A Purely Model-Theoretic Proof of the Exponential Succinctness Gap Between CTL$^+$ and CTL

Martin Lange

Institut für Informatik, Ludwig-Maximilians-Universität München

Abstract

We provide a conceptually simple and elementary proof of the exponential succinctness gap between the two branching time temporal logics CTL$^+$ and CTL. It only uses CTL's small model property instead of automata- or game-theory and combinatorics as in previous proofs by Wilke and Adler/Immerman.

Key words: specification languages, temporal logics, small model property

1 Introduction

CTL is the branching time temporal logic in which every path operator like $\mathtt{U}$ (*until*), $\mathtt{F}$ (*finally*), etc. is immediately preceded by a path quantifier $\mathtt{E}$ or $\mathtt{A}$. It is, for instance, straight-forward to say in CTL that "there is a path on which the atomic proposition q holds at some point": $\mathtt{EF}q$. CTL$^+$ is the extension of CTL that allows boolean combinations of path operators under a path quantifier. For example, $\varphi = \mathtt{E}(\mathtt{F}q_1 \wedge \mathtt{F}q_2)$ says that "there is a path on which q_1 holds somewhere and q_2 holds somewhere". Syntactically, this is not a CTL formula anymore. However, Emerson and Halpern have shown that for every CTL$^+$ formula there is an equivalent CTL formula [2]. The trick, in general, is to consider all possibilities of orders in which events occur along a path. For example, φ is equivalent to $\mathtt{EF}(q_1 \wedge \mathtt{EF}q_2) \vee \mathtt{EF}(q_2 \wedge \mathtt{EF}q_1)$ which is in CTL again and says: "there is a path on which q_1 occurs and from there, there is another path on which q_2 occurs, or vice-versa". Note that this is just a complicated way of expressing that both q_1 and q_2 must occur on some path. It is easy to imagine that a generalisation with n propositions q_i leads to an

Email address: `martin.lange@ifi.lmu.de` (Martin Lange).

$O(n)$!-blow-up in formula size. Emerson and Halpern have indeed shown that every CTL$^+$ formula of size m can be translated into a CTL formula of size at most $O(m)!$ [2].

It is known that this cannot be improved. Wilke [4] has shown that there is a family Φ_n of simple CTL$^+$ formulas such that every equivalent family Ψ_n of CTL formulas must be of size $\sqrt{\frac{3}{|\Phi_n|}} \cdot 2^{|\Phi_n|/3}$. The proof uses automata-theory and is combinatorially involved. Later, Adler and Immerman [1] have shown using Ehrenfeucht-Fraïssé games that every family of CTL formulas equivalent to these must indeed have size $(|\Phi_n|/3)!$. Finally, Johannsen and Lange [3] have obtained a weaker bound but a stronger statement about the kind of equivalence between the two logics: they presented a family of CTL$^+$ formulas Φ_n s.t. every family of CTL-formulas which are equivalent w.r.t. satisfiability only must be of size $2^{\Omega(\sqrt{|\Phi_n|})}$.

Here we provide another proof of the exponential succinctness gap. It matches Wilke's bound asymptotically (albeit having worse constants). Hence, it is not optimal. However, it is an elementary and conceptually simple proof that does not rely on "external machinery" like automata- or game-theory. It uses CTL's small model property – every satisfiable CTL formula of size m has a model of size at most exponential in m proved by Emerson and Halpern [2]. We then construct a series of CTL$^+$ formulas that are satisfiable but whose smallest models are of doubly exponential size. It follows by a simple comparison of these two terms that all equivalent CTL-formulas must be exponentially bigger.

