Automation of Termination: Abstracting Calling Contexts through Matrix-Weighted Graphs

Andréia Borges Avelar$^1$ & Mauricio Ayala-Rincón$^1$ & César A. Muñoz$^2$

$^1$Instituto de Ciências Exatas - UnB
$^2$NASA Langley Research Center

IX Seminário Informal(, mas Formal!) do Grupo de Teoria da Computação

November 29, 2013
Termination analysis is a fundamental topic in computer science. While classical results state the undecidability of various termination problems, automated methods have successfully been developed that prove termination or non-termination in practical cases. Research in termination analysis offers many challenges both in theory (mathematical logic, proof theory) and practice (software development, formal methods).
Recursive Definitions and Termination in PVS

- Recursive functions must be well defined.
- Each recursive definition have an associated measure provided by the user.
- This measure must decrease at every recursive call.
- The type checking operation generates Type Correctness Conditions $TCC$'s, related with termination of the recursively defined function.
Example - The Ackermann function

PVS specification

\[
\text{ack}(m:\text{nat}, n:\text{nat}) : \text{RECURSIVE nat} = \\
\quad \text{IF } m = 0 \text{ THEN } n+1 \\
\quad \quad \text{ELSIF } n = 0 \text{ THEN } 1:\text{ack}(m-1, 1) \\
\quad \quad \text{ELSE } 2:\text{ack}(m-1, 3:\text{ack}(m, n-1)) \\
\quad \text{ENDIF} \\
\text{MEASURE } \text{lex2}(m, n)
\]

For each recursive call there is a termination TCC:

1: \text{ack}(m-1, 1)

\[
\text{ack\_TCC2: OBLIGATION} \\
\text{FORALL } (m, n: \text{nat}): n = 0 \text{ AND NOT } m = 0 \\
\text{IMPLIES } \text{lex2}(m - 1, 1) < \text{lex2}(m, n);
\]
Size Change Principle (SCP)


- It is used to prove termination of functional programs over wellfounded data.
- It explores the digraph of all admissible path in a execution of the program.
- It is performed in two steps:
  - first: extract a *safe set of size change graphs*.
  - second: apply the following criterion:

  If every infinite computation would give rise to an infinitely decreasing value sequence, then no infinite computation is possible.
Example

Ackermann function definition

\[ ack(m, n) := \begin{cases} 
    n + 1 & \text{if } m = 0 \\
    1 : ack(m - 1, 1) & \text{if } m > 0 \land n = 0 \\
    2 : ack(m - 1, 3 : ack(m, n - 1)) & \text{if } m > 0 \land n > 0 
\end{cases} \]
Example

### Ackermann function definition

\[
ack(m, n) := \begin{cases} 
  n + 1 & \text{if } m = 0 \\
  1 : \ack(m - 1, 1) & \text{if } m > 0 \land n = 0 \\
  2 : \ack(m - 1, 3 : \ack(m, n - 1)) & \text{if } m > 0 \land n > 0
\end{cases}
\]

### Size Change Graphs for Ackermann

- \(G_1 : \ack \rightarrow \ack\)
- \(G_2 : \ack \rightarrow \ack\)
- \(G_3 : \ack \rightarrow \ack\)
Calling Context Graphs (CCG)


- It abstracts a recursive definition behavior by a digraph.
- Absorbs all the information of a SCG in a single context.
- Apply measures over well-founded domains showing that for every possible infinite sequence of contexts there is a corresponding sequence of measures that is infinitely decreasing.
Example

\[
\text{ack}(m, n) := \begin{cases} 
  n + 1 & \text{if } m = 0 \\
  1 : \text{ack}(m - 1, 1) & \text{if } m > 0 \land n = 0 \\
  2 : \text{ack}(m - 1, 3 : \text{ack}(m, n - 1)) & \text{if } m > 0 \land n > 0
\end{cases}
\]

\[
\langle \text{ack}(m, n), \{m, n \in \mathbb{N}; m, n > 0\}, \text{ack}(m - 1, \text{ack}(m, n - 1)) \rangle
\]

\[
\langle \text{ack}(m, n), \{m, n \in \mathbb{N}; m > 0; n = 0\}, \text{ack}(m - 1, 1) \rangle
\]

\[
\langle \text{ack}(m, n), \{m, n \in \mathbb{N}; m, n > 0\}, \text{ack}(m, n - 1) \rangle
\]
Some of the work done on digraphs theory

