Temporal Logics with Reference Pointers and Computation Tree Logics

Valentin Goranko
Department of Mathematics, Rand Afrikaans University
PO Box 524, Auckland Park 2006, Johannesburg, South Africa
e-mail: vfg@na.rau.ac.za

ABSTRACT. *A complete axiomatic system CTL_{rp} is introduced for a temporal logic for finitely branching ω^+-trees in a language extended with so called reference pointers. Syntactic and semantic interpretations are constructed for the branching time computation tree logic CTL^* into CTL_{rp}. In particular, that yields a complete axiomatization for the translations of all valid CTL^*-formulae. Thus, the temporal logic with reference pointers is brought forward as a simpler (with no path quantifiers), but in a way more expressive medium for reasoning about branching time.*
KEYWORDS: Computation tree logics, temporal logics, reference pointers, axiomatic system, completeness.

Introduction

The article deals with branching time temporal logics for computation trees of non-deterministic or concurrent programs. This area of research in theoretical computer science has been vigorously developing in the past twenty years, particularly since the seminal paper of [Pnu 77] which proposes the use of temporal logics for formal specification, analysis and verification of programs, in particular for reasoning about non-deterministic, concurrent and reactive programs. Since then, a number of linear and branching time temporal logical systems have been put forward with this purpose, some of the most popular ones of the latter type being UB, CTL, CTL* and variations of them (see [ES 89, Eme 90], [Pen 95], and [Sti 92] for surveys on these). The logic CTL* will be in the focus of this paper, though the results will accordingly apply to all others subsumed by it, as well. Besides the computational interpretation, branching time logics have important philosophical and purely logical aspects, in particular those related to decidability and decision procedures generated by automata-theoretic methods, complexity, model checking and axiomatizations. For some important results and further discussion on these the reader is referred to [GuS 85, Tho 88, Wol 95, ZaC 93, Zan 96] in addition to the above mentioned surveys. Of course, [Pri 67] should be added as a classical reference to philosophical aspects of time.

Here we propose syntactic and semantic interpretations of CTL* into the temporal logic with reference pointers (TL$_{rp}$) introduced in [Gor 94] and [Gor 96], where expressiveness and (un)decidability are discussed and a complete axiomatic system is constructed. The language of TL$_{rp}$ is a kind of *hybrid language*, see [BS 95, BT 98, BT 99], combining features of both propositional modal and classical first-order languages. It has a great expressive power, in particular admitting translation of CTL* therein, after an appropriate modification of its semantics into *extended computation trees.* Moreover, TL$_{rp}$ enables formalization of properties beyond the expressiveness of CTL*, like talking about the past and expressing various looping and non-looping conditions.

Of course, CTL* is known to be embeddable in another very expressive logical system, viz. Kozen's *μ-calculus* (see [Dam 94]), so the natural question arises about the relationship, advantages and disadvantages of the temporal logic with reference pointers proposed here as compared to the μ-calculus. We defer that discussion to the concluding remarks.

The main result in the paper is the construction of a complete axiomatic system (though involving an infinitary rule) CTL$_{rp}$ for extended computation trees, which is furthermore decidable, due to the general result in [GuS 85]. In particular we have obtained a complete axiomatization of the translations of all valid CTL*-formulae.

The paper begins with a preliminary section 2 which briefly describes the syntax and semantics of temporal logics with reference pointers and of the computation tree logic CTL*. In section 3 we introduce syntactic and semantic interpretations of CTL* into the temporal logic with reference pointers for extended computation trees. Section 4 presents a complete axiomatic system and completeness theorem for the temporal logics with reference pointers for extended trees CTL$_{rp}$. Finally, we discuss some open problems and directions for further research.

The reader is assumed to have some background in propositional temporal logics (syntax, semantics, deductive systems and completeness theorem) within either of [Ben 91], [Bur 84], [Gol 87].

1 Preliminaries

1.1 Temporal logic with reference pointers.

We consider a propositional temporal language with

- a set of atomic propositions $\{p_1, p_2, \ldots\}$;
- propositional connectives $\neg, \rightarrow$, and respectively definable $\wedge, \vee, \leftrightarrow$, $\top$ (*true*) and $\bot$ (*false*);

- temporal operators **A** (*always*) with dual **E** (*sometime*), **G** (*always in the future*) with dual **F** (*eventually*), and **X** (*at all immediate time-successors*), with dual **N** (*at some immediate time-successor*).

As we shall see further, the other traditional temporal operators **U** (*until*), **H** (*always in the past*), and its dual **P** (*sometime in the past*) are definable in terms of **A** and **G** using *reference pointers*. The operator **X** is definable, too, but we shall retain it in the language in order to preserve the notion of a CTL*-model.

The idea of reference pointers (see [Gor 94] or [Gor 96]) in brief is as follows. One or more pairs of new symbols $(\downarrow_k, \uparrow_k)$ are added to the temporal language; the *point of reference* $\downarrow_k$ is a unary connective, and the *reference pointer* $\uparrow_k$ is an atomic symbol like a propositional variable. In a sense, these reference pairs play a rôle of variables on states (instants, possible worlds, etc.) and quantifiers, or rather *binders* over these variables. It is therefore natural that some first-order style notions have their natural analogues in the language with reference pointers, viz.:

- The first occurrence of the binder $\downarrow_i$ in the formula $\downarrow_i \varphi$ has a *scope* φ.
- An occurrence of $\uparrow_i$ in a formula φ is *bound* if it is in the scope of an occurrence of $\downarrow_i$; otherwise it is *free*.
- If φ and ψ are formulae, $\varphi(\psi/\uparrow_i)$ denotes the result of simultaneous substitution of all free occurrences of $\uparrow_i$ in φ by ψ.
- A formula φ is *closed* if there are no free occurrences of $\uparrow$'s in φ.

