Fundamenta Informaticae 110 (2011) 1–15
DOI 10.3233/FI-2011-551
IOS Press

Typical Paths of a Graph

Cewei Cui, Zhe Dang*, Thomas R. Fischer
School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA 99164, USA
ccui@eecs.wsu.edu; zdang@eecs.wsu.edu; fischer@eecs.wsu.edu

Abstract. We introduce (finite and infinite) typical paths of a graph and prove that the typical paths carry all information with probability 1, asymptotically. An automata-theoretic characterization of the typical paths is shown: finite typical paths can be accepted by reversal-bounded multicounter automata and infinite typical paths can be accepted by counting Büchi automata (a generalization of reversal-bounded multicounter automata running on ω-words). We take a statechart example to show how to generate typical paths from a graph using SPIN model checker. The results are useful in automata theory since one can identify an information-concentrated-core of a regular language such that only words in the information-concentrated-core carry nontrivial information. When the graph is used to specify the system under test, the results are also useful in software testing by providing an information-theoretic approach to select test cases that carry nontrivial information of the system specification.

Keywords: graph, entropy, path

1. Introduction

Graph is a basic data structure in computer science. When a graph is used to specify a (blackbox) software system (e.g., control flow graphs, data flow graphs and statecharts [9]), selecting test cases is essentially equivalent to selecting paths (may contain loops) from the graph. This view is vividly illustrated and approved by several authors who understand testing as a process to "find a graph and cover it [2, 1]." A central problem of testing is how to select test cases, since test cases need be selected before tests are run and faults can only be identified after tests are run.

*Address for correspondence: School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, USA

Test cases are generated according to a given test data adequacy criterion [8] to intuitively indicate "how much"of the graph is covered. For instance, path coverage is an indicator to measure the number of paths of length n, for some given n, that are selected as test cases divided by the number of all possible paths of length n. Therefore, in path coverage, all the paths of the same length n are born equal. This is not intuitively true: some paths do carry much more Shannon information than others [16].

In information theory [5], there is a phenomenon called AEP (asymptotic equipartition property) saying that, for a random process, its information concentrates almost entirely on a small part of the sample sequences. The AEP has been shown valid for only a handful of cases; e.g., i.i.d. processes as well as stationary and ergodic Markov chains [5]. In this paper, we would like to develop a theory to show that the AEP holds on the paths (from a given initial node) of a graph; i.e., there are "typical paths"of the graphs that concentrate almost all information of all the paths. In this way, software engineers can use the result to select test cases carrying nontrivial information. However, there are difficulties in defining the typical paths. First, in AEP, one needs a random process, but in our theory, we are only given a graph without transition probabilities. We handle the difficulty as follows. Selecting a path from the graph resembles a Markov walk on the graph. Since the software specification under test is a blackbox, we know nothing except for the blackbox's interface. Hence, the walk must achieve the maximal entropy rate[1] λ^* (otherwise, we must know additional information on the blackbox). In [17], we prove that there is a Markov chain $\mathcal{X}$ achieving the maximal entropy rate $\lambda^* = \lim_{n\to\infty} \frac{\log S(n)}{n}$, where $S(n)$ is the number of paths in the graph with length n. Then, we define an ϵ-$\mathcal{X}$-typical path $x_1, \cdots, x_n$, which is a node path of the graph, of the Markov chain $\mathcal{X}$ to be one satisfying

$$\left|\frac{1}{n}\log\frac{1}{p(x_1,\cdots,x_n)} - \lambda^*\right| < \epsilon,$$

where $p(x_1, \cdots, x_n)$ is the probability of the path $x_1, \cdots, x_n$. Second, showing that the AEP holds for these ϵ-$\mathcal{X}$-typical paths is difficult. This is because of Shannon-Mcmillan-Breiman Theorem [5] saying that the existence of the self-entropy rate (in the form of a limit definition) needs a strong side condition (ergodicity). In this paper, we employ a limsup definition of entropy rate and successfully prove the AEP for ϵ-$\mathcal{X}$-typical paths. Third, Markov chain $\mathcal{X}$ that achieves the rate λ^* is not unique. In this case, we would like to know whether there is a notion that contains all typical paths of these Markov chains achieving the rate λ^*, with probability 1, asymptotically. In this paper, we prove that the following notion is as desired: a node path $x_1, \cdots, x_{n+1}$, which has n edges, on the graph is ϵ-typical if it satisfies

$$\frac{\log B(x_1,\cdots,x_{n+1})}{n} > \frac{1}{2}(\lambda^* - \epsilon),$$

where $B(x_1, \cdots, x_{n+1}) = \prod_{1\le i\le n} b(x_i)$ with $b(x_i)$ being the branching factor of node x_i. Additionally, we also extend the above results to ω-paths of the graph and show a similar AEP holds for ω-paths; i.e., the typical set of ω-paths on a graph also takes probability 1. Then, we show that ϵ-typical paths (resp. ω-ϵ-typical paths) can be accepted by reversal-bounded multicounter automata [11] (resp. counting Büchi automata – a generalization of reversal-bounded multicounter automata running on ω-words). The above characterization has some applications. First, for a given $\epsilon > 0$, we can algorithmically choose ϵ-typical

[1] In information theory, the entropy rate is used to indicate how many bits one needs to losslessly encode each sample in a stochastic process. Intuitively, a high entropy rate implies that the Markov chain has a high complexity, since one needs more resource (encoding rate) to faithfully describe the process.

paths satisfying certain patterns such as regular patterns and some nonregular patterns. Second, when the graph is used to specify the system under test, we provide an approach to select test cases that carry nontrivial information of the system specification. Third, we can define an *information-concentrated-core* (ICC) of a regular language such that only words in the ICC carry nontrivial information. The ICCs for regular and some nonregular languages deserve further investigation in automata theory.

The rest of the paper is organized as follows. Section 2 recalls the basic definitions and shows the AEP for ϵ-$\mathcal{X}$-typical and ϵ-typical (ω-)paths. Section 3 gives the definition of reversal-bounded multi-counter automata [11] and counting Büchi automata and investigates an automata-theoretic characterization of finite and infinite typical paths. Sections 4 shows an experiment to generate typical paths from a statechart using SPIN [10]. Section 5 concludes this paper.

