Deciding the weak definability of Büchi definable tree languages*

Thomas Colcombet[1], Denis Kuperberg[2], Christof Löding[3], and Michael Vanden Boom[4]

1,2 **CNRS and LIAFA, Université Paris Diderot**
Paris, France
{thomas.colcombet,denis.kuperberg}@liafa.univ-paris-diderot.fr
3 **Informatik 7, RWTH Aachen University**
Aachen, Germany
loeding@cs.rwth-aachen.de
4 **Department of Computer Science, University of Oxford**
Oxford, England
michael.vandenboom@cs.ox.ac.uk

Abstract

Weakly definable languages of infinite trees are an expressive subclass of regular tree languages definable in terms of weak monadic second-order logic, or equivalently weak alternating automata. Our main result is that given a Büchi automaton, it is decidable whether the language is weakly definable. We also show that given a parity automaton, it is decidable whether the language is recognizable by a nondeterministic co-Büchi automaton.

The decidability proofs build on recent results about cost automata over infinite trees. These automata use counters to define functions from infinite trees to the natural numbers extended with infinity. We reduce to testing whether the functions defined by certain "quasi-weak" cost automata are bounded by a finite value.

1998 ACM Subject Classification F.4.3 Formal Languages

Keywords and phrases Tree automata, weak definability, decidability, cost automata, boundedness

Digital Object Identifier 10.4230/LIPIcs.CSL.2013.215

1 Introduction

Infinite trees are often used as a model for representing the possible behaviours of a system. Various classes of automata and logic have been introduced in order to reason effectively about the properties of such systems. In particular, regular tree languages capture a rich class of properties which can be defined using logic (monadic second-order logic, or a fixpoint logic called the modal mu-calculus) and automata (including nondeterministic parity automata).

The *weakly definable languages* are a proper subclass of regular tree languages. These languages can be expressed in weak monadic second-order logic (in which second-order quantification is restricted only to finite sets), but they can be described in many other ways. For instance, Rabin [20] proved that they are precisely the languages for which the

* The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°259454 and the project ANR 2010 BLAN 0202 02 FREC.

Computer Science Logic 2013 (CSL'13).
Editor: Simona Ronchi Della Rocca; pp. 215–230
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

language and its complement are recognizable by nondeterministic Büchi automata. Arnold and Niwiński [2] showed that they can also be defined using the alternation-free mu-calculus. Muller et al. [15] showed that these languages can be equivalently defined by a form of alternating automata, called weak automata.

One reason these languages are so well studied is that they subsume temporal logics like CTL but still admit efficient (linear time) model-checking. Hence, they represent an expressive class of languages with good computational properties.

Given some regular language of infinite trees in the form of, say, an arbitrary parity automaton, it would be helpful to be able to decide whether it is weakly definable. We call this the *weak definability problem*.

Related work

The weak definability problem is related to the *nondeterministic parity index problem* or *Mostowski index problem*, which asks, given a regular language L of infinite trees and a finite set of priorities P, is there a nondeterministic parity automaton using only priorities P that recognizes L.[1] A nondeterministic parity automaton assigns to each state a priority. A run is a labelling of the tree with states, and a run is accepting if the highest priority occurring infinitely often on each branch is even. We say that a language L is P *definable* if there is a nondeterministic parity automaton using priorities P accepting exactly the trees in L. In the special case that $P = \{1, 2\}$ (respectively, $P = \{0, 1\}$), we say the language is *Büchi definable* (respectively, *co-Büchi definable*), since a nondeterministic parity automaton using priorities P can be viewed as a nondeterministic Büchi automaton (respectively, nondeterministic co-Büchi automaton).

Decidability of the nondeterministic parity index problem is known when P is restricted to a single priority (see [13]). More interesting, Niwiński and Walukiewicz [18] established decidability for any P as long as the input language is a deterministic tree language (a proper decidable subclass of regular tree languages).

This parity index problem is connected to the weak definability problem because of Rabin's characterization of weakly definable languages: a regular tree language is weakly definable if and only if the language and its complement are Büchi definable [20]. Since an automaton recognizing the complement of a regular language can be found effectively [19, 7], the weak definability problem reduces to determining whether the language and its complement are Büchi definable. Hence, one route to proving the decidability of weak definability would be to prove the decidability of Büchi definability. Although Karpiński reported such a solution already in the 1970s, there is no known written proof [9]. Hence, the weak definability problem is decidable when the input is a deterministic tree language, but is open in general.

Colcombet and Löding [5] proposed an alternative approach to solving this problem using *cost automata*. Cost automata are traditional finite state automata enriched with a finite set of counters. Instead of only accepting or rejecting some input structure, cost automata assign a value based on the evolution of counter values during the run. A cost automaton $\mathcal{A}$ can be viewed as defining a function $[\![\mathcal{A}]\!]$ from the set of structures under consideration to the natural numbers (extended with a special infinity symbol ∞). The idea is that cost automata can count some behaviour within the input.

[1] This is called the nondeterministic parity index problem since the desired automaton is nondeterministic. There are variants of this problem for different models of automata. See [13] for an overview.

The exact values of the function defined by a cost automaton do not matter. Instead, the functions are only considered up to an equivalence relation $\approx$ called the *boundedness relation* which ignores exact values but preserves all boundedness properties of the function.

Variants of these automata were famously used by Hashiguchi [8], and later Kirsten [10], to show the decidability of the star height problem over finite words. The idea is that a question about language theory like the star height problem is reduced to deciding whether the function defined by a cost automaton is limited, *i.e.* whether there is a natural number bounding the output over all accepted inputs. This is a special case of deciding whether two functions defined by cost automata are equivalent up to $\approx$. Because the boundedness relation is decidable for functions defined using cost automata over finite words ([3, 4]), this work on cost automata provided an alternative proof for a large part of the proof of decidability of the star height problem.

In this paper, we are dealing with cost automata working over infinite trees rather than finite words. Colcombet and Löding [5] provided a reduction of the decidability of the nondeterministic parity index problem to the decidability of $\approx$ for *cost-parity automata*: parity automata over infinite trees enriched with counters. Unfortunately, the decidability of $\approx$ for this richer class of cost-parity automata over infinite trees is open, so the general parity index problem remains open.

Independently, Afshari and Quickert also gave a reduction of the weak definability problem for Büchi definable languages to a question of boundedness, but decidability is also open using their approach [1].

Contributions

Our contribution is to show that recent results developed for the cost automata over infinite trees can be applied to solving a special case of the weak definability problem when the input language (or its complement) is Büchi definable. This subsumes the previously known decidability result for deterministic input. Along the way, we also point out an application to deciding whether a language is co-Büchi definable.

In other words, we show that the following problems are decidable:

- Given an alternating parity automaton over infinite trees, is there a nondeterministic co-Büchi automaton recognizing the same language? (Theorem 9)
- Given an alternating Büchi automaton, alternating co-Büchi automaton, or deterministic parity automaton over infinite trees, is there a weak automaton recognizing the same language? (Theorem 10)

The constructions make use of quasi-weak cost automata, which were introduced in [11]. This form of automaton is a new variant of weak automaton with counters, and we believe this application demonstrates the utility of this new automaton model.

