Simulation Problems for One-Counter Machines*

Petr Jančar[1], Faron Moller[2], and Zdeněk Sawa[1]

[1] Technical University of Ostrava, Czech Republic
[2] Uppsala University, Sweden

Abstract. We consider decidability questions for simulation preorder (and equivalence) for processes generated by one-counter machines. We sketch a proof of decidability in the case when testing for zero is not possible, and demonstrate the undecidability in the general case.

1 Introduction

A ***one-counter machine*** is a nondeterministic finite-state automaton acting on a single counter variable ranging over the set $\mathbb{N}$ of nonnegative integers. Formally, it is a tuple $M = \langle Q, \Sigma, \delta^{=}, \delta^{>} \rangle$ where Q is a finite set of ***control states***, Σ is a finite set of ***actions***, and $\delta^{=}, \delta^{>} : Q \times \Sigma \to \mathcal{P}(Q \times \{-1, 0, 1\})$ are ***transition functions*** (where $\mathcal{P}(A)$ denotes the set of subsets of A). To M we associate the transition system $\langle \Gamma, \{\stackrel{a}{\to}\}_{a \in \Sigma} \rangle$, where $\Gamma = \{\, p(n) \mid p \in Q, n \in \mathbb{N} \,\}$ is the set of ***states*** and each $\stackrel{a}{\to} \subseteq \Gamma \times \Gamma$ is a binary relation defined as follows:

$$p(n) \stackrel{a}{\to} p'(n+i) \quad \text{iff} \quad \begin{cases} n = 0,\ i \geq 0,\ \text{and } (p', i) \in \delta^{=}(p, a); \quad \textit{or} \\ n > 0,\ \text{and } (p', i) \in \delta^{>}(p, a) \end{cases}$$

Note that any transition increments, decrements, or leaves unchanged the counter value; and a decrementing transition is only possible if the counter value is positive. Also observe that when $n > 0$ the transitions of $p(n)$ do not depend on the actual value of n.

M is ***deterministic*** iff for any state $p(n)$ and for any action $a \in \Sigma$ there is at most one state $p'(n')$ such that $p(n) \stackrel{a}{\to} p'(n')$. M is a ***weak one-counter machine*** iff $\delta^{=} = \delta^{>}$. Thus, a weak one-counter machine can test if its counter is nonzero (that is, it can perform certain transitions on the proviso that its counter is nonzero), but it cannot test if its counter is zero.

A binary relation $\mathcal{S}$ between the states of two (weak) one-counter machines is a ***simulation*** iff, given $\langle p(m), q(n) \rangle \in \mathcal{S}$ and $p(m) \stackrel{a}{\to} p'(m')$, we have $q(n) \stackrel{a}{\to} q'(n')$ with $\langle p'(m'), q'(n') \rangle \in \mathcal{S}$. $p(m)$ is ***simulated*** by $q(n)$, written $p(m) \preccurlyeq q(n)$, iff they are related by some simulation relation $\mathcal{S}$; $p(m)$ and $q(n)$ are ***simulation equivalent***, written $p(m) \simeq q(n)$, iff $p(m) \preccurlyeq q(n)$ and $q(n) \preccurlyeq p(m)$. If two states are related by a symmetric simulation relation, then they are ***bisimilar***.

* The first and third authors are partially supported by the Grant Agency of the Czech Republic, Grant No. 201/97/0456. The second author is partially supported by TFR grant No. 221-98-103.

J. Pavelka, G. Tel, M. Bartošek (Eds.): SOFSEM'99, LNCS 1725, pp. 404–413, 1999.

Such automata, and behavioural concepts like (bi)simulation equivalence, are enjoying renewed interest within the automata and process theory communities due to the present active search for the dividing line between decidable and undecidable theories for classes of infinite state systems (see, e.g., [4]). Recently, Abdulla and Čerāns [1] outlined an extensive and involved proof of the decidability of simulation preorder over weak one-counter machines. Their 16-page extended abstract is very technical and omits the proofs of most of the crucial lemmas. Here we outline a short proof based on simple and intuitive ideas about colouring the plane. (Due to space limitations and further results communicated here, we omit the proofs of some technical lemmas; for these, we refer to the following ten page report [3].)

We then show that simulation preorder between deterministic one-counter machines is undecidable; to do this we use a reduction from the halting problem for Minsky machines. Finally we demonstrate the undecidability of simulation equivalence in the general case; this contrasts with the decidability of bisimulation equivalence [2].