- Specification of basic definitions such as:
  - equivalent pre-walks, which are general sequences of vertices;
  - cycles;
  - equivalent circuits, etc.

- Adjust some existent definitions;
Some of the work done on digraphs theory

- Formalization of crucial properties, such as:

  For all digraph $G$ and circuit $c$ in $G$: $c$ is a cycle or $c$ is equivalent to a circuit of the form $c_1 \circ c_2$, where $c_1$ is a cycle and $c_2$ is a circuit.
Some of the work done on digraphs theory

- Formalization of crucial properties, such as:

  For all digraph $G$ and circuit $c$ in $G$: $c$ is a cycle or $c$ is equivalent to a circuit of the form $c_1 \circ c_2$, where $c_1$ is a cycle and $c_2$ is a circuit.

Graphically:
The *theory* of weighted digraphs

- Introduction of important definitions, such as:
  - Digraphs with weight, which is a function defined on edges;
  - Specification of functions to treat weight of walks.

Formalization of fundamental properties, like the decomposition of the weight of a walk:

Let \( w = w_0 \ldots w_n \) be a walk of length \( n+1 \) on a digraph \( G \). For all \( j < n+1 \),

\[
\text{wgt}(w) = \text{wgt}(w(0,j)) + \text{wgt}(w(j,n))
\]
The theory weighted digraphs

- Introduction of important definitions, such as:
  - Digraphs with weight, which is a function defined on edges;
  - Specification of functions to treat weight of walks.

- Formalization of fundamental properties, like the decomposition of the weight of a walk:

Let $w = w_0 \ldots w_n$ a walk of length $n + 1$ on a digraph $G$. For all $j < n + 1$, $wgt(w) = wgt(w^{0,j}) + wgt(w^{j,n})$

Graphically:
The *theory measures*

- Specification of an algebra of matrices, with new definitions of binary operations, to treat weights of edges and walks in weighted-digraphs.
The *theory* measures

- Specification of an algebra of matrices, with new definitions of binary operations, to treat weights of edges and walks in weighted-digraphs.

- Matrices with entries \{-1, 0, 1\}

\[ U \xrightarrow{\mu_i \geqslant \mu_j} V \]
The theory measures

- Specification of an algebra of matrices, with new definitions of binary operations, to treat weights of edges and walks in weighted-digraphs.

- Matrices with entries \([-1, 0, 1]\)

\[
\begin{align*}
\mu_i &\geq \mu_j \\
U \quad \cdots \cdots \quad \rightarrow \quad V
\end{align*}
\]

\[
\begin{align*}
u \quad \_1 \quad \rightarrow \quad v &\quad - \text{The measure decreases}; \\
u \quad \_0 \quad \rightarrow \quad v &\quad - \text{The measure remains less than or equal}; \\
u \quad \_-1 \quad \rightarrow \quad v &\quad - \text{I don’t know}.
\end{align*}
\]
The *theory measures*

- Let $w = w_0 \ldots w_n$ be a walk on a digraph $G$.
  - $wgt(w) = wgt(w_0, w_1) + wgt(w_1, w_2) + \ldots + wgt(w_{n-1}, w_n)$
The theory measures

Let $w = w_0 \ldots w_n$ be a walk on a digraph $G$.

