Výzkumná skupinka VERIF na katedře informatiky FEI VŠB-TU Ostrava

Petr Jančar

katedra informatiky FEI VŠB-TU Ostrava

vystoupení na semináři Hovory s informatiky SoSIReČR
12. června 2012

Obsah

- http://verif.cs.vsb.cz/
  Petr Jančar
  Marek Běhálek, Stanislav Böhm, Martin Kot, Zdeněk Sawa
- teoretická práce v oblasti (mezí automatizované) verifikace
  aktuální výsledek:
  P. Jančar:
  Decidability of DPDA Language Equivalence via First-Order Grammars
  (nový důkaz ekvivalence deterministických zásobníkových automatů)
- praktičtěji orientovaný výzkum:
  Kaira - softwarový nástroj pro modelování a generování paralelních aplikací (hlavně S. Böhm, http://verif.cs.vsb.cz/kaira/)

Language equivalence of deterministic pushdown automata

Example of a (formal) language L over a finite alphabet Σ, so $L \subseteq \Sigma^*$:

$\Sigma = \{\, 0,1,2,3,4,5,6,7,8,9,+,-,*,(,)\,\} \cup \{\, \dashv \}$

L ... language of arithmetic expressions

e.g., the word (sequence)

$u = \boxed{5 + 28 * (318 - 5 * 24) + 562 \dashv}$ is in L,

$v = \boxed{5 + 28 * (318 - (5 * 24) + 562 \dashv}$ is not in L

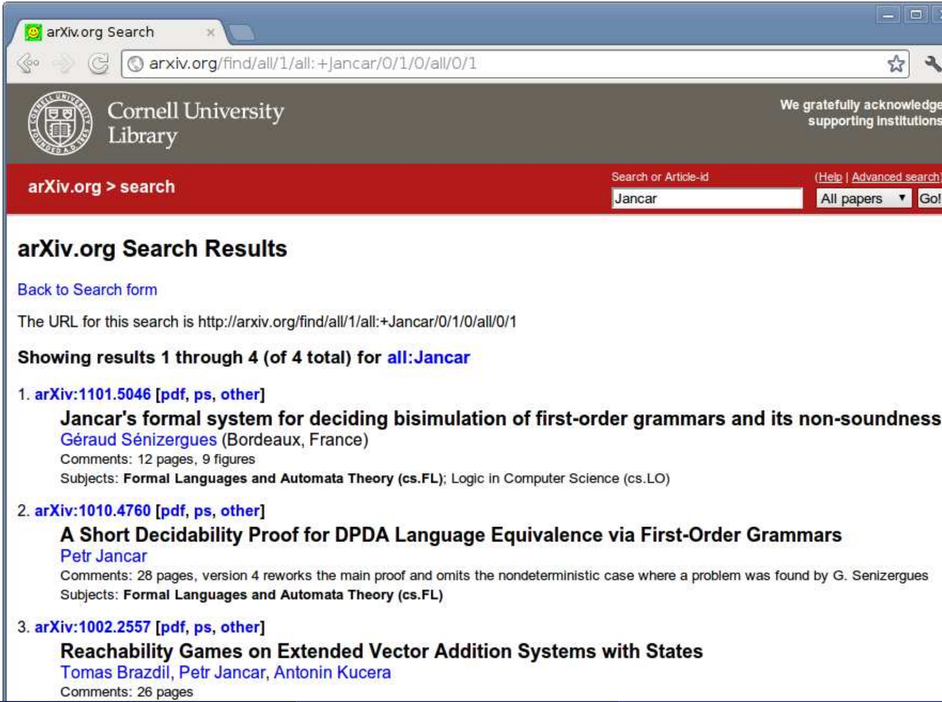
We can view a deterministic pushdown automaton M as a program

- with fixed finite memory; program+memory...finite control unit,
- with a potentially unbounded stack (LIFO, access to the top),
- reading the input word from left-to-right,
- accepting when reading the endmarker $\dashv$ and having the stack empty.

Decidability of $L(M_1) \overset{?}{=} L(M_2)$ was open since 1960s (stated in a paper by Ginsburg, Greibach). Another formulation: $L(p\alpha) \overset{?}{=} L(q\beta)$ for configurations of the same M (p ... control state, α ... stack content).

Solution

- Sénizergues G.:
  L(A)=L(B)? Decidability results from complete formal systems.
  Theoretical Computer Science 251(1-2): 1-166 (2001)
  (a preliminary version appeared at ICALP'97; Gödel prize 2002)
- Stirling C.: Decidability of DPDA equivalence.
  Theoretical Computer Science 255, 1-31, 2001
- Sénizergues G.: L(A)=L(B)? A simplified decidability proof.
  Theoretical Computer Science 281(1-2): 555-608 (2002)
- Stirling C.: Deciding DPDA equivalence is primitive recursive.
  ICALP 2002, Lecture Notes in Computer Science 2380, 821-832, Springer 2002 (longer draft paper on the author's web page)
- Sénizergues G.: The Bisimulation Problem for Equational Graphs of Finite Out-Degree.
  SIAM J.Comput., 34(5), 1025–1106 (2005)
  (a preliminary version appeared at FOCS'98)



arXiv.org Search Results

Back to Search form

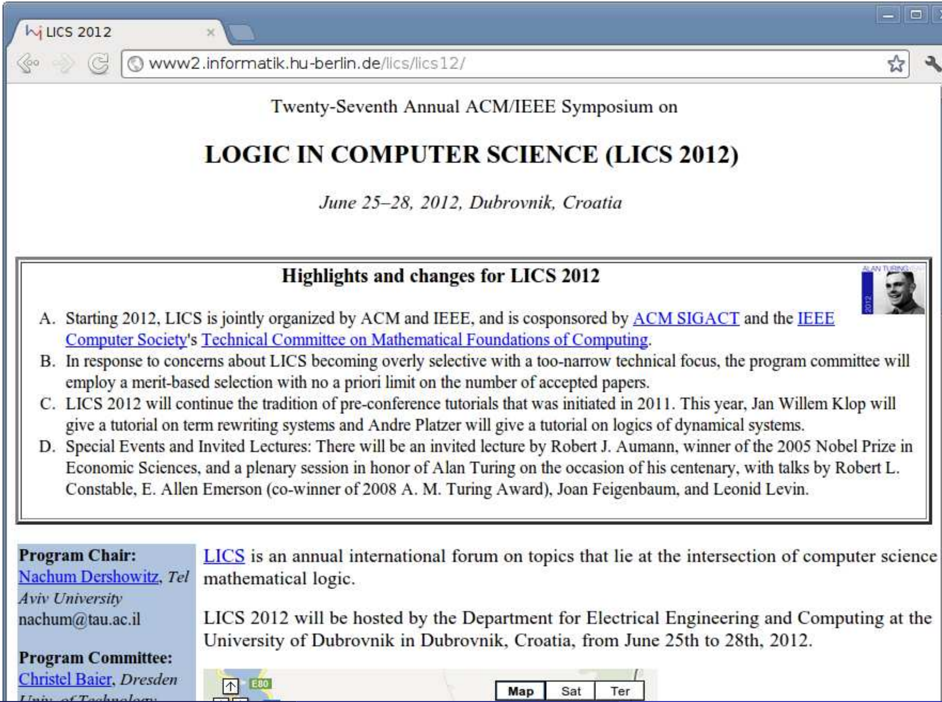
The URL for this search is http://arxiv.org/find/all/1/all:+Jancar/0/1/0/all/0/1