2 Decidability for Weak One-Counter Machines

For any pair of control states $\langle p,q\rangle \in Q_1 \times Q_2$ taken from two weak one-counter machines, we can ask for what values $m,n \in \mathbb{N}$ do we have $p(m) \preccurlyeq q(n)$. We can picture the "graphs" of the functions $\mathbb{G}_{\langle p,q\rangle} : \mathbb{N}\times\mathbb{N} \to \{\text{black},\text{white}\}$ given by

$$\mathbb{G}_{\langle p,q\rangle}(m,n) = \begin{cases} \text{black, if} & p(m) \preccurlyeq q(n); \\ \text{white, if} & p(m) \not\preccurlyeq q(n) \end{cases}$$

by appropriately colouring (black or white) the integral points in the first quadrant of the plane. Note that if $p(m) \preccurlyeq q(n)$ then $p(m') \preccurlyeq q(n')$ for all $m' \le m$ and $n' \ge n$; hence the black points are upwards- and leftwards-closed, and the white points are downwards- and rightwards-closed. For a fixed pair of states $p_0(m_0)$ and $q_0(n_0)$ of these weak one-counter machines, we decide the question "Is $p_0(m_0) \preccurlyeq q_0(n_0)$?" by effectively constructing an initial portion of the $|Q_1| \times |Q_2|$ graphs which includes the point $\langle m_0,n_0\rangle$, and look to the colour of $\mathbb{G}_{\langle p_0,q_0\rangle}(m_0,n_0)$.

Define the ***frontier function*** $f_{\langle p,q\rangle}(n) = \max\{m : \mathbb{G}_{\langle p,q\rangle}(m,n) = \text{black}\}$, that is, the greatest value m such that $p(m) \preccurlyeq q(n)$; $f_{\langle p,q\rangle}(n) = \infty$ if $\mathbb{G}_{\langle p,q\rangle}(m,n) = \text{black}$ for all m; and $f_{\langle p,q\rangle}(n) = -1$ if $\mathbb{G}_{\langle p,q\rangle}(0,n) = \text{white}$. This function is monotone nondecreasing, and the set of ***frontier points*** $\langle f_{\langle p,q\rangle}(n),n\rangle \in \mathbb{N}\times\mathbb{N}$ defines the ***frontier*** of $\mathbb{G}_{\langle p,q\rangle}$, the collection of the rightmost black points from each level. Slightly abusing notation, we use f to refer to the frontier function as well as the frontier given by the frontier function. The next theorem is the clue to our decidability result.

Belt Theorem

Every frontier lies in a linear belt with rational (or ∞) slope.

The proof of the Belt Theorem is outlined in Section 4; in the remainder of this section we describe the decision procedure which is based on this theorem. For each $k = 0, 1, 2, \ldots$, let $\mathbb{G}^k_{\langle p,q\rangle} : \mathbb{N} \times \mathbb{N} \to \{\text{black}, \text{white}\}$ be the maximally-black collection of colourings of the plane (with $\langle p, q\rangle$ ranging over $Q_1 \times Q_2$) which satisfies the following: whenever $m, n \leq k$ with $\mathbb{G}^k_{\langle p,q\rangle}(m, n) = \text{black}$, if $p(m) \xrightarrow{a} p'(m')$ then $q(n) \xrightarrow{a} q'(n')$ with $\mathbb{G}^k_{\langle p',q'\rangle}(m', n') = \text{black}$. Such a maximally-black collection of colourings exists since the collection of totally-white colourings satisfies this condition, as does the collection of colourings which colours a point black exactly when it is black in *some* collection of colourings satisfying the above condition; this final collection of colourings is the one we seek, and is effectively computable (it is black everywhere outside the initial $k \times k$ square). Note that $\mathbb{G}^k_{\langle p,q\rangle}(m, n) = \text{white}$ implies $\mathbb{G}^{k+1}_{\langle p,q\rangle}(m, n) = \text{white}$ and that $\mathbb{G}_{\langle p,q\rangle} = \lim_{k\to\infty} \mathbb{G}^k_{\langle p,q\rangle}$.

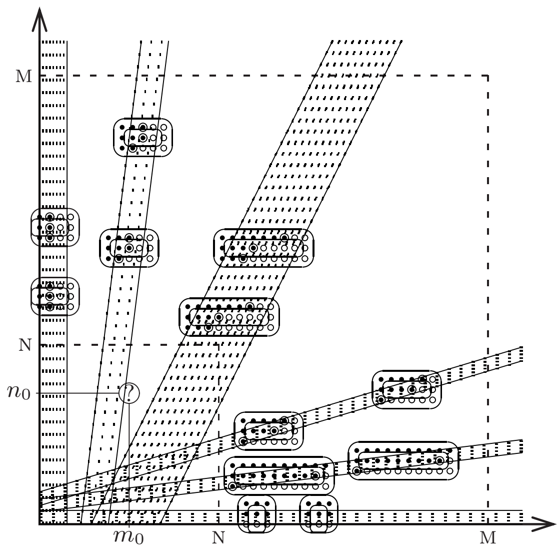


Fig. 1.