Here is the intuitive semantics of the new symbols: when a formula $\downarrow\varphi$ is being evaluated in a model, $\downarrow$ marks the state of evaluation s, and all free occurrences of $\uparrow$ in φ are rendered true at the state s and only there, thus enabling references to that state throughout φ.

The formal semantics follows below.

Let $\mathcal{L}\updownarrow_t^k$ be a temporal language as above, extended with k pairs of reference pointers. The models of $\mathcal{L}\updownarrow_t^k$ are the same as the models for the classical temporal logic: $\langle T, R, <, V\rangle$, where T is a time flow, i.e. set of moments (states), $<$ is the successor relation , R is the *immediate* successor relation, and V is a valuation of the atomic propositions in T. In order to define truth of a formula from $\mathcal{L}\updownarrow_t^k$ at a point of a model we extend the well-known standard translation ST (see [Ben 91]) as follows. Let L_1 be the first-order language containing binary predicates R and <, and a countable set of unary predicates $\{P_1, P_2, \ldots\}$. For technical convenience we split the set of individual variables of L_1 into two disjoint subsets: $W = \{x, w_1, \ldots, w_k\}$ and $Y = \{y_0, y_1, y_2, \ldots\}$, where each of x and w's plays a special rôle, viz.:

- x will represent the actual point in time (the current "now");

- w_i will represent the point of reference for the pointer $\uparrow_i$ ("then$_i$") for $i = 1, \ldots, k$.

We now define the standard translation ST of $\mathcal{L}\updownarrow_t^k$ into L_1 recursively as follows:

1. $ST(p_i) = P_i x$,
2. $ST(\uparrow_i) \ = \ (x = w_i)$,
3. $ST(\neg\varphi) = \neg ST(\varphi)$,
4. $ST(\varphi \wedge \psi) = ST(\varphi) \wedge ST(\psi)$,
5. $ST(\mathbf{X}\varphi) = \forall y(x\mathrm{R}y \rightarrow ST(\varphi)(y/x))$,
6. $ST(\mathbf{G}\varphi) = \forall y(x{<}y \rightarrow ST(\varphi)(y/x))$,
7. $ST(\mathbf{A}\varphi) = \forall y(ST(\varphi)(y/x))$,
8. $ST(\downarrow_i \varphi) = ST(\varphi)(x/w_i)$.

In 5, 6, and 7 above y is the first variable from Y, not occurring in $ST(\varphi)$; u/v means uniform substitution of u for all free occurrences of v.

Note that if φ is in $\mathcal{L}\updownarrow_t^k$ then x and $w_1, \ldots, w_k$ can only have free occurrences in $ST(\varphi)$, where they are the only possibly free variables. Furthermore, φ is closed if and only if no w_i occurs in $ST(\varphi)$.

The model $\mathcal{M} = \langle T, R, <, V\rangle$ can be regarded as an L_1-model where R is interpreted by R, < by $<$, and P_i by $V(p_i), i = 0, 1, 2, \ldots$. In order to distinguish validity in $\mathcal{M}$ as an L_1-model from validity in $\mathcal{M}$ as a temporal model we shall use the symbol $\Vdash$ for the former and $\models$ for the latter. Now, we define truth at a point for any *closed* formula φ:

$$\mathcal{M} \models \varphi[s] \ \text{ if } \ \mathcal{M} \Vdash ST(\varphi)(s/x),$$

and then validity in a model:

$$\mathcal{M} \models \varphi \ \text{ if } \ \mathcal{M} \models \varphi[t] \ \text{ for every } t \in T, \ \text{ i.e. if } \ \mathcal{M} \Vdash \forall x ST(\varphi).$$

Finally, φ is *valid in a temporal structure* if it is valid in every model on the structure, and φ is (*universally*) *valid* if it is valid in every temporal structure.

Here we have only defined validity for closed formulae since only these formulae have a determined meaning, and we shall not be interested in non-closed formulae on their own.

The reference pointers considerably increase the expressiveness of the language, as numerous examples in [Gor 96] testify. The basic *temporal logic with reference pointers* outlined here was introduced and axiomatized in [Gor 94]. Languages with reference pointers have been further studied in [BT 98, BT 99] where, inter alia, some weaker languages with pointers (binders) without universal modality have been axiomatized.

1.2 Computation Tree Logics

The full branching time computation tree logic CTL* was introduced in [EmH 83]. Here we offer a very brief summary of the syntax and semantics of CTL*, referring the reader to [ES 89], [Eme 90], or [Pen 95] for more detail.

1.2.1 Syntax

The language of CTL* is a propositional language with a set of atomic propositions (propositional variables) AP, temporal operators **X** (*nexttime*) and **U** (*until*), and path quantifiers, ∀ meaning "for all paths", and its dual ∃, meaning "for some path", which will be regarded as an abbreviation of ¬∀¬.

Two types of formulae are traditionally introduced for CTL*, *state formulae* and *path formulae*. This distinction is essential for some of its restricted versions like CTL, but is also convenient for the truth definition: the former are evaluated at states and the latter — at paths in the model. Here is the joint recursive definition of state and path formulae:

- every atomic proposition is a state formula;
- if ϕ, ψ are state formulae then so are $\neg\phi$ and $\phi \rightarrow \psi$;
- if ϕ is a path formula then $\forall\phi$ is a state formula;

- every state formula is a path formula;
- if ϕ, ψ are path formulae then so are $\neg\phi$, $\phi \rightarrow \psi$, $\mathbf{X}\phi$ and $\phi\mathbf{U}\psi$;

These definitions can be combined into a uniform definition of a *formula of* CTL*, the Backus-Naur form (BNF) of which is:

$$\phi := p \mid \neg\phi \mid \phi_1 \rightarrow \phi_2 \mid \mathbf{X}\phi \mid \phi_1\mathbf{U}\phi_2 \mid \forall\phi,$$

where p is an atomic proposition.