Showing results 1 through 4 (of 4 total) for all:Jancar

1. **arXiv:1101.5046 [pdf, ps, other]**
   Jancar's formal system for deciding bisimulation of first-order grammars and its non-soundness
   Géraud Sénizergues (Bordeaux, France)
   Comments: 12 pages, 9 figures
   Subjects: **Formal Languages and Automata Theory (cs.FL)**; Logic in Computer Science (cs.LO)

2. **arXiv:1010.4760 [pdf, ps, other]**
   A Short Decidability Proof for DPDA Language Equivalence via First-Order Grammars
   Petr Jancar
   Comments: 28 pages, version 4 reworks the main proof and omits the nondeterministic case where a problem was found by G. Senizergues
   Subjects: **Formal Languages and Automata Theory (cs.FL)**

3. **arXiv:1002.2557 [pdf, ps, other]**
   Reachability Games on Extended Vector Addition Systems with States
   Tomas Brazdil, Petr Jancar, Antonin Kucera
   Comments: 26 pages



Twenty-Seventh Annual ACM/IEEE Symposium on

LOGIC IN COMPUTER SCIENCE (LICS 2012)

June 25–28, 2012, Dubrovnik, Croatia

Highlights and changes for LICS 2012

A. Starting 2012, LICS is jointly organized by ACM and IEEE, and is cosponsored by ACM SIGACT and the IEEE Computer Society's Technical Committee on Mathematical Foundations of Computing.

B. In response to concerns about LICS becoming overly selective with a too-narrow technical focus, the program committee will employ a merit-based selection with no a priori limit on the number of accepted papers.

C. LICS 2012 will continue the tradition of pre-conference tutorials that was initiated in 2011. This year, Jan Willem Klop will give a tutorial on term rewriting systems and Andre Platzer will give a tutorial on logics of dynamical systems.

D. Special Events and Invited Lectures: There will be an invited lecture by Robert J. Aumann, winner of the 2005 Nobel Prize in Economic Sciences, and a plenary session in honor of Alan Turing on the occasion of his centenary, with talks by Robert L. Constable, E. Allen Emerson (co-winner of 2008 A. M. Turing Award), Joan Feigenbaum, and Leonid Levin.

Program Chair:
Nachum Dershowitz, *Tel Aviv University*
nachum@tau.ac.il

Program Committee:
Christel Baier, *Dresden Univ. of Technology*

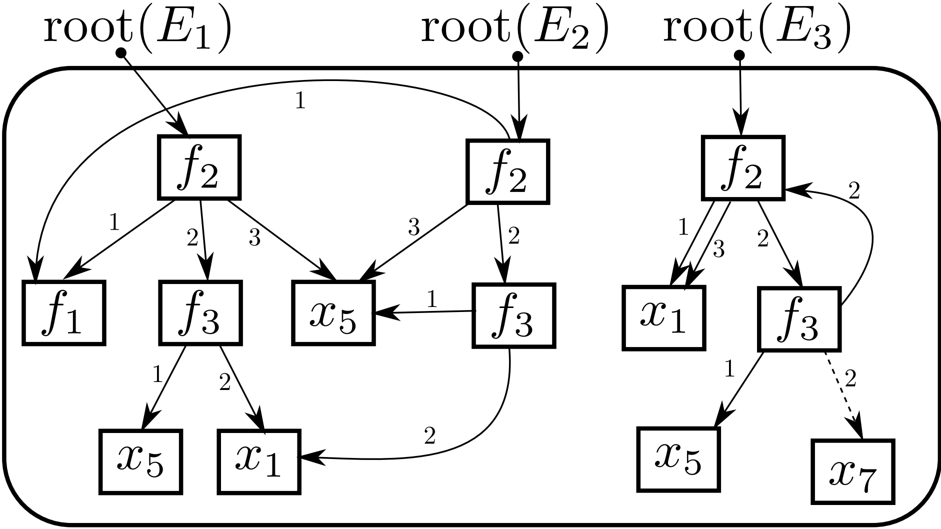
LICS is an annual international forum on topics that lie at the intersection of computer science mathematical logic.

LICS 2012 will be hosted by the Department for Electrical Engineering and Computing at the University of Dubrovnik in Dubrovnik, Croatia, from June 25th to 28th, 2012.

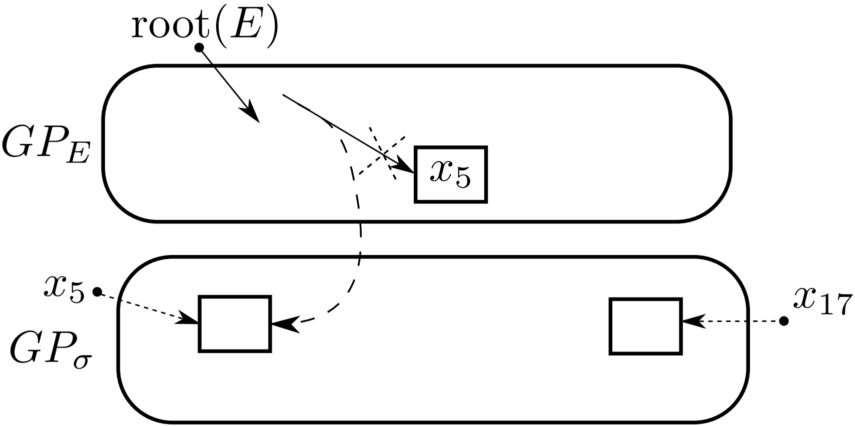
Outline (a short self-contained proof via first-order terms)

- First-order terms, substitutions; regular terms
- (Deterministic) labelled transition systems (LTSs); trace equivalence
- (D)pda configurations as terms; rules as root-rewriting
- LTSs generated by (det-)first-order grammars; semidecidability of nonequivalence
- Simple properties of $\sim$ and its "strata" $\sim_0$, $\sim_1$, $\sim_2$, ...
- Algorithm for the positive case based on a Prover-Refuter game
- Soundness of P-R game (obvious)
- Two steps for completeness
  - (n, g)-strategies for Prover are sufficient
  - A balancing strategy is an (n, g)-strategy
- Remarks
  - Bisimulation equivalence in the nondeterministic case
  - A complexity bound in the deterministic case

First-order regular terms; partial mapping $E : \mathbb{N}^* \to \mathcal{F} \cup \mathcal{V}$

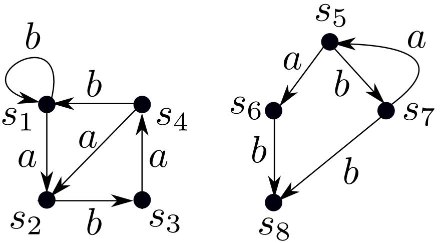


Substitutions $\sigma : \mathcal{V} \rightarrow \textsc{Terms}_{\mathcal{F}}$; terms $E\sigma$, $E\sigma_1\sigma_2$, ...