Notation and Conventions

We write $\mathbb{N}$ for the set of non-negative integers and $\mathbb{N}_\infty$ for the set $\mathbb{N} \cup \{\infty\}$, ordered by $0 < 1 < \cdots < \infty$.

We work with an arbitrary finite alphabet $\mathbb{A}$. The set of finite (respectively, infinite) words over $\mathbb{A}$ is $\mathbb{A}^*$ (respectively, $\mathbb{A}^\omega$) and the empty word is ϵ. For notational simplicity we work only with infinite binary trees. Let $\mathcal{T} = \{0,1\}^*$ be the unlabelled infinite binary tree. The set $\mathcal{T}_\mathbb{A}$ of *complete* $\mathbb{A}$*-labelled binary trees* is composed of mappings $t : \mathcal{T} \to \mathbb{A}$. A *branch* π is a word $\{0,1\}^\omega$. A *frontier* E is a set of positions in $\mathcal{T}$ such that for all branches π in $\mathcal{T}$, $E \cap \pi$ is a singleton. For frontiers E and E', we write $E < E'$ if for every branch π, if

$\{x\} = E \cap \pi$ and $\{x'\} = E' \cap \pi$, then x is a strict ancestor of x'. If x, y are nodes in $\mathcal{T}$, with x a strict ancestor of y, we write $[x, y)$ for the set of nodes that are strict ancestors of y and descendants of x, including x itself.

2 Cost automata and cost functions

2.1 Cost automata

The automata that we consider are traditional automata over infinite trees equipped with a finite set of counters Γ which can be incremented $\mathtt{ic}$, reset $\mathtt{r}$, or left unchanged ε (but cannot be used to affect the flow of the automaton). Let $\mathbb{B} := \{\mathtt{ic}, \mathtt{r}, \varepsilon\}$ denote this alphabet of counter actions. Each counter starts with value zero, and the value of a sequence of actions is the supremum of the values achieved during this sequence. For instance the finite sequence $(\mathtt{ic})(\mathtt{ic})\mathtt{r}\varepsilon(\mathtt{ic})\varepsilon$ has value 2, the infinite sequence $((\mathtt{ic})\mathtt{r})^\omega$ has value 1, and the infinite sequence $(\mathtt{ic})\mathtt{r}(\mathtt{ic})^2\mathtt{r}(\mathtt{ic})^3\mathtt{r}\ldots$ has value ∞. If there are several counters, only the sequence with the maximal value is taken into account.

Formally, an *(alternating) B-Büchi automaton* $\langle Q, \mathbb{A}, q_0, \Gamma, F, \delta\rangle$ on infinite trees has a finite set of states Q, alphabet $\mathbb{A}$, initial state $q_0 \in Q$, a finite set Γ of counters, accepting states $F \subseteq Q$, and transition function $\delta : Q \times \mathbb{A} \to \mathcal{B}^+(\{0,1\} \times \mathbb{B}^\Gamma \times Q)$, where $\mathcal{B}^+(\{0,1\} \times \mathbb{B}^\Gamma \times Q)$ is the set of positive boolean combinations, written as a disjunction of conjunctions, of elements $(d, \nu, q) \in \{0,1\} \times \mathbb{B}^\Gamma \times Q$.

We view running a B-automaton $\mathcal{A}$ on an input tree t as a game $\mathcal{A} \times t$ between two players: Eve is in charge of the disjunctive choices and tries to minimize counter values while satisfying the acceptance condition, and Adam is in charge of the conjunctive choices and tries to maximize counter values or show the acceptance condition is not satisfied. Because the transition function is given as a disjunction of conjunctions, we can consider that at each position, Eve first chooses a disjunct, and then Adam chooses a single tuple (d, ν, q) in this disjunct.

A *play* of $\mathcal{A}$ on input t is a sequence $q_0, (d_1, \nu_1, q_1), (d_2, \nu_2, q_2), \ldots$ compatible with t and δ, *i.e.* q_0 is initial, and for all $i \in \mathbb{N}$, $(d_{i+1}, \nu_{i+1}, q_{i+1})$ appears in $\delta(q_i, t(d_1 \ldots d_i))$.

A *strategy* for Eve (respectively, Adam) in the game $\mathcal{A} \times t$ is a function that fixes the next choice of Eve (respectively, Adam), based on the history of the play (respectively, the history of the play and Eve's choice of disjunct). Notice that choosing a strategy for Eve and a strategy for Adam fixes a play in $\mathcal{A} \times t$. We say a play π is *compatible* with a strategy σ for Eve if there is some strategy σ' for Adam such that σ and σ' yield the play π.

A play π is *accepting* for the *Büchi condition* specified by F if there is $q \in F$ appearing infinitely often in π. Given a play π from a B-automaton $\mathcal{A}$, the value of π is ∞ if π is not accepting and is the supremum of the counter values achieved during the play otherwise.

We assign a value to a strategy σ for Eve by taking

$$\mathrm{value}(\sigma) := \sup\{\mathrm{value}(\pi) : \pi \text{ is compatible with } \sigma\}.$$

Likewise, the value assigned to the game $\mathcal{A} \times t$ is

$$\mathrm{value}(\mathcal{A} \times t) := \inf\{\mathrm{value}(\sigma) : \sigma \text{ is a strategy for Eve in the game } \mathcal{A} \times t\}.$$

We view $\mathcal{A}$ as defining a function $[\![\mathcal{A}]\!] : \mathcal{T}_{\mathbb{A}} \to \mathbb{N}_\infty$ such that $[\![\mathcal{A}]\!](t) := \mathrm{value}(\mathcal{A} \times t)$. Hence, in a B-automaton like this, Eve's goal is to satisfy the acceptance condition while minimizing the counter values.

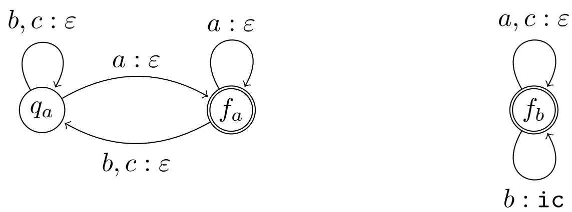
If for all $(q, a) \in Q \times \mathbb{A}$, $\delta(q, a)$ is of the form $\bigvee_i (0, \nu_i^0, q_i^0) \wedge (1, \nu_i^1, q_i^1)$, then we say the automaton is *nondeterministic*. We define a *run* to be the set of possible plays compatible with some fixed strategy of Eve. Since the only choices of Adam are in the branching, a run labels the entire binary tree with states, and choosing a branch yields a unique play of the automaton. A run is accepting if it is accepting on all branches, and the value assigned to a run of a B-automaton is the supremum of the values across all branches. For nondeterministic automata, the choices of Eve and Adam in the game described above can be viewed as Eve picking some $(0, \nu_i^0, q_i^0) \wedge (1, \nu_i^1, q_i^1)$, and Adam choosing a direction (which uniquely determines which atom Adam picks from the conjunction). Note that unless otherwise indicated, we assume automata are alternating.

