Infinitary Queries and Their Asymptotic Probabilities I: Properties Definable in Transitive Closure Logic

appeared in: *E. Börger et al. eds., Proc. CSL'91, Springer Lecture Notes in Comp. Sci. 626, pp. 396–410*

Jerzy Tyszkiewicz*

Institute of Informatics, University of Warsaw,
ul. Banacha 2, 02-097 Warszawa, Poland.
e-mail jurekty@mimuw.edu.pl

Abstract

We present new general method for proving that for certain classes of finite structures the limit law fails for properties expressible in transitive closure logic. In all such cases also all associated asymptotic problems are undecidable.

1 Introduction

The problems considered in this paper belong to the research area called *random structure theory,* and, more specifically, to its logical aspect.
To explain (very imprecisely and incompletely) what does it mean, let us imagine that we have a class of some structures (say: finite ones over some fixed signature), equipped with a probability space structure (this probability is usually assumed to be only *finitely* additive). Then we draw one structure at random and ask:
- what does the drawn structure look like?
- does the drawn structure have some particular property?

Those questions are typical in random structure theory. To turn to the logical part of it, look at the drawn structure through logical glasses: we can only notice properties definable in some particular logic. Then new questions become natural:
- does every property we can observe have a probability (is it measurable)?
- if so, what is this probability?
- can we compute this probability, and, eventually, how difficult is to do that?

It becomes clear from the above that the random structure theory is tightly connected with combinatorics, finite model theory, mathematical logic, and, last but not least, computer science. A much more complete exposition of the logical part of the random structure theory may be found in a nice survey [2].
The present paper is devoted to study of asymptotic probabilities of these properties of finite structures, which are definable in *transitive closure logic* (denoted *TC* in

*This research was completed in the Institute of Mathematics, University of Warsaw.

the sequel). The results we obtain provide a contribution to problem 8.2 in [2], mentioned above:

> *Develop techniques for showing the existence of asymptotic probabilities of* least fixed point logic[1] *sentences in slow growing classes.*

As a matter of fact, we actually deal with the other side of the coin, and present general conditions under which asymptotic probabilities of TC sentences fail to exist. In a sequel to the present paper [14] we prove results concerning similarly stated problem for properties definable in least fixed point logic. In [11] authors prove similar results for the infinitary logic $L^{\omega}_{\infty\omega}$.

2 Probability, Logic and Semi–Forests

2.1 Probability

Let $\boldsymbol{\mathcal{A}}$ be an arbitrary class of finite structures over (fixed) signature τ with equality. Let $\boldsymbol{\mathcal{A}}(n) \subseteq \boldsymbol{\mathcal{A}}$ be a subclass of $\boldsymbol{\mathcal{A}}$ containing all structures with carrier set $n = \{0, \ldots, n-1\}$, for $n \in \omega$. Let $\mu_n$be a probability distribution on $\boldsymbol{\mathcal{A}}(n)$. Then for any property φ of some structures in $\boldsymbol{\mathcal{A}}$ (often defined by (and identified with) a sentence of some logic over τ) we can define

$$\mu_n(\varphi) = \mu_n(\{\mathcal{A} \in \boldsymbol{\mathcal{A}}(n) \,|\, \mathcal{A} \models \varphi\}).$$

Examples of often considered probabilities are the *uniform labelled* probabilities (i.e. each structure in $\boldsymbol{\mathcal{A}}(n)$ is weighted equally), and *uniform unlabelled* probabilities (i.e. each isomorphism class in $\boldsymbol{\mathcal{A}}(n)$ is weighted equally).

Writing $\boldsymbol{\mu}$ for $\{\mu_n\}_{n\in\omega}$ we may consider $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mu} \rangle$ as *randomized class of finite structures* and make it an object of our study.

All results in this paper hold for any such randomized class, unless otherwise stated. We now fix a randomized class $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mu} \rangle$. All notions we define should be understood to refer to this fixed $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mu} \rangle$.

We are interested in asymptotic properties of $\mu_n(\varphi)$, and especially whether the limit $\mu(\varphi) = \lim_{n\to\infty} \mu_n(\varphi)$ exists, for φ being a sentence of the logic under consideration. If it exists, we call it an *asymptotic probability of* φ. If this is the case for every sentence of the logic L, we say that the *convergence law* holds (for L and $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mu} \rangle$). If, in addition, every sentence has probability either 0 or 1, we say that the *0–1 law* holds.

By an *asymptotic problem* (for L and $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mu} \rangle$) we mean any set of sentences of L of the form $\{\varphi \in L \mid \mu(\varphi) \geq r\}$, where r is rational, $0 < r \leq 1$. We are interested in complexities of asymptotic problems, and especially in the complexity of the *almost sure theory,* i.e. the set of all L sentences of asymptotic probability one. In this paper we consider only the decidable/undecidable distinction between complexities. Sometimes, if $\mu(\varphi) = 1$, we say that φ *holds almost surely* (we often use this convention when φ is described in an informal way).

We also say that sequence $\{a_n\}_{n\in\omega}$ *converges recursively* to limit $0 < a < \infty$ iff there exists a recursive function $g : \omega \to \omega$ such that for every positive $k \in \omega$, and

[1] Transitive closure logic is a sublogic of the least fixed point logic (J.T.).

every $n > g(k)$ we have $|a_n - a| < 1/k$. If $a = \infty$, we change the last condition to: for every positive $k \in \omega$, and for every $n > g(k)$ we have $a_n > k$.