This technique is not restricted to CTL and CTL$^+$. The essence of such a proof is presented in the following general lemma which may yield succinctness results for other logics as well. For two logics $\mathcal{L}$ and $\mathcal{L}'$ interpreted over the same class of structures we write $\mathcal{L} \le \mathcal{L}'$ if for every $\Phi \in \mathcal{L}$ there is a $\Psi \in \mathcal{L}'$ such that $\Phi \equiv \Psi$, i.e. every $\mathcal{L}$-property is also $\mathcal{L}'$-definable. We write $|\Phi|$ to denote the syntactic size of a formula Φ. We say that a logic $\mathcal{L}$ has the *small model property* of size $f : \mathbb{R} \to \mathbb{R}$ if every satisfiable $\Phi \in \mathcal{L}$ has a model of size at most $f(|\Phi|)$. $\mathcal{L}$ has the *large model property* of size $g : \mathbb{R} \to \mathbb{R}$ if there are satisfiable $\Phi_n \in \mathcal{L}$, $n \in \mathbb{N}$, such that every model of Φ_n must be of size at least $g(|\Phi_n|)$ for all $n \in \mathbb{N}$. Clearly, the large model property of a logic must provide a lower bound to the small model property of the same logic. However, this need not be the case for different logics. There is a *succinctness gap* of size $h : \mathbb{R} \to \mathbb{R}$ between $\mathcal{L}$ and $\mathcal{L}'$ if there are satisfiable $\mathcal{L}$-formulas Φ_n, $n \in \mathbb{N}$, such that for every family Ψ_n of $\mathcal{L}'$-formulas we have: if $\Phi_n \equiv \Psi_n$ then $|\Psi_n| \ge h(|\Phi_n|)$, for all $n \in \mathbb{N}$.

Lemma 1 (Gap-Lemma) *Let $\mathcal{L}, \mathcal{L}'$ be two logics such that $\mathcal{L} \le \mathcal{L}'$. If*

(1) $\mathcal{L}'$ has the small model property of size f for some invertible $f : \mathbb{R} \to \mathbb{R}$,

and

(2) $\mathcal{L}$ has the large model property of size g for some $g : \mathbb{R} \to \mathbb{R}$.

then there is an $(f^{-1} \circ g)$-succinctness gap between $\mathcal{L}$ and $\mathcal{L}'$.

PROOF. Because of (2) there are satisfiable $\mathcal{L}$-formulas Φ_n such that for all their smallest models M_n we have: $|M_n| \geq g(|\Phi_n|)$. Now suppose that Ψ_n, $n \in \mathbb{N}$, is a family of equivalent $\mathcal{L}'$-formulas. Thus, each M_n is also a model of Ψ_n, and because of (1) we have $|M_n| \leq f(|\Psi_n|)$. Since f is invertible we obtain: $|\Psi_n| \geq f^{-1}(g(|\Phi_n|))$. $\square$

In Sect. 2 we introduce CTL and CTL$^+$ formally. In Sect. 3 we apply the Gap-Lemma to these two logics. Since the small model property for CTL has already been proved we only need to show the large model property for CTL$^+$.

2 Preliminaries

A *transition system* is a tuple $\mathcal{T} = (\mathcal{S}, \rightarrow, L)$ where $\mathcal{S}$ is a set of states, $\rightarrow \subseteq \mathcal{S} \times \mathcal{S}$ a transition relation and $L : \mathcal{P} \rightarrow 2^{\mathcal{S}}$ a function that assigns to each q in some non-empty set $\mathcal{P}$ of atomic propositions the set of states $L(q)$ in which q holds. Here we assume the transition relation to be *total*: for all $s \in \mathcal{S}$ there is a $t \in \mathcal{S}$ such that $s \rightarrow t$. A *path* is an infinite sequence $\pi = s_0, s_1, \ldots \in \mathcal{S}^\omega$ such that $s_i \rightarrow s_{i+1}$ for all $i \in \mathbb{N}$. With π^k we denote the *k-th suffix* of π, namely the path $s_k, s_{k+1}, \ldots$.

Formulas of the branching time temporal logic CTL$^+$ over $\mathcal{P}$ are given by the following grammar. Let $q \in \mathcal{P}$.

$$\Phi \ ::= \ q \mid \Phi \vee \Phi \mid \neg\Phi \mid \mathtt{E}\psi \qquad\qquad \psi \ ::= \ \psi \vee \psi \mid \neg\psi \mid \mathtt{X}\Phi \mid \Phi\mathtt{U}\Phi$$