We can define other types of cost automata simply by varying the acceptance condition. Given a set $F \subseteq Q$, a play is accepting for a *co-Büchi condition* specified by F if states from F occur only finitely often on the play. Given a set of priorities P and a mapping $\Omega : Q \to P$ assigning a priority to every state, a play is accepting for the *parity condition* specified by Ω if the maximum priority occurring infinitely often on the play is even. B-automata with these acceptance conditions will be called B-parity or B-co-Büchi automata as expected. Cost automata is a more general term that includes all of these types of automata with counters.

▶ **Example 1.** Let $\mathbb{A} = \{a, b, c\}$. We describe an alternating B-Büchi automaton $\mathcal{A}$ which computes the function $g : \mathcal{T}_{\mathbb{A}} \to \mathbb{N}_\infty$ defined by $g(t) = 0$ if some branch of t has infinitely many a, and $g(t) = \sup\{|\pi|_b : \pi \text{ is a branch of } t\}$ otherwise, where $|\pi|_b$ denotes the number of b-labelled nodes on the branch π.

Informally, Eve has an initial choice between trying to find a branch with infinitely many a's, or counting the number of b's on a branch chosen by Adam. Formally, let $\mathcal{A} = \langle \{q_0, q_a, f_a, f_b\}, \mathbb{A}, q_0, \{\gamma\}, \{f_a, f_b\}, \delta\rangle$. For any letter $x \in \mathbb{A}$, we have $\delta(q_0, x) = [(0, \varepsilon, q_a) \vee (1, \varepsilon, q_a)] \vee [(0, \varepsilon, f_b) \wedge (1, \varepsilon, f_b)]$.

Then, one of the following automata is deterministically executed, the first one on a branch chosen by Eve, the second one on a branch chosen by Adam.



For instance, $\delta(q_a, a) = (0, \varepsilon, f_a) \vee (1, \varepsilon, f_a)$, and $\delta(f_b, b) = (0, \texttt{ic}, f_b) \wedge (1, \texttt{ic}, f_b)$.

Notice that if the letter labelling the root is b, the counter is not incremented. This can result in a difference of 1 between $g(t)$ and $[\![\mathcal{A}]\!](t)$. We will see in Section 2.3 that we can still consider that $\mathcal{A}$ defines g, up to some equivalence relation between functions (noted $[\![\mathcal{A}]\!] \approx g$).

2.2 Variants of weakness for cost automata

Traditional *weak automata* are alternating Büchi automata with a restriction that no cycle of the automaton visits both accepting and rejecting states (this is equivalent to the original definition in [15]).

In the cost setting, there are two natural variants of weakness. We say an alternating B-Büchi automaton is

- B-*weak* if in all cycles, either all states are accepting or all states are rejecting;
- B-*quasi-weak* if in all cycles that contain both an accepting state and a rejecting state, there is a counter that is incremented but not reset.

The weakness property is the traditional notion of weakness, and implies that every play in the game associated with this automaton has to eventually stabilise, either in a strongly connected component where all states are accepting (so the play is winning for Eve), or in a strongly connected component where no state is accepting (so the play is winning for Adam). In fact, this stabilisation occurs after at most $|Q|$-many *changes of mode* between accepting states and rejecting states (where Q is the set of states of the automaton). B-weak automata were studied in [22].

Similarly, the quasi-weakness property implies that any play that does not stabilise after kn-changes of mode (for some constant k depending on the automaton but not depending on the input structure) has a counter with value greater than n. Hence, a play with some bounded value $n \in \mathbb{N}$ must have stabilised in accepting states. B-quasi-weak automata were introduced in [11] where they were shown to be strictly more expressive than B-weak automata, but not as expressive as general B-parity automata.

Thus, the difference between these models is in the number of allowed mode changes: unrestricted for B-Büchi automata; bounded by some function of the value for B-quasi-weak automata; and bounded by some constant for B-weak automata.

▶ Remark. By definition, a B-weak or B-quasi-weak automaton is a special type of alternating B-Büchi automaton. However, a B-weak or B-quasi-weak automaton can always be converted to a B-co-Büchi automaton defining the same function. This can be accomplished by simply complementing the set F of accepting states in the original B-quasi-weak automaton, and then viewing the automaton as an alternating B-co-Büchi automaton; the transition function does not change at all.

We can switch so easily between the Büchi form and co-Büchi form because the only difference between the Büchi and co-Büchi semantics occurs when the play switches infinitely many times between accepting and rejecting states. This is impossible in B-weak automata, and occurs only in plays of infinite counter value in B-quasi-weak automata (when the value of the play is ∞, regardless of the acceptance condition).

2.3 Cost function equivalence

Let $\mathcal{D}$ be some domain of input structures, and $\mathcal{F}_{\mathcal{D}}$ the set of functions : $\mathcal{D} \to \mathbb{N}_\infty$. We say a function $f : \mathcal{D} \to \mathbb{N}_\infty$ is *bounded* on some set $U \subseteq \mathcal{D}$ if there is some $n \in \mathbb{N}$ such that $f(u) \leq n$ for all $u \in U$.

The *domination relation* $\preccurlyeq$ and *boundedness relation* $\approx$ are defined as follows. Given $f, g : \mathcal{D} \to \mathbb{N}_\infty$,

$f \preccurlyeq g$ if for all $U \subseteq \mathcal{D}$, if g is bounded on U then f is bounded on U.

Likewise, $f \approx g$ if $f \preccurlyeq g$ and $g \preccurlyeq f$. In other words,

$f \approx g$ if for all $U \subseteq \mathcal{D}$, f is bounded on U if and only if g is bounded on U.

This means that f and g satisfying $f \approx g$ may not agree on exact values but do agree on boundedness properties across all subsets of the domain of input structures.

▶ **Example 2.** Let $\mathcal{D} = \mathcal{T}_{\mathbb{A}}$ for $\mathbb{A} = \{a, b, c\}$.

- Consider the functions $|\cdot|_a$ and $|\cdot|_b$ mapping a tree $t \in \mathcal{T}_{\mathbb{A}}$ to the number of a-labelled nodes and b-labelled nodes, respectively, in t. Then $|\cdot|_a \approx 2|\cdot|_a$ since a set of trees has bounded output via $|\cdot|_a$ if and only if it has bounded output via the function $2|\cdot|_a$. However, $|\cdot|_a \not\approx |\cdot|_b$ since the family of trees $(t_i)_{i\in\mathbb{N}}$ where t_i has no occurrences of a

and i occurrences of b has a bounded output via $|\cdot|_a$ but unbounded output via $|\cdot|_b$. We can also take the singleton $\{t_b\}$ as a witness, where t_b is the full binary tree labelled only with b's: $|\cdot|_a$ is bounded on $\{t_b\}$ but $|\cdot|_b$ is not, since value ∞ is considered unbounded.

- Given $L \subseteq \mathcal{T}_{\mathbb{A}}$, let χ_L denote the *characteristic function* that maps everything in L to 0 and everything else to ∞. Then for $K, L \subseteq \mathcal{T}_{\mathbb{A}}$, we have $\chi_L \preccurlyeq \chi_K$ if and only if $K \subseteq L$. Likewise, for $L \subseteq \mathcal{T}_{\mathbb{A}}$ and $f : \mathcal{T}_{\mathbb{A}} \to \mathbb{N}_\infty$, $f \approx \chi_L$ if and only if f is bounded on L and $f(t) = \infty$ for all $t \notin L$.