Two of the most popular predecessors of CTL* are:

- UB (introduced in [BMP 81]), the language of which contains the temporal operators **X**, **G** (*always in the future*), with dual **F** (*eventually*) but not **U**. The BNF definition of formulae of UB is:

$$\phi := p \mid \neg\phi \mid \phi_1 \rightarrow \phi_2 \mid \exists\mathbf{X}\phi \mid \exists\mathbf{G}\phi \mid \forall\mathbf{G}\phi;$$

- CTL (introduced in [CE 81]), with the same language as CTL* and BNF definition of formulae:

$$\phi := p \mid \neg\phi \mid \phi_1 \rightarrow \phi_2 \mid \exists\mathbf{X}\phi \mid \exists(\phi_1\mathbf{U}\phi_2) \mid \forall(\phi_1\mathbf{U}\phi_2).$$

These will not be discussed further but, being subsystems of CTL*, the interpretations of CTL* introduced here will apply for them, too.

1.2.2 Semantics of CTL*

Intuitively, models for CTL* are *computation trees* generated by executions of non-deterministic or concurrent programs. They consist of all possible computation paths of such a program (i.e. sequences of consecutive states in an execution of the program).

Remark: In fact, computation paths may loop or meet, hence a typical model for CTL* should rather be a directed graph. Every such a model, however, is indistinguishable in the language of CTL* from a rooted tree-like model obtained from the former by *unwinding* (see [Eme 90]). Therefore the class of rooted tree-like models $\mathcal{T}$ is adequate for CTL* and we restrict our attention to that class.

The models from $\mathcal{T}$ are formalized as triples $\langle S, R, V\rangle$, where S is a nonempty set of *states*, R is the *immediate successor relation* in S, and $V : AP \to \mathcal{P}(S)$ is a *valuation* assigning to each atomic proposition the set of states at which it is true. We shall impose the additional assumption (which is quite natural in view of the interpretation of the models as computation trees) that every *R-path* (i.e. maximal linearly ordered set of states) is isomorphic to ω. Trees satisfying this condition will be called ω-trees.

The basic semantic notion for CTL* is *truth at a state s of a model $\mathcal{M}$* defined by simultaneous induction on state and path formulae as follows:

(S1) $\mathcal{M} \models_s p$ if $s \in V(p)$;

(S2) $\mathcal{M} \models_s \neg\phi$ if not $\mathcal{M} \models_s \phi$;

(S3) $\mathcal{M} \models_s \phi \to \psi$ if $\mathcal{M} \models_s \phi$ implies $\mathcal{M} \models_s \psi$;

(S4) $\mathcal{M} \models_s \forall\phi$, where ϕ is a path formula, if for every path $\mathbf{p}$ beginning from s, $\mathcal{M} \models_{\mathbf{p}} \phi$ holds.

In the following clauses $\mathbf{p}$ is a path $\{\mathbf{p}_0, \mathbf{p}_1, \ldots\}$, and $\mathbf{p}^i$ is the suffix path $\{\mathbf{p}_i, \mathbf{p}_{i+1}, \ldots\}$

(P1) $\mathcal{M} \models_{\mathbf{p}} \phi$, where ϕ is a state formula, if $\mathcal{M} \models_{\mathbf{p}_0} \phi$;

(P2) $\mathcal{M} \models_{\mathbf{p}} \neg\phi$ if not $\mathcal{M} \models_{\mathbf{p}} \phi$;

(P3) $\mathcal{M} \models_{\mathbf{p}} \phi \to \psi$ if $\mathcal{M} \models_{\mathbf{p}} \phi$ implies $\mathcal{M} \models_{\mathbf{p}} \psi$;

(P4) $\mathcal{M} \models_{\mathbf{p}} \mathbf{X}\phi$, if $\mathcal{M} \models_{\mathbf{p}^1} \phi$;

(P5) $\mathcal{M} \models_{\mathbf{p}} \phi\mathbf{U}\psi$, if there is $i \geq 0$ such that $\mathcal{M} \models_{\mathbf{p}^i} \psi$ and for every j, such that $0 \leq j < i$, $\mathcal{M} \models_{\mathbf{p}^j} \phi$ holds.

A state formula ϕ is *valid in a model* $\mathcal{M}$, denoted $\mathcal{M} \models \phi$, if ϕ is true at every state of $\mathcal{M}$. A state formula ϕ is CTL*-*valid* if it is valid in every CTL*-model.

According to my knowledge, no finite complete axiomatization for CTL* with respect to the class of standard models has been published yet (but some completeness results for more general semantics are presented in [Sti 92]).

The validity in CTL* was proved decidable in deterministic double exponential time in [EmS 84].

2 Interpretation of CTL* into the Temporal Logic with Reference Pointers

2.1 Extended Computation Trees for CTL*-models.

In this section we transform the CTL*-models from $\mathcal{T}$ into a form suitable for the language with reference pointers.