Finite-support restriction: $\textsc{supp}(\sigma) = \{x_i \mid \sigma(x_i) \neq x_i\}$ is finite
Composing substitutions: $(\sigma_1\sigma_2)(x_i) = (\sigma_1(x_i))\sigma_2$
Associativity: $(E\sigma_1)\sigma_2 = E(\sigma_1\sigma_2)$; note that $x_i\sigma = \sigma(x_i)$

(Det-)labelled transition systems (LTSs); trace equivalence



$\mathcal{L} = (\mathcal{S}, Act, (\xrightarrow{a})_{a \in Act})$

$\mathrm{EqLv}(s_1, s_2) = 0$
$\mathrm{EqLv}(s_1, s_5) = 2$
$\mathrm{EqLv}(s_1, s_4) = \omega$

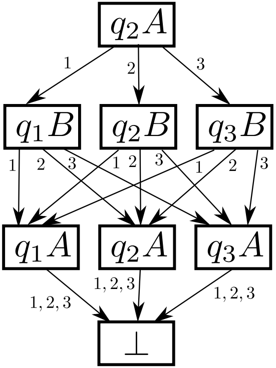
$\boxed{s \sim t}$ if $\forall w \in Act^* : s \xrightarrow{w} \Leftrightarrow t \xrightarrow{w}$; $\boxed{s \sim_k t}$ if $\forall w \in Act^{\leq k} : s \xrightarrow{w} \Leftrightarrow t \xrightarrow{w}$

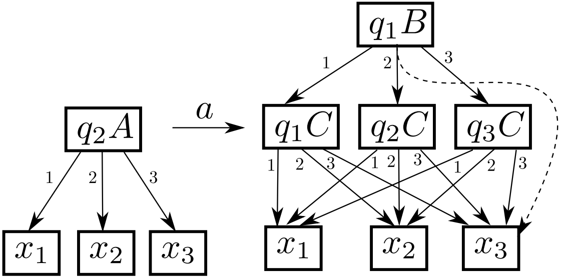
- $\mathcal{S} \times \mathcal{S} = \sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots$. $\cap_{k \in \mathbb{N}} \sim_k = \sim$.
- If $\mathrm{EqLv}(s,t) = k$ and $\mathrm{EqLv}(s,s') \geq k+1$ then $\mathrm{EqLv}(s',t) = k$ (replacing a pair-element with a "more equivalent state" does not affect the eq-level of the pair)
- In any deterministic LTS eq-level drops by at most 1 in one step:
  - If $s \xrightarrow{a} s'$, $t \xrightarrow{a} t'$ then $\mathrm{EqLv}(s',t') \geq \mathrm{EqLv}(s,t) - 1$.
  - If $\mathrm{EqLv}(s,t) = k < \omega$ then there is a such that $s \xrightarrow{a} s'$, $t \xrightarrow{a} t'$, $\mathrm{EqLv}(s',t') = \mathrm{EqLv}(s,t) - 1$.

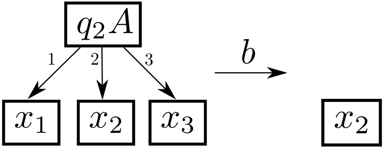
(D)pda from a first-order term perspective

$Q = \{q_1, q_2, q_3\}$
configuration q_2ABA

(pushing) rule $q_2A \xrightarrow{a} q_1BC$



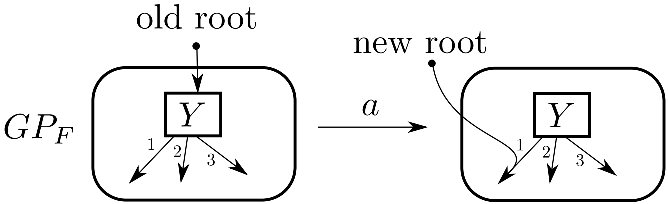


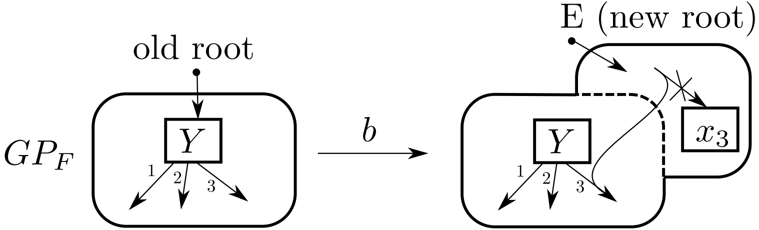


(popping) rule $q_2A \xrightarrow{b} q_2$

Applying the rules (to GP_F, for a regular F)

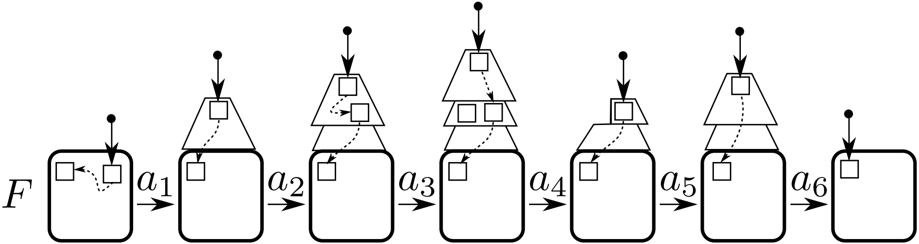
$$Yx_1x_2x_3 \xrightarrow{a} x_1$$





$$Yx_1x_2x_3 \xrightarrow{b} E$$

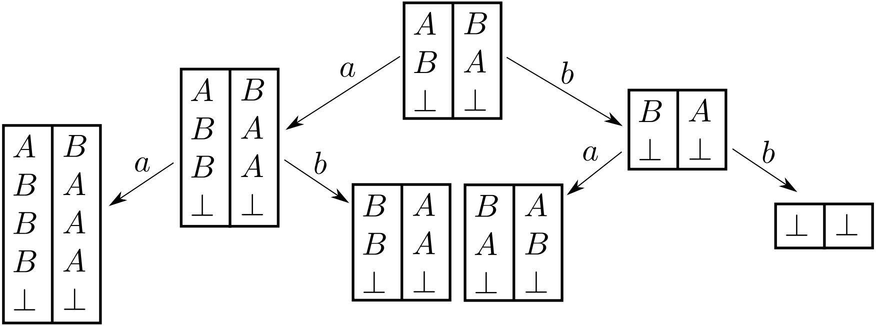
A path in $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$



