A Note on the Characterization of TL[EF] ⋆

Zhilin Wu

Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, P.O.Box 8718, Beijing, China, 100080 and Graduate School of the Chinese Academy of Sciences, 19 Yuquan Street, Beijing, China, 100049

Abstract

In this note, we give a new proof for Bojańczyk&Walukiewicz's effective characterization of TL[EF](the fragments of Computation Tree Logic(CTL), with EF modality only) following the Ehrenfeucht-Fraïssé game approach. Then, we extend the proof to the effective characterization of $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$($\mathsf{F}_{\mathsf{ns}}$ is the non-strict "future" temporal operator, while F is the strict one).

Key words: formal languages, branching-time temporal logic, tree languages, Ehrenfeucht-Fraïssé game

1. Introduction

The definability problem for logics on trees is to decide whether a given regular tree language is definable in a logic. This kind of problem has proven to be rather difficult. For instance, the definability problem for first order logic on trees (FO[<]) has been a longstanding open problem since the 80's of the last century despite several partial results [6,4,1,2].

In [3], Bojańczyk&Walukiewicz made a breakthrough in the definability problem for logics on trees by giving effective characterizations for several sublogics of CTL, namely TL[EX], TL[EF] and TL[EX,EF]. While the proofs of the characterization for TL[EX] and TL[EX,EF] in [3] were elegant and short, the proof for TL[EF] was very intricate.

One of the main reasons for the intricacy of the proof of TL[EF] in [3] is that the proof was constructive. To avoid the intricacy, in this note, we give an existential proof for the characterization of TL[EF] following the Ehrenfeucht-Fraïssé games approach, similar to the proof of the characterization of the fragment of LTL that only uses the operator "F", "sometimes in the future" [8]. Moreover, we extend this proof to the characterization of $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$ [1] (F_{ns} is the non-strict "future" operator, while F is strict), which was mentioned to be open in [3].

The remaining sections are organized as follows: in Section 2, the syntax and semantics of TL[EF] and $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$ are defined; and in Section 3, some definitions and notations are introduced; in Section 4, a new proof of characterization of TL[EF] is given; then in Section 5, the effective characterization of $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$ is established; finally in Section 6, we give some conclusions and remarks.

2. Syntax and Semantics of TL[EF] and $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$

Let Σ be a finite alphabet, then the syntax of TL[EF] is defined by the following rules.

⋆ Supported by the National Natural Science Foundation of China under Grant No. 60421001 and 60573012, and the National Grand Fundamental Research 973 Program of China under Grant No. 2002cb312200.

Email address: wuzl@ios.ac.cn (Zhilin Wu).

[1] One of the referees pointed out that the characterization of $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$ has been independently announced by Zoltan Ésik and Szabolcs Iván at a workshop of CSL'06 on formal languages: http://www.inf.u-szeged.hu/∼csl06/ws.php.

$$\varphi := p_a(a \in \Sigma) | \neg\varphi_1 | \varphi_1 \vee \varphi_2 | \mathsf{EF}\varphi_1 \quad (1)$$

A binary tree domain is a prefix closed nonempty subset of $\{0,1\}^*$ such that for all $v \in \{0,1\}^*$, $v0$ is in the domain iff $v1$ is in the domain, in other words, each inner node has two sons. Evidently ε is in all tree domains, which is called the root of the tree. The prefix relation on the tree domain is denoted by $<$.

Let Σ be a finite alphabet. A Σ-labelled finite binary tree is a function from a binary tree domain to Σ. If t is a Σ-labelled finite binary tree, then the tree domain of t is denoted by $dom(t)$. For any $v \in dom(t)$, the label of v in t is denoted by $t(v)$. In particular, $t(\varepsilon)$ is the label of the root of t.

If $v \in dom(t)$, then $t|_v$ denotes the subtree of t below v (including v).

Let T_Σ denote the set of all Σ-labelled finite binary trees.

The semantics of $\mathsf{TL[EF]}$ are defined as follows.

Let $t \in T_\Sigma$, then

- $t \models p_a$ if $t(\varepsilon) = a$, where $a \in \Sigma$.
- $t \models \neg\varphi_1$ if not $t \models \varphi_1$.
- $t \models \varphi_1 \vee \varphi_2$ if $t \models \varphi_1$ or $t \models \varphi_2$.
- $t \models \mathsf{EF}\varphi_1$ if there is some $v \in dom(t)$, $v > \varepsilon$ such that $t|_v \models \varphi_1$.

The syntax of $\mathsf{TL[EF_{ns}]}$ is defined by the same rules in (1) with $\mathsf{EF}\varphi_1$ replaced by $\mathsf{EF_{ns}}\varphi_1$.

The semantics of $\mathsf{EF_{ns}}\varphi_1$ is defined as follows:

$t \models \mathsf{EF_{ns}}\varphi_1$ if there is $v \in dom(t)$ such that $t|_v \models \varphi_1$ (note that here v may be ε).