We can effectively construct the collections $\mathbb{G}^k_{\langle p,q\rangle}$ for each $k = 1, 2, 3, \ldots$. By the Belt Theorem, eventually for some k we must be able to lay down over the graphs $\mathbb{G}^k_{\langle p,q\rangle}$ a set of linear belts with rational slopes such that (see Fig. 1)

- there is an initial (M × M) square (M $< k$) inside of which each frontier lies within some belt (we may assume that parallel belts coincide, so that two or more frontiers may appear in the same belt);
- outside of some initial (N × N) square (N < M) containing the point $\langle m_0, n_0\rangle$, the belts are separated by gaps wide enough so that no point has neighbouring points in two belts;

– within the area bounded by the initial (N × N) and (M × M) squares, looking at each horizontal level within each belt (or each vertical level, in the case of a horizontal belt) we find a pattern which repeat itself—along with all of its neighbouring points—at two different levels. (That is, the colourings of the points and neighbouring points are the same in every graph on these levels within the belt.) Furthermore, the shift from one occurrence of the pattern to the next has a slope equal to that of the belt.

Note that these belts need not *a priori* be the true frontier belts specified in the Belt Theorem; but since (by the pigeonhole principle) the true frontier belts display such a repetitive pattern, the true frontier belts must eventually appear in the above fashion if no other belts appear earlier on in the construction.

Once we recognise such belts in the graphs $\mathbb{G}^k_{\langle p,q\rangle}$ (for some k), we can define graphs $\mathbb{G}'_{\langle p,q\rangle}$ by continuing the colouring of the graphs $\mathbb{G}^k_{\langle p,q\rangle}$ by periodically repeating the colouring within the belts between the levels at which the patterns repeat, and recolouring points to the right of the belts to maintain the invariant that white points are rightwards-closed.

We can readily confirm that the set of all pairs $\langle p(m), q(n)\rangle$ such that $\mathbb{G}'_{\langle p,q\rangle}(m,n)$ is black is a simulation. Thus, all black points are correct (that is, $\mathbb{G}_{\langle p,q\rangle}(m,n)$ is black whenever $\mathbb{G}'_{\langle p,q\rangle}(m,n)$ is black), and all white points within the initial (N × N) square are correct, proving that we have correctly constructed the initial (N × N) square.

3 Undecidability for One-Counter Machines

To show undecidability, we use a reduction from the halting problem for Minsky machines with 2 counters, which is well-known to be undecidable [5]; we can even suppose the input counter values to be zero. We use the following definition:

A ***Minsky machine*** C with two nonnegative counters c_1, c_2 is a program

$$1\colon \mathit{COMM}_1;\;\; 2\colon \mathit{COMM}_2;\;\; \cdots\;\; ;\;\; n\colon \mathit{COMM}_n$$

where COMM_n is a *halt*-command and COMM_i ($i = 1, 2, \ldots, n-1$) are commands of the following two types (assuming $1 \le k, k_1, k_2 \le n, 1 \le j \le 2$)

(1) $c_j := c_j + 1;\;$ *goto* k [*action* $\mathbf{i}_j$]
(2) *if* $c_j = 0$ *then goto* k_1 [*action* $\mathbf{z}_j$] *else* ($c_j := c_j - 1;\;$ *goto* k_2 [*action* $\mathbf{d}_j$])

Note that the computation of the machine C (starting with COMM_0, the counters initialized to 0) corresponds to a sequence of actions from $\{\mathbf{i}_1, \mathbf{i}_2, \mathbf{z}_1, \mathbf{z}_2, \mathbf{d}_1, \mathbf{d}_2\}$. This sequence is finite if the computation of C halts and infinite otherwise.

Theorem 1. *The problem if $p(0) \preccurlyeq q(0)$, where $p(0)$ and $q(0)$ are states of deterministic one-counter machines M_1 and M_2, is undecidable. (This holds even in restricted cases, where one of M_1 and M_2 is fixed.)*