k-distance regions $\textsc{Reg}(T, U, k)$

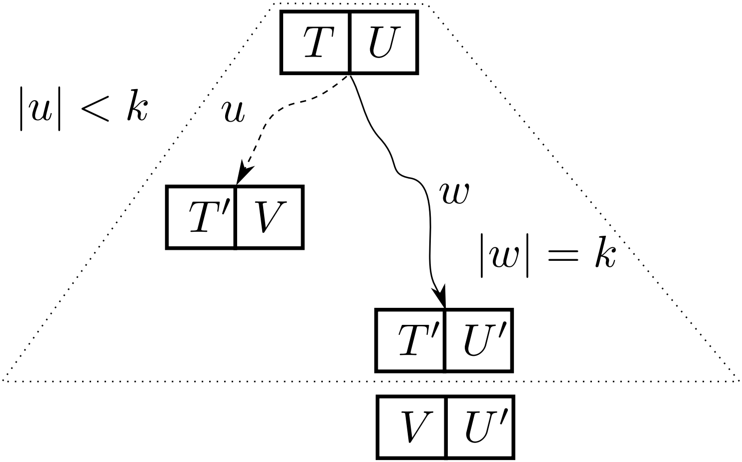
$\mathcal{G}$: $Ax_1 \xrightarrow{a} ABx_1$, $Ax_1 \xrightarrow{b} x_1$, $Bx_1 \xrightarrow{a} BAx_1$, $Bx_1 \xrightarrow{b} x_1$

The 2-distance region $\textsc{Reg}(T, U, 2)$ for $(T, U) = (AB\bot, BA\bot)$



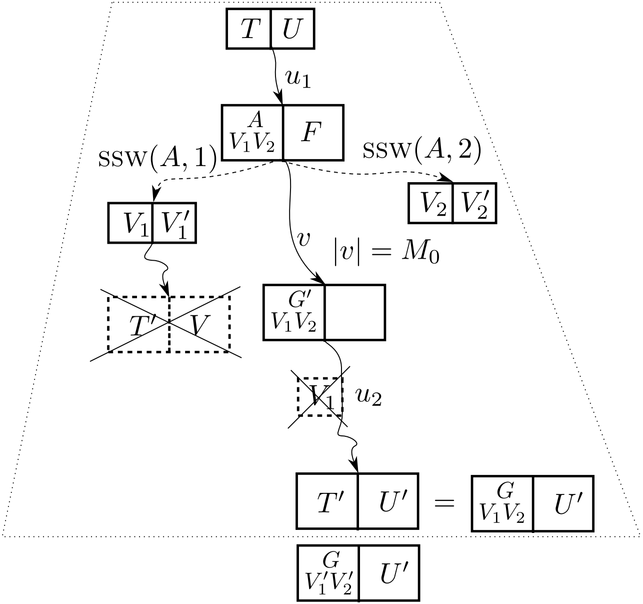
If $T \not\sim U$, $T \sim_k U$ then any least eq-level pair in $\textsc{Reg}(T, U, k)$ is at the bottom, i.e. in $\textsc{Reg}(T, U, k) \smallsetminus \textsc{Reg}(T, U, k-1)$.

Case 1 of left-balancing



(T', U') is a least eq-level pair $\implies$ $\text{EqLv}(V, U') = \text{EqLv}(T', U')$.

Case 2 of left-balancing



$V_1 \sim_{\ell+1} V_1'$,
$V_2 \sim_{\ell+1} V_2'$,

$\sigma(x_1) = V_1$,
$\sigma(x_2) = V_2$,

$\sigma'(x_1) = V_1'$,
$\sigma'(x_2) = V_2'$,

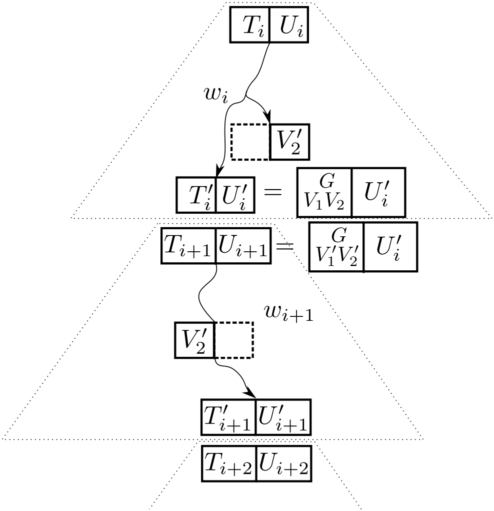
$\sigma \sim_{\ell+1} \sigma'$
$G\sigma \sim_{\ell+1} G\sigma'$

$\boxed{\mathrm{EqLv}(T', U') = \ell}$

$(T', U') = (G\sigma, U')$

$\mathrm{EqLv}(G\sigma', U') = \ell$

A left balancing phase i followed by a no-change phase $i{+}1$



(T_0, U_0)

(T_1, U_1)

(T_2, U_2)

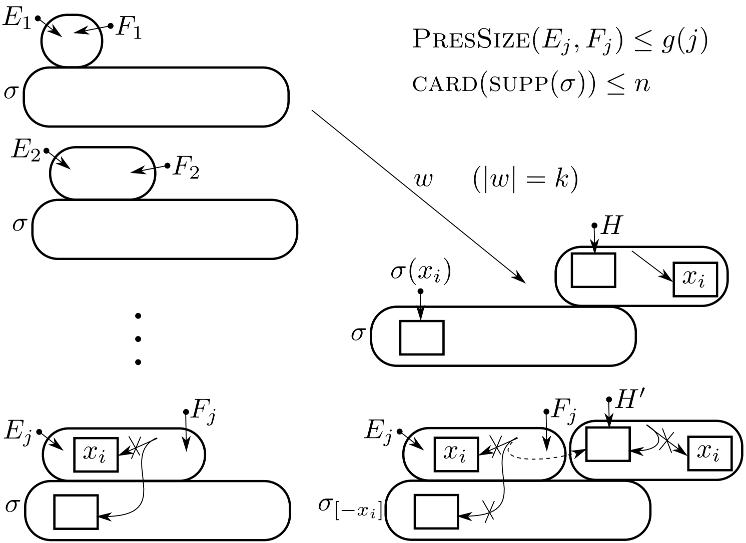
(T_3, U_3)