2.2 Logic

We replace TC with a logical formalism of the same expressive power, but much easier to deal with. This formalism is a *Dynamic Logic of While Programs with Rich Tests and Nondeterministic Assignments* [12]. A result stating equivalence of these formalisms is proved in [7]. Our choice of the formalism is motivated by simpler verification of *dynamic* formulas, which results in argumentation which is more clear and easier to pursue.
We simultaneously define the sets of *programs* and *formulae.* For the set of formulae we use symbol DL.
First we define the class of *instructions* as the least class of expressions containing

nondeterministic assignment

$$x_i := y \texttt{ such that } \varphi(y, \vec{x})$$

for all φ in DL; intended meaning of such an assignment is that just after it is executed register x_i contains nondeterministically chosen element of the universe satisfying φ with parameters – the values of registers appearing in $\vec{x}$. If there is no such an element, then this instruction loops forever. Variable y is *bounded* in this instruction.

halt instruction *halt*

abort instruction *abort*

and closed under the following formation rules (we assume S_1, S_2 to be instructions and φ to be a formula in DL, braces are *not* parts of the compound instructions):

$$\textbf{conditional instruction} \begin{cases} \texttt{if } \varphi(\vec{x}) \\ \texttt{then} \\ S_1 \\ \texttt{else} \\ S_2 \\ \texttt{fi} \end{cases}$$

$$\textbf{while loop} \begin{cases} \texttt{while } \varphi(\vec{x}) \\ \texttt{do} \\ S_1 \\ \texttt{od} \end{cases}$$

$$\textbf{sequential composition} \begin{cases} S_1 \\ S_2 \end{cases}$$

Program is just an instruction with the last line *halt* and with all lines numbered with consecutive natural numbers, starting at the top. Free variables of a program we define in standard manner.

The class of formulae is the least class of expressions containing all first order formulae over τ, *halting formulae* $\langle Q \rangle true$ for all programs Q (intended meaning of such formula is *there exists a terminating computation of Q, starting from actual valuation,*) closed under classical propositional connectives ($\neg$, $\wedge$, $\vee$, $\Rightarrow, \Leftrightarrow$), and first order quantification ($\forall$, $\exists$). Free variables of a formula we define in standard way, except that free variables of a formula $\langle Q \rangle true$ are all free variables of Q.
Semantics of programs and formulae in a τ-structure $\mathcal{A}$ is intuitively clear, and thus is moved to appendix A. The only thing we mention here is that one program may have many different computations in $\mathcal{A}, \vec{x} : \vec{a}$.
Definition of syntax of programs is oriented on formal semantics definition. In the sequel we will use more reader–oriented syntax dropping line numbering, with reading simplifying display and with no restrictions on names of registers (variables). The *halt* instructions in last lines are dropped, too. Sometimes we neglect initial values of these variables which are irrelevant to computation of a program; e.g. initial value of z in program from proof of lemma 3 does not play any role in the computation. We adopt also one syntactic abbreviation concerning programs: $x_i := x_j$ is to be understood as $x_i := y$ `such that` $y = x_j$.

2.3 Semi–Forests

Let τ_0 be a signature consisting exactly one binary relation symbol r and equality sign =. In what follows we denote by DL_0 the logic $DL(\tau_0)$.
Let $A \subseteq \omega$ be a finite set and let $R \subseteq A \times A$ be a binary relation. Of course $\mathcal{A} = \langle A, R \rangle$ may be considered as a τ_0-structure (directed graph, in other words).
We call such a τ_0-structure $\mathcal{A}$ a *semi–forest* if for every $a, b \in A$ there is at most one R-path (R-cycle if $a = b$, respectively) with no repeating vertices, between a and b. Natural examples of semi–forests are forests themselves (in directed version), as well as unary functions, permutations and many other.
Now let $\mathcal{A} = \langle A, R \rangle$ be a semi–forest, and let $a, b \in A$. Then we can naturally define an *ordered interval* $[\![a, b]\!] \subseteq A$, consisting of all elements subsequently appearing on the unique path from a to b. (It may happen that $[\![a, b]\!] = \emptyset$ or $[\![a, a]\!] \supsetneq \{a\}$.)
The *height* $h(\mathcal{A})$ of a semi–forest $\mathcal{A}$ is a maximal length of such an interval.
The following two technical lemmas will be the only facts about semi–forests we will need in further investigations:

Lemma 1 *For any two semi–forests $\mathcal{A}_1$ and $\mathcal{A}_2$ their disjoint union $\mathcal{A}_1 \sqcup \mathcal{A}_2$ is a semi–forest, too, and* $h(\mathcal{A}_1 \sqcup \mathcal{A}_2) = \max\{h(\mathcal{A}_1), h(\mathcal{A}_2)\}$.

Proof: Immediate. □

Lemma 2 (due to anonymous referee of the paper) *Let $\mathcal{A} = \langle A, R \rangle$ be any semi–forest. Let $a, b \in A$, be such that $[\![a, b]\!] \neq \emptyset$. Then $c \in [\![a, b]\!]$ iff there is no R-path from a to b on which c does not appear.*

Proof: Suppose that $c \in [\![a, b]\!]$, and let p be an arbitrary R-path from a to b in a semi–forest $\mathcal{A}$. Then after cutting off all repetitions p becomes $[\![a, b]\!]$, so c was present on p.
To prove the converse observe that $[\![a, b]\!]$ is an R-path from a to b, so c appears on it. □

3 Expressive Power of DL_0 Over Semi–Forests

Lemma 3 *There exists a DL formula $Path(x,y)$ such that for any τ–structure $\mathcal{A}$*

$$\mathcal{A}, x\!:\!a, y\!:\!b \models Path \qquad \text{iff} \qquad \text{there is an } R\text{-path from } a \text{ to } b.$$

Proof: Take

$$Path \equiv \langle z := x \ \mathtt{while}\ z \neq y \ \mathtt{do}\ \ z := w \ \mathtt{such\ that}\ r(z,w)\ \mathtt{od}\rangle true.$$

□

Lemma 4 *There is DL_0 formula $Memb(x;t,u)$ such that for any semi–forest $\mathcal{A}$:*

$$\mathcal{A}, x\!:\!c, t\!:\!d, u\!:\!e \models Memb \qquad \text{iff} \qquad c \in [\![d,e]\!].$$