$\triangleright wgt(w) = wgt(w_0, w_1) + wgt(w_1, w_2) + \ldots + wgt(w_{n-1}, w_n)$

\[ w_0 \rightarrow^{1\lor 0} w_i \rightarrow^1 w_{i+1} \rightarrow^{1\lor 0} w_n \quad \Rightarrow \quad w_0 \rightarrow^1 w_n \]
The theory measures

Let \( w = w_0 \ldots w_n \) be a walk on a digraph \( G \).

\[ \text{wgt}(w) = \text{wgt}(w_0, w_1) + \text{wgt}(w_1, w_2) + \ldots + \text{wgt}(w_{n-1}, w_n) \]

\[ \begin{array}{c}
  w_0 \rightarrow_{1 \lor 0} w_i \rightarrow_1 w_{i+1} \rightarrow_{1 \lor 0} w_n \Rightarrow w_0 \rightarrow_1 w_n \\
  w_0 \rightarrow_0 w_i \rightarrow_0 w_{i+1} \rightarrow_0 w_n \Rightarrow w_0 \rightarrow_0 w_n
\end{array} \]
The **theory measures**

- Let \( w = w_0 \ldots w_n \) be a walk on a digraph \( G \).

\[ \text{wgt}(w) = \text{wgt}(w_0, w_1) + \text{wgt}(w_1, w_2) + \ldots + \text{wgt}(w_{n-1}, w_n) \]

\[ \begin{array}{c}
\text{w_0} \xrightarrow{1 \lor 0} \text{w_i} \xrightarrow{1} \text{w_{i+1}} \xrightarrow{1 \lor 0} \text{w_n} \Rightarrow \text{w_0} \xrightarrow{1} \text{w_n} \\
\text{w_0} \xrightarrow{0} \text{w_i} \xrightarrow{0} \text{w_{i+1}} \xrightarrow{0} \text{w_n} \Rightarrow \text{w_0} \xrightarrow{0} \text{w_n} \\
\text{w_0} \xrightarrow{1 \lor 0} \text{w_i} \xrightarrow{-1} \text{w_{i+1}} \xrightarrow{1 \lor 0} \text{w_n} \Rightarrow \text{w_0} \xrightarrow{-1} \text{w_n}
\end{array} \]
The *theory* measures

- Let \( w = w_0 \ldots w_n \) be a walk on a digraph \( G \).

\[
\text{\textbf{\footnotesize{wgt}}}(w) = \text{\textbf{\footnotesize{wgt}}}(w_0, w_1) + \text{\textbf{\footnotesize{wgt}}}(w_1, w_2) + \ldots + \text{\textbf{\footnotesize{wgt}}}(w_{n-1}, w_n)
\]

\[
\begin{align*}
\text{w}_0 & \xrightarrow{1 \lor 0} w_i \xrightarrow{1} w_{i+1} \xrightarrow{1 \lor 0} w_n \Rightarrow \text{w}_0 \xrightarrow{1} w_n \\
\text{w}_0 & \xrightarrow{0} \text{w}_i \xrightarrow{0} w_{i+1} \xrightarrow{0} w_n \Rightarrow \text{w}_0 \xrightarrow{0} w_n \\
\text{w}_0 & \xrightarrow{1 \lor 0} w_i \xrightarrow{-1} w_{i+1} \xrightarrow{1 \lor 0} w_n \Rightarrow \text{w}_0 \xrightarrow{-1} w_n \\
\end{align*}
\]

\[
+: \{-1, 0, 1\} \times \{-1, 0, 1\} \rightarrow \{-1, 0, 1\}
\]

\[
+(x, y) := \begin{cases} 
-1 & \text{if } x = -1 \lor y = -1 \\
1 & \text{if } (x \neq -1 \lor y \neq -1) \land (x = 1 \lor y = 1) \\
0 & \text{if } x = 0 \land y = 0
\end{cases}
\]
The *theory* measures

- But, in general, we are dealing with more than only one measure...
The *theory* measures