For any model $T = \langle S, R, V \rangle$ from $\mathcal{T}$ we define an *extended computation tree of* T: $T^e = \langle S^e, R^e, <, V^e \rangle$ as follows. Let

$$\mathcal{F} = \{s_x : x \text{ is a maximal path in } S\}$$

be a set disjoint from S. Then we put:

$S^e = S \cup \mathcal{F}$; $R^e = R$;

$< = R^+ \cup \{(t, s_x) : t \in S \text{ and } x \text{ is a maximal path in } S \text{ containing } t\}$,
where R^+ is the transitive closure of R;

Finally, V^e coincides with V over S and is arbitrarily extended over all states from $\mathcal{F}$, e.g. by declaring all atomic propositions false at every $s \in \mathcal{F}$.

Intuitively, T^e extends T by adding a ”frontier” $\mathcal{F}$ of transfinite ”terminal ends” for all maximal paths, each terminal end $<$-seen from all states of the corresponding path and only from them. Thus, every maximal path in an extended ω-tree will be of ordinal type ω^+, so we shall refer to such trees as *ω^+-trees.*

The class of extended computation trees for CTL*-models will be denoted by $\mathcal{ECT}$.

A (extended) computation tree is called *finitely branching* if every node has finitely many immediate successors. The class of finitely branching extended computation trees will be denoted by $\mathcal{FECT}$.

Let us note that every formula from $\mathcal{L}\updownarrow_t^k$ can be translated into a Π^1_1 formula of the monadic second-order language L_2 obtained from L_1 by treating the unary predicates as variables. The following result implies that every such a formula is valid in $\mathcal{ECT}$ iff it is valid in $\mathcal{FECT}$, and therefore $\mathcal{ECT}$ and $\mathcal{FECT}$ provide equivalent semantics for any temporal logic with reference pointers.

Theorem 1 *If a Π^1_1 formula of L_2 is falsifiable in an ω^+-tree then it is falsifiable in a finitely branching ω^+-tree.*

This theorem follows from the fact that for every positive integer n, every ω^+-tree T is n-equivalent to a finitely branching one. The proof of this result is rather long and technically involved. It goes through 3 major steps. First, using Ehrenfeucht games it can be shown that T is n-equivalent to an ω^+-tree finitely branching at the first k levels, which in turn is proved n-equivalent to an "almost ω^+-tree" which is finitely branching at *all finite* levels, and every state has a successor which is a terminal state. Finally, every such a tree is n-equivalent to a finitely branching ω^+-tree. For a detailed proof see [Gor 99].

2.2 Syntactic Translation of CTL* into the Temporal Logic with Reference Pointers.

We define a translation τ of the formulae of CTL* into the temporal logic with reference pointers, assuming that both languages share the same set of atomic propositions, inductively as follows. (Note that the states without successors (the terminal ends) in an extended computation tree are precisely the added states s_x; the pointer $\downarrow_1$ will be used to indicate the current state of evaluation, while $\downarrow_2$ indicates the terminal end of the path on which the evaluation is being done.)

- $\tau(p) = p$;
- $\tau(\neg\phi) = \neg\tau(\phi)$;
- $\tau(\phi \to \psi) = \tau(\phi) \to \tau(\psi)$;
- $\tau(\forall\phi) = \downarrow_1 \mathbf{G} \downarrow_2 (\mathbf{G}\bot \to \mathbf{A}(\uparrow_1 \to \tau(\phi)))$;
- $\tau(\mathbf{X}\phi) = \mathbf{X}(\mathbf{F}\uparrow_2 \to \downarrow_1 \tau(\phi))$;
- $\tau(\phi\mathbf{U}\psi) = \downarrow_1 \tau(\psi) \vee \mathbf{F}(\mathbf{F}\uparrow_2 \wedge \tau(\psi) \wedge \mathbf{H}(\mathbf{P}\uparrow_1 \to \downarrow_1 \tau(\phi)))$.

The translation τ is faithful in the following sense:

Theorem 2 *For any state formula ϕ in the language of CTL* the following are equivalent:*
(i) ϕ is CTL-valid;*
(ii) $(\mathbf{F}\top \to \tau(\phi))$ is valid in the class $\mathcal{ECT}$;
(iii) $(\mathbf{F}\top \to \tau(\phi))$ is valid in the class $\mathcal{FECT}$;

Proof:
(ii) ⇔ (iii) follows from directly from theorem 1 or, using the equivalence between (i) and (ii), from the fact (see [Wol 95], Lemma 3.5) that every satisfiable CTL*-formula is satisfiable in a finitely branching tree.
(i) ⇔ (ii):
Recall that every CTL*-formula ϕ can be regarded as a path formula. A state formula regarded as a path formula is valid in a model if and only if it is true on every path of the model.

It can be proved by induction on ϕ that for every CTL*-model $\mathcal{M}$ and a path $\mathbf{p}$ in it,

$$\mathcal{M} \models_{\mathbf{p}} \phi \ \text{ iff } \ \mathcal{M}^e \Vdash \exists y(x < y) \to ST(\tau(\phi))(x,z)$$

where the beginning of $\mathbf{p}$ (the current state) is assigned to x and the terminal end of $\mathbf{p}$ is assigned to z.

For that purpose it suffices to note that:

- $ST(\tau(\forall\phi))(x,z)$ is logically equivalent to $\forall w(\forall u(\neg w < u) \to ST(\tau(\phi))(x,w/z))$;
- $ST(\tau(\mathbf{X}\phi))(x,z)$ is logically equivalent to $\forall y((x\mathrm{R}y \wedge y < z) \to ST(\tau(\phi))(y/x,z))$;
- $ST(\tau(\phi\mathbf{U}\psi))(x,z)$ is logically equivalent to $ST(\tau(\psi))(x,z)) \vee \exists y(x < y \wedge y < z \wedge ST(\tau(\psi))(y/x,z) \wedge \forall u((x < u \wedge u < y) \to ST(\tau(\phi))(u/x,z))$.