Let φ be a $\mathsf{TL[EF]}$ or $\mathsf{TL[EF_{ns}]}$ formula. then the closure of φ, denoted by $cl(\varphi)$, is defined to be the smallest set of formulas containing φ and closed under negations and subformulas.

A tree language L is said to be $\mathsf{TL[EF]}$ ($\mathsf{TL[EF_{ns}]}$ respectively)-definable if there is a formula φ in $\mathsf{TL[EF]}$ ($\mathsf{TL[EF_{ns}]}$ respectively) such that $L = \{t \in T_\Sigma | t \models \varphi\}$.

Since $\mathsf{EF_{ns}}\varphi \equiv \varphi \vee \mathsf{EF}\varphi$, $\mathsf{TL[EF_{ns}]}$ can be seen as a sublogic of $\mathsf{TL[EF]}$. Moreover, $\mathsf{TL[EF]}$ is more expressive than $\mathsf{TL[EF_{ns}]}$. For instance, the property "the tree has at least depth two and all its nodes are labelled by a" can be expressed by $\mathsf{TL[EF]}$ formula: $p_a \wedge \mathsf{EFp_a} \wedge \neg\mathsf{EF}\neg\mathsf{p_a}$, which, nevertheless, is not expressible in $\mathsf{TL[EF_{ns}]}$.

3. Notations and definitions

Basically, we follow the notations in [3]. But for the reader's convenience, we recall the relevant notations and definitions.

Throughout this section, let Σ be a finite alphabet and $L \subseteq T_\Sigma$.

A multicontext is a tree in $T_{\Sigma\cup\{*\}}$ such that at least one leaf is labelled by $*$, and no inner nodes are labelled by $*$. The leaves labelled by $*$ are called holes of the multicontext. In particular, each $a \in \Sigma$ can be seen as a multicontext with two holes. A context is a multicontext with exactly one leaf labelled by $*$.

Let C be a multicontext with holes $v_1, ..., v_n$ and $t_1, ..., t_n \in T_\Sigma$. Then $C\langle t_1, ..., t_n\rangle$ denotes the tree obtained by replacing the $v_1, ..., v_n$ by $t_1, ..., t_n$ respectively. In particular, $a\langle s, t\rangle$ denotes the tree with the root labelled by a and s, t as the left and right subtree respectively.

Let $s, t \in T_\Sigma$. Then $s \sim_L t$ iff for all contexts C, ($C\langle s\rangle \in L$ iff $C\langle t\rangle \in L$).

The equivalence classes of $\sim_L$ are called types of L, denoted by $Types(L)$. The type of a tree s is denoted by $type(s)$.

A tree language L is regular iff L has only a finite number of types, namely, $Types(L)$ is a finite set.

As a matter of fact, $\sim_L$ is a congruence on T_Σ in the sense that if $s \sim_L s'$ and $t \sim_L t'$, then $a\langle s, t\rangle \sim_L a\langle s', t'\rangle$ for all $a \in \Sigma$. Then it is easy to see that for all multicontexts C with holes $v_1, ..., v_n$, if $s_i \sim_L t_i$ for all $1 \leq i \leq n$, then $C\langle s_1, ..., s_n\rangle \sim_L C\langle t_1, ..., t_n\rangle$. Consequently for multicontext C with holes $v_1, ..., v_n$, $\alpha_1, ..., \alpha_n \in Types(L)$, we can write $C\langle \alpha_1, ..., \alpha_n\rangle$ to denote the type of any tree $C\langle s_1, ..., s_n\rangle$ with $type(s_i) = \alpha_i$ for all $1 \leq i \leq n$. In particular, $C\langle\alpha\rangle$ denotes the type of any tree $C\langle s\rangle$ with $type(s) = \alpha$, and $a\langle\alpha, \beta\rangle$ denotes the type of any tree $a\langle s, t\rangle$ with $type(s) = \alpha$ and $type(t) = \beta$.

Let $\alpha, \beta \in Types(L)$. Then $\alpha \preceq \beta$ if there is a context C such that $C\langle\alpha\rangle = \beta$. Moreover, if $\alpha \preceq \beta$ and $\beta \preceq \alpha$, we say that $\alpha \approx \beta$. If $\alpha \preceq \beta$ and not $\alpha \approx \beta$, then we say that $\alpha \prec \beta$.

It is easy to see that $\preceq$ is a preorder relation and $\approx$ is an equivalence relation on $Types(L)$.

The equivalence classes of $\approx$ are called strongly-connected-components (SCC's) of L, denoted by $SCCS(L)$. For any type α, the SCC (namely equivalence class of $\approx$) of α is denoted by $[\alpha]$.