2. Graph, Markov chain, and typical path

As usual, a (finite) graph G has a set of nodes Q and a set of directed edges E. It has a designated initial node, say, q_1. Without loss of generality, we assume that the graph is reduced; i.e., every node is reachable from the initial node. A *path* of G is a finite sequence of nodes in G, $q^1 \cdots q^n$, for some n, which starts from the initial node (i.e., $q^1 = q_1$) and, for each $1 \leq i < n$, $\langle q^i, q^{i+1} \rangle \in E$. An *$\omega$-path* in G is an infinite sequence of nodes in G where each (finite) prefix is a path of G.

In this paper, a (finite state) Markov chain $\mathcal{X}$ is a discrete stochastic process $X_1, \cdots, X_n, \cdots$ where the sample space for each random variable X_n is Q, and the conditional probability of $\mathcal{X}$ needs to satisfy

$$\mathrm{Prob}(X_n = x_n | X_{n-1} = x_{n-1}, \cdots, X_1 = x_1) = \mathrm{Prob}(X_n = x_n | X_{n-1} = x_{n-1}),$$

for all $x_1, \cdots, x_n \in Q$. Together with the *initial distribution* $\mathrm{Prob}(X_1 = x_1) = 1$ (the process always starts from the initial node of G), the probability of a particular sequence $\pi = x_1 \cdots x_n$ for some $n \geq 1$, is

$$p(\pi) = \mathrm{Prob}(X_1 = x_1)\mathrm{Prob}(X_2 = x_2 | X_1 = x_1) \cdots \mathrm{Prob}(X_n = x_n | X_{n-1} = x_{n-1}).$$

Hence, the (finite state and time-invariant) Markov chain can also be represented in the form of *probability transition matrix* $T = [T_{ij}]$ where each entry T_{ij} indicates the *transition probability* $\mathrm{Prob}(X_{n+1} = q_j \mid X_n = q_i)$ from node q_i to q_j. The Markov chain $\mathcal{X}$ is called a *G-represented* Markov chain if, for each T_{ij}, $T_{ij} = 0$ when there is no edge from node q_i to node q_j in G.

In information theory, entropy rate indicates the growth rate of the entropy of a stochastic process. The entropy rate of $\mathcal{X}$ is defined as

$$\lambda_{\mathcal{X}} = \lim_{n \to \infty} \frac{1}{n} H(X_1, \cdots, X_n), \tag{1}$$

where $H(X_1, X_2, \cdots, X_n)$ is the joint entropy of $X_1, \cdots, X_n$, defined as, according to Shannon,

$$\sum_{x_1, \cdots, x_n} p(x_1, \cdots, x_n) \log \frac{1}{p(x_1, \cdots, x_n)}.$$

Note that throughout this paper, the base of logarithm is 2. Traditionally, research in information theory focuses on random processes when the entropy rate, as a limit in (1), exists; e.g., when the processes

are stationary. However, in general, the limit may not exist. In this paper, we use the upper limit as the entropy rate if the limit does not exist. The entropy rate now is defined as follows:

$$\lambda_{\mathcal{X}} = \limsup_{n\to\infty} \frac{1}{n} H(X_1, X_2, \cdots, X_n). \tag{2}$$

Notice that the upper limit always exists (and is $\leq \log |Q|$, and hence finite), when the Markov chain $\mathcal{X}$ is G-represented. In this case, what would be the maximal rate λ^* among all possible G-represented $\mathcal{X}$? In [17], it is shown that

$$\lim_{n\to\infty} \frac{\log S(n)}{n} = \lambda^*, \tag{3}$$

where $S(n)$ is the number of paths in G with length n, and the rate λ^* is achievable by a G-represented $\mathcal{X}$ (however, such Markov chain $\mathcal{X}$ is not necessarily unique); i.e., $\lambda_{\mathcal{X}} = \lambda^*$. The existence [3] of the limit in (3) is the source of the well-known fact that the count of words with length n in a regular language grows either polynomially or exponentially. Furthermore, in [17] (also implicitly in [3]), an efficient numerical algorithm to compute the rate λ^* from the graph G is provided. In the sequel, the rate λ^* is also called the *rate* of the graph G. That is all for our technical preparation in this section.

Intuitively, λ^* refers to the entropy rate of a most "random" Markov walk on the graph. In the sequel, $\mathcal{X}$ refers to a Markov process that achieves the maximal rate λ^*.

In information theory, there is an interesting principle called AEP (asymptotic equipartition property) [5] whose intuitive meaning is as follows: for a random process $X_1, \cdots, X_n$, the information is almost entirely concentrated on a "small" (not necessarily small in practice) number of sequences; these sequences, called *typical sequences*, almost take probability 1. In other words, non-typical sequences take almost zero amount of information, combined! The word "small" refers to the fact that the number of typical sequences is only inverse-exponential ratio to the number of all sequences, in many cases. Hence, the word "typical" here catches the real-world meaning of "carrying nontrivial amount of information."

However, the principle has only been shown its validity for a handful of cases (e.g., i.i.d. process and ergodic Markov process, etc.). This is because the difficulty of the mathematics involved to show the cases when AEP does hold.

In below, we will define typical paths for the Markov chain $\mathcal{X}$. Intuitively, a typical path on the G-represented $\mathcal{X}$ means that the self-entropy rate [5] of the path approximates the rate of G, which is also the entropy rate that $\mathcal{X}$ achieves.

Definition 2.1. Let $\epsilon > 0$ and $\mathcal{X}$ be the G-represented Markov chain specified earlier achieving the rate λ^* of the graph G. A path $x_1, \cdots, x_n$ is ϵ-$\mathcal{X}$-typical if the probability of this path satisfies

$$\Big|\frac{1}{n} \log \frac{1}{p(x_1, \cdots, x_n)} - \lambda^*\Big| < \epsilon.$$

We are going to prove that AEP does hold for typical paths defined in above. In other words, for sufficiently large n, non-ϵ-$\mathcal{X}$-typical paths hold almost zero information by showing that non-ϵ-$\mathcal{X}$-typical paths take probability almost zero asymptotically, as follows.