A *cost function over* $\mathcal{D}$ is an equivalence class of $\mathcal{F}_{\mathcal{D}}/\approx$, so we also refer to $\approx$ as *cost function equivalence*. In practice, a cost function (denoted $f, g, \ldots$) will be represented by one of its elements in $\mathcal{F}_{\mathcal{D}}$. In this paper, $\mathcal{D}$ will usually be $\mathcal{T}_{\mathbb{A}}$. The function $[\![\mathcal{A}]\!]$ defined by a cost automaton $\mathcal{A}$ will always be considered as a cost function, *i.e.* only considered up to $\approx$.

3 Expressivity of traditional automata on infinite trees

For readers who are unfamiliar with regular tree languages, we briefly review in this section some results about traditional automata over infinite trees. We refer the reader to [21, 6] for more information.

By setting $\Gamma = \emptyset$, the definitions in the previous section correspond to the traditional definitions of automata over infinite trees. In this case, $L(\mathcal{A}) \subseteq \mathcal{T}_{\mathbb{A}}$ denotes the set of trees t for which there exists a strategy for Eve in $\mathcal{A} \times t$ such that every play π in this strategy satisfies the acceptance condition. We say $\mathcal{A}$ *recognizes* the *language* $L(\mathcal{A})$. The function $[\![\mathcal{A}]\!]$ defined by $\mathcal{A}$ is $\chi_{L(\mathcal{A})}$ (recall that χ_L is the characteristic function of L mapping every tree in L to 0 and all other trees to ∞). Given some language $L \subseteq \mathcal{T}_{\mathbb{A}}$, we write $\overline{L}$ for the complement $\mathcal{T}_{\mathbb{A}} \setminus L$ of L.

In terms of expressivity, parity automata (in either their alternating or nondeterministic form) capture all regular languages of infinite trees. Languages recognized by alternating and nondeterministic Büchi and co-Büchi automata are strict subclasses of the regular tree languages. Deterministic parity automata are strictly less expressive than alternating co-Büchi automata.

Alternating automata can be easily complemented through *dualization*. The dual $\widetilde{\mathcal{U}}$ of an alternating automaton $\mathcal{U}$ is obtained by switching conjunctions and disjunctions in the transition formulas, and dualizing the acceptance condition. For the parity condition, this amounts to incrementing each priority by 1. Likewise, the dual of a Büchi (respectively, co-Büchi) condition specified by F is a co-Büchi (respectively, Büchi) condition specified by F. In each case, the dual automaton $\widetilde{\mathcal{U}}$ recognizes the complement of $\mathcal{U}$ [16].

Thanks to the so-called "breakpoint construction" [14], alternating Büchi automata are expressively equivalent to nondeterministic Büchi automata. Therefore, a tree language is Büchi definable if it is recognizable using either a nondeterministic or alternating Büchi automaton. On the other hand, alternating co-Büchi automata are strictly more expressive than nondeterministic co-Büchi automata (what we call co-Büchi definable). This means that the complement of a co-Büchi definable language is Büchi definable. The complement of a Büchi definable language can be recognized by an alternating co-Büchi automaton but is not necessarily co-Büchi definable.

A language is weakly definable if it is recognizable by a weak automaton, *i.e.* an alternating Büchi automaton satisfying the weakness condition described earlier. Equivalently, a language L is weakly definable if and only if L and $\overline{L}$ are Büchi definable [20, 12]. Stated in terms of alternating automata, the weakly definable languages are precisely the languages recognizable by both alternating Büchi and alternating co-Büchi automata. The dual of a weak automaton is weak, so weakly definable languages are closed under complement [15].

We pause to mention some well-known examples of weakly definable and Büchi definable languages.

▶ **Example 3.** Let $\mathbb{A} = \{a, b\}$. Consider $L_1, L_2 \subseteq \mathcal{T}_{\mathbb{A}}$ such that L_1 is the language of trees with infinitely many a's on every branch and L_2 is the language of trees with infinitely many a's on some branch.

- L_1 and $\overline{L_1}$ are weakly definable and Büchi definable.
- L_2 is Büchi definable but not weakly definable.
- $\overline{L_2}$ is neither Büchi definable nor weakly definable, but is co-Büchi definable.

4 Reducing to cost function equivalence

4.1 Description of the contribution

We seek to reduce questions about language definability to deciding cost function equivalence for functions defined by cost automata.

Colcombet and Löding [5] provided such a reduction for the parity index problem.

▶ **Theorem 4** ([5][2])**.** *Given a parity automaton $\mathcal{A}$ and a desired set of priorities P, there exists effectively a nondeterministic B-parity automaton $\mathcal{E}$ using priorities P such that the following are equivalent:*

- *$L(\mathcal{A})$ is recognizable by a nondeterministic parity automaton using priorities P,*
- *$[\![\mathcal{E}]\!] \approx \chi_{L(\mathcal{A})}$.*

The rough idea behind the construction is that the automaton $\mathcal{E}$ "guesses" a run of $\mathcal{A}$ (in fact, a run of a special normalized form for $\mathcal{A}$) and tries to map the priorities in $\mathcal{A}$ to the desired set of priorities P. The counters are used as a way to measure mistakes in this mapping, and it turns out that the function defined by $\mathcal{E}$ has a bounded value for all $t \in L(\mathcal{A})$ if and only if $L(\mathcal{A})$ is actually recognizable by a nondeterministic parity automaton using priorities P.

Our main contribution in this paper is to provide a reduction of the weak definability problem for Büchi definable languages to the decidability of $\approx$ for B-quasi-weak automata.

▶ **Theorem 5.** *Given a nondeterministic Büchi automaton $\mathcal{U}$ with $L = L(\mathcal{U})$, there exists effectively a B-quasi-weak automaton $\mathcal{B}$ such that the following are equivalent:*

- *L is weakly definable,*
- *$[\![\mathcal{B}]\!] \approx \chi_{\overline{L}}$.*

Recall that early work by Rabin [20] characterized weak definability in terms of Büchi definability of the language and its complement. Kupferman and Vardi [12] built on Rabin's work by providing an explicit construction of a weak automaton from two complementary nondeterministic Büchi automata.

Our construction is derived from this work. However, here we are only given a single Büchi automaton to start, so the constructed cost automaton "guesses" information about the

[2] The notation and terminology used in [5] is different than in this paper. The "distance-parity" automata in that work are B-parity automata here. The reduction is to the uniform universality problem of an automaton $\mathcal{U}_{ij}$, which asks whether $[\![\mathcal{U}_{ij}]\!]$ is equivalent to the constant function 0 (*i.e.* whether it is universal, and has a bounded value across all inputs). The construction of $\mathcal{U}_{ij}$ uses an automaton $\mathcal{A}_{ij}$, which corresponds to $\mathcal{E}$ here.

complementary automaton, and uses the counters to determine whether this complementary automaton is also Büchi (and therefore if the language is weak). Although this reduction was inspired by Theorem 4, it does not rely on it and the proof ideas are quite different.

We start by giving a brief description of the construction in [12], before proceeding to our reduction.