Proof. Given a Minsky machine C with 2 counters c_1, c_2, we describe the construction of two deterministic one-counter machines M_1 and M_2, with specified control states p, q respectively, such that $p(0) \preccurlyeq q(0)$ iff C does not halt. In an obvious way, C can be transformed to a deterministic one-counter machine M_1 with n control states (n being the number of commands of C), with the set of actions $\{\mathbf{i}_1, \mathbf{i}_2, \mathbf{z}_1, \mathbf{z}_2, \mathbf{d}_1, \mathbf{d}_2\}$, and such that its counter 'behaves' like c_1 (actions $\mathbf{i}_1, \mathbf{z}_1, \mathbf{d}_1$ depend on and change it) while the actions $\mathbf{i}_2, \mathbf{z}_2, \mathbf{d}_2$ ignore the counter (c_2 is 'missing'). Thus a computation of M_1 can digress from that of C by performing an action $\mathbf{d}_2$ instead of $\mathbf{z}_2$ or vice versa. M_2 can be constructed similarly, now with the counter corresponding to c_2 while c_1 is ignored. Moreover, for each control state of M_2 with 'outgoing arcs' labelled $\mathbf{z}_2$ (enabled when the counter is zero) and $\mathbf{d}_2$ (enabled when the counter is positive) we add new 'complementary' arcs labelled $\mathbf{z}_2$ (for positive) and $\mathbf{d}_2$ (for zero) which lead to a special control state q_* which has a loop for any action (ignoring the counter). Note that M_2 remains deterministic. Finally we add a new outgoing arc to the halting control state of M_1, and we obviously have $p(0) \preccurlyeq q(0)$ iff C does not halt. □

Below we briefly describe two modifications of this construction which show that M_1 or M_2 can be fixed (not depending on C), even with just one or two control states respectively. Here δ represents both $\delta^=$ and $\delta^>$ (which are thus equal).

M_1 **fixed:** M_1 has only one state, $Q_1 = \{p\}$, M_2 has control states $Q_2 = \{q_1, q_2, \ldots, q_n, q_*\}$. The common alphabet is $\Sigma = \{\mathbf{i}_1, \mathbf{z}_1, \mathbf{d}_1, \mathbf{a}_2\}$ (we merge $\mathbf{i}_2, \mathbf{z}_2, \mathbf{d}_2$ into one symbol $\mathbf{a}_2$). The transition function in q_i depends on the type of the i-th command of C. δ_1 gives $\emptyset$ and δ_2 gives $\{(q_*, 0)\}$ in all cases not mentioned in the table.

Commands of C	**Transitions of M_1**	**Transitions of M_2**
i: $c_1 := c_1 + 1$; *goto* j	$\delta_1(p, \mathbf{i}_1) = \{(p, 1)\}$	$\delta_2(q_i, \mathbf{i}_1) = \{(q_j, 0)\}$
i: *if* $c_1 = 0$ *then goto* j *else* $c_1 := c_1 - 1$; *goto* k	$\delta_1^=(p, \mathbf{z}_1) = \{(p, 0)\}$ $\delta_1^>(p, \mathbf{d}_1) = \{(p, -1)\}$	$\delta_2(q_i, \mathbf{z}_1) = \{(q_j, 0)\}$ $\delta_2(q_i, \mathbf{d}_1) = \{(q_k, 0)\}$
i: $c_2 := c_2 + 1$; *goto* j	$\delta_1(p, \mathbf{a}_2) = \{(p, 0)\}$	$\delta_2(q_i, \mathbf{a}_2) = \{(q_j, 1)\}$
i: *if* $c_2 = 0$ *then goto* j *else* $c_2 := c_2 - 1$; *goto* k		$\delta_2^=(q_i, \mathbf{a}_2) = \{(q_j, 0)\}$ $\delta_2^>(q_i, \mathbf{a}_2) = \{(q_k, -1)\}$
i: *halt*		$\delta_2(q_i, \alpha) = \emptyset \quad (\forall \alpha \in \Sigma)$
		$\delta_2(q_*, \alpha) = \{(q_*, 0)\} \quad (\forall \alpha \in \Sigma)$

M_2 **fixed:** M_1 has control states $Q_1 = \{p_1, p_2, \ldots, p_n\}$, M_2 has control states $Q_2 = \{q_0, q_*\}$. The common alphabet is $\Sigma = \{\mathbf{a}_1, \mathbf{i}_2, \mathbf{z}_2, \mathbf{d}_2, \mathbf{h}\}$ (we merge $\mathbf{i}_1, \mathbf{z}_1, \mathbf{d}_1$ into one symbol $\mathbf{a}_1$, and added a new symbol $\mathbf{h}$). δ_1 gives $\emptyset$ in all cases not mentioned in the table.