Lemma 2.1. Let $\epsilon > 0$ and $\mathcal{X}$ be the G-represented Markov chain specified earlier achieving the rate λ^* of the graph G. Then,

$$\liminf_{n\to\infty} \mathrm{Prob}\Big(\Big|\frac{1}{n}\log \frac{1}{p(X_1, \cdots, X_n)} - \lambda^*\Big| \geq \epsilon\Big) = 0.$$

Proof:
Let $X_1, \cdots, X_n, \cdots$ be the Markov chain $\mathcal{X}$. For each n, the sample space size of $X_1, \cdots, X_n$, according to [17], is $S(n)$. By definition

$$\limsup_{n\to\infty} \frac{H(X_1, \cdots, X_n)}{n} = \lambda^*. \tag{4}$$

Clearly, it suffices for us to show, using (3), that, for any $\epsilon > 0$,

$$\liminf_{n\to\infty} \text{Prob}(|\log \frac{1}{p(X_1, \cdots, X_n)} - \log S(n)| \geq n\epsilon) = 0. \tag{5}$$

Let $\tau > 0$. For notational convenience in the sequel, we define $C1(n)$ as the condition that

$$\log \frac{1}{p(x_1, \cdots, x_n)} \geq n\tau + \log S(n).$$

We first notice that, for all n and $\tau > 0$, using Theorem 5.10.1 in [5],

$$\text{Prob}(\log \frac{1}{p(X_1, \cdots, X_n)} \geq n\tau + \log S(n)) \leq \frac{1}{2^{n\tau-1}}. \tag{6}$$

From (6), we immediately have

$$\liminf_{n\to\infty} \text{Prob}(\log \frac{1}{p(X_1, \cdots, X_n)} - \log S(n) \geq n\epsilon) = 0.$$

Hence, in order to prove (5), we need only show

$$\liminf_{n\to\infty} \text{Prob}(\log \frac{1}{p(X_1, \cdots, X_n)} \leq \log S(n) - n\epsilon) = 0. \tag{7}$$

Take $p^* > 0$ to be the minimal positive transition probability of the Markov chain $X_1, \cdots, X_n, \cdots$. Clearly, from above,

$$\sum_{C1(n)} p(x_1, \cdots, x_n) \log \frac{1}{p(x_1, \cdots, x_n)} \leq \frac{1}{2^{n\tau-1}} \cdot n \log \frac{1}{p^*}. \tag{8}$$

Notice that, as $n \to \infty$, the RHS $\to 0$. Using (3) and (4), there are $n_1 < \cdots < n_k < \cdots$ such that

$$\lim_{k\to\infty} \frac{\log S(n_k) - H(X_1, \cdots, X_{n_k})}{n_k} = 0. \tag{9}$$

We prove (7) by contradiction. That is, we assume that there are $\epsilon > 0, \delta > 0$, and an infinite set K such that for each $k \in K$,

$$\text{Prob}(\log \frac{1}{p(X_1, \cdots, X_{n_k})} \leq \log S(n_k) - n_k\epsilon) \geq \delta$$

holds. Again, we use $C2(n)$ to denote the condition

$$\log \frac{1}{p(x_1, \cdots, x_n)} \leq \log S(n) - n\epsilon.$$

Now, for each $k \in K$, $H(X_1, \cdots, X_{n_k})$

$$\begin{aligned}
&\leq \sum_{C2(n_k)} p(x_1, \cdots, x_{n_k}) \cdot [\log S(n_k) - n_k\epsilon] \\
&\quad + \sum_{\neg C2(n_k) \wedge \neg C1(n_k)} p(x_1, \cdots, x_{n_k}) \log \frac{1}{p(x_1, \cdots, x_{n_k})} \\
&\quad + \sum_{\neg C2(n_k) \wedge C1(n_k)} p(x_1, \cdots, x_{n_k}) \log \frac{1}{p(x_1, \cdots, x_{n_k})} \\
&\quad \text{(Using (8))} \\
&\leq \delta \cdot [\log S(n_k) - n_k\epsilon] + (1-\delta) \cdot [\log S(n_k) + n_k\tau] + \frac{1}{2^{n_k\tau - 1}} \cdot n_k \log \frac{1}{p^*}.
\end{aligned}$$

Take $\tau < \frac{\epsilon \cdot \delta}{1-\delta}$ and, since K is infinite, we have

$$\liminf_{k\to\infty} \frac{\log S(n_k) - H(X_1, \cdots, X_{n_k})}{n_k} \geq \epsilon \cdot \delta - (1-\delta) \cdot \tau > 0,$$

a contradiction to (9). The result follows. □

The following AEP theorem, directly from Lemma 2.1, claims that, asymptotically (i.e., for infinitely many large n), typical paths take probability 1.

Theorem 2.1. Let $\epsilon > 0$. The ϵ-$\mathcal{X}$-typical paths take probability 1, asymptotically; i.e.,

$$\limsup_{n\to\infty} \mathrm{Prob}(\Big|\frac{1}{n} \log \frac{1}{p(X_1, \cdots, X_n)} - \lambda^*\Big| < \epsilon) = 1.$$

Now, we extend our results to ω-paths of G.

Definition 2.2. Let $\mathcal{X}$ be the G-represented Markov chain mentioned earlier. An ω-path is ω-ϵ-$\mathcal{X}$-typical if there are infinitely many prefixes such that each prefix is ϵ-$\mathcal{X}$-typical.

To show that AEP holds for ω-ϵ-$\mathcal{X}$-typical (ω-)paths, we first need to define a probabilistic measure μ over ω-sequences of nodes, using notation in [15]. In the sequel, we use $\pi = x_1, \cdots, x_n$ to denote a sequence in Q^* and use $\pi^\omega = x_1, \cdots, x_n, \cdots$ to denote an ω-sequence in Q^ω. Define

$$cylinder(\pi) = \{\pi^\omega \mid \pi \text{ is a prefix of } \pi^\omega\}$$

and define

$$\mu(cylinder(x_1, \cdots, x_n)) = p(x_1, \cdots, x_n).$$

Now, μ can be extended to the σ-algebra generated by the cylinder sets, as usual [15]. In particular, the set of all ω-ϵ-$\mathcal{X}$-typical paths can be written as

$$\bigcap_{k\in\mathbf{N}} \bigcup_{n\geq k} \{(x_1, \cdots, x_n \cdots) \in Q^\omega \mid \Big|\frac{1}{n} \log \frac{1}{p(x_1, \cdots, x_n)} - \lambda^*\Big| < \epsilon\}.$$

Clearly, this set is measurable under μ. We can show that the set of all ω-ϵ-typical paths takes probability 1 using measure theory [18].

