamma_n)$
  - $\vec{w} = \text{particularSol}(T, \gamma_1, \ldots, \gamma_n)$

Computing a Linear Set for a Homogeneous Equation

$\text{linearSet}(\gamma_1, \ldots, \gamma_n) = (\vec{s}_1, \ldots, \vec{s}_{n-1})$

Theorem

$$\{\vec{y} \mid \gamma_1 y_1 + \ldots + \gamma_n y_n = 0\} = \{\lambda_1 \vec{s}_1 + \ldots + \lambda_n \vec{s}_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{Z}\}$$

$$= \left\{ \lambda_1 \begin{pmatrix} K_{11} \\ \vdots \\ K_{n1} \end{pmatrix} + \ldots + \lambda_{n-1} \begin{pmatrix} K_{1(n-1)} \\ \vdots \\ K_{n(n-1)} \end{pmatrix} \middle| \lambda_i \in \mathbb{Z} \right\}$$

where the integers K_{ij} are computed as follows:

- *if $i < j$, $K_{ij} = 0$ (the matrix K is lower triangular)*
- $K_{jj} = \dfrac{\gcd((\gamma_k)_{k \geq j+1})}{\gcd((\gamma_k)_{k \geq j})}$
- *the other K_{ij} are computed as follows* ...

Computing a Linear Set for a Homogeneous Equation

$\text{linearSet}(\gamma_1, \ldots, \gamma_n) = (\vec{s}_1, \ldots, \vec{s}_{n-1})$
... the integers K_{ij} are computed as follows:

- for each index j, $1 \leq j \leq n-1$, we compute K_{ij} as follows: Consider

$$\gamma_j K_{jj} + \sum_{i=j+1}^{n} \gamma_i u_{ij} = 0$$

  and find any solution.

$(K_{(j+1)j}, \ldots, K_{nj}) = \text{particularSol}(-\gamma_j K_{jj}, \gamma_{j+1}, \ldots, \gamma_n)$

Finding a Particular Solution of an Equation

General form:
$\text{particularSol}(T, \gamma_1, \ldots, \gamma_n) = (w_1, \ldots, w_n)$,
where

- $(w_1, w_N) = \text{particularSol}(T, \gamma_1, \text{gcd}((\gamma_k)_{k \geq 2}))$
- $(w_2, \ldots, w_n) = \text{particularSol}(\text{gcd}((\gamma_k)_{k \geq 2}) w_N, \gamma_2, \ldots, \gamma_n)$

Finding a Particular Solution of an Equation

For two variables:

- based on Extended Euclidean Algorithm: given the integers a_1 and a_2, the EEA finds two integers v_1 and v_2 such that $a_1 v_1 + a_2 v_2 = \gcd(a_1, a_2)$. It also computes the value of $\gcd(a_1, a_2)$

Finding a Particular Solution of an Equation

For two variables:

- based on Extended Euclidean Algorithm: given the integers a_1 and a_2, the EEA finds two integers v_1 and v_2 such that $a_1 v_1 + a_2 v_2 = \gcd(a_1, a_2)$. It also computes the value of $\gcd(a_1, a_2)$
- $\text{particularSol}(T, \gamma_1, \gamma_2) = (v_1 * r, v2 * r)$, where
  - $(d, v_1, v_2) = \text{ExtendedEuclid}(\gamma_1, \gamma_2)$
  - $r = T/d$

Solving Inequality Constraints

- $A_i \leq \alpha_i x$
  $\beta_j x \leq B_j$

Solving Inequality Constraints

- $A_i \leq \alpha_i x$
  $\beta_j x \leq B_j$
- $a = \max_i \lceil A_i / \alpha_i \rceil$
  $b = \min_j \lfloor B_j / \beta_j \rfloor$
  if b is defined, return $x = b$ else return $x = a$

Solving Inequality Constraints

- $A_i \leq \alpha_i x$
  $\beta_j x \leq B_j$
- $a = \max_i \lceil A_i / \alpha_i \rceil$
  $b = \min_j \lfloor B_j / \beta_j \rfloor$
  if b is defined, return $x = b$ else return $x = a$
-
$$\bigwedge_{i,j} \lceil A_i / \alpha_i \rceil \leq \lfloor B_j / \beta_j \rfloor$$

Solving Inequality Constraints - Example

Example

- consider the formula $2y - b \leq 3x + a \wedge 2x - a \leq 4y + b$

Solving Inequality Constraints - Example

Example

- consider the formula $2y - b \leq 3x + a \wedge 2x - a \leq 4y + b$
- resulting formula: $\lceil (2y - b - a)/3 \rceil \leq \lfloor (4y + a + b)/2 \rfloor$

Solving Inequality Constraints - Example

Example

- consider the formula $2y - b \leq 3x + a \wedge 2x - a \leq 4y + b$
- resulting formula: $\lceil (2y - b - a)/3 \rceil \leq \lfloor (4y + a + b)/2 \rfloor$
- $\Leftrightarrow \lceil (2y - b - a) * 2/6 \rceil \leq \lfloor (4y + a + b) * 3/6 \rfloor$