Theorem 2. *The problem if $p(m) \simeq q(n)$, where $p(m)$ and $q(n)$ are states of two (nondeterministic) one-counter machines, is undecidable.*

Commands of C	**Transitions of M_1**	**Transitions of M_2**
$i\colon c_1 := c_1 + 1;$ *goto* j	$\delta_1(p_i, \mathbf{a}_1) = \{(p_j, 1)\}$	$\delta_2(q_0, \mathbf{a}_1) = \{(q_0, 0)\}$
$i\colon$ *if* $c_1 = 0$ *then goto* j *else* $c_1 := c_1 - 1;$ *goto* k	$\delta_1^{=}(p_i, \mathbf{a}_1) = \{(p_j, 0)\}$ $\delta_1^{>}(p_i, \mathbf{a}_1) = \{(p_k, -1)\}$	
$i\colon c_2 := c_2 + 1;$ *goto* j	$\delta_1(p_i, \mathbf{i}_2) = \{(p_j, 0)\}$	$\delta_2(q_0, \mathbf{i}_2) = \{(q_0, 1)\}$
$i\colon$ *if* $c_2 = 0$ *then goto* j *else* $c_2 := c_2 - 1;$ *goto* k	$\delta_1(p_i, \mathbf{z}_2) = \{(p_j, 0)\}$ $\delta_1(p_i, \mathbf{d}_2) = \{(p_k, 0)\}$	$\delta_2^{=}(q_0, \mathbf{z}_2) = \{(q_0, 0)\}$ $\delta_2^{>}(q_0, \mathbf{z}_2) = \{(q_*, 0)\}$ $\delta_2^{>}(q_0, \mathbf{d}_2) = \{(q_0, -1)\}$ $\delta_2^{=}(q_0, \mathbf{d}_2) = \{(q_*, 0)\}$
$i\colon$ *halt*	$\delta_1(p_i, \mathbf{h}) = \{(p_i, 0)\}$	$\delta_2(q_0, \mathbf{h}) = \emptyset$
		$\delta_2(q_*, \alpha) = \{(q_*, 0)\} \quad (\forall \alpha \in \Sigma)$

Proof. For control states p and q, taken from M_1 and M_2 respectively, we give a construction of M_1' with r_1 and M_2' with r_2 so that $p(0) \preccurlyeq q(0)$ iff $r_1(0) \simeq r_2(0)$; recalling the previous theorem we shall be done.

We take M_1' to be the disjoint union of M_1 and M_2, adding a new control state r_1 and putting $\delta_1'(r, a) = \{(p, 0), (q, 0)\}$ (a is an arbitrary symbol; here is the only use of nondeterminism when M_1 and M_2 are deterministic). M_2' arises from M_2 by adding a new control state r_2 and putting $\delta_2'(r_2, a) = \{(q, 0)\}$.

It is easily seen that $r_2(0) \preccurlyeq r_1(0)$, and that $r_1(0) \preccurlyeq r_2(0)$ iff $p(0) \preccurlyeq q(0)$. $\square$

Remark. Decidability of simulation equivalence for deterministic one-counter machines follows from decidability of equality for one-counter languages [6]. Undecidability of simulation preorder is slightly stronger than the well-known fact that inclusion for deterministic context-free languages is undecidable.

4 Proof of the Belt Theorem

By an ***area*** we mean a set $A \subseteq \mathbb{N} \times \mathbb{N}$. We define its ***interior*** and ***border*** as follows.

$$\begin{aligned} \mathbf{interior}(A) &= \Big\{ \langle m, n\rangle \ : \ \{m-1, m, m+1\} \times \{n-1, n, n+1\} \ \subseteq \ A \Big\}; \\ \mathbf{border}(A) &= A - \mathbf{interior}(A). \end{aligned}$$

Given an area A and a vector $v \in \mathbb{Z} \times \mathbb{Z}$ (where $\mathbb{Z}$ denotes the set of integers), we let $\mathbf{shift}(A, v) = (A + v) \cap (\mathbb{N} \times \mathbb{N})$ denote the area A shifted by vector v. We say that the shift of an area A by a vector v is ***safe*** wrt $B \subseteq \mathbf{shift}(A, v)$ iff for all graphs $\mathbb{G}_{\langle p,q\rangle}$ and all $u \in B$ we have that $\mathbb{G}_{\langle p,q\rangle}(u)$ is black whenever $\mathbb{G}_{\langle p,q\rangle}(u - v)$ is black. We say that such a shift is ***safe*** iff it is safe wrt $\mathbf{shift}(A, v)$, that is, if it never shifts a black point to a white point.

By a ***line*** ℓ we mean a line with a finite rational slope $\beta > 0$; however, we occasionally refer explicitly to horizontal or vertical lines. We also view a line as a function, writing $\ell(y)$ to represent the value x such that the point $\langle x, y\rangle$ is on the line. We often refer to areas determined by a horizontal line at level $b \in \mathbb{N}$ and one or two lines. For this, we use the following notation: $A[b, \overrightarrow{\ell}, \overleftarrow{\ell'}\,]$ denotes the set of all points of $\mathbb{N}\times\mathbb{N}$ which lie on or above level b, on or to the right of ℓ, and on or to the left of ℓ'. We omit b when $b=0$. Finally, by a ***belt*** we mean the set of points on or between two parallel lines; here we also allow horizontal and vertical lines. Thus we may have a horizontal belt, or a vertical belt, or a belt of the form $A[\overrightarrow{\ell}, \overleftarrow{\ell'}]$ where ℓ and ℓ' are parallel lines with ℓ' to the right of ℓ.