Here we use capital Greek letters to denote *state formulas* and little ones for *path formulas.* The latter only occur as genuine subformulas of the former. We also use the usual abbreviations for $\wedge, \rightarrow, \leftrightarrow$, etc. from propositional logic, and from temporal logic: $\mathtt{A}\psi := \neg\mathtt{E}\neg\psi$, $\mathtt{F}\Phi := \mathtt{tt}\mathtt{U}\Phi$ where $\mathtt{tt} := q \vee \neg q$ for some $q \in \mathcal{P}$, and $\mathtt{G}\Phi := \neg\mathtt{F}\neg\Phi$. The logic CTL is obtained as a fragment of CTL$^+$ by removing the clauses for the boolean operators from the definition of ψ in the grammar above. The set of subformulas $Sub(\varphi)$ of a CTL$^+$ formula is defined in the usual way, and is used to measure the size of a formula: $|\varphi| := |Sub(\varphi)|$. Note that $|\mathtt{E}(\bigwedge_{i=1}^{n} \mathtt{F}q_i)| \leq 3n$ for example. [1]

[1] . . . if we allow the binary $\wedge$ as a first-class operator. If it counts as an abbreviation the bound is $4n + 1$.

Formulas of CTL$^+$ are interpreted over states and paths of a transition system $\mathcal{T} = (\mathcal{S}, \to, L)$, reflecting the two types of formulas.

$$
\begin{array}{lll}
\mathcal{T}, s \models q & \text{iff} & q \in L(s) \\
\mathcal{T}, s \models \Phi_1 \vee \Phi_2 & \text{iff} & \mathcal{T}, s \models \Phi_1 \text{ or } \mathcal{T}, s \models \Phi_2 \\
\mathcal{T}, s \models \neg\Phi & \text{iff} & \mathcal{T}, s \not\models \Phi \\
\mathcal{T}, s \models \mathtt{E}\psi & \text{iff} & \text{there is a path } \pi = s, \ldots \text{ with } \mathcal{T}, \pi \models \psi \\
\mathcal{T}, \pi \models \psi_1 \vee \psi_2 & \text{iff} & \mathcal{T}, \pi \models \psi_1 \text{ or } \mathcal{T}, \pi \models \psi_2 \\
\mathcal{T}, \pi \models \neg\psi & \text{iff} & \mathcal{T}, \pi \not\models \psi \\
\mathcal{T}, \pi \models \mathtt{X}\Phi & \text{iff} & \pi^1 = s, \ldots \text{ and } \mathcal{T}, s \models \Phi \\
\mathcal{T}, \pi \models \Phi_1 \mathtt{U} \Phi_2 & \text{iff} & \pi = s_0, s_1, \ldots \text{ and there is a } k \in \mathbb{N} \text{ with } \mathcal{T}, s_k \models \Phi_2 \\
 & & \text{and for all } j < k : \mathcal{T}, s_j \models \Phi_1
\end{array}
$$

Two (state) formulas are equivalent, written $\Phi \equiv \Psi$ iff for all $\mathcal{T}$ and all states s we have: $\mathcal{T}, s \models \Phi$ iff $\mathcal{T}, s \models \Psi$. Even though CTL is a strict syntactic fragment of CTL$^+$ the two logics are equi-expressive as shown by Emerson and Halpern.

Theorem 2 ([2]) *For every family Φ_n, $n \in \mathbb{N}$, of CTL$^+$ formulas there are CTL formulas Ψ_n, $n \in \mathbb{N}$, such that $\Psi_n \equiv \Phi_n$ and $|\Psi_n| \leq O(|\Phi_n|)!$.*

3 Proof of the Succinctness Gap

As noted above, we proceed as follows. First we will show that there are satisfiable CTL$^+$ formulas Φ_n which do not have small models, i.e. all of their models must have at least doubly-exponential size. These models must have a path which is divided into sequences of some $f(n)$ many states for some suitable function $f(n)$. Each such sequence will then be used to model the state of a binary counter such that adjacent sequences model successive counter values. In other words, Φ_n postulates the existence of a path with labels of the form

$$\underbrace{000\ldots0}_{f(n)}\underbrace{100\ldots0}_{f(n)}\underbrace{010\ldots0}_{f(n)}\underbrace{110\ldots0}_{f(n)}\ldots\ldots\underbrace{111\ldots1}_{f(n)}\ldots$$

Of course this could easily be done using a formula of size $O(f(n) \cdot 2^{f(n)})$ but it would be worthless for any meaningful function f. In order to achieve the best possible succinctness gap result, $f(n)$ must be as large as possible while the formula Φ_n describing it should be as small as possible.