Solving Inequality Constraints - Example

Example

- consider the formula $2y - b \leq 3x + a \wedge 2x - a \leq 4y + b$
- resulting formula: $\lceil (2y - b - a)/3 \rceil \leq \lfloor (4y + a + b)/2 \rfloor$
- $\Leftrightarrow \lceil (2y - b - a) * 2/6 \rceil \leq \lfloor (4y + a + b) * 3/6 \rfloor$
- $\Leftrightarrow (4y - 2b - 2a)/6 \leq [(12y + 3a + 3b) - (12y + 3a + 3b) \bmod 6]/6$

Solving Inequality Constraints - Example

Example

- consider the formula $2y - b \leq 3x + a \wedge 2x - a \leq 4y + b$
- resulting formula: $\lceil (2y - b - a)/3 \rceil \leq \lfloor (4y + a + b)/2 \rfloor$
- $\Leftrightarrow \lceil (2y - b - a) * 2/6 \rceil \leq \lfloor (4y + a + b) * 3/6 \rfloor$
- $\Leftrightarrow (4y - 2b - 2a)/6 \leq [(12y + 3a + 3b) - (12y + 3a + 3b) \mod 6]/6$
- $\Leftrightarrow (12y + 3a + 3b) \mod 6 \leq 8y + 5a + 5b$

Solving Inequality Constraints - Example

Example

- consider the formula $2y - b \leq 3x + a \wedge 2x - a \leq 4y + b$
- resulting formula: $\lceil (2y - b - a)/3 \rceil \leq \lfloor (4y + a + b)/2 \rfloor$
- $\Leftrightarrow \lceil (2y - b - a) * 2/6 \rceil \leq \lfloor (4y + a + b) * 3/6 \rfloor$
- $\Leftrightarrow (4y - 2b - 2a)/6 \leq [(12y + 3a + 3b) - (12y + 3a + 3b) \mod 6]/6$
- $\Leftrightarrow (12y + 3a + 3b) \mod 6 \leq 8y + 5a + 5b$
- $\Leftrightarrow 12y + 3a + 3b = 6 * l + k \wedge k \leq 8y + 5a + 5b$

Solving Inequality Constraints - Example

Example

- $12y + 3a + 3b = 6 * l + k \wedge k \leq 8y + 5a + 5b$

Solving Inequality Constraints - Example

Example

- $12y + 3a + 3b = 6 * l + k \wedge k \leq 8y + 5a + 5b$
- $\text{eqSyn}(l, y), 12y - 6l + 3a + 3b - k = 0) =$
  $$(6|3a + 3b - k, \lambda, \left\{ \left(\begin{array}{c} 2 \\ 1 \end{array} \right) \right\}, \left(\begin{array}{c} (3a + 3b - k)/6 \\ 0 \end{array} \right))$$

Solving Inequality Constraints - Example

Example

- $12y + 3a + 3b = 6 * l + k \wedge k \leq 8y + 5a + 5b$
- eqSyn$(l, y), 12y - 6l + 3a + 3b - k = 0) =$
  $$(6|3a + 3b - k, \lambda, \left\{ \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right\}, \begin{pmatrix} (3a + 3b - k)/6 \\ 0 \end{pmatrix})$$
- this way, formula becomes: $k - 5a - 5b \leq 8\lambda$

Solving Inequality Constraints - Example

Example

- $12y + 3a + 3b = 6 * l + k \wedge k \leq 8y + 5a + 5b$
- eqSyn$(l, y), 12y - 6l + 3a + 3b - k = 0) =$
  $$(6|3a + 3b - k, \lambda, \left\{\begin{pmatrix} 2 \\ 1 \end{pmatrix}\right\}, \begin{pmatrix} (3a + 3b - k)/6 \\ 0 \end{pmatrix})$$
- this way, formula becomes: $k - 5a - 5b \leq 8\lambda$
- solution: $\lambda = \lceil (k - 5a - 5b)/8 \rceil$

Solving Inequality Constraints - Example

```
val kFound = false
for k = 0 to 5 do {
  val v1 = 3 * a + 3 * b − k
  if (v1 mod 6 == 0) {
    val lambda = ((k − 5 * a − 5 * b)/8).ceiling
    val l = (v1 / 6) + 2 * lambda
    val y = lambda
    val kFound = true
    break } }
if (kFound)
  val x = ((4 * y + a + b)/2).floor
else
  throw new Exception("No solution exists")
```

Conclusions

- Complete Functional Synthesis - input and output parameters
- produces a code which computes output variables as a function of input variables
- outputs preconditions which have to be fulfilled for a problem to have a solution
- efficient implementation as a Scala-plugin: `http://lara.epfl.ch/dokuwiki/comfusy`