Universidade de Brasília

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

Motivation

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

```
ack(m:nat, n:nat) :  RECURSIVE nat =
   IF m = 0 THEN n+1
      ELSIF n = 0 THEN 1:ack(m-1, 1)
      ELSE 2:ack(m-1, 3:ack(m, n-1))
   ENDIF
MEASURE lex2(m, n)
```

- For each recursive call there is a termination TCC:

1: ack(m-1, 1)

```
ack_TCC2: OBLIGATION
  FORALL (m, n: nat): n = 0 AND NOT m = 0
      IMPLIES lex2(m - 1, 1) < lex2(m, n);
```

Size Change Principle (SCP)

→ Introduced by C. S. Lee, N. D. Jones and A. M. Ben-Amram, *The Size-Change Principle for Program Termination*, *POPL*, 81–92, 2001.

- 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

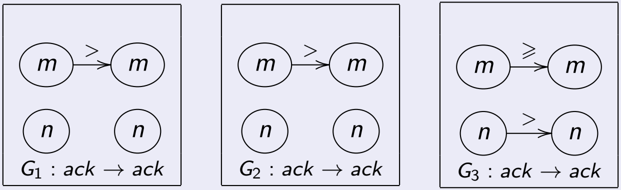
$$ack(m,n) := \begin{cases} n+1 & \text{if } m = 0 \\ 1 : ack(m-1,1) & \text{if } m > 0 \wedge n = 0 \\ 2 : ack(m-1, 3 : ack(m, n-1)) & \text{if } m > 0 \wedge n > 0 \end{cases}$$

Example

Ackermann function definition

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

Size Change Graphs for Ackermann



Calling Context Graphs (CCG)

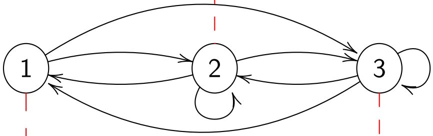
→ Introduced by P. Manolios and D. Vroon, *Termination Analysis with Calling Context Graphs*, *CAV*, 401–414, 2006.

- 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

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

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



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

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

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:

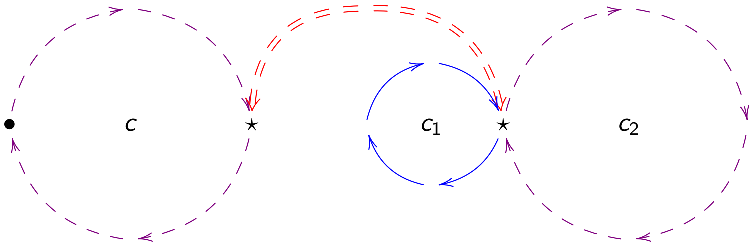
> For all digraph G and circuit c in G: c is a cycle or c is equivalent to a circuit of the form $c_1 \circ c_2$, where c_1 is a cycle and c_2 is a circuit.

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* `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.

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 $w = w_0 \dots w_n$ 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:

$$\begin{tikzcd} w_0 \arrow[r] \arrow[rr, bend left, dashed, no head, "wgt(w)"] & w_j \arrow[r] & w_n \end{tikzcd} = \begin{tikzcd} w_0 \arrow[r] \arrow[r, bend left, dashed, no head, "wgt(w^{(0,j)})"] & w_j \arrow[r] \arrow[r, bend left, dashed, no head, "wgt(w^{(j,n)})"] & w_n \end{tikzcd}$$

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 $\{-1, 0, 1\}$

$$u \xrightarrow{\mu_i \overset{?}{\geqslant} \mu_j} v$$

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 $\{-1, 0, 1\}$

$$u \xrightarrow{\mu_i \overset{?}{\geqslant} \mu_j} v$$

$u \xrightarrow{1} v$ - The measure decreases;

$u \xrightarrow{0} v$ - The measure remains less than or equal;

$u \xrightarrow{-1} v$ - I don't know.