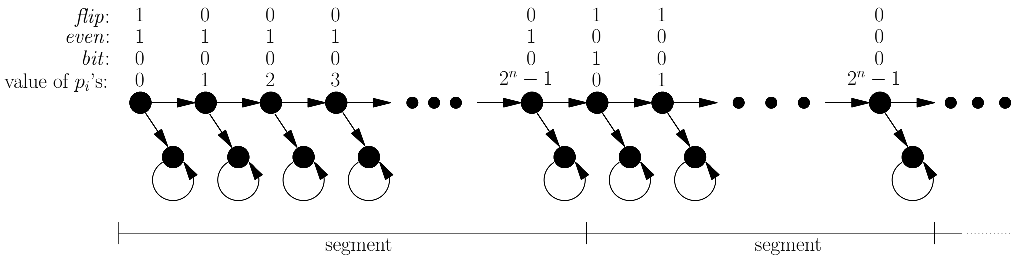

Fig. 1. A model of Φ_n.

It is possible to build a Φ_n of size linear in n that yields an $f(n) = 2^n$. Let $\mathcal{P}_n = \{p_0, \ldots, p_{n-1}, on, even, bit, flip\}$. We use proposition *on* to mark a certain path, propositions $p_0, \ldots, p_{n-1}$ to model a counter that is increased successively along that path, and *even*,*bit* and *flip* in order to model a second counter with 2^n many bits which can therefore count up to 2^{2^n}. Φ_n will ensure that all counter values "occur" somewhere along that path which, hence, has to be of doubly-exponential length.

Formulas Φ_n will consist of eight conjuncts, to be defined in the following.

$$\Phi_n \;:=\; \mathit{Path} \wedge \mathit{Min}_n \wedge \mathit{Next}_n \wedge \mathit{Even}_n \wedge \mathit{Cache}_n \wedge \mathit{Init}_n \wedge \mathit{Flip}_n \wedge \mathit{Inc}_n$$

The first one describes the existence of the path mentioned above. We furthermore require that every state on this path has a successor which is not on this path. This successor will be called the *cache* of the corresponding state on this path, see below for details.

$$\mathit{Path} \;:=\; on \;\wedge\; \mathtt{AG}\Big(\big(on \;\rightarrow\; (\mathtt{EX}on \wedge \mathtt{EX}\neg on)\big) \;\wedge\; \big(\neg on \;\rightarrow\; \mathtt{AX}\neg on\big)\Big)$$

Models of formula *Path* look like the one that is schematically depicted in Fig. 1.[2] The states in the upper line are those satisfying *on*, the ones below are the cache states not satisfying *on*. Their labels will be required to differ from their predecessors' only in the proposition *on*. Thus, we only show the labels of the states on the *on*-path.

Propositions $p_0, \ldots, p_{n-1}$ will be used to assign to each state on this path a natural number in the range $0, \ldots, 2^n - 1$ represented in binary coding. These should then act as a counter, i.e. the values should be increased successively along the *on*-path. It is easy to assert the minimal and maximal value in this,

[2] Since CTL$^+$ is bisimulation-invariant one can only characterise the models of a formula upto bisimilarity. Furthermore, models can have other parts (not just unreachable ones) which the formula *Path* makes no assertions about. Later we will be interested in the smallest models of Φ_n anyway for which such parts are obsolete.

marking the beginning and the end of each segment on this path.

$$Min_n \ := \ \bigwedge_{i=0}^{n-1} \neg p_i \qquad\qquad Max_n \ := \ \bigwedge_{i=0}^{n-1} p_i$$

Incrementation of the counter values can be formulated in CTL$^+$ as follows. Note that in binary increment the least significant digit always flips and the others flip only if all lower bits are set. Equivalently, a bit flips only if the one below flips from 1 to 0.