- But, in general, we are dealing with more than only one measure...
- ...then matrices come into play. This way, if we have \( n \) measures, we can encode in a \( n \times n \) matrix all the \( n \times n \) possible combinations between the measures.
The *theory* measures

- But, in general, we are dealing with more than only one measure...
- ...then matrices come into play. This way, if we have $n$ measures, we can encode in a $n \times n$ matrix all the $n \times n$ possible combinations between the measures.

\[
\begin{pmatrix}
\mu_1(\vec{x}) \geq \mu_1(\vec{y}) & \cdots & \mu_1(\vec{x}) \geq \mu_n(\vec{y}) \\
\mu_2(\vec{x}) \geq \mu_1(\vec{y}) & \cdots & \mu_2(\vec{x}) \geq \mu_n(\vec{y}) \\
\vdots & & \vdots \\
\mu_n(\vec{x}) \geq \mu_1(\vec{y}) & \cdots & \mu_n(\vec{x}) \geq \mu_n(\vec{y})
\end{pmatrix}
\]

- $\vec{x} = \{x_1, \ldots, x_n\}$ are the formal parameters at the call $v_i$ and $\vec{y} = \{y_1, \ldots, y_n\}$ are the actual parameters in $v_i$. 
The *theory measures*

\[
\begin{pmatrix}
  a_{11} & a_{12} & \cdots & a_{1n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{i1} & a_{i2} & \cdots & a_{in} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{n1} & a_{n2} & \cdots & a_{nn}
\end{pmatrix}
\rightarrow
\begin{pmatrix}
  b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\
  b_{21} & \cdots & b_{2j} & \cdots & b_{2n} \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  b_{n1} & \cdots & b_{nj} & \cdots & b_{nn}
\end{pmatrix}
\]

\[c_{ij} = \max \{ a_{ik} + b_{kj} | 1 \leq k \leq n \} \]
The theory measures

\[
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{i1} & a_{i2} & \cdots & a_{in} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{pmatrix} \rightarrow
\begin{pmatrix}
b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\
b_{21} & \cdots & b_{2j} & \cdots & b_{2n} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
b_{i1} & \cdots & b_{ij} & \cdots & b_{in} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
b_{n1} & \cdots & b_{nj} & \cdots & b_{nn}
\end{pmatrix}
\rightarrow
\begin{pmatrix}
c_{11} & \cdots & c_{1j} & \cdots & c_{1n} \\
c_{21} & \cdots & c_{2j} & \cdots & c_{2n} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
c_{i1} & \cdots & c_{ij} & \cdots & c_{in} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
c_{n1} & \cdots & c_{nj} & \cdots & c_{nn}
\end{pmatrix}
\]

\[c_{ij} = \max\{a_{ik} + b_{kj} \mid 1 \leq k \leq n\}\]
The *theory* measures

The notion of *termination*

\[
\begin{pmatrix}
1 & \ast \\
\ast & \ast
\end{pmatrix} \rightarrow \begin{pmatrix}
\ast & 0 \\
\ast & \ast
\end{pmatrix}
\]

\[
\begin{pmatrix}
\ast & \ast \\
0 & \ast
\end{pmatrix} \rightarrow \begin{pmatrix}
\ast & \ast \\
\ast & 1
\end{pmatrix}
\]
The *theory* measures

- The notion of **termination**

\[
\begin{align*}
(1 & \ast) & \quad (\ast & 0) \\
(* & *) & \quad (\ast & \ast)
\end{align*}
\]

\[
\begin{align*}
\mu_1 > \mu_1 & \land \mu_1 \geq \mu_2 \land \mu_2 > \mu_2 \land \mu_2 \geq \mu_1
\end{align*}
\]
The theory measures