Proof: Let $Q_1(y;t,u)$ be the following program:

$$v := t;\ \mathtt{while}\ \ v \neq u\ \mathtt{do}\ \ v := w\ \ \mathtt{such\ that}\ \ (R(v,w) \wedge w \neq y)\ \mathtt{od}.$$

Then let $Memb(x;t,u)$ be $\langle Q\rangle true$ with

$$Q \equiv v := t;\ \mathtt{while}\ \ v \neq u\ \mathtt{do}\ \ v := y\ \ \mathtt{such\ that}\ \ (R(v,y) \wedge \neg\langle Q_1\rangle true)\ \mathtt{od}.$$

Verification that this is a proper choice is based on lemma 2, and is left for the reader. □

We immediately infer the following:

Lemma 5 *There is a DL_0 formula $Succ(x,y;t,u)$ such that for any semi–forest $\mathcal{A}$*

$$\mathcal{A}, x\!:\!a, y\!:\!b, t\!:\!d, u\!:\!e \models Succ \ \ \text{iff}\ \ a \in [\![d,e]\!] \text{ is an immediate successor of } a \in [\![d,e]\!].$$

Lemma 6 *There exist DL_0 formulae $Nat(x;t,u)$ and $Zero(x;t,u)$ such that for any semi–forest $\mathcal{A}$ and any $d,e \in A$ the set $I_{de} = \{a \in M \mid \mathcal{A}, x\!:\!a, t\!:\!d, u\!:\!e \models Nat\}$ is either empty or is an interval $[\![d,e]\!]$ of length $h = h(\mathcal{A})$, and, moreover, then*

$$\langle I_{de},\ Zero(x;d,e), Succ(x,y;d,e)\rangle \simeq \langle [0,h-1]\ ,0\ ,succ\rangle.$$

Proof: Consider a program $Q(p,q,t,u)$

```
if ¬(∃z Memb(z;t,u) ∧ ∃z Memb(z;p,q))
then
    abort
else
    w := p;  s := t
    while w ≠ q ∧ s ≠ u
    do
        w := v such that Succ(v,w;p,q);
        s := v such that Succ(v,s;t,u)
    od
    if s ≠ u then  abort else halt fi
fi
```

It is not hard to see that Q has a terminating computation in $\mathcal{A}, p:a, q:b, t:d, u:e$ iff the length of $[\![d,e]\!]$ is no less than the length of $[\![a,b]\!]$, and both intervals are nonempty. It follows that we can take

$$Nat(x;t,u) \equiv (\forall pq\ (\neg\exists z\ Memb(z;p,q) \vee \langle Q\rangle true)) \wedge Memb(x;t,u)$$

$$Zero(x;t,u) \equiv Nat(x;t,u) \wedge x = t.$$

$\square$

The last tool we may (eventually) need is the following proposition, useful when semi–forest structure is defined by some formula, possibly with parameters.

Proposition 1 *Suppose that for a finite τ–structure $\mathcal{A} = \langle n, \vec{R}\rangle$ and a DL formula $\varphi(\vec{x}, \vec{y}; \vec{z})$ with $\vec{x}$ and $\vec{y}$ of length k and $\vec{z}$ of length m, for every choice of $\vec{c} \in n^m$ to be the value of $\vec{z}$, the structure*

$$\mathcal{A}_{\varphi,\vec{z}:\vec{c}} = \langle n^k, \varphi^{\mathcal{A}}(\vec{x}, \vec{y}; \vec{c})\rangle = \langle n^k, \{(\vec{a}, \vec{b}) \in n^k \times n^k \mid \mathcal{A}, \vec{x}:\vec{a}, \vec{y}:\vec{b}, \vec{z}:\vec{c} \models \varphi\}\rangle$$

is a semi–forest of height $h(\mathcal{A}_{\varphi,\vec{z}:\vec{c}})$. Then there is a formula $\widehat{\varphi}(\vec{x}, \vec{z}; \vec{y}, \vec{z}\,')$ in DL which defines a semi–forest $\mathcal{A}_{\widehat{\varphi}}$ of height $h(\mathcal{A}_{\widehat{\varphi}}) = \max_{\vec{c}\in n^m} h(\mathcal{A}_{\varphi,\vec{z}:\vec{c}})$.

Proof: Taking $\widehat{\varphi}(\vec{x}, \vec{z}; \vec{y}, \vec{z}\,') \equiv \varphi(\vec{x}, \vec{y}, \vec{z}) \wedge \vec{z} = \vec{z}\,'$ we get structure

$$\mathcal{A}_{\widehat{\varphi}} = \langle n^{k+m}, \{((\vec{a}, \vec{c}), (\vec{b}, \vec{c})) \in n^{k+m} \times n^{k+m} \mid \mathcal{A}, \vec{x}:\vec{a}, \vec{y}:\vec{b}, \vec{z}:\vec{c} \models \varphi\}\rangle,$$

(isomorphic to) disjoint union of all $\mathcal{A}_{\varphi,\vec{z}:\vec{c}}$, so the claim follows by lemma 1. $\square$

4 Tools: Expressibility Over Structures with Successor

Now we turn to the main technical tools we will use to prove our results.
But first we need some auxiliary terminology. Let us consider the class $\boldsymbol{\mathcal{W}}$ of all finite models of the form $\boldsymbol{\mathcal{W}}(n) \ni \mathcal{W} = \langle n, Succ, U\rangle$, where $Succ$ is a standard successor relation, and U is an unary predicate. Then every element $\mathcal{W} \in \boldsymbol{\mathcal{W}}(n)$ may be treated as a binary string of length n. We say that the logic L over $\langle \leq, u, =\rangle$ *captures complexity class C* iff every subset of $\boldsymbol{\mathcal{W}}$ that belongs to C as set of words is definable by some sentence of L. In our situation unary predicate is usually absent, so we have only structures which correspond to strings over one letter alphabet.
Let us also adopt that a recursive function $G : \omega \longrightarrow \omega$ is called *space constructible*[2] iff there exists an on-line Turing Machine M which computes G, and for some constant c_G and all $n \in \omega$, machine M uses at most $c_G \cdot length(G(n))$ tape cells during computation with input n (input and output are represented as binary strings).
Let $\widetilde{G}$ be a function from ω into ω defined as follows:

- $\widetilde{G}(0) = 1$;
- $\widetilde{G}(m) = GG(\widetilde{G}(m-1) + 1)$ for $m > 0$.