■

Note that other important versions of temporal logic of programs can be accordingly translated into the temporal logic with reference pointers, e.g. the *anchored version* proposed in [MaP 89].

Furthermore, the temporal logic with reference pointers is in a way more expressive than CTL*. For instance, the fact that no execution path of the program will ever loop can be simply expressed by

$$\downarrow_i \mathbf{G}\neg \uparrow_i,$$

while this fact is not expressible in CTL*, and therefore the language with reference pointers can distinguish unwound structures from general ones.

3 Temporal Logic with Reference Pointers for $\mathcal{ECT}$

3.1 Syntax and Semantics.

We fix a propositional temporal language with (at least) four pairs of reference pointers $\mathcal{L}\updownarrow_t$.

The temporal operators **H** (with dual **P**), **X**, and **U** are defined in terms of **G** and **A** as follows:

$$\mathbf{H}\phi = \downarrow_3 \mathbf{A}(\mathbf{F} \uparrow_3 \to \phi),$$

$$\phi\mathbf{U}\psi = \psi \vee (\phi \wedge \downarrow_4 \mathbf{F}(\psi \wedge \mathbf{H}(\mathbf{P} \uparrow_4 \to \phi))),$$

$$\mathbf{X}\phi = \downarrow_4 \mathbf{G}(\mathbf{H}\mathbf{H}\neg \uparrow_4 \to \phi).$$

where the formulae ϕ, ψ have no free occurrences of $\uparrow_3, \uparrow_4$. Actually, these two pointers will only be used for the purpose of expressing the operators **H**, **X**, and **U**. (Alternatively, a language with only two reference pointers and these operators added to the basic ones can be considered, and axioms corresponding to the above definitions should be included in the deductive system.) We shall, however, retain **X** in the language in order to comply with the notion of a model introduced earlier. The dual operator $\neg\mathbf{X}\neg$ will be denoted by **N**.

Note that every extended computation tree $(S^e, R^e, <, V^e)$ can be regarded as a model for $\mathcal{L}\updownarrow_t$ in terms of the previous section.

3.2 Axiomatic System: $\mathbf{CTL}_{rp}$

We now propose a complete axiomatization of the class $\mathcal{ECT}$ in the language $\mathcal{L}\updownarrow_t$ thus introducing the logic CTL_{rp}.

AXIOMS:

0. A recursive set of axioms for the classical propositional logic.
I. **Axioms for the temporal operators:**
In all axioms and rules below j, k range over all indices for reference pointers.

I.1 The K-axiom for $\mathbf{G}$.

I.2 The S5-axioms for $\mathbf{A}$.

I.3 $\mathbf{A}p \to \mathbf{G}p$.

I.4 $\downarrow_k \mathbf{X}\downarrow_i \mathbf{A}(\uparrow_k \leftrightarrow (\mathbf{F}\uparrow_i \wedge \mathbf{G}\mathbf{G}\neg\uparrow_i))$.

II. **Axioms for the reference pointers:**

II.1 $\downarrow_k\uparrow_k$,

II.2 $\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p) \to (q \to \mathbf{A}(p \to q))$

II.3 $\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p) \to (\downarrow_j \psi \leftrightarrow \psi(p/\uparrow_j))$, for any closed formula $\downarrow_j \psi$

where p, q are propositional variables.

III. **Axioms for the model structures:**

III.1 $\downarrow_k \mathbf{G}\neg\uparrow_k$ (*irreflexivity*);

III.2 $\downarrow_k \mathbf{A}(\mathbf{F}\mathbf{F}\uparrow_k \to \mathbf{F}\uparrow_k)$ (*transitivity*);

III.3 $\mathbf{E}(\downarrow_k \mathbf{A}(\uparrow_k \vee \mathbf{P}\uparrow_k) \wedge \mathbf{F}\top)$ (*there is a root and a non-root*);

III.4 $\downarrow_1 \mathbf{H}\downarrow_2 \mathbf{A}(\mathbf{F}\uparrow_1 \to (\uparrow_2 \vee \mathbf{P}\uparrow_2 \vee \mathbf{F}\uparrow_2))$ (*every state has a linear past*);

III.5 $\mathbf{F}\top \to (\mathbf{F}\mathbf{G}\bot \wedge \mathbf{N}\top \wedge \mathbf{X}\mathbf{F}\top)$ (*every state which is not a terminal end sees one, has an immediate successor, and no immediate successor is a terminal end*);

III.6 $\downarrow_k \mathbf{A}((\mathbf{G}\bot \wedge \mathbf{H}\mathbf{F}\uparrow_k) \to \uparrow_k)$ (*every path has at most one terminal end*).

RULES:

1. *MP*:

$$\frac{\varphi, \varphi \to \psi}{\psi};$$

2. *NEC*$_{\mathbf{A}}$:

$$\frac{\varphi}{\mathbf{A}\varphi};$$

3. *CLSUB*:

$$\frac{\varphi}{clsub(\varphi)},$$

where $clsub(\varphi)$ is a result of uniform substitution of *closed* formulae for propositional variables in φ.

4. *WITNESS*:

$$\frac{\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p) \to \varphi \quad \text{for some propositional variable } p \text{ not occurring in } \varphi}{\varphi},$$

5. *PATH*:

$$\frac{\varphi \to \mathbf{N}^m \mathbf{H}\psi \quad \text{for every} \quad m = 0, 1, 2, \ldots}{\varphi \to \mathbf{F}(\mathbf{G}\bot \wedge \mathbf{H}\psi).}$$

The last two rules deserve some comments.