- The notion of **termination**

\[
\begin{align*}
(1 & \ast) & \rightarrow & (\ast & 0) \\
(\ast & \ast) & & (\ast & 1)
\end{align*}
\]

\[
\begin{align*}
\mu_1 > \mu_1 & \land \mu_1 \geq \mu_2 \land \mu_2 > \mu_2 & \land \mu_2 \geq \mu_1 \\
\mu_1 > \mu_1 & \geq \mu_2 \geq \mu_2 \geq \mu_1
\end{align*}
\]
The theory measures
The *theory* measures

\[
\begin{pmatrix}
1 & \ast \\
\ast & \ast
\end{pmatrix}
\quad \leftrightarrow 
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\rightarrow
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\quad \Rightarrow
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\rightarrow
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & \ast \\
\ast & \ast
\end{pmatrix}
\quad \leftrightarrow 
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\rightarrow
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\quad \Rightarrow
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\rightarrow
\begin{pmatrix}
\ast & \\
\ast & \ast
\end{pmatrix}
\]

**termination**

\[
\begin{pmatrix}
\ast & \ast & \cdots & \ast \\
\ast & \ast & \cdots & \ast \\
\vdots & \vdots & \ddots & \vdots \\
\ast & \ast & \cdots & \ast
\end{pmatrix}
\]

**positive measure-matrix**
The *theory matrix* \(_{wdg}\)

- Two criteria to verify termination:
The *theory matrix* \textit{wdg}

- Two criteria to verify termination:
  - The first one is based on lexicographic order, where there is a controlling measure \( k \).

\[
\forall e \in G : M^e(k, k) \geq 0
\]
\[
\forall c \in G, \text{ such that } c \text{ is a double cycle : } M^c(k, k) = 1
\]
The theory matrix\textsubscript{wdg}

- Two criteria to verify termination:
  - The first one is based on lexicographic order, where there is a controlling measure \(k\).

\[
\forall e \in G : M^e(k, k) \geq 0
\]
\[
\forall c \in G, \text{ such that } c \text{ is a double cycle : } M^c(k, k) = 1
\]

- The second is based on a fixed labeling of vertexes that gives rise to a combination of measures that must be “limiting”. Each vertex \(v_i\) has a label \(k_i\).

\[
\forall e = (v_i, v_j) \in G : M^e(k_i, k_j) \geq 0
\]
\[
\forall c \in G, \text{ such that } c = v_\alpha \ldots v_\alpha \text{ is a cycle : } M^c(k_\alpha, k_\alpha) = 1
\]
Example: an application of the first criterion

\[ \text{gcd} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \]

\[
gcd(m, n) := \begin{cases} 
  m + n & \text{if } m = 0 \lor n = 0 \\
  1 : \gcd(m - n, n) & \text{if } m \geq n \land m \neq 0 \land n \neq 0 \\
  2 : \gcd(n, m) & \text{if } n > m \land m \neq 0 \land n \neq 0 
\end{cases}
\]

- measures: \( \mu_1(m, n) = m \) and \( \mu_2(m, n) = n \)
A non-Terminating Example

\[
f(\{n \mid n \in \mathbb{N} \land n \leq 100\}) := \begin{cases} 
  1 : f(n - 1) & \text{if } n \geq 50 \\
  2 : f(n + 1) & \text{if } n < 50 
\end{cases}
\]

Measures: \( \mu_1(n) := |n| \) and \( \mu_2(n) := |100 - n| \)
Example: an application of the first criterion

\[ \text{gcd} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \]

\[
gcd(m, n) := \begin{cases} 
  m + n & \text{if } m = 0 \lor n = 0 \\
  1 : \gcd(m - n, n) & \text{if } m \geq n \land m \neq 0 \land n \neq 0 \\
  2 : \gcd(n, m) & \text{if } n > m \land m \neq 0 \land n \neq 0
\end{cases}
\]

- The limiting measure: \( \mu_2(m, n) = n \)
Example: an application of the second criterion

\[
p : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}
\]

\[
p(m, n, r)) := \begin{cases} 
1 : p(m, r - 1, n) & \text{if } r > 0 \\
2 : p(r, n - 1, m) & \text{if } r = 0 \land n > 0 \\
m & \text{if } r = 0 \land n = 0
\end{cases}
\]