Let s be a tree. Then the strongly-connected-component-set of s, denoted by $SCCS(s)$, is $\{[type(s|_v)] : v \in dom(s)\}$. And the delayed strongly-connected-component-set of s, denoted by $DSCCS(s)$, is $\{[type(s|_v)] : v \in dom(s) \text{ and } v > \varepsilon\}$. The rank of s, $rank(s)$, is defined to be the cardinality of $SCCS(s)$, namely $|SCCS(s)|$. The delayed rank, $drank(s)$, is defined to be the cardinality of $DSCCS(s)$, namely $|DSCCS(s)|$.

Let s be a tree and $a \in \Sigma$. Then $s[a]$ denotes the tree the same as s except that the root of s is labelled by a now.

Let s be a tree. Then the delayed type of s, denoted by $dtype(s)$, is a function from Σ to $Types(L)$ such that $dtype(s)(a) = type(s[a])$.

It is evident that $dtype(a\langle s,t\rangle)$ has nothing to do with a, consequently we can write $dtype(s,t)$ simply. It is also easy to see that if $s \sim_L s'$ and $t \sim_L t'$, then $dtype(s,t) = dtype(s',t')$. Thus we can write $dtype(\alpha,\beta)$ to denote the delayed type $dtype(s,t)$ with $type(s) = \alpha$ and $type(t) = \beta$.

4. New proof of characterization of TL[EF]

Before the proof, we give some definitions, propositions and lemmas.

Definition 1 (TL[EF] EF-Game) *Let s,t be two trees. Then the k-round Ehrenfeucht-Fraïssé game on s,t is played by two players spoiler and duplicator in turn:*

0-round game: if $s(\varepsilon) \neq t(\varepsilon)$, then spoiler wins, otherwise duplicator wins.

k-round game ($k > 0$): if $s(\varepsilon) \neq t(\varepsilon)$, then spoiler wins.

Otherwise spoiler should select some non-root node in one of the two trees, say v in s. If he fails to do so (namely both trees have only one node), then duplicator wins.

Otherwise duplicator should select some non-root node in the other tree, say w in t. If she fails to do so (namely exactly one of the two trees has only one node), then spoiler wins.

Otherwise spoiler and duplicator play the (k-1)-round game on $s|_v$ and $t|_w$.

We say that spoiler or duplicator has a winning strategy in the k-round TL[EF] EF-game on s,t if he or she can win regardless of the moves by the opponent.

It is not hard to see that if spoiler has a winning strategy in the k-round TL[EF] EF-game on s,t, then he has a winning strategy in the $(k+1)$-round game as well. Similarly if duplicator has a winning strategy in the $(k+1)$-round TL[EF] EF-game on s,t, then she has a winning strategy in the k-round game as well.

Similar to Corollary 2.2 in [5], we have the following proposition.

Proposition 1 *Let L be a tree language. If there is some $k \geq 0$ such that for all $s \in L$ and $t \notin L$, spoiler has a winning strategy in the k-round TL[EF] EF-game on s,t; then L is TL[EF] definable.*

Definition 2 (EF-admissible) *L is said to be EF-admissible if the following three properties are satisfied:*

[P1] *$dtype(\alpha,\beta) = dtype(\beta,\alpha)$*

[P2] *if $\alpha_0 \approx \beta_0$ and $\alpha_1 \approx \beta_1$, then $dtype(\alpha_0,\alpha_1) = dtype(\beta_0,\beta_1)$*

[P3] *if $\alpha \preceq \beta$, then $dtype(\alpha,\beta) = dtype(\beta,\beta)$*

where $\alpha_i, \beta_i (i = 0,1), \alpha, \beta \in Types(L)$.

Definition 3 (DSCCS-dependent) *L is said to be delayed-strongly-connected-component-set dependent (DSCCS-dependent) if for any trees s and t such that $DSCCS(s) = DSCCS(t)$, we have that $dtype(s) = dtype(t)$.*

Theorem 1 ([3,1]) *Let L be a regular tree language. Then the following three conditions are equivalent:*

(i) L is TL[EF]-definable.

(ii) L is EF-admissible.

(iii) L is DSCCS-dependent.

Proof.

(i) $\Rightarrow$ (ii):

Suppose that L is TL[EF]-definable.

[**P1**] is evident since TL[EF] can't distinguish between the left and right sons.

The proof of [**P2**] is exactly Lemma 3.3.7 in [1].

The proof of [**P3**] is exactly Lemma 3.3.6 in [1].

(ii) $\Rightarrow$ (iii):

Essentially the proof has been given in section 3.3.1 of [1].

Suppose that L is EF-admissible.

Let s,t be two trees such that $DSCCS(s) = DSCCS(t)$.

If $DSCCS(s) = DSCCS(t) = \emptyset$, then evidently $dtype(s) = dtype(t)$.

Now we assume $DSCCS(s) = DSCCS(t) \neq \emptyset$.