Theorem 2.2. Let $\epsilon > 0$. Then,

$$\mu(\bigcap_{k\in\mathbf{N}}\bigcup_{n\geq k}\{(x_1,\cdots,x_n,\cdots)\in Q^\omega \mid |\frac{1}{n}\log\frac{1}{p(x_1,\cdots,x_n)}-\lambda^*|<\epsilon\})=1.$$

However, the Markov chain $\mathcal{X}$ that achieves λ^* is not necessarily unique, as mentioned earlier. Therefore, we need a way to define a set that contains all ϵ-$\mathcal{X}$-typical paths for all such $\mathcal{X}$, asymptotically, as follows. In this way, the set does not depend on any particular choice of the $\mathcal{X}$ and, hence, only depends on the graph G. Let $x_1,\cdots,x_{n+1}$ be a path of G, which has n edges. Define

$$B(x_1,\cdots,x_{n+1})=\prod_{1\leq i\leq n} b(x_i),$$

where $b(x_i)$ is the branching factor (i.e., out-degree) of node x_i. We define an ϵ-typical path of a graph G as follows. Notice that this definition depends only on the graph (not on any Markov chain).

Definition 2.3. Let $\epsilon > 0$. A path $x_1,\cdots,x_{n+1}$ on G is ϵ-typical if it satisfies

$$\frac{\log B(x_1,\cdots,x_{n+1})}{n} > \frac{1}{2}(\lambda^*-\epsilon), \tag{10}$$

where λ^* is the rate of the graph G.

Intuitively, (10) says that the average logarithmic branching factors on the path is higher than almost half of λ^*. Indeed, this is a good approximation as shown in the following Theorem 2.3. Intuitively, the theorem states that, asymptotically with probability 1, ϵ-typical paths include all ϵ-$\mathcal{X}$-typical paths. We first need a technical lemma.

Lemma 2.2. The number $K(n,\epsilon)$ of paths $x_1,\cdots,x_{n+1}$ of G satisfying

$$\frac{\log B(x_1,\cdots,x_{n+1})}{n} \leq \frac{1}{2}(\lambda^*-\epsilon) \tag{11}$$

is bounded by $2^{n(\lambda^*-\frac{\epsilon}{3})}$, for any n that is large enough.

Proof:
Define a random variable $X=\frac{1}{B(X_1,\cdots,X_{n+1})}$, where $X_1,\cdots,X_{n+1}$ has uniform joint distribution over node paths with n edges on G; i.e., $p(x_1,\cdots,x_{n+1})=\frac{1}{S(n)}$ for every path $x_1,\cdots,x_{n+1}$. Observe that

$$\sum_{x_1,\cdots,x_{n+1}\text{ is a path}}\frac{1}{B(x_1,\cdots,x_{n+1})}=1.$$

Hence, the mean $\mathrm{E}(X)=\frac{1}{S(n)}$ and the deviation $\mathrm{E}(X-\mathrm{E}(X))^2\leq\frac{1}{S(n)}$.Using Chebyshev inequality, we have

$$\mathrm{Prob}(|\frac{1}{B(X_1,\cdots,X_{n+1})}-\frac{1}{S(n)}|\geq(\frac{2^{n\frac{2}{3}\epsilon}}{S(n)})^{\frac{1}{2}})\leq\frac{1}{2^{n\frac{2}{3}\epsilon}}. \tag{12}$$

Therefore, since $X_1, \cdots, X_{n+1}$ has uniform joint distribution, the number $\hat{K}(n,\epsilon)$ of paths $x_1, \cdots, x_{n+1}$ of G satisfying

$$|\frac{1}{B(x_1,\cdots,x_{n+1})} - \frac{1}{S(n)}| \geq (\frac{2^{n\frac{2}{3}\epsilon}}{S(n)})^{\frac{1}{2}}, \tag{13}$$

using (12), is bounded as

$$\hat{K}(n,\epsilon) \leq \frac{S(n)}{2^{n\frac{2}{3}\epsilon}}. \tag{14}$$

From (3),

$$2^{n(\lambda^* - \frac{\epsilon}{3})} \leq S(n) \leq 2^{n(\lambda^* + \frac{\epsilon}{3})}, \tag{15}$$

when n large enough. Now (13) can be restricted to, when n is large,

$$B(x_1,\cdots,x_{n+1}) \leq [\frac{2^{n(\lambda^* - \frac{1}{3}\epsilon)}}{2^{n\frac{2}{3}\epsilon}}]^{\frac{1}{2}}. \tag{16}$$

That is, a sequence satisfying (16) also satisfies (13), when n large. Notice that (16) is exactly (11). Hence, $K(n,\epsilon) \leq \hat{K}(n,\epsilon)$. The lemma follows, using (14) and (15). □

Using Lemma 2.2 and Theorem 2.1,we can show:

Theorem 2.3. Let $\epsilon > 0$. For any G-represented $\mathcal{X}$ that achieves λ^*,

$$\limsup_{n\to\infty} \mathrm{Prob}(\frac{\log B(X_1,\cdots,X_{n+1})}{n} > \frac{1}{2}(\lambda^* - \epsilon)) = 1.$$

One can analogously define ω-ϵ-typical paths and, similar to Theorem 2.2, show that the ω-ϵ-typical paths take probability 1, asymptotically.

Definition 2.4. Let $\epsilon > 0$. An ω-path $x_1, \cdots, x_n, \cdots$ on G is ω-ϵ-typical if it has infinitely many ϵ-typical prefixes.

Theorem 2.4. For any G-represented $\mathcal{X}$ that achieves λ^*, μ(the set of ω-ϵ-typical paths) = 1.