The rule WITNESS is similar by idea to some quantifier rules in first-order logic (e.g.: if $\vdash A(c)$ for some constant c not occurring in $A(x)$ then $\vdash \forall x A(x)$) and especially to a type of rules discussed in detail in [Ven 93], originating from Gabbay's "irreflexivity rule", see [Ga 81]. In the presence of the substitution rule WITNESS is deductively equivalent to

$$\frac{\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p) \to \varphi \quad \text{for every propositional variable } p}{\varphi}.$$

The effect of WITNESS is this: if a formula φ is not valid, i.e. it is falsified at a point t of a model in which validity is preserved under WITNESS, then there will be a propositional variable p which is a witness of the failure φ by being a "name" for the falsifying state t, i.e. being true at that state only. In temporal setting the variable p says: "it is t o'clock now (when φ is false)". For analogues and further discussion of WITNESS see [PaT 91, GaG 93, Gor 94].

The rule PATH will ensure that in the canonical model every path is of type ω, and it either has a terminal end or can be extended with one.

While the rule WITNESS plays a very important rôle in the derivations in CTL$_{rp}$, the main reason (so far) for the introduction of PATH is the proof of completeness, though, it is used for derivations too, e.g. of the induction scheme $\phi \wedge \mathbf{N}\phi \wedge \mathbf{G}(\phi \to \mathbf{N}\phi) \to \mathbf{F}(\mathbf{G}\bot \wedge \mathbf{H}\phi)$.

Some related open problems are formulated at the end of the paper.

Theorem 3 *The logic CTL$_{rp}$ is sound and complete with respect to the class of extended computation trees $\mathcal{ECT}$.*

Proof:

I. SOUNDNESS.

The validity of the first two groups of axioms (except for I.4) and all rules except for PATH is shown in [Gor 96]. As for I.4 and the third group of axioms, it is a routine task to check that their ST translations are respectively equivalent to the following first-order conditions:

i.4) $\forall y(xRy \leftrightarrow (x < y \wedge \neg\exists z(x < z \wedge z < y)))$;

iii.1) $\neg x < x$;

iii.2) $\forall y \forall z((z < y \wedge y < x) \rightarrow z < x)$;

iii.3) $\exists y(\forall z(y = z \vee y < z) \wedge \exists z(y < z))$;

iii.4) $\forall y \forall z((y < x \wedge z < x) \rightarrow (y = z \vee y < z \vee z < y))$;

iii.5) $\exists y(x < y) \rightarrow (\exists y(x < y \wedge \neg\exists z(y < z)) \wedge \exists y(x < y \wedge \neg\exists z(x < z \wedge z < y)) \wedge \forall y((x < y \wedge \neg\exists z(x < z \wedge z < y)) \rightarrow \exists z(y < z)))$;

iii.6) $\forall y((\neg\exists z(y < z) \wedge \forall z(z < y \rightarrow z < x)) \rightarrow y = x)$;

which are valid in every extended computation tree.

Finally, the rule PATH preserves validity, too. Indeed, if the conclusion of PATH is falsifiable in some ω^+-tree then it is falsified at a state s in a model $\mathcal{M}$ over a finitely branching one, by theorem 1. If all premises are valid there, then there are arbitrarily long finite paths starting from s, along which the formula ψ is true, hence by König's lemma there is an infinite path starting from s along which ψ is true, and therefore there is a terminal end of such a path — a contradiction.

II. COMPLETENESS.

The proof of completeness consists of two major parts: in the first part, given a consistent formula ϕ we construct a model satisfying ϕ, which is almost an extended computation tree. In the second part, we modify this model into a proper extended computation tree in such a way that the resulting model will still satisfy ϕ.

Part II.1. This part closely follows, *mutatis mutandis*, the proof of completeness for the basic temporal logic with reference pointers presented in [Gor 94] (for the logic with one pair of reference pointers) and [Gor 96] (for logics with more pairs of pointers). Nevertheless, it will be outlined here in some detail in order to make the paper more self-contained and to demonstrate that the infinitary rule PATH presents no additional complications.

II.1.i. We first introduce the syntactic notion of *universal forms of* $*$ *in* $\mathcal{L}\updownarrow_t$ (originating from the *admissible forms* in [Gol 82], see also [GaG 93]), recursively as follows:

- $*$ is a universal form of $*$.
- If $u(*)$ is a universal form of $*$, φ is a closed formula in $\mathcal{L}\updownarrow_t$, and $\mathbf{L}$ is a box-modality in $\mathcal{L}\updownarrow_t$ (i.e. $\mathbf{A}$, $\mathbf{G}$, $\mathbf{X}$) then $\varphi \to u(*)$ and $\mathbf{L}u(*)$ are universal forms of $*$ in $\mathcal{L}\updownarrow_t$.

Every universal form of $*$ in $\mathcal{L}\updownarrow_t$ can be represented (up to tautological equivalence) in a uniform way:

$$u(*) = \varphi_0 \to \mathbf{L}_1(\varphi_1 \to \ldots \mathbf{L}_n(\varphi_n \to *) \ldots)$$

where $\mathbf{L}_1, \ldots, \mathbf{L}_n$ are box-modalities in $\mathcal{L}\updownarrow_t$ and some of $\varphi_1, \ldots, \varphi_n$ may be $\top$ if necessary.

For every universal form $u(*)$ and a formula θ we denote by $u(\theta)$ the result of substitution of θ for $*$ in $u(*)$. Obviously, if θ is a closed formula then $u(\theta)$ is a closed formula, too.