Let $\alpha_i = type(s|_i)$ and $\beta_i = type(t|_i)$, where $i = 0,1$. Then $[\alpha_i] \in DSCCS(t)$ and $[\beta_i] \in DSCCS(s)$, where $i = 0,1$. There are three cases.

Case I: there is i such that $[\alpha_0], [\alpha_1] \in SCCS(t|_i)$.

Then $\alpha_0, \alpha_1 \preceq \beta_i$ and $\beta_i \preceq \alpha_j$, $\beta_{1-i} \preceq \alpha_{j'}$ for some j, j'. Thus $\alpha_j \approx \beta_i$ and $\alpha_{1-j} \preceq \beta_i$, $\beta_{1-i} \preceq \beta_i$. So

$$\begin{aligned} dtype(\alpha_0,\alpha_1) &= dtype(\alpha_j,\alpha_{1-j}) = dtype(\beta_i,\alpha_{1-j}) \\ &= dtype(\beta_i,\beta_i) = dtype(\beta_i,\beta_{1-i}) = dtype(\beta_0,\beta_1) \end{aligned}$$

The first and last equations above are according to [**P1**] in the definition 2; the second equation is according to [**P2**]; the third and fourth equations are according to [**P3**].

Case II: there is i such that $[\beta_0], [\beta_1] \in SCCS(s|_i)$.

Similar to Case I.

Case III: neither I nor II holds.

Then there is i such that $[\alpha_0] \in SCCS(t|_i)$ and $[\alpha_1] \in SCCS(t|_{1-i})$ and there is j such that $[\beta_0] \in SCCS(s|_j)$ and $[\beta_1] \in SCCS(s|_{1-j})$. There are four subcases.

Subcase III.I: $i = j = 0$

Then $\alpha_0 \preceq \beta_0$, $\alpha_1 \preceq \beta_1$, $\beta_0 \preceq \alpha_0$ and $\beta_1 \preceq \alpha_1$. So we have $\alpha_0 \approx \beta_0$ and $\alpha_1 \approx \beta_1$.

Then $dtype(\alpha_0, \alpha_1) = dtype(\beta_0, \beta_1)$ according to [**P2**].

Subcase III.II: $i = j = 1$

Then $\alpha_0 \preceq \beta_1$, $\alpha_1 \preceq \beta_0$, $\beta_0 \preceq \alpha_1$ and $\beta_1 \preceq \alpha_0$. So we have $\alpha_0 \approx \beta_1$ and $\alpha_1 \approx \beta_0$.

Then according to [**P2**] and [**P1**], $dtype(\alpha_0, \alpha_1) = dtype(\beta_1, \beta_0) = dtype(\beta_0, \beta_1)$.

Subcase III.III: $i = 1 - j = 0$

Then $\alpha_0 \preceq \beta_0$, $\alpha_1 \preceq \beta_1$, $\beta_0 \preceq \alpha_1$ and $\beta_1 \preceq \alpha_0$. So we have $\alpha_0 \approx \beta_0 \approx \alpha_1 \approx \beta_1$.

Then $dtype(\alpha_0, \alpha_1) = dtype(\beta_0, \beta_1)$ according to [**P2**].

Subcase III.IV: $i = 1 - j = 1$

Then $\alpha_0 \preceq \beta_1$, $\alpha_1 \preceq \beta_0$, $\beta_0 \preceq \alpha_0$ and $\beta_1 \preceq \alpha_1$. So we have $\alpha_0 \approx \beta_1 \approx \alpha_1 \approx \beta_0$.

Then $dtype(\alpha_0, \alpha_1) = dtype(\beta_0, \beta_1)$ according to [**P2**].

(iii) $\Rightarrow$ (i):

Suppose L is DSCCS-dependent.

According to Proposition 1, it suffices to prove that for all s, t such that $type(s) \neq type(t)$, spoiler has a winning strategy in the $(drank(s)+drank(t))$-round game on s, t (because if this is true, then spoiler has a winning strategy in the $(2 \cdot |\mathrm{SCCS(L)}|)$-round game on s, t for all $s \in L$ and $t \notin L$).

Induction on $drank(s) + drank(t)$.

Let $s(\varepsilon) = a$ and $t(\varepsilon) = b$.

Induction base: $drank(s) + drank(t) = 0$.

Then both s and t have only one node. Since $type(s) \neq type(t)$, then $a \neq b$, consequently spoiler wins in the 0-round game.

Induction step: $drank(s) + drank(t) > 0$

If $a \neq b$, then spoiler wins.

Otherwise $DSCCS(s) \neq DSCCS(t)$ since $type(s) \neq type(t)$ and L is DSCCS-dependent. Consequently there is $[\gamma] \in DSCCS(s) \backslash DSCCS(t)$ or $[\gamma] \in DSCCS(t) \backslash DSCCS(s)$. Here we consider the former case, the latter case can be considered similarly.