$$Next_n \ := \ \mathtt{AGA}\Big(\ \mathtt{X}on \ \rightarrow \ \big((p_0 \leftrightarrow \mathtt{X}\neg p_0) \ \wedge \ \bigwedge_{i=1}^{n-1} \mathtt{X}p_i \leftrightarrow (p_i \leftrightarrow (p_{i-1} \rightarrow \mathtt{X}p_{i-1}))\big)\Big)$$

This results in the numbers called "value of p_i's" in Fig. 1. It also partitions the path outlined with the proposition *on* into segments of length 2^n. The proposition *even* is now used to mark every second of these segments. This will be used later on when the second counter is being modelled.

$$Even_n \ := \ even \ \wedge \ \mathtt{AGA}\Big(\ \mathtt{X}on \ \rightarrow \ \big(\ (even \leftrightarrow \mathtt{X}even) \leftrightarrow \mathtt{X}\neg Min_n \ \big) \ \Big)$$

Next we axiomatise the cache states in the sense that they mirror their predecessor on that path w.r.t. all propositions other than *on*. Let $\mathcal{P}'_n := \mathcal{P}_n \setminus \{on\}$.

$$Cache_n \ := \ \mathtt{AGA}\big(\ \mathtt{X}\neg on \ \rightarrow \ \bigwedge_{q \in \mathcal{P}'_n} (q \leftrightarrow \mathtt{X}q) \ \big)$$

Note that this even implies that once the *on*-path is left all reachable states carry the same $\mathcal{P}'$ label.

We now use the proposition *bit* to model a counter with 2^n many bits – one for each state in such a segment. The first such state, i.e. the one representing the number 0 carries the least significant bit of this counter. It is easy to require this counter to have minimal value in the first segment.

$$Init_n \ := \ \mathtt{A}\big(\mathtt{G}on \ \rightarrow \ (\neg bit)\mathtt{U}(\neg bit \wedge \mathtt{X}Min_n) \ \big)$$

Next we need to say that the counter value is increased by one when moving from one segment to the next one along *on*. In order to do so we use another atomic proposition *flip* that marks all those bits which need to be flipped in the next incremental step. This is done in the usual way: the least significant bit is always flipped, and another bit is flipped only if the one below is set and flipped as well.

$$\begin{aligned} Flip_n \ := \ \mathtt{AGA}\Big(\ \mathtt{G}on \ \rightarrow \ & (Min_n \rightarrow flip) \ \wedge \\ & \neg \mathtt{X}Min_n \ \rightarrow \ \big(\ \mathtt{X}flip \leftrightarrow (flip \wedge bit) \ \big) \ \Big) \end{aligned}$$

This can then be used to model the increment of that counter. We need the values of the counter bits to flip or remain the same according to the value of the proposition *flip* when moving to the same bit in the next segment. This requires an assertion of the kind "something holds in 2^n many steps" which is in general not easy to express in CTL$^+$ with a formula of less than exponential size. Here we employ two tricks to model such an assertion via a statement of the weaker form "something holds eventually".

- In a position representing an arbitrary bit of the counter and a value $i \in \{0, \dots, 2^n - 1\}$ we quantify over a path that will *eventually* reach a state representing the same value i. This can be done using the cache states.
- There are in general several of these paths starting in an *on*-state representing i. The first one runs into its cache state immediately, the second one runs into the cache state of the i-th state in the next segment, the third one does the same for the segment after that etc. In order to restrict the quantification to the second one in this enumeration – the only one that connects a bit to the same bit in the next segment – we require the value of the proposition *even* to change exactly once on this path.

First of all we build a path formula that restricts the quantification accordingly.

$$
\begin{aligned}
res_n \ := \ & \mathtt{X}on \wedge \Big(\ Min_n \ \rightarrow \ \neg\big(\ \mathtt{F}(even \wedge Max_n) \ \wedge \ \mathtt{F}(\neg even \wedge Max_n) \ \big) \ \Big) \\
& \wedge \ \Big(\ \neg Min_n \ \rightarrow \ \neg\big(\ \mathtt{F}(even \wedge Min_n) \ \wedge \ \mathtt{F}(\neg even \wedge Min_n) \ \big) \ \Big) \\
& \wedge \ \bigwedge_{i=0}^{n-1} \Big(\big(\ p_i \rightarrow \mathtt{F}(\neg on \wedge p_i) \ \big) \ \wedge \ \big(\ \neg p_i \rightarrow \mathtt{F}(\neg on \wedge \neg p_i) \ \big) \Big)
\end{aligned}
$$