Measures:

\[
\mu_1(m, n, r) = m + n + r
\]
\[
\mu_2(m, n, r) = m + r
\]
\[
\mu_3(m, n, r) = m + n
\]
Example: an application of the second criterion

\[ p : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \]

\[ p(m, n, r)) := \begin{cases} 
1 : p(m, r - 1, n) & \text{if } r > 0 \\
2 : p(r, n - 1, m) & \text{if } r = 0 \land n > 0 \\
m & \text{if } r = 0 \land n = 0 
\end{cases} \]

\[
\begin{pmatrix}
1 & 1 & 1 \\
-1 & -1 & 1 \\
-1 & 0 & -1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 & 1 \\
-1 & 0 & -1 \\
-1 & -1 & -1
\end{pmatrix}
\]
Example: an application of the second criterion

\[ p : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \]

\[ p(m, n, r) := \begin{cases} 
1 : p(m, r - 1, n) & \text{if } r > 0 \\
2 : p(r, n - 1, m) & \text{if } r = 0 \land n > 0 \\
m & \text{if } r = 0 \land n = 0 
\end{cases} \]
The *theory* CCG

- Specification of CCG as a digraph with a family of measures.

- Definition of combination of measures for a walk, which can be:
  - A greater than or equal measure-combination
  - Or a greater than measure-combination

- Formalization of properties of such measure-combinations.
The theory \texttt{ccg\_to\_mwg}

\[ dg(\text{CCG}) = dg(\text{MWG}) \]

\[ \text{measure\_combination} \iff \text{measure\_matrix} \]

Let \( G \) be a CCG, \( c \) a circuit on \( G \) and \( G' \) the MWG corresponding to \( G \). Then, \( M_c G' \) is not positive for all \( m_c \): measure\_combination \( G(c), m_c \) is not greater than...
The *theory ccg_to_mwg*

Let $G$ be a CCG, $c$ a circuit on $G$ and $G'$ the MWG corresponding to $G$. Then,

$M_{G'}^c$, is not positive

$\forall mc : measure\_combination_G(c)$, $mc$ is not greater than
Summarizing

**ccg**
digraph with a family of measures

**ccg_to_mwg**
transformation from CCG to MWG

**digraphs**
cycles, circuits, weighted digraphs, etc.

**measures**
algebra of matrices

**matrix_wdg**
criteria to verify termination
Future Work

- Add more interesting criteria in theory matrix_wdg to verify termination;
Future Work

- Add more interesting criteria in theory matrix_wdg to verify termination;

Define a PVS language $L$ inside PVS

Construct a CCG for terms of $L$

termination criterion in CCG $\uparrow$

termination for $L$
Thank you!
Example: an application of the first criterion

\[
ack(m, n) := \begin{cases} 
  n + 1 & \text{if } m = 0 \\
  1 : \ack(m - 1, 1) & \text{if } m > 0 \land n = 0 \\
  2 : \ack(m - 1, 3 : \ack(m, n - 1)) & \text{if } m > 0 \land n > 0 
\end{cases}
\]

measures: \( \mu_1(m, n) = m \) and \( \mu_2(m, n) = n \)
Example: an application of the first criterion
Referências

C.S. Lee, N.D. Jones and A.M Ben-Amram.
The Size-Change Principle for Program Termination.

P. Manolios and D. Vroon
Termination Analysis with Calling Context Graphs.

C. Muñoz and M. Ayala-Rincón
Automating Termination in PVS by the Size-Change Principle. Personal communication on the implementation of the size-change principle by calling context graphs in PVS.

P.C. Dillinger, P. Manolios, D. Vroon and J.S. Moore
ACL2s: The ACL2 Sedan.

S. Owre, J.M. Rushby and N. Shankar
PVS: A Prototype Verification System.