4.2 Proof from Kupferman and Vardi

The proofs in [20] and [12] begin with an analysis of composed runs of two nondeterministic Büchi automata $\mathcal{U} = \langle Q_{\mathcal{U}}, \mathbb{A}, q_0^{\mathcal{U}}, F_{\mathcal{U}}, \delta_{\mathcal{U}} \rangle$ and $\mathcal{U}' = \langle Q_{\mathcal{U}'}, \mathbb{A}, q_0^{\mathcal{U}'}, F_{\mathcal{U}'}, \delta_{\mathcal{U}'} \rangle$. Let $M := |Q_{\mathcal{U}}| \cdot |Q_{\mathcal{U}'}|$.

Recall that a frontier E is a set of nodes of t such that for any branch π of t, $E \cap \pi$ is a singleton. Frontiers can be compared: we write $E < E'$ if for any branch π, the only node in $\pi \cap E$ is a strict ancestor of the one in $\pi \cap E'$. Kupferman and Vardi [12] define a *trap* for $\mathcal{U}$ and $\mathcal{U}'$ to be a strictly increasing sequence of frontiers $\{\epsilon\} = E_0 < E_1 < \cdots < E_M$ such that there exists a tree t, a run R of $\mathcal{U}$ on t, and a run R' of $\mathcal{U}'$ on t satisfying the following properties: for all $0 \leq i < M$ and for all branches π in t, there exists $x, x' \in [e_i^\pi, e_{i+1}^\pi)$ such that $R(x) \in F_{\mathcal{U}}$ and $R'(x') \in F_{\mathcal{U}'}$ where $e_0^\pi < \cdots < e_M^\pi$ is the set of nodes from $E_0, \ldots, E_M$ induced by π. The set of positions $[e_i^\pi, e_{i+1}^\pi)$ can be viewed as an *accepting block* that witnesses an accepting state from both $\mathcal{U}$ and $\mathcal{U}'$.

A pumping argument from [20] shows that the existence of a trap implies $L(\mathcal{U}) \cap L(\mathcal{U}') \neq \emptyset$. We use this property in our proof below and therefore state it as a lemma.

▶ **Lemma 6** ([20],[12]). *If there is a trap for two nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{U}'$, then $L(\mathcal{U}) \cap L(\mathcal{U}') \neq \emptyset$.*

Now assume that $\mathcal{U}$ and $\mathcal{U}'$ are nondeterministic Büchi automata such that $L(\mathcal{U}')$ is the complement of $L(\mathcal{U})$. In this case, the existence of a trap implies a contradiction (which is why it is called a trap in [12]).

By taking advantage of this trap condition, Kupferman and Vardi [12] show how to use the complementary nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{U}'$ in order to construct a weak automaton $\mathcal{W}$ recognizing $L(\mathcal{U})$. The general idea is that Eve (respectively, Adam) selects a run of $\mathcal{U}$ (respectively, $\mathcal{U}'$). The acceptance condition in $\mathcal{W}$ requires that any time an accepting state from $\mathcal{U}'$ is seen, an accepting state from $\mathcal{U}$ is eventually seen. Because the existence of a trap is impossible, these accepting blocks only need to be counted up to M times before the automaton is allowed to enter an accepting sink state. Hence, the automaton $\mathcal{W}$ keeps track of the number of blocks (up to M) in the state. Since each block contains at most two changes of mode between accepting and rejecting states, this means there is no cycle visiting both accepting and rejecting states, so $\mathcal{W}$ is weak.

The idea for the proof of correctness is that if $t \in L(\mathcal{U})$, then Eve has a strategy to play in the game $\mathcal{W} \times t$ (*i.e.* play the accepting run of $\mathcal{U}$). On the other hand, if $t \in L(\mathcal{U}')$ and we assume for the sake of contradiction that Eve has a winning strategy in $\mathcal{W} \times t$, then this strategy and the accepting run of $\mathcal{U}'$ can be used to build a trap, which is impossible.

We now turn to our construction and the proof of Theorem 5.

4.3 Construction

Let us now describe the construction of the automaton $\mathcal{B}$ in Theorem 5, which bears a resemblance to the construction in [12]. However, the construction from [12] explicitly uses a Büchi automaton $\mathcal{U}'$ for the complement of $\mathcal{U}$, and furthermore uses an explicit counter in the state space to count up to value M (the value used in the definition of a trap, see above).

We only use the existence of such an automaton $\mathcal{U}'$ in one direction of the correctness proof, and replace the explicit counting by a single B-counter.

Let $\mathcal{U} = \langle Q_{\mathcal{U}}, \mathbb{A}, q_0^{\mathcal{U}}, F_{\mathcal{U}}, \delta_{\mathcal{U}}\rangle$ be a nondeterministic Büchi automaton without counters. Let $\widetilde{\mathcal{U}}$ be the dual of $\mathcal{U}$, obtained by exchanging conjunctions and disjunctions in the transition formulas of $\mathcal{U}$, and changing the acceptance condition from a Büchi condition specified by F to a co-Büchi condition specified by the same F. The automaton $\widetilde{\mathcal{U}}$ recognizes the complement of $\mathcal{U}$ [16]. Note that dualization of an automaton $\mathcal{U}$ also results in the dualization of the game $\mathcal{U} \times t$: the roles of Adam and Eve exchange (we use this fact in the proof below).

The construction of $\mathcal{B}$ can roughly be described as follows. It consists of two copies of $\widetilde{\mathcal{U}}$, the states of the first of these copies are non-accepting, the states of the second copy are accepting. In the first copy, the transition function of $\widetilde{\mathcal{U}}$ is extended to allow a nondeterministic choice between staying inside the first copy or jumping to the second copy (so each transition of $\widetilde{\mathcal{U}}$ is doubled in a nondeterministic way). In the second copy, the transitions from states in $F_{\mathcal{U}}$ go back to the first copy and perform an increment on the counter. The other transitions of the second copy, from states not in $F_{\mathcal{U}}$, stay in the second copy and do not increment the counter.

Formally, define the B-Büchi automaton $\mathcal{B} := \langle Q_{\mathcal{B}}, \mathbb{A}, q_0^{\mathcal{B}}, \{\gamma_{\mathrm{alt}}\}, F_{\mathcal{B}}, \delta_{\mathcal{B}}\rangle$ with the following components. $Q_{\mathcal{B}} = Q_{\mathcal{U}} \times \{R, A\}$ (R and A for rejecting and accepting), $q_0^{\mathcal{B}} = \langle q_0^{\mathcal{U}}, R\rangle$, and $F_{\mathcal{B}} = Q_{\mathcal{U}} \times \{A\}$.