We can partition the frontiers according to whether or not they lie in a horizontal or a vertical belt. To this end we make the following definitions.

(i) **HF** is the set of frontiers f such that $f(n)=\infty$ for some $n\in\mathbb{N}$. We let $\textsc{hb}\in\mathbb{N}$ (the "horizontal bound") be the least value such that $f(\textsc{hb})=\infty$ for all $f\in$ **HF**. The frontiers of **HF** are those which lie in a horizontal belt.
(ii) **VF** is the set of frontiers f such that $\lim_{n\to\infty} f(n)<\infty$. We let $\textsc{vb}\in\mathbb{N}$ (the "vertical bound") be the least value such that $f(n)<\textsc{vb}$ for all $f\in$ **VF** and all $n\in\mathbb{N}$. The frontiers of **VF** are those which lie in a vertical belt.
(iii) **IF** is the set of interior frontiers, those not appearing in **HF** nor in **VF**.

We now formalize the notion of a line *separating* frontiers. For this, we need the following notions. We refer to a (horizontal) shift of a line ℓ by an amount $i\in\mathbb{Z}$ by $\mathbf{shift}(\ell,i)$; this is the line ℓ' such that $\ell'(y)=\ell(y)+i$. Given $\beta>0$, we let $\mathbf{step}(\beta)\in\mathbb{N}$ be the least integral horizontal distance which two lines with slope β must be separated so as to fit a unit square between them; this ensures that, given two such lines ℓ and $\ell'=\mathbf{shift}(\ell,\mathbf{step}(\beta))$ we have $A[\overleftarrow{\ell}]\cap\mathbf{interior}(\mathbb{N}\times\mathbb{N})\subseteq\mathbf{interior}(A[\overleftarrow{\ell'}])$. Note that $\mathbf{step}(\alpha)\le\mathbf{step}(\beta)$ whenever $\alpha\ge\beta$.

Definition 1. *A line ℓ with rational slope $\beta>0$* ***separates frontiers above level*** *$b\in\mathbb{N}$ iff:*

(i) *for all $f\in$ **HF**, $f(b)=\infty$; that is, $b\ge\textsc{hb}$;*
(ii) *for all f, if $f(b)=-1$ then $f(n)=-1$ for all n;*
(iii) *for all $f\in$ **IF**, $f(b)>\textsc{vb}$;*
(iv) *for all f, if $f(b)\le\ell(b)$ then $f(n)<\ell(n)-\mathbf{step}(\beta)$ for all $n\ge b$ (in which case we call f an ℓ-**left frontier**);*
(v) *for all f, if $f(b)\ge\ell(b)$ then $f(n)>\ell(n)+\mathbf{step}(\beta)$ for all $n\ge b$ (in which case we call f an ℓ-**right frontier**).*

Thus the ℓ-left and ℓ-right frontiers are separated by a belt with (horizontal) width $2\cdot\mathbf{step}(\beta)$ centered on the line ℓ. We say simply that a line ***separates frontiers*** *if it separates frontiers above some level.*

The next Lemma shows that there always exists such a separating line.

Lemma 1. *There is a line ℓ (with rational slope $\beta>0$) which separates frontiers, in which the ℓ-right frontiers are exactly those of* **HF**.

We now outline the proof of our Belt Theorem.

Proof of The Belt Theorem: Suppose we have a line ℓ with rational slope β which separates frontiers above level b in such a way that ℓ-right frontiers lie in belts and their number cannot be increased by choosing a different ℓ. That such a separating line exists is ensured by Lemma 1.

Let $\mathcal{L}$ be the set of ℓ-left frontiers, and suppose for the sake of contradiction that $\mathcal{L} - \textsf{VF} \neq \emptyset$ (otherwise we have nothing to prove).