The requirement on the bit values to change correctly is then only a simple case distinction.

$$
\begin{aligned}
Inc_n \ := \ \mathtt{AGA}\Big(\ res_n \ \rightarrow \ \Big(\ & flip \ \rightarrow \ \big(\ bit \rightarrow \mathtt{F}(\neg on \wedge \neg bit) \ \big) \\
& \qquad \wedge \big(\ \neg bit \rightarrow \mathtt{F}(\neg on \wedge bit) \ \big) \ \Big) \\
\wedge \Big(\ & \neg flip \ \rightarrow \ \big(\ bit \rightarrow \mathtt{F}(\neg on \wedge bit) \ \big) \\
& \qquad \wedge \big(\ \neg bit \rightarrow \mathtt{F}(\neg on \wedge \neg bit) \ \big) \ \Big) \ \Big)
\end{aligned}
$$

Clearly, the size of the Φ_n is only linear in n since so is the size of each conjunct.

Lemma 3 $|\Phi_n| = O(n)$.

To show that the Φ_n witness the large model property of CTL$^+$ of doubly-exponential size we need to show that they are satisfiable and that their models must have that size indeed.

Lemma 4 *For all $n \in \mathbb{N}$: Φ_n is satisfiable.*

PROOF. Let $\mathcal{T}_n = (\mathcal{S}, \rightarrow, L)$ be the transition system shown in Fig. 1. Formally, $\mathcal{S} = \{s_k, s'_k \mid k \in \mathbb{N}\}$; the transitions are $s_k \rightarrow s_{k+1}$, $s_k \rightarrow s'_k$, and $s'_k \rightarrow s'_k$ for all $k \in \mathbb{N}$; and the labeling is given as:

$$\begin{aligned} L(on) &= \{s_k \mid k \in \mathbb{N}\} \\ L(p_i) &= \{s_k, s'_k \mid \text{the } i\text{-th bit of } k \ mod \ 2^n \text{ is } 1\} \\ L(even) &= \{s_k, s'_k \mid k \ div \ 2^n \text{ is even }\} \\ L(bit) &= \{s_k, s'_k \mid \text{the } (k \ mod \ 2^n)\text{-th bit of } (k \ div \ 2^n) \text{ is } 1\} \\ L(flip) &= \{s_k, s'_k \mid \forall h = 0, \ldots, (k \ mod \ 2^n) - 1 : \text{the } h\text{-th bit of } (k \ div \ 2^n) \text{ is } 1\} \end{aligned}$$

It is not hard to verify that $\mathcal{T}, s_0 \models \Phi_n$ holds indeed. □

Lemma 5 *For all $n \geq 1$: every model of Φ_n has at least $2^n \cdot 2^{2^n}$ many states.*

PROOF. Suppose that $\mathcal{T} = (\mathcal{S}, \rightarrow, L)$ with some state $s_0 \in \mathcal{S}$ is a model of Φ_n. Since $\mathcal{T}, s_0 \models Path$ there is a path $\pi = s_0, s_1, \ldots$ that satisfies *on* everywhere. Because of this and $Cache_n$, every s_i has a cache s'_i that does not differ in the atomic labels except for proposition *on*. Hence, we must have $s_i \neq s'_j$ for all $i, j \in \mathbb{N}$.

The *small type* assigns to each state s_k in π an $st(s_k) \in \mathbb{N}$. It is defined as

$$st(s_k) \ := \ \sum_{i=0}^{n-1} b_i \cdot 2^i \qquad \text{with } b_i \ = \ \begin{cases} 1 & , \text{ if } s_k \in L(p_i) \\ 0 & , \text{ otherwise} \end{cases}$$

Clearly, we have $st(s_k) \in \{0, \ldots, 2^n - 1\}$ for all $k \in \mathbb{N}$. Now for all $j, k \in \mathbb{N}$ we have $st(s_j) \neq st(s_k) \Rightarrow s_j \neq s_k$ because binary representations are unique, and equal states must have equal labels. Formula $Next_n$ ensures that $st(s_k) = st(s_j)$ only if $k \equiv j \ mod \ 2^n$.