[2]Note that our definition slightly differs from the usual one.

It is easy to see that if G is space constructible and strictly growing, then so is $\widetilde{G}$. If $G : \omega \to \omega$ is nondecreasing and unbounded, then for such a function we denote by $\overline{G}^{-1}$ an *upper converse of* G, i.e., a function $\omega \to \omega$ which maps n onto the least natural number m such that $G(m) \geq n$. Similarly we define *lower converse* $\underline{G}^{-1}$ which maps n onto the greatest natural number m such that $G(m) \leq n$.

Lemma 7 *For every space constructible, strictly growing function $G : \omega \to \omega$ and for every logic L which captures DLOGSPACE over structures with successor relation, there is a sentence φ in L such that for any finite structure $\mathcal{A}$ with successor and of size n, one has*

$$\mathcal{A} \models \varphi \quad \text{iff} \quad \text{there is a value of } \widetilde{G} \text{ in the integer interval } [\underline{G}^{-1}(n), n].$$

Proof: It suffices to show that the property "there is a value of $\widetilde{G}$ in the integer interval $[\underline{G}^{-1}(n), n]$" is in DLOGSPACE (recall that in our situation input value n is given as *unary* string).
The machine we need cycles through all input numbers m of at most $c_{\widetilde{G}} \cdot \log n$ bits and for each of them
1. it tries to compute $r = \widetilde{G}(m)$ in $c_{\widetilde{G}} \cdot \log n$ tape cells to check whether $r \leq n$;
2. it tries to compute $G(r)$ (and eventually $G(r+1)$, if $G(r) = n$) in $c_G \cdot \log n$ tape cells to check whether $r \geq \underline{G}^{-1}(n)$;
3. if $\underline{G}^{-1}(n) \leq r \leq n$ it accepts.
4. if it does not accept for any input of at most $c_{\widetilde{G}} \cdot \log n$ bits it rejects. □

Lemma 8 *With assumptions as above there exist L formulae $Zero(x)$, $Max(x)$, $Add(x, y, z)$ and $Mult(x, y, z)$ such that for any finite $\mathcal{A}$ with successor and of size n one has*

$$\langle M, \preceq, Zero, Max, Add, Mult \rangle \simeq \langle n, \leq, 0, n-1, +, \times \rangle.$$

Proof: Addition and multiplication of numbers bounded by n are in DLOGSPACE. □

5 Main Theorem

The main theorem is formulated and proved in version essentially weaker than it could be. We decided to do that for three major reasons: eventual stronger formulations are essentially more complicated (e.g. existence of asymptotic probabilities and complexity of the almost sure theory should be treated separately); the current one is still powerful enough to be applied in many interesting situations; finally, similar theorem is proved in the strongest known to author version in the sequel to the current paper.

Theorem 1 *Suppose that for a class $\boldsymbol{\mathcal{A}}$ of finite structures and a sequence $\boldsymbol{\mu}$ of probabilities there exists a DL formula $\varphi(\vec{x}, \vec{y})$ with $\vec{x}$ and $\vec{y}$ of length k, such that for every $\mathcal{A} \in \boldsymbol{\mathcal{A}}(n)$ the structure*

$$\mathcal{A}_\varphi = \langle n^k, \varphi^{\mathcal{A}}(\vec{x}, \vec{y}) \rangle = \langle n^k, \{(\vec{a}, \vec{b}) \in n^k \times n^k \mid \mathcal{A}, \vec{x} : \vec{a}, \vec{y} : \vec{b} \models \varphi\} \rangle$$

is a semi–forest and for some nondecreasing, unbounded and recursive function $g : \omega \longrightarrow \omega$ *one has*

$$\lim_{n\to\infty} \mu_n(\{\mathcal{A} \in \boldsymbol{\mathcal{A}}(n) \mid g(n) \le h(\mathcal{A}_\varphi)\}) = 1. \tag{1}$$

Then the limit law fails for DL *and* $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mu}\rangle$ *and no associated asymptotic problem is decidable; moreover, there exists a sentence* ψ *with* $\liminf_{n\to\infty} \mu_n(\psi) = 0$ *and* $\limsup_{n\to\infty} \mu_n(\psi) = 1$.

Proof: We assume $k = 1$. Generalization to higher arities causes no problems. In the examples of applications k is always 1.
Let us note that w.l.o.g. we can assume g to be $\overline{G}^{-1}$ for some space constructible strictly growing function G. Indeed, g is nondecreasing and unbounded, so there exists recursive $\overline{g}^{-1}$, and then there is a space constructible strictly growing G such that $G \ge \overline{g}^{-1}$. Then $\overline{G}^{-1} \le g$, so $\overline{G}^{-1}$ satisfies (1) as well.
Lemma 6 guarantees that we can define a structure with successor of size $h(\mathcal{A}_\varphi)$ in every $\mathcal{A}$, by means of DL formulae. By a famous theorem in [9] DL captures NLOGSPACE over structures with successor, so there exists, by lemma 7, a DL sentence expressing property "there is a value of $\widetilde{G}$ in the integer interval $[\underline{G}^{-1}(h(\mathcal{A}_\varphi)), h(\mathcal{A}_\varphi)]$". Let us call this sentence Θ. Below we use h to denote $h(\mathcal{A}_\varphi)$.
Now let $\varepsilon > 0$ be arbitrary.
Choose n_0 large enough to have for every $n > n_0$:

$$\mu_n(\{\mathcal{A} \in \boldsymbol{\mathcal{A}}(n) \mid g(n) \le h\}) \ge 1 - \varepsilon \tag{2}$$

- For all $n > n_0$ such that $n = G\widetilde{G}(p)$ for some p (there are infinitely many such n), we have $g(n) = \overline{G}^{-1}(n) = \underline{G}^{-1}(n) = \widetilde{G}(p)$ since G is strictly growing, and therefore

  $$\mu_n(\Theta) \ge \mu_n(\{g(n)\} \subseteq [\underline{G}^{-1}(h), h]) = \mu_n(g(n) \le h \le n) \ge 1 - \varepsilon,$$

  by (2).