3. Automata-theoretic characterization of typical paths

Automata theory studies the relationship between formal languages and automata accepting the languages. Finite automata form a simple class of automata which only have a finite amount of memory. They can be further equipped with unbounded storage devices such as counters. A counter is a nonnegative integer variable that can add one, subtract one or test against zero. A (nondeterministic) finite automata augmented with a finite number of counters is called a *nondeterministic multicounter automata* (NCA). It is well-known that NCAs with two counters are equivalent to Turing machines, which, in almost all cases, are too powerful to draw decidable results. To this end, it is necessary to make restrictions on the counters' behaviors. One such restriction is a reversal-bounded counter, proposed by Ibarra in his seminal paper [11] in 1978. A counter is k-reversal-bounded if it changes modes between nondecreasing and nonincreasing for at most k times. For example, the sequence of a counter values: 0, 0,

1, 1, 2, 2, 3, 3, 2, 2, 1, 1, 2, 2, 3, 3, 4, 4, $\cdots$ is 2-reversal-bounded. An NCA where each counter is k-reversal-bounded for some k in every execution is called a *nondeterministic reversal-bounded multi-counter automata* (NRCA). Note that an NRCA does not necessarily limit the number of moves to be finite.

The notion of reversal-bounded counters has found its applications in areas like verification, Diophantine equations, and P systems [13] (see [12] for a survey). For the purpose of this paper, we need a fundamental result [11] that links NRCAs with Presburger formulas. We first need some definitions.

Let **N** be the set of nonnegative integers and **Z** be the set of integers. Let **Y** = $\{y_1, \ldots, y_n\}$, for some n, denote a finite set of integer variables. An *atomic linear constraint* is a formula in the form of $\sum_{1 \leq i \leq n} a_i y_i \# b$, where the a_i's and b are integers and $\# \in \{>, =\}$. When $\#$ is $\equiv_d$ (for some constant d), the formula is called an *atomic mod-constraint*. A *Presburger formula* is a Boolean combination of atomic linear constraints and atomic mod-constraints. A set $\subseteq \mathbf{Z}^n$ is Presburger-definable if there exists a Presburger formula P on **Y** such that the set is exactly the set of the solutions for **Y** that make P true. It is well-known that Presburger formulas are closed under quantification.

An integer v can be represented by a unary string $[v]$, i.e., +5 as "$\dot{a}\dot{a}\dot{a}\dot{a}\dot{a}$", -5 as"$\hat{a}\hat{a}\hat{a}\hat{a}\hat{a}$", and 0 as the empty string. In this way, a tuple of integers $\langle v_1, \cdots, v_n \rangle$ can be represented as an n-tuple string $[v_1] \# \cdots \# [v_n]$, where symbol $\#$ is the delimiter. Therefore, a set $\subseteq \mathbf{Z}^n$ can be represented as an n-tuple language. In this paper, we need the following fundamental result in [11].

Theorem 3.1. For each n, a set $\subseteq \mathbf{Z}^n$ is Presburger definable iff the n-tuple language representing the set is accepted by a nondeterministic reversal-bounded multicounter automaton. The result remains when the automaton is further augmented with a counter (that is not necessarily reversal-bounded).

To study an automata-theoretic characterization of ω-ϵ-typical paths, we need a generalization, inspired by the machine model in [6], of NRCA to run over ω-words. A *counting Büchi automaton* M is a tuple

$$\langle \Sigma, S, Y, F, \delta, s_0 \rangle, \tag{17}$$

where Σ is a finite alphabet, S is a finite set of states with $s_0 \in S$ being the initial state, and $Y = \{y_1, \cdots, y_n\}$, for some n, is a finite set of nonnegative integer variables, which are called (monotonic) counters. Additionally, F is a finite set of *accepting conditions*; each condition is a finite set of *Presburger tests*. A Presburger test is a pair (s, P) of a state $s \in S$ and a Presburger formula P on the counters, (this definition of accepting conditions ensures Theorem 3.2 in below), and $\delta \subseteq S \times \Sigma \times \mathbf{N}^{|Y|} \times S$ specifies a finite set of transitions or edges. An edge in δ, $\langle s, a, \mathbf{incr}, s' \rangle$ denotes a transition from state s to state s' with $a \in \Sigma$ being the (input) symbol on the transition. The vector $\mathbf{incr} \in \mathbf{N}^{|Y|}$ specifies how the counters change after firing the transition: each counter y_i in Y is incremented by the i-th value $\mathbf{incr}[i]$ in the vector **incr**.

A *run* of M over an ω-word $\alpha = a_1 a_2 \cdots \in \Sigma^\omega$ is an infinite sequence of configurations

$$C_0 C_1 C_2 \cdots,$$

where each configuration $C_i = (s_i, \mathbf{V}_i)$ is a pair of a state s_i and counter values (as a vector) $\mathbf{V}_i$, satisfying the following conditions:

- In C_0, the state is the initial state and the counter values are all 0,

- for each $i \geq 0$, C_i can reach C_{i+1} by firing a transition in M while reading the input symbol a_i. That is, $(s_i, a_i, \mathbf{V}_{i+1} - \mathbf{V}_i, s_{i+1}) \in \delta$. In this case, we also write $C_i \stackrel{a_i}{\rightarrow} C_{i+1}$.

We say that a test (s, P) *satisfies* a configuration if the configuration is at state s and the counter values satisfy P; i.e., the configuration is in the form of $(s, \mathbf{V})$ with $P(\mathbf{V})$ holds. The run is *accepting* if there is an accepting condition in F such that each test (s, P) in the condition satisfies C_i for infinitely many i. The infinite word α is *accepted by* M if there is an accepting run of M over α. The (ω-) language accepted by M, denoted by $L^{\omega}(M)$, is the set of infinite words accepted by M.

Clearly, classic Büchi automata are a special case of counting Büchi automata where an accepting condition contains only one test $(s, \textbf{true})$. The following result is straightforward and analogous to the same result of Büchi automata (Lemma 1.2 in Chapter 4 in [14]).

Theorem 3.2. ω-Languages accepted by counting Büchi automata are closed under union and intersection.