For $q \in Q_{\mathcal{U}}$ and $a \in \mathbb{A}$ with $\delta_{\mathcal{U}}(q, a) = \bigvee_{i=0}^{n_q}(0, q_i^0) \wedge (1, q_i^1)$, we define $\delta_{\mathcal{B}}$ as follows:

$$\delta_{\mathcal{B}}(\langle q, R\rangle, a) = \textstyle\bigwedge_{i=0}^{n_q}(0, \varepsilon, \langle q_i^0, R\rangle) \vee (0, \varepsilon, \langle q_i^0, A\rangle) \vee (1, \varepsilon, \langle q_i^1, R\rangle) \vee (1, \varepsilon, \langle q_i^1, A\rangle).$$

If $q \notin F_{\mathcal{U}}$ then

$$\delta_{\mathcal{B}}(\langle q, A\rangle, a) = \textstyle\bigwedge_{i=0}^{n_q}(0, \varepsilon, \langle q_i^0, A\rangle) \vee (1, \varepsilon, \langle q_i^1, A\rangle)$$

and if $q \in F_{\mathcal{U}}$ then

$$\delta_{\mathcal{B}}(\langle q, A\rangle, a) = \textstyle\bigwedge_{i=0}^{n_q}(0, \mathtt{ic}, \langle q_i^0, R\rangle) \vee (1, \mathtt{ic}, \langle q_i^1, R\rangle).$$

Note that the two copies of $\mathcal{U}$ that are used in $\mathcal{B}$ have the structure of $\widetilde{\mathcal{U}}$ with the additional branching between the two copies, as described above. We use this fact in the proof below to copy strategies of Eve or Adam in $\widetilde{\mathcal{U}} \times t$ to $\mathcal{B} \times t$.

4.4 Proof of correctness

It is easy to see that $\mathcal{B}$ is quasi-weak: any cycle going through both accepting and rejecting states has to take a transition from the second copy of $\widetilde{\mathcal{U}}$ to the first one, thereby incrementing the single counter γ_{alt} of $\mathcal{B}$ (which is never reset). It remains to prove that $\mathcal{B}$ has the property claimed in Theorem 5, namely that $[\![\mathcal{B}]\!] \approx \chi_{\overline{L}}$ if and only if L is weakly definable.

For one direction, assume that $[\![\mathcal{B}]\!] \approx \chi_{\overline{L}}$. Then there exists $N \in \mathbb{N}$ such that for all $t \in \overline{L}$, there is an accepting run ρ of $\mathcal{B}$ on t with value$(\rho) \leq N$. Hence, there is a weak automaton $\mathcal{B}'$ based on $\mathcal{B}$ that simulates γ_{alt} in the state up to value N, and enters a special rejecting state as soon as it would exceed this bound. Then $L(\mathcal{B}') = \overline{L}$, and the dual of $\mathcal{B}'$ is a weak automaton defining L. Notice that this part does not depend on how $\mathcal{B}$ is built: the existence of any quasi-weak automaton $\mathcal{A}$ with $[\![\mathcal{A}]\!] \approx \chi_L$ or $[\![\mathcal{A}]\!] \approx \chi_{\overline{L}}$ would imply that L is weakly definable.

For the other direction, assume that L is weakly definable. We first show that $\mathcal{B}$ has value ∞ on all trees from L (this does not use the fact that L is weakly definable). So let $t \in L$ be a tree. Then Eve has a strategy in $\mathcal{U} \times t$ that ensures that $F_{\mathcal{U}}$ is visited infinitely often. It follows that in the dual game $\widetilde{\mathcal{U}} \times t$ Adam has a strategy σ that ensures that $F_{\mathcal{U}}$ is visited infinitely often. Adam can use this strategy in $\mathcal{B} \times t$, regardless of which copy is currently used. Note that if the current copy is R, it is Eve who can choose to stay in R or to switch to A. However when the current copy is A, the plays goes back to R as soon as a state from $F_{\mathcal{U}}$ is seen, which depends on Adam's choices. A resulting play will either stay in R eventually, or it alternates infinitely often between A and R because σ ensures that $F_{\mathcal{U}}$ is visited infinitely often. In both cases, the value of the resulting play is ∞.

We now finish the correctness proof by showing that $\mathcal{B}$ is bounded on $\overline{L}$. Since L is weakly definable, there is a nondeterministic Büchi automaton $\mathcal{U}'$ for $\overline{L}$. Let $M := |Q_{\mathcal{U}}| \cdot |Q_{\mathcal{U}'}|$.

Fix some $t \in \overline{L}$. We consider a game that in some sense corresponds to the game $(\mathcal{U}' \times \widetilde{\mathcal{U}}) \times t$. Accordingly, the positions of the game are $Q_{\mathcal{U}'} \times Q_{\mathcal{U}} \times \{0,1\}^*$. One round of the game consists of the following moves, assuming the current position is (q', q, x):

1. Eve chooses a transition of $\mathcal{U}'$, that is, a conjunction $(0, p_0') \wedge (1, p_1')$ that appears in $\delta_{\mathcal{U}'}(q', t(x))$.
2. Adam chooses a transition of $\mathcal{U}$, that is, a conjunction $(0, p_0) \wedge (1, p_1)$ that appears in $\delta_{\mathcal{U}}(q, t(x))$.
3. Eve chooses a direction $d \in \{0, 1\}$.

As usual, the game continues from (p_d', p_d, xd). Note that the moves 2 and 3 are the moves from $\mathcal{U} \times t$ with the roles of Eve and Adam exchanged, and thus correspond to the moves in $\widetilde{\mathcal{U}} \times t$.

Because $t \in \overline{L}$, Eve has a strategy σ to ensure that $F_{\mathcal{U}'}$ is visited infinitely often. Assume that Adam has a strategy τ against such a σ that enforces in each compatible play more than M many alternations between $F_{\mathcal{U}'}$ and $F_{\mathcal{U}}$. Since Eve is choosing the directions, τ works along all branches of t, and therefore would define a trap for $\mathcal{U}$, $\mathcal{U}'$, and t. Since $L(\mathcal{U}) \cap L(\mathcal{U}') = \emptyset$, this would contradict Lemma 6. We can conclude that Eve has a strategy σ that visits $F_{\mathcal{U}'}$ infinitely often and at the same time keeps the number of alternations between $F_{\mathcal{U}'}$ and $F_{\mathcal{U}}$ bounded by M (here, we use the determinacy of the game with the corresponding winning condition for Eve).

From σ we can derive a strategy in $\mathcal{B} \times t$ for Eve that accepts t with value bounded by M. Recall that the moves 2 and 3 correspond to the game $\widetilde{\mathcal{U}} \times t$ and thus Eve can use σ in $\mathcal{B}$ (keeping track of her moves in $\mathcal{U}'$ in her memory). We only have to define when Eve switches from $Q_{\mathcal{U}} \times \{R\}$ to $Q_{\mathcal{U}} \times \{A\}$. She does this whenever the state of $\mathcal{U}'$ in the original game is in $F_{\mathcal{U}'}$. If she plays according to this strategy, then each play switches from $Q_{\mathcal{U}} \times \{R\}$ to $Q_{\mathcal{U}} \times \{A\}$ eventually, because σ ensures infinitely many visits to $F_{\mathcal{U}'}$. Furthermore, Adam can switch back to $Q_{\mathcal{U}} \times \{R\}$ at most M times, because σ ensures at most M alternations between $F_{\mathcal{U}'}$ and $F_{\mathcal{U}}$. Hence, each play eventually remains in $Q_{\mathcal{U}} \times \{A\}$ and has value at most $M \in N$. Since this is true for all $t \in \overline{L}$, $\mathcal{B}$ is bounded on $\overline{L}$ as desired.