- For all $n > n_0$ such that $n = \widetilde{G}(p) - 1$ for some p (there are infinitely many such n), we have no value of $\widetilde{G}$ in $[gg(n), n]$ since $\widetilde{G}(p) > n$ and $\widetilde{G}(p-1) < gg(\widetilde{G}(p) - 1) - 1 < gg(n)$ (recall that $\widetilde{G}$ is strictly growing, too) and thus

  $$\mu_n(\Theta) \le 1 - \mu_n([\underline{G}^{-1}(h), h] \subseteq [gg(n), n]) = 1 - \mu_n(g(n) \le h \le n) \le \varepsilon,$$

  by (2).

Therefore, as ε is arbitrary, we get immediately that $\liminf_{n\to\infty} \mu_n(\Theta) = 0$ and $\limsup_{n\to\infty} \mu_n(\Theta) = 1$. In particular $\mu_n(\Theta)$ does not converge.
Now we turn to the undecidability result.
In the paper [3] one may find proof of the following lemma, which combined with lemmas 6 and 8 immediately gives the desired consequence:

Lemma 9 *If arbitrarily large initial segments of arithmetics are almost surely definable by means of formulae of logic* L *closed under classical propositional connectives and first order quantification, then sets* $\{\varphi \in L \mid \mu(\varphi) = 1\}$ *and* $\{\varphi \in L \mid \mu(\varphi) = 0\}$ *are recursively inseparable, and hence no asymptotic problem for* L *is decidable.* $\square$

6 Examples of Applications

The interest of the main theorem is that the semi–forest structure is relatively easy to define, and the function g which should approximate from below the height of this semi–forest may grow very slowly. This allows us to find many interesting situations where we may apply the main theorem to get interesting, new results.
The examples we have chosen are intended to show how we can define the semi–forest structure, and how easy is to find a nondecreasing, unbounded and recursive g to approximate its height. However, we believe that these examples are interesting for their own.

6.1 Slow Growing Classes

We deal with randomized classes considered in [1].
Suppose that a recursive class $\mathcal{A}$ of semi–forests is closed under disjoint unions and components, and that $\boldsymbol{\mu}$ is a sequence of uniform labelled probabilities, and $\boldsymbol{\nu}$ is a sequence of uniform unlabelled probabilities. Let $\sum_{n=0}^{\infty}(a_n/n!)x^n$ be the exponential generating series for $\mathcal{A}$ with radius of convergence $0 < R \leq \infty$, and let $\sum_{n=0}^{\infty} b_n x^n$ be the ordinary generating series for $\mathcal{A}$ with radius of convergence $0 < S \leq 1$.
Suppose that the situation is nontrivial, i.e. asymptotic uniform labelled (unlabelled) probabilities of first order sentences do exist. Then, as it has been proved in [1], the following equalities hold:

labelled case

$$\lim_{n\to\infty} \frac{a_{n-m}/(n-m)!}{a_n/n!} = R^m \tag{3}$$

unlabelled case

$$\lim_{n\to\infty} \frac{b_{n-m}}{b_n} = S^m \tag{4}$$

for every m being a cardinality of some connected structure in $\mathcal{A}$.

Theorem 2 *Suppose that the convergence in the above is recursive. Then any of the following conditions is sufficient for the DL limit law to fail, as well as for undecidability of all associated asymptotic problems (below $\mathcal{H}^d$ denotes the subclass of all connected structures of height at least d, symbol $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$ and $\sigma(\mathcal{A})$ – the number of automorphisms of $\mathcal{A}$):*

labelled case *1. $R = \infty$ and there are connected elements of $\mathcal{A}$ of arbitrary large height*

2. $0 < R < \infty$ and for every natural d

$$\sum_{\mathcal{A}\in\mathcal{H}^d} \frac{R^{|\mathcal{A}|}}{\sigma(\mathcal{A})} = \infty,$$

unlabeled case *1. $S = 1$ and there are connected elements of $\mathcal{A}$ of arbitrary large height*

2. $0 < S < 1$ *and for every natural* d

$$\prod_{\mathcal{A} \in \mathcal{H}^d} (1 - S^{|\mathcal{A}|}) = 0.$$

Proof: We defer it to the extended version of the paper.

Example 1 (permutations) Let $\boldsymbol{\mathcal{P}}$ be a class of finite permutations, and let $\boldsymbol{\mu}$ be uniform labelled probabilities, and $\boldsymbol{\nu}$ uniform unlabelled probabilities.
In that case convergence in (3) and (4) is clearly recursive, $R = 1$, $S = 1$, so we would like to apply labelled 2 and unlabelled 1. Indeed, $\mathcal{H}^d$ contains at least one cycle of each cardinality $i \geq d$, which is a structure with $i - 1$ automorphisms, and thus

$$\sum_{\mathcal{P} \in \mathcal{H}^d} \frac{R^{|\mathcal{P}|}}{\sigma(\mathcal{P})} \leq \sum_{i \geq d} 1/(i-1) = \infty,$$

so there are DL sentences with no uniform labelled asymptotic probability with respect to $\boldsymbol{\mathcal{P}}$, as well as no associated asymptotic problem is decidable.
In the unlabelled case we get the same conclusion even without any calculations.
This sharpens result of [10], where it is proved that in the unlabelled case the 0–1 law does not hold, for least fixed point logic. $\square$