The *theory* `measures`

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

$$w_0 \overset{1 \vee 0}{\dashrightarrow} w_i \xrightarrow{1} w_{i+1} \overset{1 \vee 0}{\dashrightarrow} w_n \implies w_0 \overset{1}{\dashrightarrow} w_n$$

The *theory* `measures`

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

$$w_0 \overset{1 \vee 0}{\dashrightarrow} w_i \xrightarrow{1} w_{i+1} \overset{1 \vee 0}{\dashrightarrow} w_n \implies w_0 \overset{1}{\dashrightarrow} w_n$$

$$w_0 \overset{0}{\dashrightarrow} w_i \xrightarrow{0} w_{i+1} \overset{0}{\dashrightarrow} w_n \implies w_0 \overset{0}{\dashrightarrow} w_n$$

The *theory* `measures`

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

$$w_0 \overset{1\vee 0}{\dashrightarrow} w_i \xrightarrow{1} w_{i+1} \overset{1\vee 0}{\dashrightarrow} w_n \implies w_0 \overset{1}{\dashrightarrow} w_n$$

$$w_0 \overset{0}{\dashrightarrow} w_i \xrightarrow{0} w_{i+1} \overset{0}{\dashrightarrow} w_n \implies w_0 \overset{0}{\dashrightarrow} w_n$$

$$w_0 \overset{1\vee 0}{\dashrightarrow} w_i \xrightarrow{-1} w_{i+1} \overset{1\vee 0}{\dashrightarrow} w_n \implies w_0 \overset{-1}{\dashrightarrow} w_n$$

The *theory* measures

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

$$w_0 \xrightarrow{1 \vee 0} w_i \xrightarrow{1} w_{i+1} \xrightarrow{1 \vee 0} w_n \Longrightarrow w_0 \xrightarrow{1} w_n$$

$$w_0 \xrightarrow{0} w_i \xrightarrow{0} w_{i+1} \xrightarrow{0} w_n \Longrightarrow w_0 \xrightarrow{0} w_n$$

$$w_0 \xrightarrow{1 \vee 0} w_i \xrightarrow{-1} w_{i+1} \xrightarrow{1 \vee 0} w_n \Longrightarrow w_0 \xrightarrow{-1} w_n$$

$$+ : \{-1, 0, 1\} \times \{-1, 0, 1\} \to \{-1, 0, 1\}$$

$$+(x, y) := \begin{cases} -1 & \text{if } x = -1 \vee y = -1 \\ 1 & \text{if } (x \neq -1 \vee y \neq -1) \wedge (x = 1 \vee y = 1) \\ 0 & \text{if } x = 0 \wedge y = 0 \end{cases}$$

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.

$$v_i \xrightarrow{\begin{pmatrix} \mu_1(\vec{x}) \overset{?}{\geqslant} \mu_1(\vec{y}) & \dots & \mu_1(\vec{x}) \overset{?}{\geqslant} \mu_n(\vec{y}) \\ \mu_2(\vec{x}) \overset{?}{\geqslant} \mu_1(\vec{y}) & \dots & \mu_2(\vec{x}) \overset{?}{\geqslant} \mu_n(\vec{y}) \\ \vdots & & \vdots \\ \mu_n(\vec{x}) \overset{?}{\geqslant} \mu_1(\vec{y}) & \dots & \mu_n(\vec{x}) \overset{?}{\geqslant} \mu_n(\vec{y}) \end{pmatrix}} v_j$$

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

The *theory* `measures`

$$v_1 \xrightarrow{\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & a_{in} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}} v_2 \xrightarrow{\begin{pmatrix} b_{11} & \dots & b_{1j} & \dots & b_{1n} \\ b_{21} & \dots & b_{2j} & \dots & b_{2n} \\ \vdots & & \vdots & & \vdots \\ b_{n1} & \dots & b_{nj} & \dots & b_{nn} \end{pmatrix}} v_3$$

The *theory* `measures`

$$
v_1 \xrightarrow{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{in} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}} v_2 \xrightarrow{\begin{pmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\ b_{21} & \cdots & b_{2j} & \cdots & b_{2n} \\ \vdots & & \vdots & & \vdots \\ b_{n1} & \cdots & b_{nj} & \cdots & b_{nn} \end{pmatrix}} v_3
$$

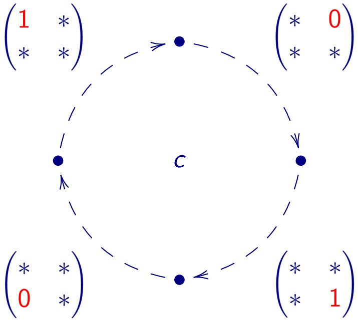
$$\Downarrow$$

$$
v_1 \xrightarrow{\begin{pmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1n} \\ \vdots & & \vdots & & \vdots \\ c_{i1} & \cdots & c_{ij} & \cdots & c_{in} \\ \vdots & & \vdots & & \vdots \\ c_{n1} & \cdots & c_{nj} & \cdots & c_{nn} \end{pmatrix}} v_3
$$