Then spoiler selects $v > \varepsilon$ in s such that $[type(s|_v)] = [\gamma]$ and v is maximal in this sense.

If t has only one node, then spoiler wins.

Otherwise duplicator selects $w > \varepsilon$ in t.

Since $[type(s|_v)] = [\gamma]$ and v is maximal, $[\gamma] \notin DSCCS(s|_v)$, so $drank(s|_v) < drank(s)$. Then $drank(s|_v) + drank(t|_w) < drank(s) + drank(t)$.

We also have that $type(s|_v) \neq type(t|_w)$ for otherwise $[\gamma] = [type(s|_v)] = [type(t|_w)] \in DSCCS(t)$, a contradiction.

Then according to the induction hypothesis, spoiler has a winning strategy in the $(drank(s|_v) + drank(t|_w))$-round game on $s|_v$ and $t|_w$. Consequently spoiler has a winning strategy in the $(drank(s) + drank(t) - 1)$-round game on $s|_v$ and $t|_w$.

Thus we conclude that spoiler has a winning strategy in the $(drank(s)+drank(t))$-round game on s, t. $\square$

5. Characterization of $\mathsf{TL}[\mathsf{EF_{ns}}]$

Before giving the characterization of $\mathsf{TL}[\mathsf{EF_{ns}}]$, we give some definitions, propositions and lemmas.

Definition 4 ($\mathsf{TL}[\mathsf{EF_{ns}}]$ EF-Game) *Let s, t be two trees. Then the k-round Ehrenfeucht-Fraïssé game on s, t is played by two players spoiler and duplicator in turn:*

0-round game: if $s(\varepsilon) \neq t(\varepsilon)$, then spoiler wins, otherwise duplicator wins.

k-round game($k > 0$): if $s(\varepsilon) \neq t(\varepsilon)$, then spoiler wins.

Otherwise spoiler selects some node in one of the two trees, say v in s. And duplicator selects some node in the other tree, say w in t. Then spoiler and duplicator play the (k-1)-round game on $s|_v$ and $t|_w$.

Similar to the $\mathsf{TL}[\mathsf{EF}]$ EF-game, we have that if spoiler has a winning strategy in the k-round $\mathsf{TL}[\mathsf{EF_{ns}}]$ EF-game on s, t, then he has a winning strategy in the $(k+1)$-round game as well. Similarly if duplicator has a winning strategy in the $(k+1)$-round $\mathsf{TL}[\mathsf{EF_{ns}}]$ EF-game on s, t, then she has a winning strategy in the k-round game as well.

Similar to Proposition 1, we have the following proposition for $\mathsf{TL}[\mathsf{EF_{ns}}]$.

Proposition 2 *Let L be a tree language. If there is some $k \geq 0$ such that for all $s \in L$ and $t \notin L$, spoiler has a winning strategy in the k-round $\mathsf{TL}[\mathsf{EF_{ns}}]$ EF-game on s, t; then L is $\mathsf{TL}[\mathsf{EF_{ns}}]$ definable.*

Let L be a tree language and $\alpha \in Types(L)$. The root letters of α, denoted by $rletters(\alpha)$, is defined to be $\{a \in \Sigma | \text{there is } t \text{ such that } t(\varepsilon) = a, type(t) = \alpha\}$.

Definition 5 ($\mathsf{EF_{ns}}$-admissible) *A tree language L is said to be $\mathsf{EF_{ns}}$-admissible if it is EF-admissible* *and satisfies the following condition* [**P4**].

[**P4**] *if $a \in rletters(\alpha)$, then $a\langle\alpha,\alpha\rangle = \alpha$, where $\alpha \in Types(L)$.*

Lemma 1 *Let L be $\mathsf{EF_{ns}}$-admissible and s,t be trees. If $SCCS(s) = SCCS(t)$ and $s(\varepsilon) = t(\varepsilon)$, then $type(s) = type(t)$.*

Proof.

Let $a = s(\varepsilon) = t(\varepsilon)$, $\alpha = type(s)$ and $\beta = type(t)$.

Since L is $\mathsf{EF_{ns}}$-admissible, then it is EF-admissible, so DSCCS-dependent according to Theorem 1. Consequently $dtype(a\langle s,s\rangle) = dtype(a\langle t,t\rangle)$ since $DSCCS(a\langle s,s\rangle) = SCCS(s) = SCCS(t) = DSCCS(a\langle t,t\rangle)$. So $type(a\langle s,s\rangle) = type(a\langle t,t\rangle)$.

Because $a \in rletters(\alpha)$ and $a \in rletters(\beta)$, according to [**P4**] in Definition 5, $\alpha = a\langle\alpha,\alpha\rangle = type(a\langle s,s\rangle) = type(a\langle t,t\rangle) = a\langle\beta,\beta\rangle = \beta$. □

The following lemma is obvious.