This finishes the correctness proof for the construction of $\mathcal{B}$ and thus the proof of Theorem 5.

5 Deciding special cases of cost function equivalence

5.1 Special cases of cost function equivalence over infinite trees

Although we do not know how to decide $\preccurlyeq$ for all cost functions over infinite trees, we can show it is decidable in some cases.

▶ **Theorem 7.** *Let f, g be cost functions over infinite trees. Then $f \preccurlyeq g$ is decidable if these two conditions are satisfied:*

- *f is given by a parity automaton (without counters) or a B-co-Büchi automaton;*
- *g is given by a parity automaton (without counters), B-Büchi automaton, or nondeterministic B-parity automaton.*

This is a slight extension of the decidability of cost function equivalence reported in [22] and [11], but the proof method is the same. The procedure tries to find a witness showing $f \not\preccurlyeq g$. Such a witness is a family of trees with bounded value via g but unbounded value via f. In order to do this, the automata are first converted into nondeterministic forms (so these witnesses can be "guessed"). B-automata are good for g because a single run of a nondeterministic B-automaton can witness a low value. There is a dual form called S-automata which are good for f since they can witness a high value with a single run. We will not describe this dual form here (we refer to the appendix for some details on this model). The types of automata appearing in Theorem 7 describe when we know when we can convert alternating cost automata into these required forms. From there, deciding $\preccurlyeq$ can be reduced to solving parity games on finite graphs. We refer the interested reader to [23] for more information.

Because B-quasi-weak automata can be viewed as either B-co-Büchi or B-Büchi automata (see the Remark at the end of Section 2.2), this means that $f \approx g$ is decidable when f and g are given by B-quasi-weak automata.

▶ **Corollary 8.** *Let f, g be cost functions over infinite trees. Then $f \approx g$ is decidable if each function is given by either a parity automaton (without counters) or a B-quasi-weak automaton.*

In the remainder of this section, we state the decidability results that follow from Theorem 7 and Corollary 8 and the reductions given in Theorem 4 and Theorem 5.

5.2 Deciding co-Büchi definability

In this section we show that it is decidable for a given parity automaton whether there exists an equivalent nondeterministic co-Büchi automaton.

The decidability of co-Büchi definability is a corollary of Theorem 7 and Theorem 4.

▶ **Theorem 9.** *Given a parity automaton $\mathcal{A}$ over infinite trees, it is decidable whether or not there is a nondeterministic co-Büchi automaton $\mathcal{A}'$ such that $L(\mathcal{A}) = L(\mathcal{A}')$.*

Proof. Given $\mathcal{A}$ we apply Theorem 4 with $P = \{0, 1\}$ to obtain a nondeterministic B-co-Büchi automaton $\mathcal{E}$ such that $[\![\mathcal{E}]\!] \approx \chi_{L(\mathcal{A})}$ if and only if $L(\mathcal{A})$ is recognizable by a nondeterministic co-Büchi automaton. According to Theorem 7 it is decidable whether $[\![\mathcal{E}]\!] \approx \chi_{L(\mathcal{A})}$ because the types of the automata $\mathcal{A}$ and $\mathcal{E}$ are such that both directions, $[\![\mathcal{E}]\!] \preccurlyeq \chi_{L(\mathcal{A})}$ and $\chi_{L(\mathcal{A})} \preccurlyeq [\![\mathcal{E}]\!]$ can be decided. ◀

One route to proving the decidability of the full parity index problem would be to prove a more general decidability result for $\approx$, for arbitrary alternating B-parity automata, but this problem remains open.

5.3 Deciding weak definability of Büchi definable languages

We now turn to the main result, deciding weakness for Büchi definable languages. This follows from Corollary 8 and the reduction in Theorem 5.

▶ **Theorem 10.** *Given an alternating Büchi automaton, alternating co-Büchi automaton, or deterministic parity automaton $\mathcal{A}$ with $L = L(\mathcal{A})$, it is decidable whether there is a weak automaton $\mathcal{W}$ such that $L(\mathcal{W}) = L$, or equivalently whether L is definable in weak monadic second-order logic.*

Proof. Note that Corollary 8 implies that $[\![\mathcal{B}]\!] \approx \chi_{\overline{L}}$ is decidable when $\mathcal{B}$ is B-quasi-weak. Hence, Theorem 5 implies the decidability of the weak definability problem when the input is a nondeterministic Büchi automaton. The other inputs can all be transformed into a nondeterministic Büchi automaton for the language or its complement.

If the input is an alternating Büchi automaton, then an equivalent nondeterministic Büchi automaton can be constructed ([14, 17]), and Theorem 5 can be applied. If the input is an alternating co-Büchi automaton, then the complement is an alternating Büchi automaton and we can use the previous case to decide whether $\overline{L}$ is weak. Since weakly definable languages are closed under complement, $\overline{L}$ is weak if and only if L is weak.

Finally, if the input is a deterministic parity automaton (not necessarily Büchi), then we reduce to other cases since a nondeterministic Büchi automaton can be constructed for the complement language: the nondeterministic Büchi automaton guesses a branch in the run of the deterministic parity automaton and checks that it is rejecting. As mentioned earlier, the decidability of this case was known already from [18] using different methods. ◀

6 Conclusion

We showed that given a Büchi automaton, it is decidable whether the language it recognizes is weak. We also showed that it is decidable whether a regular tree language is recognizable by a nondeterministic co-Büchi automaton, for arbitrary input. These are particular cases of the Mostowski index problem, for which very few intermediate results are known.

We used the recent formalism of quasi-weak automata to prove these results. We believe this shows that this model is flexible and well-suited for the weak definability problem. Moreover, it provides additional motivation to develop the theory of cost functions, since we demonstrated that it can give rise to new decidability procedures.

The current understanding of the theory of cost functions over infinite trees does not allow us to state a more general theorem, because decidability of the boundedness problem for such cost functions is still open in general. The inherent difficulty of this problem can be explained by the complex interplay between classic acceptance conditions and counter behaviour.

References

1 Bahareh Afshari and Sandra Quickert. Personal communication, 2012.
2 André Arnold and Damian Niwiński. Fixed point characterization of weak monadic logic definable sets of trees. In *Tree Automata and Languages*, pages 159–188. 1992.
3 Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *LICS*, pages 285–296. IEEE Computer Society, 2006.
4 Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *ICALP (2)*, volume 5556 of *LNCS*, pages 139–150. Springer, 2009.
5 Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In Luca Aceto, Ivan Damgard, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *LNCS*, pages 398–409. Springer, 2008.

6 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002.

7 Yuri Gurevich and Leo Harrington. Trees, automata, and games. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *STOC*, pages 60–65. ACM, 1982.

8 Kosaburo Hashiguchi. Relative star height, star height and finite automata with distance functions. In Jean-Éric Pin, editor, *Formal Properties of Finite Automata and Applications*, volume 386 of *LNCS*, pages 74–88. Springer, 1988.

9 Karpiński. Personal communication, 2008.

10 Daniel Kirsten. Distance desert automata and the star height problem. *ITA*, 39(3):455–509, 2005.

11 Denis Kuperberg and Michael Vanden Boom. Quasi-weak cost automata: a new variant of weakness. In Supratik Chakraborty and Amit Kumar, editors, *FSTTCS*, volume 13 of *LIPIcs*, pages 66–77. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011. Online at `http://www.liafa.jussieu.fr/~dkuperbe/`.