$$c_{ij} = \max\{a_{ik} + b_{kj} \mid 1 \leqslant k \leqslant n\}$$

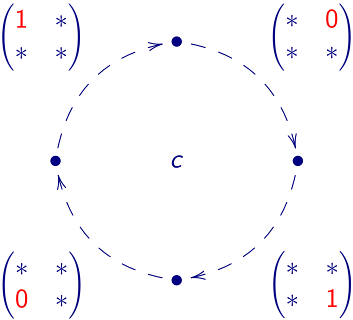
The *theory* `measures`

- The notion of **termination**



The *theory* `measures`

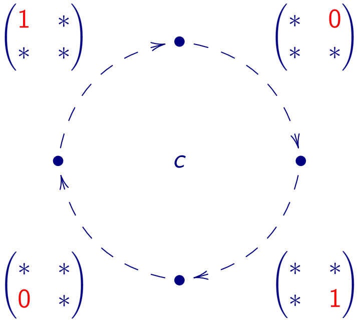
- The notion of **termination**



$$\mu_1 > \mu_1 \wedge \mu_1 \geqslant \mu_2 \wedge \mu_2 > \mu_2 \wedge \mu_2 \geqslant \mu_1$$

The *theory* `measures`

- The notion of **termination**

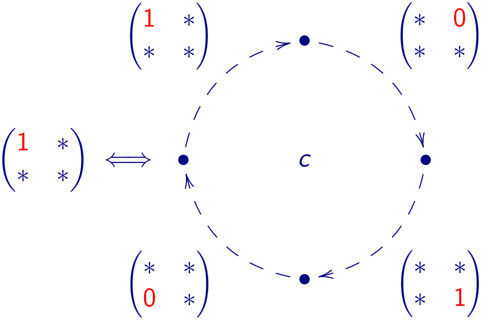


$$\mu_1 > \mu_1 \wedge \mu_1 \geqslant \mu_2 \wedge \mu_2 > \mu_2 \wedge \mu_2 \geqslant \mu_1$$

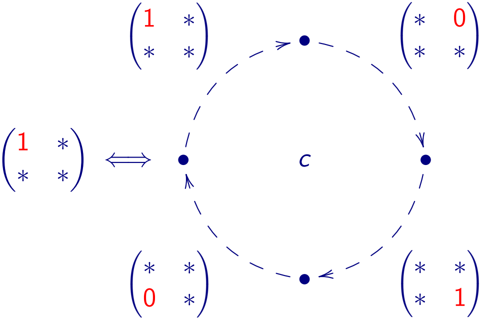
$$\Downarrow$$

$$\mu_1 > \mu_1 \geqslant \mu_2 > \mu_2 \geqslant \mu_1$$

The *theory* `measures`



The *theory* `measures`



termination

$\Updownarrow$

positive measure-matrix

$\Updownarrow$

$$\begin{pmatrix} \ddots & * & \dots & * \\ * & \ddots & \dots & * \\ \vdots & \vdots & 1 & \vdots \\ * & * & \dots & \ddots \end{pmatrix}$$

The *theory* `matrix_wdg`

- Two criteria to verify termination:

The *theory* `matrix_wdg`

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

$$\forall e \in G : M^{e}(k,k) \geqslant 0$$

$$\forall c \in G, \text{ such that } c \text{ is a double cycle} : M^{c}(k,k) = 1$$

The *theory* matrix_wdg

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

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

  - 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.

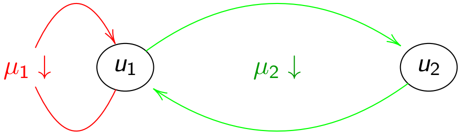
$$\forall e = (v_i, v_j) \in G : M^e(k_i, k_j) \geqslant 0$$
$$\forall c \in G, \text{ such that } c = v_\alpha \dots v_\alpha \text{ is a cycle} : M^c(k_\alpha, k_\alpha) = 1$$

Example: an application of the first criterion

$$gcd : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

$$gcd(m,n) := \begin{cases} m+n & \text{if } m = 0 \vee n = 0 \\ 1 : gcd(m-n,n) & \text{if } m \geqslant n \wedge m \neq 0 \wedge n \neq 0 \\ 2 : gcd(n,m) & \text{if } n > m \wedge m \neq 0 \wedge n \neq 0 \end{cases}$$

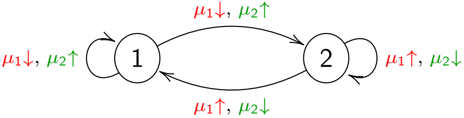
- measures: $\mu_1(m,n) = m$ and $\mu_2(m,n) = n$