A basic automata theory problem concerns emptiness. In this context, the emptiness problem for counting Büchi automata is to decide whether $L^{\omega}(M) = \emptyset$ for a given M.

To prove the emptiness problem of counting Büchi automata is decidable, we need more definitions. We say configuration $(s, \mathbf{V})$ can reach configuration $(s',\ \mathbf{V}')$, written

$$(s, \mathbf{V}) \rightsquigarrow_M (s', \mathbf{V}'),$$

if there is a word $a_1 \cdots a_{k-1}$ (for some k-1) such that

$$C_1 \stackrel{a_1}{\rightarrow} C_2 \stackrel{a_2}{\rightarrow} \cdots \stackrel{a_{k-1}}{\rightarrow} C_k \tag{18}$$

where $C_1 = (s, \mathbf{V})$ and $C_k = (s', \mathbf{V}')$. In this case, the execution in (18) *witnesses* the reachability.

Let R be a Presburger formula which defines a set $\subseteq S \times \mathbf{N}^n \times S \times \mathbf{N}^n$ (note that the finite state set S can be encoded as a finite range of integers). This R can be used to represent certain relation between configurations. We say that R is *transitive* if, for any $s, \mathbf{V}, s', \mathbf{V}', s'', \mathbf{V}''$, we have

$$R(s, \mathbf{V}, s', \mathbf{V}') \wedge R(s', \mathbf{V}', s'', \mathbf{V}'') \rightarrow R(s, \mathbf{V}, s'', \mathbf{V}'').$$

An infinite sequence of configurations $(s^1, \mathbf{V}^1) \cdots (s^i, \mathbf{V}^i) \cdots$ is an *ω-chain* of R if $R(s^i, \mathbf{V}^i, s^{i+1}, \mathbf{V}^{i+1})$ holds, for each $i \geq 1$. We need the following result whose proof can be found in [6].

Theorem 3.3. It is decidable whether a transitive Presburger formula has an ω-chain.

Now, we are ready to claim the following theorem, using Theorem 3.1 and Theorem 3.2.

Theorem 3.4. The emptiness problem for counting Büchi automata is decidable.

In practice, one may select a typical path that satisfies a specific pattern from a graph (e.g., a typical path on a graph satisfying that the number of times that nodes *green*, *yellow*, and *red* are passed are all the same). It will be awkward (if not possible) to perform Matrix operations on a Markov chain while maintaining the counting constraints, while searching for such a typical path. However, once the set of typical paths is characterized as a language accepted by an automaton M, the selection problem is

roughly asking if $L(M) \bigcap L_{\text{pattern}} \neq \varnothing$, where L_{pattern} is the set of all sequences (not being necessarily paths) satisfying the pattern. This is an emptiness problem, which has been a central research topic throughout automata theory. In below, we will show M is an NRCA.

We will show that ϵ-typicalness is roughly Presburger in below. Recall that a path $x_1, \cdots, x_{n+1}$ is ϵ-typical if

$$\log B(x_1, \cdots, x_{n+1}) > \frac{1}{2} n(\lambda^* - \epsilon). \tag{19}$$

We use count $\#_i$ to denote the number of appearances of the node q_i in $x_1, \cdots, x_n$ (in other words, $\#_i$ denotes the number of times that the path $x_1, \cdots, x_n$ passes the node q_i in the graph). We use b_i to denote the branching factor of the node q_i given in the graph. Notice that $B(x_1, \cdots, x_{n+1}) = \prod_{1 \leq i \leq |Q|} (b_i)^{\#_i}$. For convenience, let a_i denote $\log b_i$. Then, it is not hard to translate (19) into

$$\sum_{1 \leq i \leq |Q|} a_i \#_i > \frac{1}{2} n(\lambda^* - \epsilon), \tag{20}$$

where the a_i's are some nonnegative reals.

We use $\tau = 10^{-j}$, for some j, to denote a precision accurate to the jth decimal place. We call

$$\sum_{1 \leq i \leq |Q|} \lfloor \frac{a_i}{\tau} \rfloor \#_i > n \lceil \frac{\lambda^* - \epsilon}{2\tau} \rceil \tag{21}$$

the *lower-approximation* of (20), and

$$\sum_{1 \leq i \leq |Q|} \lceil \frac{a_i}{\tau} \rceil \#_i > n \lfloor \frac{\lambda^* - \epsilon}{2\tau} \rfloor \tag{22}$$

the *upper-approximation* of (20). Notice that the floor ($\lfloor \cdot \rfloor$) and the ceiling ($\lceil \cdot \rceil$) are a little different than usual in here: $\lfloor 2.01 \rfloor = \lfloor 2.00 \rfloor = 2$ and $\lceil 2.01 \rceil = \lceil 2.00 \rceil = 3$. Observe that (21) implies (20) and (20) implies (22). Therefore, a path $x_1 \cdots x_{n+1}$ is ϵ-lower-typical (resp. ϵ-upper-typical) if it satisfies (21) (resp. (22)). Clearly, the path is ϵ-typical if it is ϵ-upper-typical; the path is ϵ-lower-typical then it is ϵ-typical. In below, we will show that the set of ϵ-lower-typical (ϵ-upper-typical as well) paths is accepted by NRCA.

Theorem 3.5. The set of ϵ-lower-typical (ϵ-upper-typical as well) paths is accepted by an NRCA.

Proof:
For a path $x_1, \cdots, x_n$ on G, the Presburger constraints in (21) as well as in (22) can be checked using reversal-bounded counters. The result follows. □

Similarly, we can define an ω-ϵ-lower-typical (resp. ω-ϵ-upper-typical) path to be an ω-path such that there are infinitely many prefixes, each of which is an ϵ-lower-typical (resp. ϵ-upper-typical) path. The proof of the following theorem is straightforward.

Theorem 3.6. The set of ω-ϵ-lower-typical (ω-ϵ-upper-typical as well) paths is accepted by a counting Büchi automaton.

The significance of the above two characterizations (Theorem 3.5 and Theorem 3.6) is that, from [11] and Theorem 3.4, the emptiness problems for NRCAs and counting Büchi automata are decidable.