II.1.ii. Now we introduce the rules

$$\mathbf{WITNESS}_U : \frac{u(\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p) \to \varphi) \quad \text{for every propositional variable } p}{u(\varphi)},$$

and

$$\mathbf{PATH_U} : \frac{u(\mathbf{N}^m \mathbf{H}\psi) \quad \text{for every } m = 0, 1, 2, \ldots}{u(\mathbf{F}(\mathbf{G}\bot \wedge \mathbf{H}\psi))}$$

where u is an arbitrarily fixed universal form.

Although the new rules $\mathbf{WITNESS}_U$ and $\mathbf{PATH}_U$ seem much stronger than **WITNESS** and **PATH**, in fact they are respectively derivable from the latter in CTL_{rp} (cf. [Gor 96]), and therefore deductively equivalent to them.

II.1.iii. We now introduce and prove some necessary facts about an appropriately strengthened notion of a maximal theory, which will eventually serve as a building block of the canonical model for ϕ.

Definition 1 *A* theory *in $\mathcal{L}\updownarrow_t$ is a set of closed formulae of $\mathcal{L}\updownarrow_t$, which contains all theorems of CTL_{rp} and is closed with respect to* **MP**.

A good theory *is a theory in $\mathcal{L}\updownarrow_t$ which is closed with respect to* $\mathbf{WITNESS}_U$ *and* $\mathbf{PATH}_U$.

Note that for every set of closed formulae Γ there is a minimal good theory $\mathrm{GTh}(\Gamma)$ /resp. a minimal theory $\mathrm{Th}(\Gamma)$/ containing Γ. Indeed, the set of all closed formulae is a good theory. Furthermore, the intersection of every family of good theories is a good theory. Then $\mathrm{GTh}(\Gamma)$ is the intersection of all good theories containing Γ. Likewise for theories.

Definition 2 *A theory (resp. good theory) is* consistent *if it does not contain the falsity $\bot$. A set of closed formulae Δ is* well-consistent *if $GTh(\Delta)$ is consistent.*

Lemma 4 (Deduction theorem for good theories (cf. [Gor 96])) *If* Γ *is a good theory and* φ, ψ *are closed formulae then* $\varphi \to \psi \in \Gamma$ *iff* $\psi \in GTh(\Gamma \cup \{\varphi\})$

Proof: The proof follows the standard lines of an inductive proof of deduction theorem in modal logic, using in addition the fact that $\phi \to u(*)$ is an universal form whenever $u(*)$ is. For more detail, see [Gor 94]). ■

As a corollary to the Deduction theorem, note that for every consistent formula ϕ, the set $\{\phi\}$ is well-consistent.

Definition 3 *A (good) theory* Γ *is* maximal *if for every closed formula* φ, *either* $\varphi \in \Gamma$ *or* $\neg\varphi \in \Gamma$.

Every maximal theory is consistent and cannot be extended to another consistent theory. The most important property of a maximal good theory Γ is that it contains a "witness" $\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow q)$ for some propositional variable q. Indeed, otherwise all $\neg \downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p)$ would be in Γ, and hence, by $\mathbf{WITNESS}_U$, $\bot$ would belong to Γ.

Lemma 5 (Lindenbaum lemma) *Every well-consistent set* Γ_0 *can be extended to a maximal good theory.*

Proof: First, note that $\Gamma = \mathrm{GTh}(\Gamma_0)$ is a consistent good theory. Let $\psi_1, \psi_2, \ldots$ be a list of all closed formulae of $\mathcal{L}\updownarrow_t$ and $u_1, u_2, \ldots$ be a list of all universal forms in $\mathcal{L}\updownarrow_t$. Then we can list all combinations $\{u_i(\psi_j)\}_{i,j=1}^{\infty}$ in a sequence $\theta_1, \theta_2, \ldots$. (obviously, with repetitions, but that does not matter). We define a sequence of consistent good theories $T_0 \subseteq T_1 \subseteq \ldots$ as follows: $T_0 = \Gamma$; suppose that T_n is defined and consider $\mathrm{GTh}(T_n \cup \{\theta_n\})$. If it is consistent, this is T_{n+1}. Otherwise let $\theta_n = u_i(\psi_j)$. Then $\neg u_i(\psi_j) \in T_n$ by the Deduction theorem. Therefore $u_i(\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p) \to \psi_j)$ does not belong to T_n for some propositional variable p and some k (and therefore, due to Ax. II.3, for *all* k). Then we put

$$S_{n+1} = \mathrm{GTh}(T_n \cup \{\neg u_i(\downarrow_k \mathbf{A}(\uparrow_k \leftrightarrow p) \to \psi_j)\}).$$

By the Deduction theorem, S_{n+1} is a consistent good theory.

Now, if ψ_j is not of the form $\mathbf{F}(\mathbf{G}\bot \wedge \mathbf{H}\psi)$ then $T_{n+1} = S_{n+1}$. Otherwise, let $\psi_j = \mathbf{F}(\mathbf{G}\bot \wedge \mathbf{H}\psi)$. Then, by closedness of S_{n+1} under PATH, $u_i(\mathbf{N}^m\mathbf{H}\psi)$ does not belong to S_{n+1} for some $m \in \mathcal{N}$ and we put

$$T_{n+1} = \mathrm{GTh}(S_{n+1} \cup \{\neg u_i(\mathbf{N}^m\mathbf{H}\psi)).$$

Again, by the Deduction theorem, T_{n+1} is a consistent good theory.