$\ldots$

eq-level decreasing
if Refuter's claims
are true

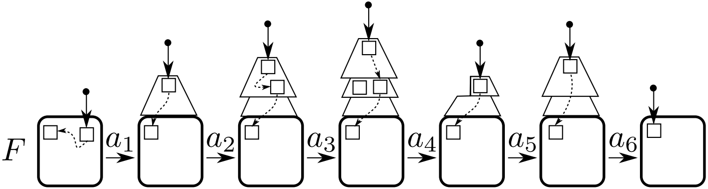
An (n, g)-strategy for $\mathcal{G}$ implies a sufficient basis

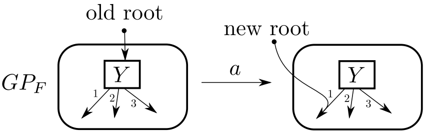
$\textsc{Basis} = \{(E, F) \mid E \sim F, \textsc{PresSize}(E, F) \leq \mathcal{B}\}$ for large $\mathcal{B} \in \mathbb{N}$

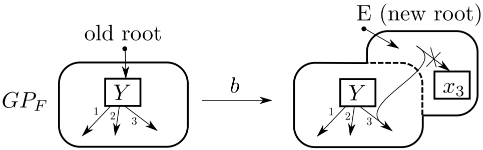


(Y,j)-sink-words; shortest sink-words $\mathrm{SSW}(Y,j)$

$w \in Act^*$ is a (Y,j)-sink-word, $1 \le j \le m = \text{arity}(Y)$, if $Yx_1 \dots x_m \xrightarrow{w} x_j$

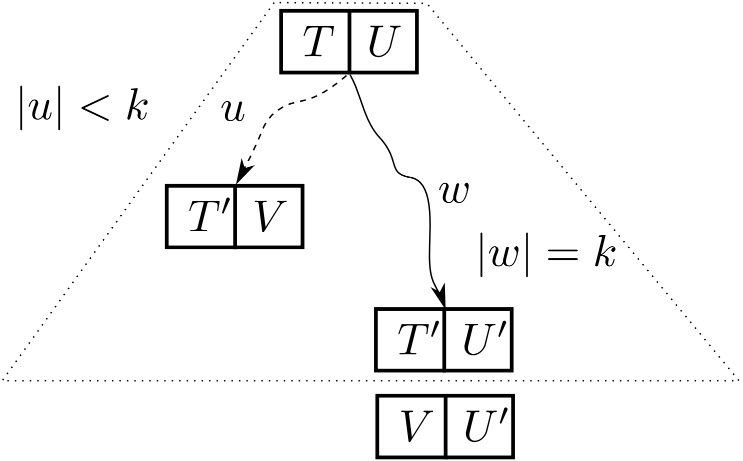




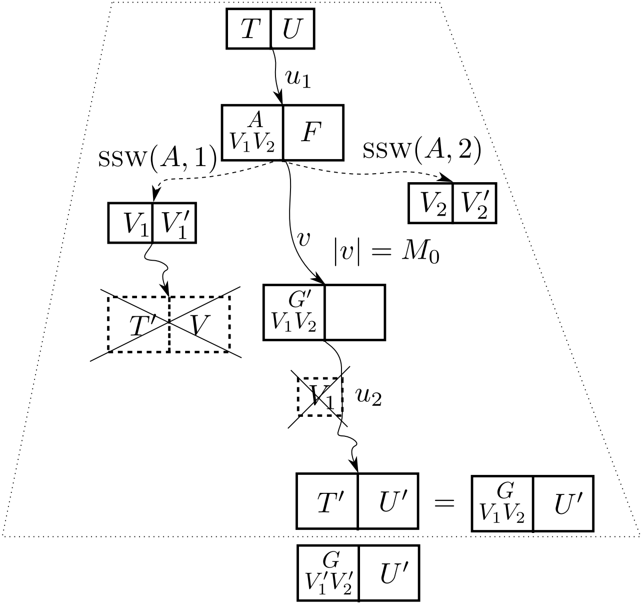


$M_0 = 1 + \max\{\, |\mathrm{SSW}(Y,j)| \mid Y \in \mathcal{N}, 1 \le j \le \text{arity}(Y) \}$

Case 1 of left-balancing

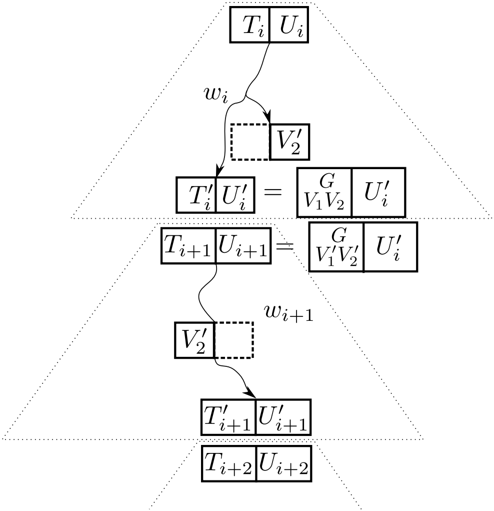


Case 2 of left-balancing



A left balancing phase *i* followed by a no-change phase $i{+}1$

U_i $\downarrow u$ V_2' $\downarrow u'$ T_{i+2}

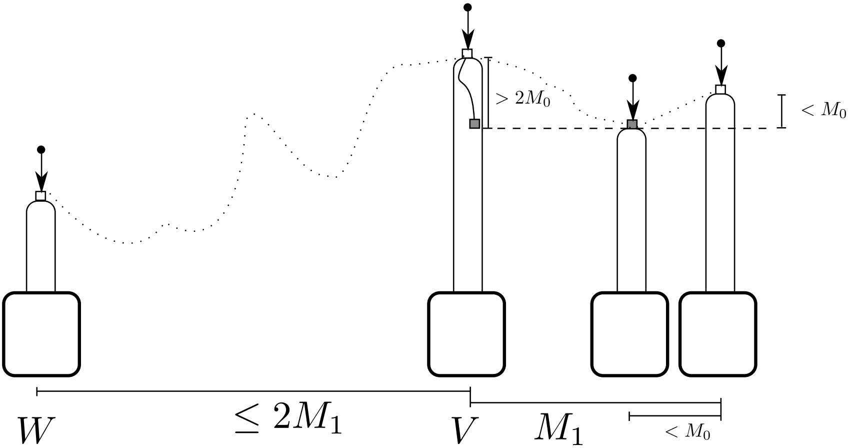


T_i U_i
w_i
V_2'
T_i' U_i' = G $V_1 V_2$ U_i'
T_{i+1} U_{i+1} = G $V_1' V_2'$ U_i'
w_{i+1}
V_2'
T_{i+1}' U_{i+1}'
T_{i+2} U_{i+2}