The characterizations immediately imply the decidability on whether there is a (lower- or upper-) typical path following certain patterns. For instance, consider a graph G and an NRCA M accepting a set of paths in G. For a given ϵ and a given precision, we use $L_{\text{lower}}(G)$ and $L_{\text{upper}}(G)$ to denote the set of ϵ-lower typical paths and the set of ϵ-upper typical paths of G, respectively. Clearly, whether $L_{\text{lower}}(G) \bigcap L(M) = \varnothing$ and whether $L_{\text{upper}}(G) \bigcap L(M) = \varnothing$ are all decidable (since languages accepted by NRCAs are closed under intersection [11]). $L(M)$ can be used to specify fairly complex and nonregular patterns: e.g.,

$$(\#_\pi(q_1) - \#_\pi(q_2) > 5) \wedge (\#_\pi(q_3) - \#_\pi(q_4) > 7) \tag{23}$$

where $\#_\pi(q)$ indicates the number of times that π passes node q in G. Hence, both of the following problems are decidable:

- Is there a lower-ϵ-typical path in G that satisfies (23)?
- Is there an upper-ϵ-typical path in G that satisfies (23)?

Similarly, let M be a counting Büchi automaton to accept a set of ω-paths of G. We use $L^\omega_{\text{lower}}(G)$ and $L^\omega_{\text{upper}}(G)$ to denote the ω-ϵ-lower-typical paths and ω-ϵ-upper-typical paths of G, respectively. Clearly, whether $L^\omega_{\text{lower}}(G) \bigcap L^\omega(M) = \varnothing$ and whether $L^\omega_{\text{upper}}(G) \bigcap L^\omega(M) = \varnothing$ are both decidable (using Theorem 3.2). Notice that $L^\omega(M)$ can also be used to specify fairly complex and non-ω-regular patterns: e.g., ω-paths π^ω of G such that

- there are infinitely many times that π^ω passes through node q_1 and, at each such time, the number of times node q_1 has been passed is greater than the number of times node q_2 has been passed, and,
- there are infinitely many times that π^ω passes through node q_2 and, at each such time, the number of times node q_2 has been passed is greater than the number of times node q_3 has been passed.

Hence, the following problems are decidable:

- Is there an ω-ϵ-lower-typical path in G that satisfies the aforementioned ω-pattern?
- Is there an ω-ϵ-upper typical path in G that satisfies the aforementioned ω-pattern?

Furthermore, when

$$L^\omega_{\text{upper}}(G) \bigcap L^\omega(M) = \varnothing, \tag{24}$$

it really says that the ω-paths contained in the ω-pattern $L^\omega(M)$ take probability 0 (Theorem 2.4). In other words, the ω-pattern $L^\omega(M)$ does not carry "essential information" (Notice that the ω-pattern is also able to specify safety properties that concern finite prefixes of ω-paths only.). The ω-pattern can also be specified using a generalized LTL formula [7, 6]. Such a formula is often employed in a property specification in a specification language such as Promela [10]. In this case, the graph G serves as the system specification. Hence, from (24), a software engineer may conclude that the property specification does not catch any "essential" information of the system specification.

4. Experiment

Theoretically, generating a typical path is difficult. This is because there is no known polynomial time algorithm to solve the emptiness problem of NRCAs. To this end, in this section, we use a practically efficient model-checking [4] tool SPIN [10] to generate ϵ-typical paths.

In this experiment, we are to choose ϵ-typical paths from a statechart. In the original statechart paper [9], the author uses Citizen Quartz Multi-Alarm III wristwatch as an example to exemplify the way of generating statecharts to explain the semantics of the system. In this section, for simplicity, we only use the $\mathrm{displays}$ unit of the watch statechart as an instance to illustrate how to generate ϵ-typical and non-ϵ-typical paths (i.e., a path that is not ϵ-typical) from a statechart.

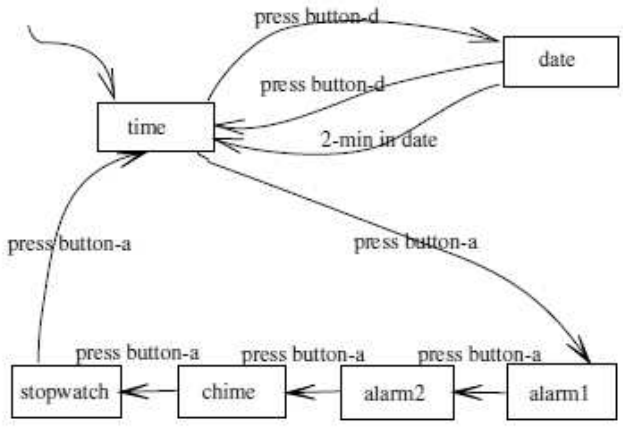

Figure 1. The statechart S of the $\mathrm{displays}$ units (Figure 9 in [9])

Figure 1 shows the statechart S of the $\mathrm{displays}$ unit. In S, a node denotes a state and a labeled edge denotes a transition with its label being the input symbol from one state to another. For example, the transition $\mathrm{time} \overset{\mathrm{button-a}}{\longrightarrow} \mathrm{alarm1}$ represents that the $\mathrm{displays}$ unit can transit from the state time to the state $\mathrm{alarm1}$ when button-a is pressed. Figure 1 indicates two functions: first, continuing pressing button-a, the state can transit in the state sequence of $\mathrm{time}, \mathrm{alarm1}, \mathrm{alarm2}, \mathrm{chime}, \mathrm{stopwatch}$; second, pressing button-d, the system can transit from state time to state date. If pressing the button-d again or waiting more than 2 minutes, the system goes back to state time again.

We use Promela (a specification language provided with SPIN) to code the statechart S and write LTL formulas to specify a property on a path being ϵ-typical (resp. non-ϵ-typical). Hence, a non-ϵ-typical path and an ϵ-typical path can be generated using the counter example generation function of SPIN.