6.2 Trees and forests

Example 2 (trees) *Let* $\boldsymbol{\mathcal{T}}$ *be* any *class of undirected (or directed) trees with degree of vertices almost surely bounded by* m. *It doesn't matter whether* $\boldsymbol{\mathcal{T}}$ *contains all such trees or not. The only important assumption is that* $\boldsymbol{\mathcal{T}}(n) \neq \emptyset$ *for all* n. *Let* $\boldsymbol{\mu}$ *be* any *sequence of probabilities. Then the limit law for DL fails and no associated asymptotic problem is decidable.*

Proof: Directed trees are themselves semi–forests, and their height almost surely satisfies $\log_m n \leq h(\mathcal{T})$, for $\mathcal{T} \in \boldsymbol{\mathcal{T}}(n)$. Now the thesis immediately follows by theorem 1.
In the undirected case we have more problems. Undirected trees are not semi–forests in general, so we need a DL formula which will define semi–forest structure over them, so that its height is equal to the maximal path length in the undirected tree. We leave this for the reader. $\square$

Remark 1 The above example may be, without any additional effort, extended to the case when almost surely all vertices of a random $\mathcal{T} \in \boldsymbol{\mathcal{T}}(n)$ have degree no greater than $g(n)$, with g satisfying:
for every rational $\varepsilon > 0$ the sequence $g(n)/n^{\varepsilon}$ recursively (with respect to both n and ε) converges to 0. $\square$

Example 3 (oriented (rooted) forests) We consider the class $\langle \boldsymbol{\mathcal{L}}, \boldsymbol{\mu} \rangle$ of *oriented undirected forests* with labelled, uniform probabilities, as they are described in example 7.8 in [1]. We prove that DL limit law fails in this case, and no associated asymptotic problem is decidable.

This is not hard to see that, similarly as above, we are able to define by DL means a semi–forest from undirected forest, and the height of the resulting semi–forest will be no less than the maximal path length in the source structure. Therefore, by theorem 2, in order to establish our result it suffices to show that

$$\sum_{\mathcal{L}\in\mathcal{H}^d} R^{|\mathcal{L}|}/\sigma(\mathcal{L}) = \infty$$

(note that the suitable sequences converge recursively to $R = 1/e$). Consider the following elements in $\mathcal{H}^d$: for every $i \geq d$ the set of $i!$ chains with root in one of the ends. Each of them has only trivial automorphism. Then have indeed

$$\sum_{\mathcal{L}\in\mathcal{H}^d} R^{|\mathcal{L}|}/\sigma(\mathcal{L}) \geq \sum_{i\geq d} i!(1/e)^i = \infty.$$

6.3 Random graphs

Here we investigate the random graph construction, initiated in [6]. Let $p = p(n)$ be a function from ω into the real interval [0,1]. Then we define the probability model $\boldsymbol{\mathcal{G}}(n,p) = \langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mu}^p \rangle$ with $\boldsymbol{\mathcal{G}}$ the class of all labelled, undirected finite graphs, and $\mu^p_n(\{\mathcal{G}\}) = p^m(1-p)^{\binom{n}{2}-m}$, where m is the number of edges in $\mathcal{G}$. Those graphs we treate as structures over the signature $\langle \sim, = \rangle$, following tradition.
Our aim is to prove the following result, which extends knowledge about limit laws for various logics with respect to models $\boldsymbol{\mathcal{G}}(n,p)$.

Example 4 *If $p = p(n)$ satisfies*

1. *$p(n)/n^{-1}\log n$ converges recursively to 0;*
2. *for every positive rational ε the sequence $n^{-1-\varepsilon}/p(n)$ converges to 0, recursively with respect to both n and ε.*

then the limit law fails for DL and $\boldsymbol{\mathcal{G}}(n,p)$, and no associated asymptotic problem is decidable.

Proof: Of course we want to apply theorem 1. In order to do that we need two things: a formula which will define the semi–forest structure over almost every $\mathcal{G}$ and a proof that the height of this semi–forest is bounded from the below by some nondecreasing, unbounded and recursive function of n. The idea of finding them comes from the following result (and its proof):

Proposition 2 ([6]) *If $\lim_{n\to\infty} p(n)/n^{-1}\log n = 0$ and $\lim_{n\to\infty} n^{-1-\varepsilon}/p(n) = 0$ for every rational $\varepsilon > 0$, then each finite tree almost surely appears as a component of a random $\mathcal{G} \in \boldsymbol{\mathcal{G}}(n)$.* □

Following the proof of this proposition and using information about recursive convergence of both sequences, we can in standard way show that there is nondecreasing, unbounded and recursive function $g = g(n)$ such that $\mathcal{G} \in \boldsymbol{\mathcal{G}}(n)$ contains almost surely each finite tree of size $g(n)$ as a component.

Now we need a method to separate elements that belong to components that are trees to be able to define a semi–forest from such structure, as in previous example. Observe that the component of a is a tree iff for every b in the same component as a, $b \not\sim b$ and, moreover, all cycles $b \sim \ldots \sim b$ have the same the second and the last but one element. This allows us to write easily needed formula. □

7 Deterministic While–Programs

In this section we will work with one fixed class: the class of finite unary functions with uniform labelled probabilities: $\langle \boldsymbol{\mathcal{F}}, \boldsymbol{\mu} \rangle$, over signature $\tau = \langle f, = \rangle$. Our aim is to improve the proof technique in order to get similar result as in §5, but working in a formalism of smaller expressive power.