U_i $\downarrow w_i$ U_{i+1} $\downarrow w_{i+2}$ U_{i+2}

Balancing pivots are on a special path in $\mathcal{L}^{\mathrm{A}}_{\mathcal{G}}$

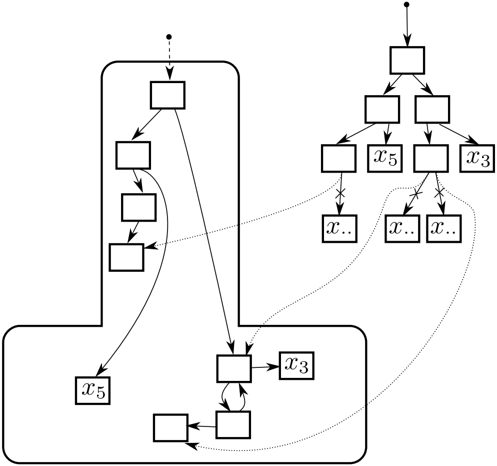
$$W_1 \xrightarrow{v_1} W_2 \xrightarrow{v_2} W_3 \xrightarrow{v_3} \cdots$$



Recall $M_1 = M_0 \cdot (2 + (2M_0 - 1) \cdot \textsc{StepDepthInc})$

Presenting V as $V = (\textsc{Top}^V_d)\sigma$

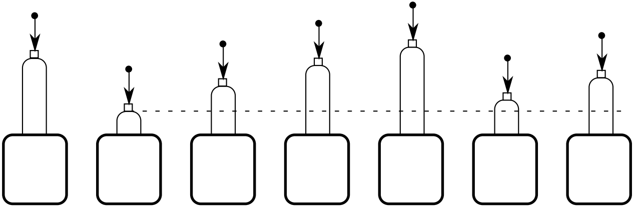
$\textsc{supp}(\sigma) \subseteq \{x_1, x_2, \ldots, x_{c^d}\}$ where $c = \max\{\,\text{arity}(Y) \mid Y \in \mathcal{N}\,\}$



$d = 3$ in the example

Balancing strategy is an (n, g)-strategy (for the given $\mathcal{G}$)

If the pivot path $W_1 \xrightarrow{v_1} W_2 \xrightarrow{v_2} W_3 \xrightarrow{v_3} \cdots$ is finite or visits some V' infinitely often then we have a repeat ($(T_j, U_j) = (T_i, U_i)$ for $j > i$). Otherwise we have a "stair-base":

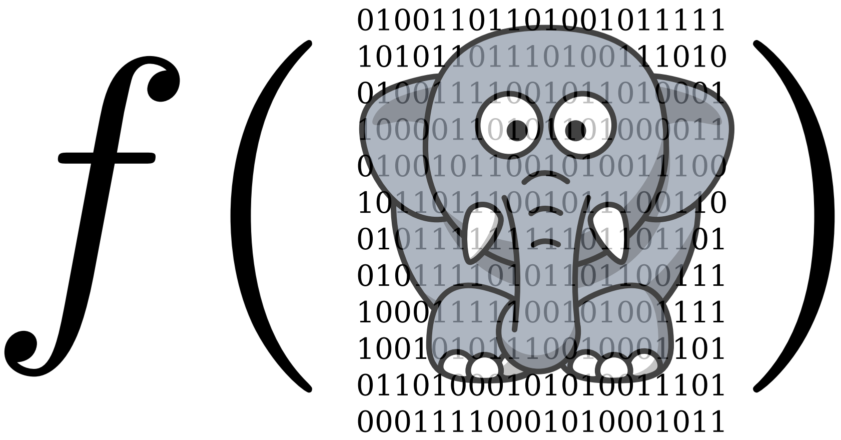


$$W_1 \xrightarrow{u} V = ({\color{red}Yx_1 \ldots x_m}){\color{blue}\sigma'} \xrightarrow{u'} {\color{red}H_1}{\color{blue}\sigma'} \xrightarrow{v_{k+1}} {\color{red}H_2}{\color{blue}\sigma'} \xrightarrow{v_{k+2}} \cdots$$

$$\text{where } ({\color{red}Yx_1 \ldots x_m}) \xrightarrow{u'} {\color{red}H_1} \xrightarrow{v_{k+1}} {\color{red}H_2} \xrightarrow{v_{k+2}} \cdots,$$

and ${\color{red}H_j}{\color{blue}\sigma'} = W_{k+j}$ ($j = 1, 2, \ldots$) are the pivots after $V = ({\color{red}Yx_1 \ldots x_m}){\color{blue}\sigma'}$. Putting $V = (\textsc{Top}^{V}_{M_1}){\color{red}\sigma} = (({\color{red}Yx_1 \ldots x_m})\sigma''){\color{red}\sigma}$, we have $W_{k+j} = {\color{red}H_j}\sigma''{\color{red}\sigma}$, and the bal-result with W_{k+j} is $({\color{blue}E_j}{\color{red}\sigma}, {\color{blue}F_j}{\color{red}\sigma})$, where ${\color{blue}E_j}, {\color{blue}F_j}$ are finite terms; we can easily find function $g : \mathbb{N} \to \mathbb{N}$ such that $\textsc{PresSize}({\color{blue}E_j}, {\color{blue}F_j}) \leq g(j)$.