Lemma 2 *Let L be defined by $\mathsf{TL}[\mathsf{EF_{ns}}]$ formula φ and s,t be two trees. If s and t satisfy the same formulas in $cl(\varphi)$, then $type(s) = type(t)$.*

Theorem 2 *Let L be a regular tree language. Then the following two conditions are equivalent:*

(i) L is $\mathsf{TL}[\mathsf{EF_{ns}}]$-definable.

(ii) L is $\mathsf{EF_{ns}}$-admissible.

Proof.

(i) ⇒ (ii):

Suppose that L is TL[$\mathsf{EF_{ns}}$] definable.

Since $\mathsf{TL}[\mathsf{EF_{ns}}]$ can be seen as a sublogic of $\mathsf{TL}[\mathsf{EF}]$, we know that L is EF-admissible from Theorem 1.

Now we consider [**P4**].

Let s be a tree such that $type(s) = \alpha$, $s(\varepsilon) = a$.

Let $t = a\langle s,s\rangle$. We can prove that for all $\mathsf{TL}[\mathsf{EF_{ns}}]$ formula φ, $s \models \varphi$ iff $t \models \varphi$ by induction on the structure of φ.

Because L is TL[$\mathsf{EF_{ns}}$] definable, then according to Lemma 2, we have $type(s) = type(t)$, $a\langle\alpha,\alpha\rangle = \alpha$.

(ii) ⇒ (i):

Suppose that L is $\mathsf{EF_{ns}}$-admissible.

According to Proposition 2, it suffices to prove that for all s,t with $type(s) \neq type(t)$, spoiler has a winning strategy in the $(3 \cdot (rank(s) + rank(t)))$-round game on s and t.

Induction on $rank(s) + rank(t)$.

Let $type(s) = \alpha$, $type(t) = \beta$, $s(\varepsilon) = a$, $t(\varepsilon) = b$.

Base case: $rank(s) + rank(t) = 2$ (because $rank(s), rank(t) \geq 1$).

Then $rank(s) = rank(t) = 1$, $SCCS(s) = \{\alpha\}$ and $SCCS(t) = \{\beta\}$.

Spoiler selects some leaf v in s. And duplicator selects w in t.

Spoiler selects some leaf w' in t such that $w' \geq w$. And duplicator has no choice but to select v in s.

Because $type(s|_v) = \alpha$ and $type(t|_{w'}) = \beta$, we have $s(v) \neq t(w')$, spoiler wins. Consequently spoiler has a winning strategy in the 2-round game on s,t. Thus spoiler has a winning strategy in the $(3 \cdot (rank(s) + rank(t)))$-round game on s,t.

Induction step: $rank(s) + rank(t) > 2$.

If $a \neq b$, then spoiler wins.

Otherwise we have that $SCCS(s) \neq SCCS(t)$ according to Lemma 1. There are three cases.

Case I: there is $[\gamma] \in SCCS(s) \backslash \{[\alpha]\}$ and $[\gamma] \notin SCCS(t)$.

Evidently $\gamma \prec \alpha$.

Spoiler selects some $v > \varepsilon$ in s such that $[type(s|_v)] = [\gamma]$ and v is maximal in this sense.

Duplicator selects some w in t.

Since $[\alpha] \notin SCCS(s|_v)$, we have $rank(s|_v) < rank(s)$, thus $rank(s|_v) + rank(t_w) < rank(s) + rank(t)$.

We also have $type(s|_v) \neq type(t|_w)$ because otherwise $[\gamma] = [type(s|_v)] = [type(t|_w)] \in SCCS(t)$, a contradiction.

Then according to the induction hypothesis, spoiler has a winning strategy in the $(3 \cdot (rank(s|_v) + rank(t|_w)))$-round game on $s|_v$ and $t|_w$. Thus spoiler has a winning strategy in the $(3 \cdot (rank(s) + rank(t)) - 1)$-round game on $s|_v$ and $t|_w$.

We conclude that spoiler has a winning strategy in the $(3 \cdot (rank(s) + rank(t)))$-round game on s,t.

Case II: there is $[\gamma] \in SCCS(t) \backslash \{[\beta]\}$ and $[\gamma] \notin SCCS(s)$.

Similar to Case I.

Case III: neither I nor II holds.

Then $SCCS(s) \backslash \{[\alpha]\} \subseteq SCCS(t)$ and $SCCS(t) \backslash \{[\beta]\} \subseteq SCCS(s)$.

We must have that $[\alpha] \notin SCCS(t)$ or $[\beta] \notin SCCS(s)$. For otherwise $SCCS(s) \subseteq SCCS(t)$ and $SCCS(t) \subseteq SCCS(s)$, $SCCS(s) = SCCS(t)$, then according to Lemma 1, $\alpha = \beta$, a contradiction.