To describe this formalism let us define the set of *deterministic while–programs* (DWP) to be the set of these programs in which the only assignments are of form $x_i := x_j$ and $x_i := f(x_j)$ (which is to be understood as shorthand for $x_i := y$ `such that` $y = f(x_j)$), and in which formulae in conditional instructions and while loops are restricted to be only $x_i = x_j$ and $x_i \neq x_j$. Then we are interested in formulae of the form $\langle Q \rangle true$ for deterministic while–programs Q. They play very important role in investigation of properties of programs, especially in the theory of program correctness. The reader may find more complete information about this formalism in [5, 12]. Therefore the aim of this section is to remove nondeterministic assignments, rich tests and as many quantifiers as possible from formulae resulting in our proofs, keeping them working for $\langle \boldsymbol{\mathcal{F}}, \boldsymbol{\mu} \rangle$.

First we cite a result proved in [13] (but in notation from [12]):

Proposition 3 *For a language with one unary function symbol and one unary predicate symbol one has*

$$SP(DWP) \approx DLOGSPACE.$$

□

We do not attempt to explain the sense of this proposition in details here.

We present a result being a corollary to the above and expressed in our terminology, instead:

Corollary 1 1. *Over finite structures with total successor function[3] and no other functions formulae of the form $\langle \vec{x} := \vec{0};\ Q \rangle true$ with $Q \in DWP$ can express every DLOGSPACE–computable property.*

2. *There is a program $TSucc(x, y) \in DWP$ which in every $\mathcal{F} = \langle n, F \rangle \in \boldsymbol{\mathcal{F}}(n)$ and for every $a \in n$, computes (defines) some total successor function on $C_a = \{F^n(a) \mid n \in \omega\}$, with a corresponding to 0.*

□

Now we can state our theorem:

[3]By a total successor function on $\{0, \ldots, n-1\}$ we mean the following total function: $\lambda m.$ `if` $m \neq n-1$ `then` $m+1$ `else` m `fi`.

Theorem 3 *In the case of $\langle \mathcal{F}, \boldsymbol{\mu} \rangle$ the limit law fails for formulae of the form $\exists x \forall \vec{y} \langle Q \rangle true$ with $Q \in DWP$, as well as no associated asymptotic problem is decidable. Moreover, there is a program $Q \in DWP$ with*

$$\liminf_{n \to \infty} \mu_n (\exists x \forall \vec{y} \langle Q \rangle true) = 0$$

and

$$\limsup_{n \to \infty} \mu_n (\exists x \forall \vec{y} \langle Q \rangle true) = 1.$$

Proof (sketch): For fixed strictly growing space constructible $G : \omega \longrightarrow \omega$ we have a program $Q_G(x, \vec{z}) \in DWP$ which run in $\mathcal{F}, x{:}a$ checks whether there is a value of $\widetilde{G}$ in the integer interval $[\underline{G}^{-1}(|C_a|), |C_a|]$; if so it stops and otherwise loops forever. Its existence is assured by proposition 1 and lemma 7. Let us cite now
Now we construct a program $Q_1(x, y, \vec{z}) \in DWP$ which run in $\mathcal{F}, x : a, y : b$ checks whether $|C_a| \geq |C_b|$; if so it stops, otherwise loops forever. Additionally, it should not change the value of x. It is very easy when we use program $TSucc$ as a subroutine to compute total successors on C_a and C_b.
Let a program Q_{1G} to be sequential composition of Q_1 and Q_G. Thus the sentence $\theta \equiv \exists x \forall y \forall \vec{z} \langle Q_{1G} \rangle true$ expresses:
"the cardinality of $|C_a|$ is maximal possible and there is a value of $\widetilde{G}$ in the integer interval $[\underline{G}^{-1}(|C_a|), |C_a|]$".
A result in [8] states that in random $\mathcal{F}$ almost surely $|C_0| > \sqrt[3]{n}$.
From now on we can follow the proof of theorem 1, taking θ as Θ, to get the thesis. The only subtle moment is that the set of formulae we are talking about is not closed under propositional connectives and quantification, which is necessary to apply lemma 9. Fortunately, they are expressible by the program itself (one may, roughly speaking, put a "model checker" subroutine inside the program). $\Box$
The last result suffices to derive the following corollary which, roughly speaking, asserts that *monadic second order logic MSO* is unable, in a very strong way, to express properties of programs.

Corollary 2 *There exists a program $Q \in DWP$ so that for every monadic second order formula φ (both without constant for 0)*

$$\liminf_{n \to \infty} \mu_n (\varphi \Leftrightarrow \langle Q \rangle true) \leq 1/2.$$

In particular, MSO $\not\geq \{\langle Q \rangle true \mid Q \in DWP\}$.

Proof: Take the program Q with

$$\liminf_{n \to \infty} \mu_n (\exists x \forall \vec{y} \langle Q \rangle true) = 0$$

and

$$\limsup_{n \to \infty} \mu_n (\exists x \forall \vec{y} \langle Q \rangle true) = 1.$$

Then let φ be arbitrary formula in monadic second order logic. It is easy to see that

$$\mu_n (\varphi \Leftrightarrow \langle Q \rangle true) \leq \mu_n (\exists x \forall \vec{y} \varphi \Leftrightarrow \exists x \forall \vec{y} \langle Q \rangle true) \leq$$

$$\leq 1 - |\mu_n(\exists x \forall \vec{y} \varphi) - \mu_n(\exists x \forall \vec{y} \langle Q \rangle true)|.$$

By a theorem proved in [4] $\lim_{n\to\infty} \mu_n(\exists x \forall \vec{y} \varphi)$ exists, so $\liminf$ of the last number above is no greater than $1/2$, which immediately implies the claim. $\square$

Note the strength of this result. It is more than just an assertion that monadic second order logic is unable to express halting properties of programs: our result is not only qualitative, but quantitative, as well, and says that there are many counterexamples. In particular it forces that, although finite models are expressive for DWP (i.e. for every $Q \in DWP$ and every $\mathcal{F} \in \boldsymbol{\mathcal{F}}$ there is first order formula φ such that $\mathcal{F} \models \varphi \Leftrightarrow \langle Q \rangle true$), they are not uniformly expressive (i.e. the φ above has to depend on $\mathcal{F}$), even if we allow φ to be in MSO. For discussion of expressiveness issues see [5].