Problem

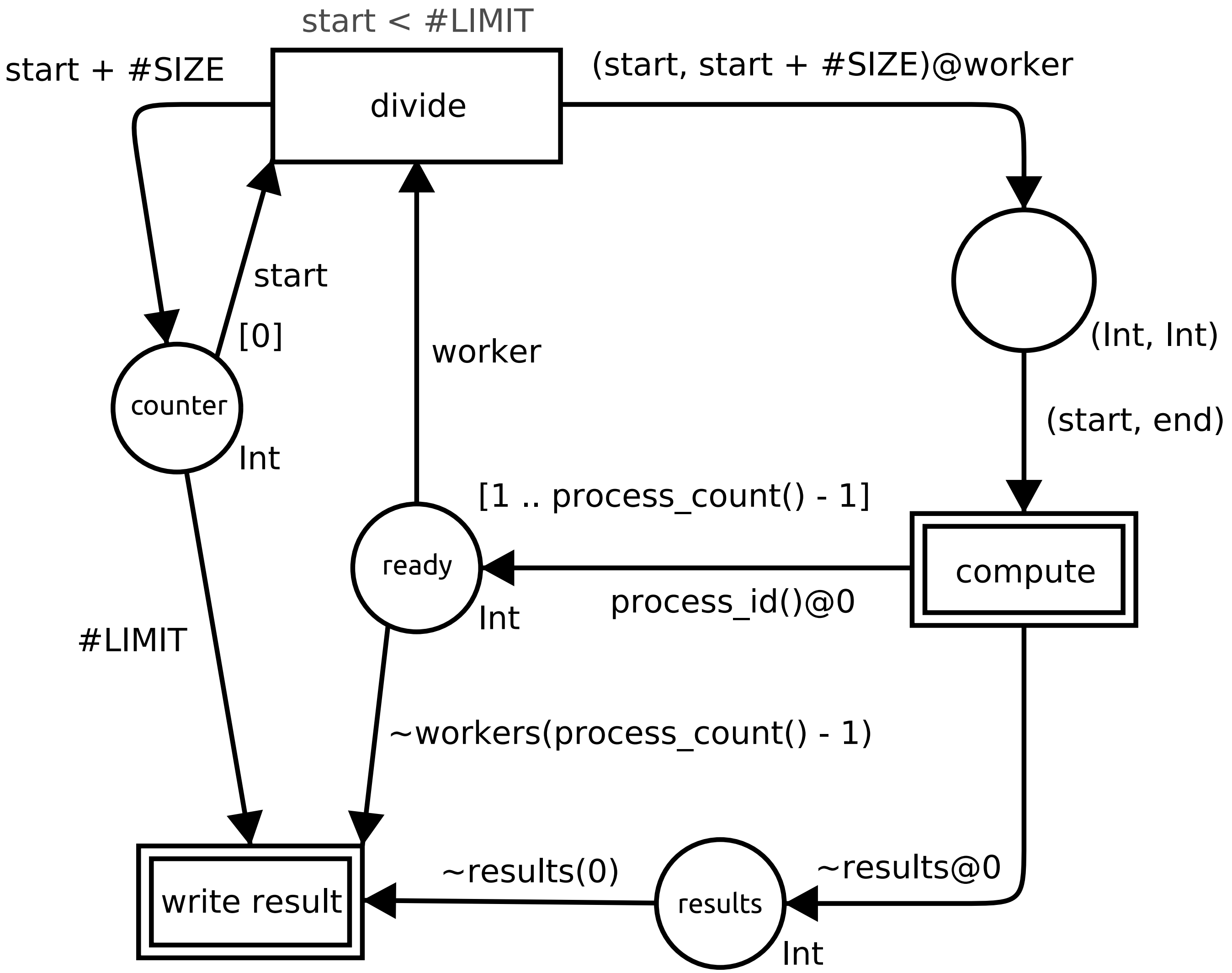


Big data

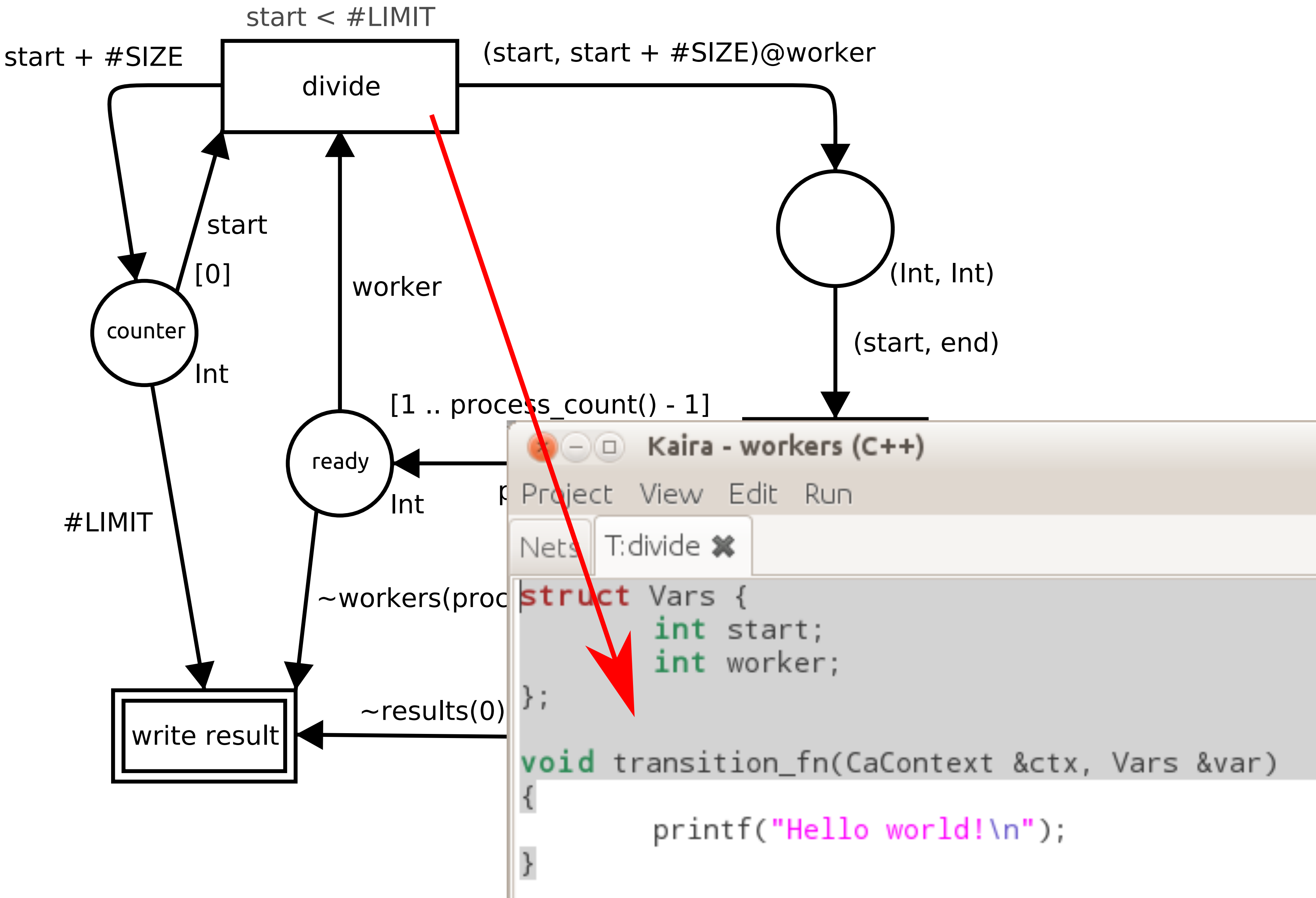
+

Parallel computer

Visual model of parallel aspects and communication



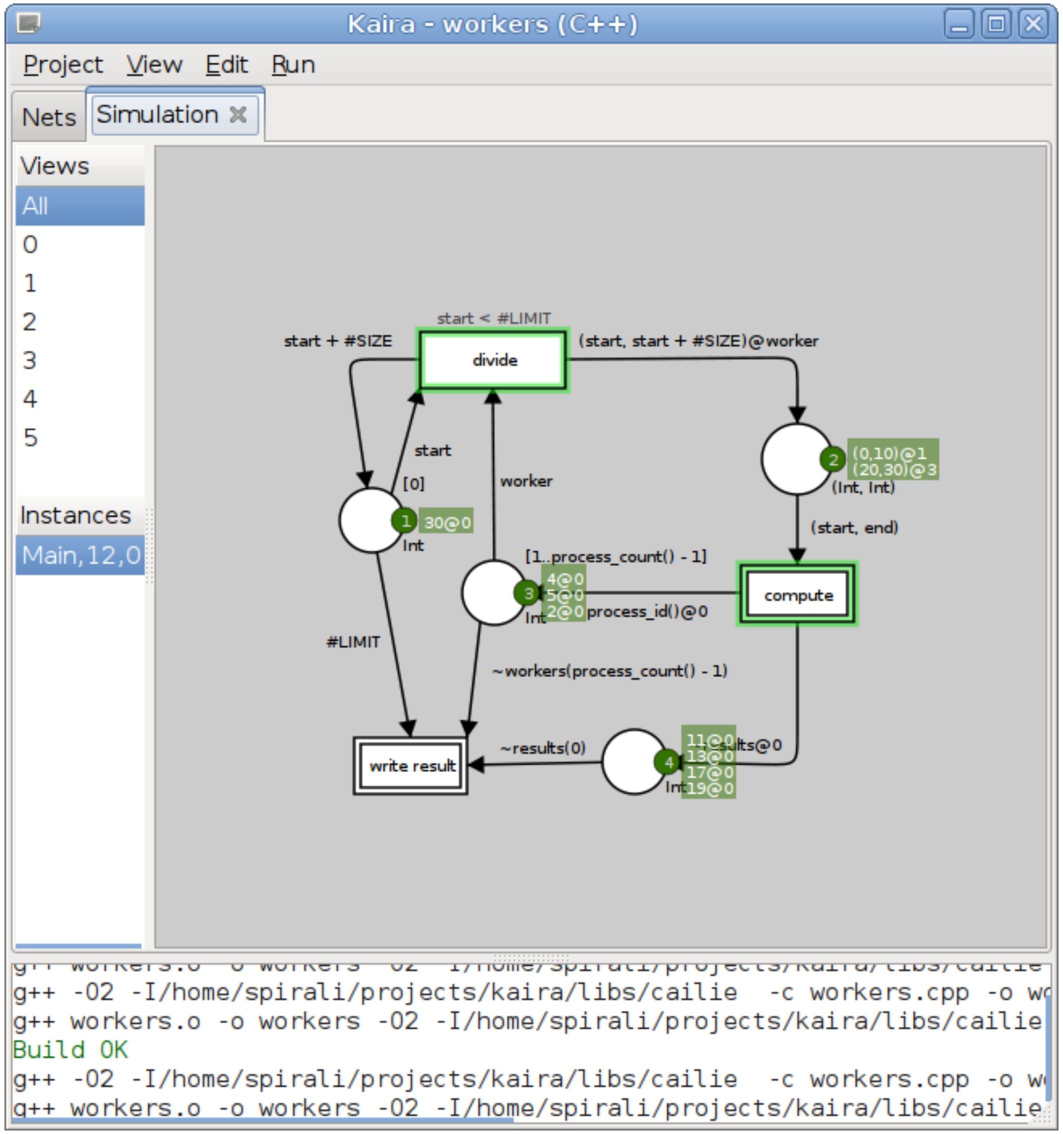
Visual model of parallel aspects and communication



```
struct Vars {
        int start;
        int worker;
};

void transition_fn(CaContext &ctx, Vars &var)
{
        printf("Hello world!\n");
}
```

- Visual simulations & debugging

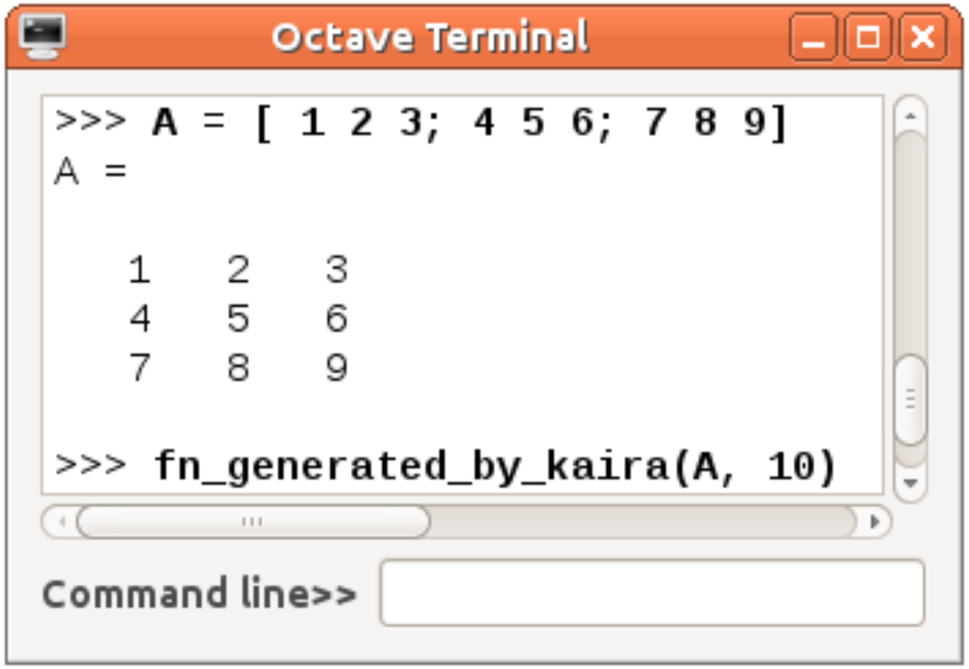


```
g++ workers.o -o workers -O2 -I/home/spirali/projects/kaira/libs/cailie
g++ -O2 -I/home/spirali/projects/kaira/libs/cailie  -c workers.cpp -o wo
g++ workers.o -o workers -O2 -I/home/spirali/projects/kaira/libs/cailie
Build OK
g++ -O2 -I/home/spirali/projects/kaira/libs/cailie  -c workers.cpp -o wo
g++ workers.o -o workers -O2 -I/home/spirali/projects/kaira/libs/cailie
```

- Visual simulations & debugging
- Running on parallel computers

```
user@bigcomputer  ~/project$ mpirun -np 64 ./project
```

- Visual simulations & debugging
- Running on parallel computers
- Modules for high level tools



```
>>> A = [ 1 2 3; 4 5 6; 7 8 9]
A =


   1   2   3
   4   5   6
   7   8   9

>>> fn_generated_by_kaira(A, 10)
```