Without loss of generality, suppose that $[\alpha] \notin SCCS(t)$.

Then spoiler selects some v in s such that $[type(s|_v)] = [\alpha]$ and v is maximal in this sense.

Duplicator selects some w in t.

If $s(v) \neq t(w)$, then spoiler wins.

Otherwise if $type(t|_w) \prec type(t)$, then we have that $rank(t|_w) < rank(t)$, and $rank(s|_v) + rank(t|_w) < rank(s) + rank(t)$.

Moreover, $type(s|_v) \neq type(t|_w)$ because otherwise $[\alpha] = [type(s|_v)] = [type(t|_w)] \in SCCS(t)$, a contradiction.

Then according to the induction hypothesis, spoiler has a winning strategy in the $(3 \cdot (rank(s|_v) + rank(t|_w)))$-round game on $s|_v$ and $t|_w$. Thus spoiler has a winning strategy in the $(3 \cdot (rank(s) + rank(t)) - 1)$-round game on $s|_v$ and $t|_w$.

So spoiler has a winning strategy in the $(3 \cdot (rank(s)+rank(t)))$-round game on s, t in this case.

Otherwise $[type(t|_w)] = [type(t)] = [\beta]$.

If v is a leaf in s, then spoiler can select some leaf $w' \geq w$ in t, duplicator has to select v in s. Since $[type(s|_v)] = [\alpha]$, we have $s(v) \neq t(w')$ for otherwise $[\alpha] = [type(s|_v)] = [type(t|_{w'})] \in SCCS(t)$, a contradiction. So spoiler wins in this case.

Otherwise v is an inner node of s. Let $a' = s(v) = t(w)$, $type(s|_v) = \alpha'$, $type(t|_w) = \beta'$ and $\alpha_i = type(s|_{vi})$ where $i = 0, 1$ (see Fig. 1).

Because v is maximal, $\alpha_0, \alpha_1 \prec \alpha$. Since $SCCS(s) \backslash \{[\alpha]\} \subseteq SCCS(t)$, it must be that $[\alpha_0], [\alpha_1] \in SCCS(t)$, so $\alpha_0, \alpha_1 \preceq \beta$. In fact, we must have that $\alpha_0, \alpha_1 \prec \beta$.

To the contrary, suppose that $[\alpha_0] = [\beta]$ (or $[\alpha_1] = [\beta]$).

Because $[\alpha_0] = [\beta]$ and $\alpha_1 \preceq \beta$, we have that $dtype(\alpha_0, \alpha_1) = dtype(\beta, \alpha_1) = dtype(\beta, \beta) = dtype(\beta', \beta')$ (The first and third equations are by [**P2**], second by [**P3**]). Consequently

$$\alpha' = a'\langle \alpha_0, \alpha_1 \rangle = dtype(\alpha_0, \alpha_1)(a') =$$
$$dtype(\beta', \beta')(a') = a'\langle \beta', \beta' \rangle = \beta'$$

(The last equation above holds because $a' \in rletters(\beta')$ and [**P4**]).

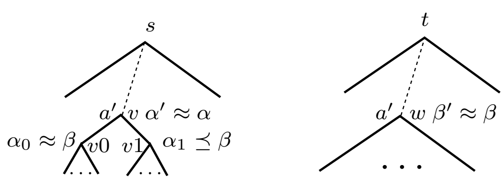

Fig. 1. $[\alpha_0] = [\beta]$

Then $[\alpha] = [\alpha'] = [\beta'] = [\beta] \in SCCS(t)$, a contradiction.

So it must be that $\alpha_0, \alpha_1 \prec \beta$. Then $[\beta] \notin SCCS(s|_v)$.

Now spoiler can select $w' \geq w$ in t such that $[type(t|_{w'})] = [\beta]$ and w' is maximal in this sense.

Duplicator selects $v' \geq v$ in s.

If $s(v') \neq t(w')$, then spoiler wins.

Otherwise if $v' > v$, then $type(s|_{v'}) \prec \alpha$ since v is maximal. In this case, we can get that $rank(s|_{v'}) + rank(t|_{w'}) < rank(s) + rank(t)$ and $type(s|_{v'}) \neq type(t|_{w'})$ and spoiler has a winning strategy in the $(3 \cdot (rank(s) + rank(t)))$-round game on s, t by the induction hypothesis.

Otherwise $v' = v$. In this case $type(s|_{v'}) \neq type(t|_{w'})$ for otherwise $[\alpha] = [type(s|_{v'})] = [type(t|_{w'})] \in SCCS(t)$, a contradiction.

Since L is DSCCS-dependent according to Theorem 1, we have that $DSCCS(s|_{v'}) \neq DSCCS(t|_{w'})$.