In the experiments, we take $\epsilon = 0.007$. The rate $\lambda^* = 0.5973$ of the statechart S is computed using a Matlab script written by Linmin Yang. SPIN generates two sequences with length of 19: one is an ϵ-typical sequence,

$$\text{(press button-a)}^{15}\text{(press button-d)}^{2}\text{(press button-a)}^{2},$$

and the other is a non-ϵ-typical sequence,

$$(\text{press button-a})^{19}.$$

Our Promela code runs on the Ubuntu 10.10 OS which is installed on an Oracle VirutalBox. The hosting computer is a window platform PC with Intel P8600 processor and 2GB memory. The running times of the experiments are all negligible.

5. Conclusion and discussion

We have introduced the concepts of ϵ-typical paths and ω-ϵ-typical paths on a graph, and the typical paths (ω-typical as well) take probability 1, asymptotically. We also provide an automata-theoretic characterization for those paths; e.g., typical paths accepted by NRCAs and ω-typical paths accepted by counting Büchi automata. Finally, we use SPIN to show how to generate ϵ-typical paths in practice.

A magic number $\frac{1}{2}$ is used to define the ϵ-typical paths in Definition 2.3. Can we use other larger numbers instead of $\frac{1}{2}$ in order to generate a smaller typical set taking probability 1? We believe that the answer is no. The reasons are explained as follows. First, if we use 1 instead of $\frac{1}{2}$ in the formula, (10) can be satisfied only when the transition probabilities of the graph are uniform. The uniform distribution is not consistent with our intended meaning of typical paths. Second, we conjecture that $\frac{1}{2}$ is the maximal value that can be used in the range $[\frac{1}{2}, 1)$ to achieve a similar AEP result. ($\frac{1}{2}$ used in Lemma 2.2 comes from the Chebyshev inequality in its proof. The inequality is the root of the law of large numbers, which is also "equivalent" to AEP.)

Additionally, typical paths of a graph can be generalized to "typical words" in a language. Let L be a language (not necessarily regular) and $w \in L$ be a word. Define $B(w)$, called the B-value of w, to be

$$\prod_{w' \prec w} \text{branch}(w'),$$

where w' is a proper prefix of w (i.e., $w' \prec w$) and branch(w') is $\big|\{a \in \Sigma : \exists w'', w'aw'' \in L\}\big|$. For example, Let $L = \{\text{abc}, \text{cde}, \text{adf}\}$. Then, $B(\text{a}) = 2$ and $B(\text{ab}) = 4$. Suppose now that L is regular. For a given $\epsilon > 0$, we say that a word $w \in L$ is ϵ-typical if its B-value satisfies

$$\frac{\log B(w)}{n} > \frac{1}{2}(\lambda^* - \epsilon), \tag{25}$$

where λ^* is computed as the rate of a graph representing a deterministic finite automaton accepting L. In this way, we can obtain an *information-concentrated-core* (ICC) of a regular language. The ICC of a regular language is defined as the set of all ϵ-typical words in this regular language for a given $\epsilon > 0$. From the results of this paper, the ICC is accepted by an NRCA after applying a precision to (25) (actually, it is context free), and satisfies Theorem 2.3: only words in the ICC carry nontrivial information. We conjecture that a similar notion of (25) can be created for some nonregular languages (e.g., context-free languages and languages accepted by NRCAs) and hence, the ICCs deserve further investigation in automata theory.

Acknowledgements. The authors would like to thank Oscar H. Ibarra, Eric Wang and Linmin Yang for valuable discussions. The authors also thank the anonymous referees for detailed comments.

References

[1] Ammann, P., Offutt, J.: *Introduction to Software Testing*, Cambridge University Press, 2008.

[2] Beizer, B.: *Software Testing Techniques*, second edition, Van Nostrand Reinhold, 1990.

[3] Chomsky, N., Miller, G. A.: Finite state languages, *Information and Control*, **1**(2), 1958, 91–112.

[4] Clarke, E. M., Grumberg, O., Peled, D. A.: *Model Checking*, MIT Press, 1999.

[5] Cover, T. M., Thomas, J. A.: *Elements of Information Theory*, second edition, Wiley-Interscience, 2006.

[6] Dang, Z., Ibarra, O. H.: On the existence of ω-chains for transitive mixed linear relations and its applications, *International Journal of Foundations of Computer Science*, **13**(6), 2002, 911–936.

[7] Dang, Z., Ibarra, O. H., Pietro, P. S.: Liveness verification of reversal-bounded multicounter machines with a free counter, *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science* (R. Hariharan, M. Mukund, V. Vinay, Eds.), LNCS 2245, Springer, Berlin, 2001.

[8] Goodenough, J. B., Gerhart, S. L.: Toward a theory of test data selection, *Proc. International Conference on Reliable Software*, ACM, 1975.

[9] Harel, D.: Statecharts: A visual formalism for complex systems, *Sci. Comput. Program.*, **8**(3), 1987, 231–274.

[10] Holzmann, G. J.: The model checker SPIN, *IEEE Transactions on Software Engineering*, **23**(5), 1997, 279–295.

[11] Ibarra, O. H.: Reversal-bounded multicounter machines and their decision problems, *J. ACM*, **25**(1), 1978, 116–133.

[12] Ibarra, O. H., Dang, Z., Yang, L.: On counter machines, reachability problems, and Diophantine equations, *Int. J. Found. Comput. Sci.*, **19**(4), 2008, 919–934.

[13] Păun, G.: Computing with membranes, *J. Comput. Syst. Sci.*, **61**(1), 2000, 108–143.

[14] Thomas, W.: Automata on infinite objects, in: *Handbook of Theoretical Computer Science: Formal Models and Semantics* (J. Van Leeuwen, Ed.), MIT Press, 1990, 133–192.

[15] Vardi, M. Y.: Probabilistic linear-time model checking: an overview of the automata-theoretic approach, *Formal Methods for Real-Time and Probabilistic Systems* (J. Katoen, Ed.), LNCS 1601, Springer, Berlin, 1999.

[16] Yang, L., Dang, Z., Fischer, T. R.: Information gain of black-box testing, (To appear in *Formal Aspects of Computing*.).

[17] Yang, L., Dang, Z., Fischer, T. R.: An information-theoretic complexity metric, Submitted, 2011.

[18] Yeh, J.: *Real Analysis: Theory of Measure and Integration*, World Scientific, 2006.