For any $n \geq b$, let $\mathbf{gap_1}(n)$ be the (horizontal) distance from ℓ to the rightmost ℓ-left frontier point on level n; that is, $\mathbf{gap_1}(n) = \min\{\ell(n) - f(n) : f \in \mathcal{L}\}$. Note that, since β is rational, the fractional part of $\mathbf{gap_1}(n)$ ranges over a finite set. Hence we cannot have an infinite sequence of levels $i_1, i_2, i_3, \ldots$ above b such that $\mathbf{gap_1}(i_1) > \mathbf{gap_1}(i_2) > \mathbf{gap_1}(i_3) > \cdots$. We can thus take an infinite sequence $i_1 < i_2 < i_3 < \cdots$ of levels above b such that

1. $\mathbf{gap_1}(i) \leq \mathbf{gap_1}(n)$ for all $i \in \{i_1, i_2, i_3, \ldots\}$ and all $n \geq i$;
2. either $\mathbf{gap_1}(i_1) = \mathbf{gap_1}(i_2) = \mathbf{gap_1}(i_3) = \cdots$
   or $\mathbf{gap_1}(i_1) < \mathbf{gap_1}(i_2) < \mathbf{gap_1}(i_3) < \cdots$;
3. for some fixed ℓ-left frontier $f_{\max} \in \mathcal{L}$: $\mathbf{gap_1}(i) = \ell(i) - f_{\max}(i)$ for all $i \in \{i_1, i_2, i_3, \ldots\}$.

The above conditions can be satisfied by starting with the infinite sequence $b+1, b+2, b+3, \ldots$, and first extracting an infinite subsequence which satisfies the first condition, then extracting from this a further infinite subsequence which satisfies (also) the second condition, and then extracting from this a further infinite subsequence which satisfies (also) the third condition.

For $i \in \{i_1, i_2, i_3, \ldots\}$, we let $\mathbf{offset}_i : \mathcal{L} \to \mathbb{N}$ be defined by $\mathbf{offset}_i(f) = f_{\max}(i) - f(i)$. We can then assume that our infinite sequence further satisfies the following condition.

4. For each $f \in \mathcal{L}$: either $\mathbf{offset}_{i_1}(f) = \mathbf{offset}_{i_2}(f) = \mathbf{offset}_{i_3}(f) = \cdots$
   or $\mathbf{offset}_{i_1}(f) < \mathbf{offset}_{i_2}(f) < \mathbf{offset}_{i_3}(f) < \cdots$.
   In the first case, we call f a ***fixed-offset frontier***; and in the second case, we call f an ***increasing-offset frontier***.

This condition can be satisfied by repeatedly extracting an infinite subsequence to satisfy the condition for each $f \in \mathcal{L}$ in turn. Finally, we assume our sequence satisfies the following condition.

5. We have a maximal number of fixed-offset frontiers; no other sequence satisfying conditions 1–4 can have more ℓ-left frontiers $f \in \mathcal{L}$ with $\mathbf{offset}_{i_1}(f) = \mathbf{offset}_{i_2}(f) = \cdots$.

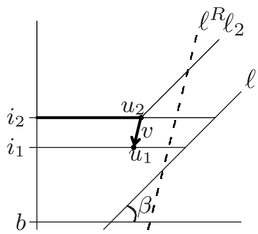
For technical reasons, we also suppose the next two conditions which can be satisfied by dropping some number of initial levels (that is, sequence elements).

6. $\mathbf{gap_2}(i_1) > |\mathcal{L}| \cdot \mathbf{step}(\beta)$, where $\mathbf{gap_2}(i_j)$ is defined as

$$\min\{\mathbf{offset}_{i_j}(f) : f \text{ is an increasing-offset frontier}\} \\ -\ \max\{\mathbf{offset}_{i_j}(f) : f \text{ is a fixed-offset frontier}\}$$

7. $f_{\max}(i_1) < f_{\max}(i_2)$.

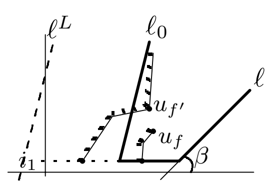
The line going through the points $u_1 = \langle f_{\max}(i_1), i_1\rangle$ and $u_2 = \langle f_{\max}(i_2), i_2\rangle$ has some slope $\alpha \geq \beta$. If we let **left-of**(u_2) denote the set of points consisting of u_2 along with all points to its left (that is, all (m, i_2) with $m \leq f_{\max}(i_2)$) then the shift of **left-of**(u_2) by $v = u_1 - u_2$ is safe: for the shift of the point onto the y-axis, this is assured by condition (ii) of Definition 1; and for the remaining points, this is assured since frontier offsets cannot shrink (condition 4 above). We can thus invoke the following.



Right Lemma *Consider a line ℓ with slope β separating frontiers above level b, and take two points $u_1 = \langle m_1, i_1\rangle$ and $u_2 = \langle m_2, i_2\rangle$ in $A[b, \overleftarrow{\ell}\,]$ with $m_1 < m_2$ and $b < i_1 < i_2$ such that the slope α of the vector $v = u_1 - u_2$ is at least β. Let ℓ_2 be the line parallel to ℓ which goes through u_2, and suppose that all ℓ-left frontier points in $A[i_2]$ are in $A[\overleftarrow{\ell_2}]$, and that the shift of* **left-of**(u_2) *by v is safe. Then there is a line ℓ^R with slope α separating the same frontiers as ℓ does above level i_2.*

This gives us a line ℓ^R with slope α separating the same frontiers as ℓ above i_2.