Without loss of generality, suppose that there is $[\gamma] \in DSCCS(s|_{v'}) \backslash DSCCS(t|_{w'})$, the other case is similar.

We know that $[\alpha] \notin SCCS(t)$ and $[\beta] \notin SCCS(s|_v)$. Thus we have that $[type(s|_{v'})] = [\alpha] \notin SCCS(t|_{w'})$ and $[type(t|_{w'})] = [\beta] \notin SCCS(s|_{v'})$.

So

$$[\gamma] \notin DSCCS(t|_{w'}) \cup \{[type(t|_{w'})]\} = SCCS(t|_{w'}).$$

Then spoiler can select $v'' > v'$ in s such that $[type(s|_{v''})] = [\gamma]$.

Duplicator selects $w'' \geq w'$ in t (see Fig. 2).

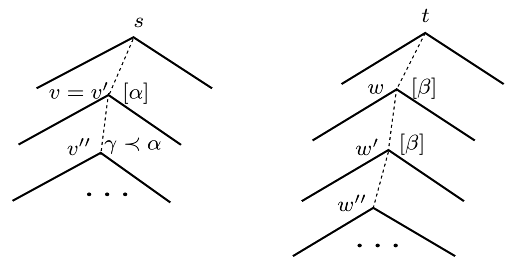

Fig. 2. game positions

If $s(v'') \neq t(w'')$, then spoiler wins.

Otherwise $type(s|_{v''}) \neq type(t|_{w''})$, for otherwise

$$[\gamma] = [type(s|_{v''})] = [type(t|_{w''})] \in SCCS(t|_{w'}),$$

a contradiction.

Since v' is maximal and $v'' > v'$, we have that

$$rank(s|_{v''}) + rank(t|_{w''}) < rank(s) + rank(t).$$

Then according to induction hypothesis, spoiler has a winning strategy in the $(3 \cdot (rank(s|_{v''}) + rank(t|_{w''})))$-round (thus $(3 \cdot (rank(s) + rank(t)) - 3)$-round) game on $s|_{v''}$ and $t|_{w''}$.

Consequently we conclude that spoiler has a winning strategy in the $(3 \cdot (rank(s) + rank(t)))$-round game on s, t. □

6. Conclusions and Remarks

In this note, we give a new proof of characterization of TL[EF] following the Ehrenfeucht-Fraïssé games approach and extend this proof to the characterization of $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$.

The property [**P4**] in the characterization of $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$(Definition 5) can be seen as one kind of binary-tree extension of stutter-invariance concept of words [7].

We define the binary-stutter-invariance as follows:

Let L be a tree language, t be a tree and t' be a tree obtained from t by applying the following operation: replacing subtree $t|_v$ $(v \in dom(t))$ by $a\langle t|_v, t|_v\rangle(a = t(v))$ in t (see Fig. 3). If ($t \in L$ iff $t' \in L$ for any t and t' stated above), then we say that L is binary-stutter-invariant.

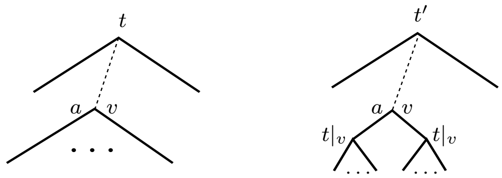

Fig. 3. binary stutter

From Theorem 1 and Theorem 2, it is easy to see that a $\mathsf{TL}[\mathsf{EF}]$-definable tree language L is $\mathsf{TL}[\mathsf{EF}_{\mathsf{ns}}]$-definable iff L is binary-stutter-invariant.

Acknowledgements I want to thank Prof. Wenhui Zhang for his reviews on this paper and discussions with me. Moreover, I would like to thank anonymous referees for their valuable comments and suggestions.

References

[1] M. Bojańczyk, Decidable properties of tree languages, PHD thesis, Warsaw University, 2004.

[2] M.Benedict, L.Segoufin. Regular tree languages definable in FO, STACS'05, LNCS 3404, 327-339, 2005.

[3] M. Bojańczyk, I. Walukiewicz. Characterizing EF and EX tree logics, CONCUR'04, LNCS 3170, 131-145, 2004.

[4] Z. Ésik, P. Weil. On certain logically defined tree languages. FSTTCS'03, LNCS 2914, 195-207, 2003.

[5] K. Etessami, T. Wilke. An until hierarchy for temporal logic, LICS'96, 108-117, IEEE Computer Society Press, 1996.

[6] A. Pothoff. First-order logic on finite trees. In Theory and Practice of Software Development, LNCS 915, 125-139, 1995.

[7] D. Peled, T. Wilke. Stutter-invariant temporal properties are expressible without the next-time operators, Information Processing Letters, V.63(5), 243-246,1997.

[8] T. Wilke. Classifying discrete temporal properties. STACS'99, LNCS 1563, 32-46, Springer-Verlag, 1999.