Let $ran(s_k) = \{j \mid k \ div \ 2^n = j \ div \ 2^n\}$ be the *range* of a state s_k on π – the set of all indices of states in its segment. Clearly $|ran(s_k)| = 2^n$ for all $k \in \mathbb{N}$, and the range of a state is a closed segment of the path π. Furthermore, this induces an equivalence relation: $s_k \approx s_j$ iff $ran(s_k) = ran(s_j)$. Hence, these segments are non-overlapping, of fixed size 2^n, cover the entire path π, and are always of the form $s_k, s_{k+1}, \ldots, s_{k+2^n-1}$ such that $k \equiv 0 \ mod \ 2^n$. This also means that for all s_k and all pairwise different $i, j \in ran(s_k)$ we have $st(s_i) \neq st(s_j)$. We write $ran(s_k) = ran(s_j) + 1$ iff $1 \leq j - k \leq 2^n$ and there is an h with $k \leq h \leq j$ such that $st(s_h) = 0$.

The *big type* of a state s_k in π is

$$bt(s_k) \; := \sum_{i \in ran(s_k)} b_i \cdot 2^{i \, mod \, 2^n} \qquad \text{with } b_i \; = \; \begin{cases} 1 & , \text{ if } s_i \in L(bit) \\ 0 & , \text{ otherwise} \end{cases}$$

It is obtained by reading the value of the proposition *bit* in the range of s_k as the binary representation of a natural number. Since $|ran(s_k)| = 2^n$ for all s_k we have $0 \leq bt(s_k) < 2^{2^n}$. We also clearly have $bt(s_k) \neq bt(s_j) \Rightarrow s_k \neq s_j$ for all $k, j \in \mathbb{N}$ by contraposition. More interestingly, we have $bt(s_k) \neq bt(s_j) \Rightarrow ran(s_k) \neq ran(s_j)$. Since ranges are non-overlapping, two states with different big types must occur in two different segments on π.

Now formula $Init_n$ ensures $bt(s_k) = 0$ for all $0 \leq k < 2^n$, i.e. for all states in the first segment. Formulas $Flip_n$ and $Init_n$ ensure that for all $k, j \in \mathbb{N}$ such that $ran(s_k) = ran(s_j) + 1$ we have $bt(s_k) = bt(s_j) + 1 \; mod \; 2^{2^n}$. This means that π contains at least 2^{2^n} many different segments each of size 2^n, and each with a different big type. Furthermore, within each segment, all states are pairwise different because they have pairwise different small types. Hence, π must contain at least $2^n \cdot 2^{2^n}$ many different states. □

This establishes part (2) of the Gap-Lemma. Part (1) was, as said above, already proved by Emerson and Halpern.

Theorem 6 ([2]) *CTL has the small model property of size* $f(m) = m \cdot 2^{3m}$.

Corollary 7 *There is a* $2^{\Omega(m)}$*-succinctness gap between CTL*$^+$ *and CTL.*

PROOF. Thm. 6 yields the small model property for CTL of size $f(m) = m \cdot 2^{3m}$ which is strictly monotonically increasing. Lem. 3–5 yield the large model property for CTL$^+$ of size $g(m) = 2^{c \cdot m} \cdot 2^{2^{c \cdot m}}$ for some constant c. Hence, we have $f^{-1}(g(m)) = 2^{\Omega(m)}$ which is the succinctness gap according to Lemma 1. □

References

[1] M. Adler and N. Immerman. An $n!$ lower bound on formula size. *ACM Transactions on Computational Logic*, 4(3):296–314, 2003.

[2] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30:1–24, 1985.

[3] J. Johannsen and M. Lange. CTL^+ is complete for double exponential time. In *Proc. 30th Int. Coll. on Automata, Logics and Programming, ICALP'03*, volume 2719 of *LNCS*, pages 767 – 775. Springer, 2003.

[4] T. Wilke. CTL^+ is exponentially more succinct than CTL. In *Proc. 19th Conf. on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'99*, volume 1738 of *LNCS*, pages 110–121. Springer, 1999.