A non-Terminating Example

$$f(\{n \mid n \in \mathbb{N} \wedge n \leqslant 100\}) := \begin{cases} 1: f(n-1) & \text{if } n \geqslant 50 \\ 2: f(n+1) & \text{if } n < 50 \end{cases}$$

Measures: $\mu_1(n) := |n|$ and $\mu_2(n) := |100 - n|$

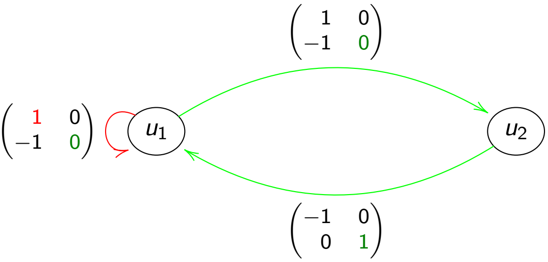


Example: an application of the first criterion

$$gcd : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

$$gcd(m, n) := \begin{cases} m + n & \text{if } m = 0 \vee n = 0 \\ 1 : gcd(m - n, n) & \text{if } m \geqslant n \wedge m \neq 0 \wedge n \neq 0 \\ 2 : gcd(n, m) & \text{if } n > m \wedge m \neq 0 \wedge n \neq 0 \end{cases}$$

- The limiting measure: $\mu_2(m, n) = n$



Example: an application of the second criterion

$$p : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

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

- Measures:

$$\mu_1(m, n, r) = m + n + r$$

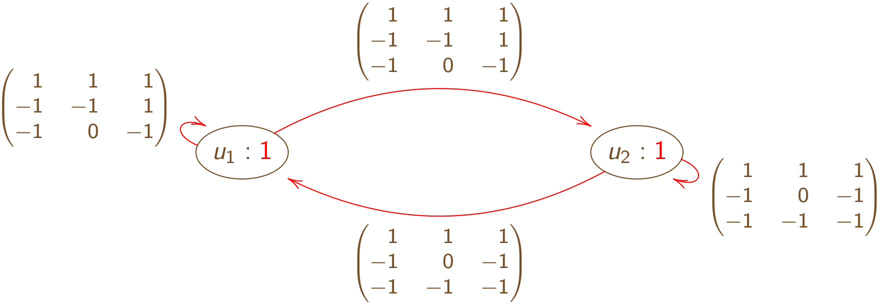
$$\mu_2(m, n, r) = m + r$$

$$\mu_3(m, n, r) = m + n$$

Example: an application of the second criterion

$$p : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

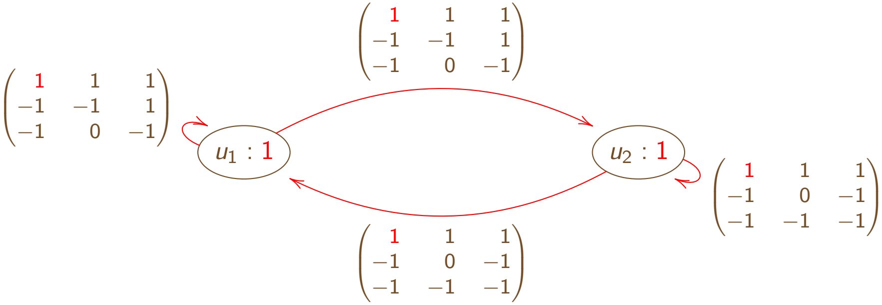
$$p(m,n,r)) := \begin{cases} 1 : p(m, r-1, n) & \text{if } r > 0 \\ 2 : p(r, n-1, m) & \text{if } r = 0 \wedge n > 0 \\ m & \text{if } r = 0 \wedge n = 0 \end{cases}$$



Example: an application of the second criterion

$$p : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

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



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* `ccg_to_mwg`

$$\boxed{CCG} \xleftrightarrow[measure-combination \iff measure-matrix]{dg(CCG)=dg(MWG)} \boxed{MWG}$$

matrix

The *theory* `ccg_to_mwg`

$$\boxed{CCG} \xleftrightarrow[measure-combination \iff measure-matrix]{dg(CCG)=dg(MWG)} \boxed{MWG}$$