Acknowledgment I wish to thank Damian Niwiński for helpful discussions and improving my poor English. Thanks are also due to an anonymous referee, whose suggestions helped me very much.

References

[1] K.J. Compton. A logical approach to asymptotic combinatorics I. First order properties, *Advances in Mathematics* **65**(1987), 65-96.

[2] K.J. Compton. 0–1 laws in logic and combinatorics, *Proc. NATO Advanced Study Institute on Algorithms and Order* (I. Rival, ed.), Reidel, Dordrecht (1988).

[3] K.J. Compton, C.W. Henson, and S. Shelah. Nonconvergence, undecidability and intractability in asymptotic problems, *Annals of Pure and Applied Logic*, **36**(1987), 207-224.

[4] K.J. Compton, and S. Shelah. A convergence theorem for unary functions, *in preparation.*

[5] P. Cousot. Methods aand logics for proving programs, *in: J. van Leeuven (ed.) Handbook of theoretical computer science*, North Holland, Amsterdam, 1990.

[6] P. Erdős, and A. Rényi. On the evolution of random graphs, *Magyar Tud. Akad. Mat. Kutató Int. Kőzl.* **5**(1960), 17-61.

[7] D. Harel, and D. Peleg. On static logics, dynamic logics, and complexity classes, *Infor. and Control* **60**(1984), 86-102.

[8] B. Harris. Probability distributions related to random mappings, *The Annals of Mathematical Statistics* **31**(1960), 1045-1062.

[9] N. Immerman. Languages that capture complexity classes, *SIAM Journal of Computing* **16**, No. 4(1987).

[10] Ph. Kolaitis. On asymptotic probabilities of inductive queries and their decision problem, *in: Logics of Programs, Brooklyn, June 1985,* Lecture Notes in Comp. Sci. 193, Springer Verlag, 1985, 153-166.

[11] Ph. Kolaitis, and M.Y. Vardi. Zero–one laws for infinitary logics, *Proc. 5th IEEE Symp. on Logic in Computer Science*, 1990, 156-167.

[12] D. Kozen, and J. Tiuryn. Logics of programs, *in: J. van Leeuven (ed.) Handbook of theoretical computer science*, North Holland, Amsterdam, 1990.

[13] J. Tiuryn, and P. Urzyczyn. Some relationships between logics of programs and complexity theory, *Theoretical Computer Science*, **60**(1988), 83-108.

[14] J. Tyszkiewicz. Infinitary Queries and Their Asymptotic Probabilities II: Properties Definable in Least Fixed Point Logic, *to appear in Proc. Random Graphs'91, Poznań, Poland, August 1991.*

A Semantics of programs and DL formulae.

A.1 Semantics of programs

Let $\mathcal{A} = \langle A, \vec{R} \rangle$ be any structure, and let $\vec{a} \in A^k$. Then a computation of a program $Q(\vec{x})$ (with k registers) in $\mathcal{A}, \vec{x} : \vec{a}$ is any (note that one program may have many different computations in $\mathcal{A}, \vec{x}:\vec{a}$), finite or infinite, sequence

$$Comp(Q, \mathcal{A}, \vec{a}) = (\vec{a}_0, l_0)(\vec{a}_1, l_1)\ldots$$

of *states* $(\vec{a}_i, l_i) \in A^k \times \omega$ such that:

1. $\vec{a}_0 = \vec{a}$, $l_0 = 0$
2. $(\vec{a}_{m+1}, l_{m+1})$ satisfies the following conditions relatively to $(\vec{a}_m, l_m)$
   (a) if l_m is a number of line with *abort*, then $(\vec{a}_{m+1}, l_{m+1}) = (\vec{a}_m, l_m)$
   (b) if l_m is a number of line with `do` or `fi` then $(\vec{a}_{m+1}, l_{m+1}) = (\vec{a}_m, l_m + 1)$
   (c) if l_m is a number of line with `if` $\varphi(\vec{x})$ then $\vec{a}_{m+1} = \vec{a}_m$ and if $\mathcal{A}, \vec{x}: \vec{a}_m \models \varphi(\vec{x})$ then l_{m+1} is the number of line just after the corresponding `then` line, and otherwise the number of line just after corresponding `else` line
   (d) if l_m is a number of line with `else` , then $\vec{a}_{m+1} = \vec{a}_m$ and l_{m+1} is the number of corresponding `fi` line
   (e) if l_m is a number of line with `while` $\varphi(\vec{x})$, then $\vec{a}_{m+1} = \vec{a}_m$ and if $\mathcal{A}, \vec{x}: \vec{a}_m \models \varphi(\vec{x})$ then $l_{m+1} = l_m + 1$, and otherwise it is the number of the line just after the corresponding `od`line
   (f) if l_m is a number of line with `od` then $\vec{a}_{m+1} = \vec{a}_m$ and l_{m+1} is the number of corresponding `while` line
   (g) if l_m is a number of line with $x_i := y$ `such that` $\varphi(y, \vec{x})$ then $\vec{a}_{m+1}$ is identical with $\vec{a}_m$ except i-th coordinate, which is now equal to some element $b \in A$ such that $\mathcal{A}, \vec{x}:\vec{a}_m, y:b \models \varphi$ and $l_{m+1} = l_m + 1$, if such b exists, and otherwise $(\vec{a}_{m+1}, l_{m+1}) = (\vec{a}_m, l_m)$
3. if l_m is a number of line with *halt*, then it is the last element in $Comp(Q, \mathcal{A}, \vec{a})$.

A.2 Semantics of *DL* Formulae

Semantics of *DL* formulae is standard, except the case of the formula $\langle Q \rangle true$. In this case we define

$\mathcal{A}, \vec{x}:\vec{a} \models \langle Q \rangle true$ iff there exists finite computation $Comp(Q, \mathcal{A}, \vec{a})$ of Q in $\mathcal{A}, \vec{x}:\vec{a}$.