12 Orna Kupferman and Moshe Y. Vardi. The weakness of self- complementation. In Christoph Meinel and Sophie Tison, editors, *STACS*, volume 1563 of *LNCS*, pages 455–466. Springer, 1999.

13 Christop Löding. Logic and automata over infinite trees. Habilitation thesis, RWTH Aachen University, 2009.

14 Satoru Miyano and Takeshi Hayashi. Alternating finite automata on omega-words. *Theor. Comput. Sci.*, 32:321–330, 1984.

15 David E. Muller, Ahmed Saoudi, and Paul E. Schupp. Alternating automata. The weak monadic theory of the tree, and its complexity. In Laurent Kott, editor, *ICALP*, volume 226 of *LNCS*, pages 275–283. Springer, 1986.

16 David E. Muller and Paul E. Schupp. Alternating automata on infinite trees. *Theor. Comput. Sci.*, 54:267–276, 1987.

17 David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theor. Comput. Sci.*, 141(1&2):69–107, 1995.

18 Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electr. Notes Theor. Comput. Sci.*, 123:195–208, 2005.

19 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

20 Michael O. Rabin. Weakly definable relations and special automata. In *Mathematical Logic and Foundations of Set Theory (Proc. Internat. Colloq., Jerusalem, 1968)*, pages 1–23. North-Holland, Amsterdam, 1970.

21 Wolfgang Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.

22 Michael Vanden Boom. Weak cost monadic logic over infinite trees. In Filip Murlak and Piotr Sankowski, editors, *MFCS*, volume 6907 of *LNCS*, pages 580–591. Springer, 2011.

23 Michael Vanden Boom. *Weak Cost Automata over Infinite Trees.* PhD thesis, University of Oxford, 2012. Online at `http://www.cs.ox.ac.uk/people/michael.vandenboom/`.

A Appendix

There is another form of cost automata called S-automata that is dual to B-automata. We briefly describe the definition of S-automata and its relationship to the B-automata used in the body of the paper.

A.1 S-automata

S-automata have atomic actions increment $\mathtt{i}$, no change ε, reset $\mathtt{r}$, and check-reset $\mathtt{cr}$. Let $\mathbb{S} := \{\mathtt{i}, \varepsilon, \mathtt{cr}, \mathtt{r}\}$. Given a sequence of counter actions $u \in \mathbb{S}^\omega$, let $C(u)$ denote the values of the counter at the moment(s) when a check-reset $\mathtt{cr}$ occurs. For instance, if $u = \mathtt{i}^{100}\mathtt{cri}\varepsilon\mathtt{icr}(\mathtt{ir})^\omega$, then $C(u) = \{2, 100\}$. Likewise, for a set of counters Γ, and a word $u \in (\mathbb{S}^\Gamma)^\omega$, let $C(u) = \bigcup_{\gamma\in\Gamma} C(u_\gamma)$ where u_γ is the γ-projection of u. The S-value of such a sequence is $\inf C(u)$ (the minimum checked counter value).

An *(alternating) S-Büchi automaton* $\langle Q, \mathbb{A}, q_0, \Gamma, F, \delta\rangle$ on infinite trees has a finite set of states Q, alphabet $\mathbb{A}$, initial state $q_0 \in Q$, accepting states F, finite set Γ of counters, and transition function $\delta : Q \times \mathbb{A} \to \mathcal{B}^+(\{0,1\} \times \mathbb{S}^\Gamma \times Q)$. S-automata with other acceptance conditions can be defined as expected. The notion of plays and strategies is the same as in B-automata.

Given a play π from a S-automaton $\mathcal{A}$, the value of π is 0 if π is not accepting and $\inf C(u)$ otherwise, where u is the sequence of counter actions from π. For a strategy σ for Eve, value(σ) is the infimum over the values from all plays consistent with the strategy. For the corresponding game $\mathcal{A} \times t$, value($\mathcal{A} \times t$) is

$$\sup\{\text{value}(\sigma) : \sigma \text{ is a strategy for Eve in the game } \mathcal{A} \times t\}$$

and $[\![\mathcal{A}]\!](t) = \text{value}(\mathcal{A} \times t)$. The idea is that in an S-automaton Eve is trying to satisfy the acceptance condition while maximizing the values of the counters (whereas, in a B-automaton Eve is trying to satisfy the acceptance condition and minimize the values of the counters).

A.2 Duality and simulation

Switching between the B and S forms of cost automata takes the place of complementing classical automata.

Indeed, let L be the language recognized by some traditional automaton $\mathcal{A}$ without counters. If we view this automaton as a B-automaton then it defines the function χ_L (the characteristic function for the language), and if we view it as an S-automaton, then it defines the function $\chi_{\overline{L}}$ (the characteristic function for the complement of the language). This means that we can always find both a B and S form for cost functions of the form χ_L for L a regular language recognized by some automaton $\mathcal{A}$ (the S-form for χ_L can be obtained by complementing $\mathcal{A}$).

Using alternating cost automata, we can switch between B and S forms too.

▶ **Theorem 11** ([22]). *Alternating B-parity and alternating S-parity automata are effectively equivalent.*

A closer examination of the proof shows that in the conversion between B-parity and S-parity automata, the priorities are shifted by one (see [23] for more information), which parallels the classical complementation procedure for alternating parity automata. This means that Büchi automata become co-Büchi automata, and vice versa.

▶ **Corollary 12.** *Alternating B-Büchi and alternating S-co-Büchi automata are effectively equivalent.*

Alternating S-Büchi automata and alternating B-co-Büchi automata are effectively equivalent.

Alternating B-Büchi and alternating S-Büchi automata can also be simulated with nondeterministic versions. This parallels the classical result that alternating Büchi automata are equivalent to nondeterministic Büchi automata. The B-part of this result was stated in [22]. The S-part was not stated clearly there, but the proof technique is the same. The details of both of these proofs can be found in [23].

▶ **Theorem 13** ([22],[23]). *Alternating B-Büchi and nondeterministic B-Büchi automata are effectively equivalent.*

Alternating S-Büchi and nondeterministic S-Büchi automata are effectively equivalent.

The following corollary summarizes these results.

▶ **Corollary 14.** *Alternating B-co-Büchi automata, alternating S-Büchi automata, and nondeterministic S-Büchi automata are effectively equivalent.*

Alternating S-co-Büchi automata, alternating B-Büchi automata, and nondeterministic B-Büchi automata are effectively equivalent.

In [11], B-quasi-weak automata were characterized as precisely the cost functions that are definable by both nondeterministic B-Büchi and nondeterministic S-Büchi automata.

A.3 Decidability

The decidability of cost function equivalence over infinite trees was originally stated in terms of B and S automata.

▶ **Theorem 15** ([22]). *For cost functions f and g over infinite trees, the relation $f \preccurlyeq g$ is decidable if*

- *f is given by a nondeterministic S-parity automaton, and*
- *g is given by a nondeterministic B-parity automaton.*

Theorem 7 can be viewed as a restatement of these results, using Corollary 14 to write it in terms of B-automata only.