▸ matrix

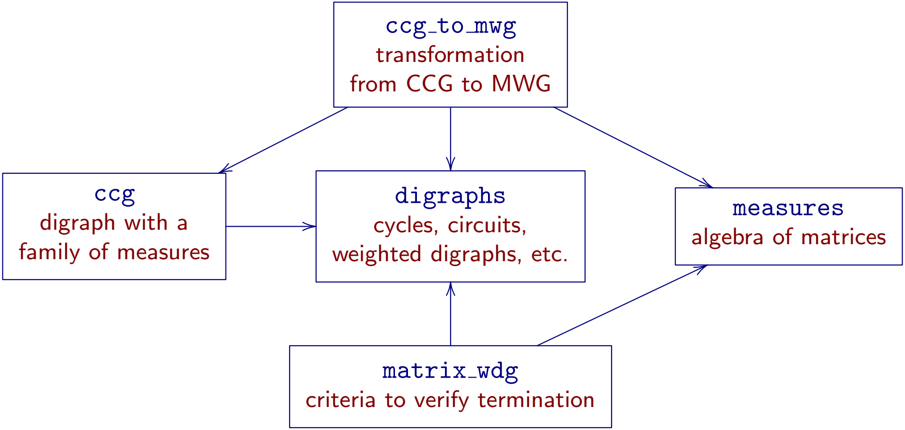
Let G be a CCG, c a circuit on G and G' the MWG corresponding to G. Then,

$$M^c_{G'} \text{ is not positive}$$

$$\Updownarrow$$

$$\forall\ mc : measure\_combination_G(c),\ mc \text{ is not greater than}$$

Summarizing

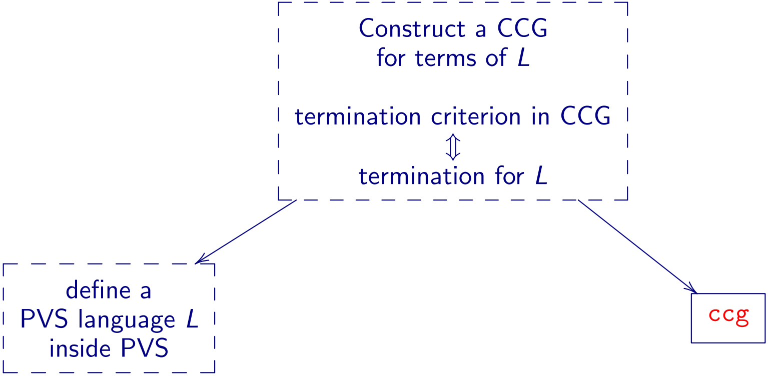


Future Work

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

Future Work

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

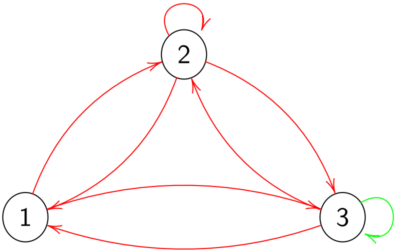


Thank you!

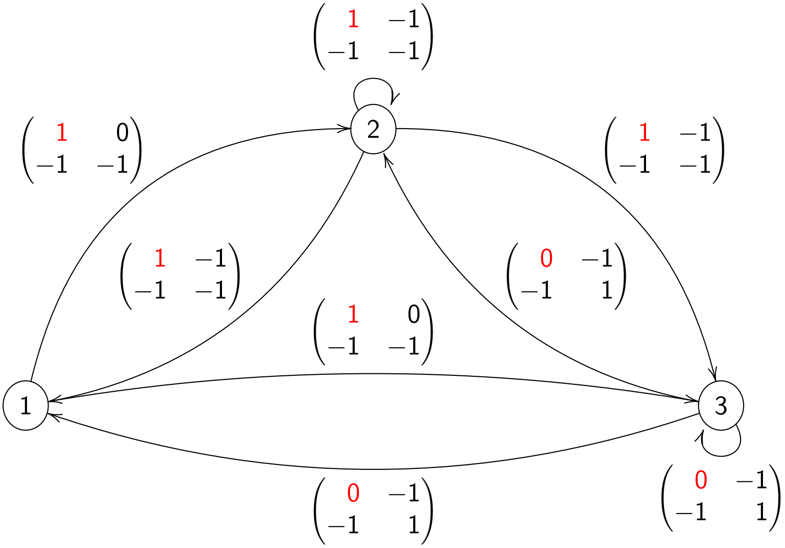
Example: an application of the first criterion

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

- 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.
POPL, 81–92, 2001.

P. Manolios and D. Vroon
Termination Analysis with Calling Context Graphs.
CAV, 401–414, 2006.

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.
International Conference on Software Engineering, 2007.

S. Owre, J.M. Rushby and N. Shankar
PVS: A Prototype Verification System.
cade, volume 607, 748–752, 1992.