Now, there must then be a line ℓ_0 with slope α to the left of ℓ above level i_1 such that every fixed-offset frontier appears in (that is, intersects with) $A = A[i_1, \overrightarrow{\ell_0}, \overleftarrow{\ell}\,]$, and such that whenever a frontier f appears in A there is a frontier point $u_f = \langle f(n), n\rangle \in$ **interior**(A) such that $f(n) - \ell_0(n) \geq f(i_1) - \ell_0(i_1)$; that is, u_f is at least as far to the right of ℓ_0 as the frontier point of f on level i_1. (We can first consider ℓ_0 to be the line going through the frontier points at levels i_1 and i_2 of the fixed-offset frontier with the greatest offset value; if some frontier f is on the border but not in the interior of A, then we can instead take ℓ_0 to be the shift of this line by $-$**step**(α). If there is now some other frontier which is on the border but not in the interior of A, then we again shift the line by $-$**step**(α). We need only shift (at most) once for each frontier in $\mathcal{L}$ before being guaranteed to arrive at a suitable choice for ℓ_0, so we shift by at most $|\mathcal{L}| \cdot$ **step**$(\alpha) \leq |\mathcal{L}| \cdot$ **step**(β), and hence by condition 5 we don't reach the increasing-offset frontiers on level i_1.) We may then invoke the following.



Left Lemma *Suppose we have a line ℓ with rational slope β separating frontiers above level i_1, and a line ℓ_0 with rational slope $\alpha \geq \beta$ to the left of ℓ above level i_1. Suppose further that whenever a frontier f appears in $A = A[i_1, \overrightarrow{\ell_0}, \overleftarrow{\ell}\,]$, there is a frontier point $u_f = \langle f(n), n\rangle \in$* **interior**$(A)$ *such that $f(n) - \ell_0(n) \geq f(i_1) - \ell_0(= i_1)$. Then there is a line ℓ^L to the left of ℓ_0 with slope α such that $f \subseteq A[\overrightarrow{\ell^L}]$ for each such frontier f.*

The premise of this is thus satisfied, so all frontiers with frontier points in A are in $A[\overrightarrow{\ell^L}]$ for some line ℓ^L with slope α. Hence they are in the belt

$A[\overrightarrow{\ell^L}, \overleftarrow{\ell^R}]$ above i_2; and in fact only the fixed-offset frontiers can (and do) have frontier points in $A = A[i_1, \overrightarrow{\ell_0}, \overleftarrow{\ell}]$, for otherwise they would not correspond to increasing-offset frontiers.

It remains to demonstrate that we can choose ℓ^L so that it separates frontiers. This can only fail if an increasing-offset frontier appears infinitely often in $A[\overrightarrow{\ell'}]$ where $\ell' = \mathbf{shift}(\ell^L, -2 \cdot \mathbf{step}(\alpha))$. But then there would be two levels i_{j_1} and i_{j_2} where $f_{\max}(i_{j_1}) - \ell^L(i_{j_1}) = f_{\max}(i_{j_2}) - \ell^L(i_{j_2}) = d$ and $\mathbf{gap_2}(i_{j_1}) > d + |\mathcal{L}| \cdot \mathbf{step}(\alpha)$. We could then find a contradiction using Left Lemma, by considering now the area $A[i_{j_1}, \overrightarrow{\ell^L}, \overleftarrow{\ell^R}]$. □

References

1. Abdulla, P., K. Čerāns (1998). Simulation is decidable for one-counter nets (Extended Abstract.) In Proceedings of CONCUR'98, *Lecture Notes in Computer Science* **1466**:253–268. 405
2. Jančar, P. (1997). Decidability of bisimilarity for one-counter processes. In Proceedings of ICALP'97, *Lecture Notes in Computer Science* **1256**:549–559. (Revised version to appear in *The Journal of Information and Computation*.) 405
3. Jančar, P., F. Moller (1999). Simulation on one-counter nets via colouring. Uppsala Univ. CSD Report No. 159, February 1999. http://www.csd.uu.se/papers/reports.html 405
4. Moller, F. (1996). Infinite results. Proceedings of CONCUR'96, *Lecture Notes in Computer Science* **1119**:195–216. 405
5. Minsky, M. (1967). Computation: Finite and Infinite Machines. Prentice-Hall, Englewood Cliffs, NJ. 407
6. Valiant, L.G., M.S. Paterson (1975). Deterministic one-counter automata. *Journal of Computer and System Science* **10**:340–350. 409