Finally, we put $T = \bigcup_{n=0}^{\infty} T_n$. By construction, T is a maximal good theory. ■

Now, for any set of formulae Δ we define

$$\mathbf{G\Delta} = \{\varphi : \mathbf{G}\varphi \in \mathbf{\Delta}\},\ \mathbf{X\Delta} = \{\varphi : \mathbf{X}\varphi \in \mathbf{\Delta}\} \text{ and } \mathbf{A\Delta} = \{\varphi : \mathbf{A}\varphi \in \mathbf{\Delta}\}.$$

Lemma 6 *If Δ is a maximal good theory then* $\mathbf{G\Delta}, \mathbf{X\Delta}$ *and* $\mathbf{A\Delta}$ *are good theories.*

Proof: The proof hinges on the fact that $\mathbf{L}u(*)$ is a universal form whenever $u(*)$ is, where $\mathbf{L} \in \{\mathbf{G}, \mathbf{X}, \mathbf{A}\}$. ■

Lemma 7 *If Δ is a maximal good theory and $\mathbf{F}\theta \in \Delta$ (resp. $\mathbf{N}\theta \in \Delta, \mathbf{E}\theta \in \Delta$) then there is a maximal good theory Δ' such that $\theta \in \Delta'$ and $\mathbf{G}\Delta \subseteq \Delta'$ (resp. $\mathbf{H\Delta} \subseteq \mathbf{\Delta'}, \mathbf{A\Delta} \subseteq \mathbf{\Delta'}$).*

Proof: By Lemma 6 $\mathbf{G\Delta}$ is a good theory. Moreover, $\mathbf{G}\neg\theta \notin \mathbf{\Delta}$ since Δ is consistent. Therefore $\neg\theta \notin \mathbf{G\Delta}$, hence $\mathrm{GTh}(\mathbf{G\Delta} \cup \{\theta\})$ is consistent. Then, by Lemma 5 it can be extended to a maximal good theory Δ'. The other cases are analogous. ■

II.1.iv We are now prepared to construct a "canonical" model for any well-consistent set of formulae, in particular for the consistent formula ϕ.

Definition 4 *A model $\langle S, R, <, V\rangle$ is called* clock-model *if for every $t \in S$ there is a "t o'clock-variable" p_t such that $V(p_t) = \{t\}$.*

Lemma 8 *Every well-consistent set Γ_0 in CTL_{rp} is satisfiable in a clock-model.*

Proof:

The proof follows, *mutatis mutandis*, the standard canonical model construction. First, we extend Γ_0 to a maximal good theory Γ. Then we define a *good canonical model* $\mathcal{M} = \langle S, R, <, V\rangle$ as follows:

- $S = \{\Delta : \Delta$ is a maximal good theory and $\mathbf{A}\Gamma \subseteq \Delta\}$;
- For any $\Delta_1, \Delta_2 \in S$: $R\Delta_1\Delta_2$ if $\mathbf{X}\Delta_1 \subseteq \Delta_2$, and $\Delta_1 < \Delta_2$ if $\mathbf{G}\Delta_1 \subseteq \Delta_2$;
- For any propositional variable $p, V(p) = \{\Delta \in S : p \in \Delta\}$.

$\mathcal{M}$ is a clock model since every maximal good theory contains a "witness". This completes the proof of the lemma. ■

Note that for any Δ_1, Δ_2 from the canonical model $\mathcal{M}$ constructed above, $\mathbf{A}\Delta_1 \subseteq \Delta_2$.

The model $\mathcal{M}$ satisfies the following *truth lemma*, the proof of which essentially repeats the one in [Gor 96] and crucially depends on the fact that $\mathcal{M}$ is a clock-model:

Lemma 9 *For every closed formula θ and $\Delta \in T$,*

$$\mathcal{M} \models \theta[\Delta] \textit{ iff } \theta \in \Delta.$$

In particular, Γ_0 is satisfied at the point Γ of the model $\mathcal{M}$.

It also follows from the truth lemma that $\mathcal{M}$ is a model for all theorems of CTL_{rp}. In particular, all axioms for the structure, hence their corresponding first-order conditions listed in the proof of soundness, are valid in $\mathcal{M}$. Also, the axiom I.4 guarantees that the relations R and $<$ in $\mathcal{M}$ are related accordingly. Indeed, that axiom, being a *pure* formula (with no atomic propositions) , can be easily seen to be "canonical" i.e. to hold in $\mathcal{M}$ and to impose there the condition for R to be the immediate successor relation associated with the partial ordering $<$.

Therefore, $\mathcal{M}$ is a rooted tree where every path is of type ω or ω^+. Indeed, every path will have an initial segment of type ω; suppose there is a state s beyond that initial segment which is not a terminal end and let p be the variable which is true at s and only s. Then $\mathbf{N}^m\mathbf{HF}p$ is true at the root for every natural m, hence, by closedness under PATH, $\mathbf{F}(\mathbf{G}\bot \wedge \mathbf{HF}p)$ is true at the root, so $\mathbf{HF}p$ is true at some terminal end, implying (by axiom III.6 and WITNESS) that s is the terminal end itself — a contradiction.

Thus $\mathcal{M}$ is *almost* an extended computation tree, since not (necessarily) every path in that tree has a terminal end.

It now remains to note that, since $\{\phi\}$ is well-consistent, ϕ is satisfied in a good canonical model $\mathcal{M}$.

Part II.2. Given a good canonical model $\mathcal{M} = \langle S, R, <, V\rangle$ satisfying the formula ϕ we shall modify it into an extended computation tree by adding the missing terminal ends to all maximal paths in such a way that the resulting model will still satisfy ϕ.

II.2.i. Let $p_1, \ldots, p_k$ be the atomic propositions occurring in ϕ. We restrict the temporal language by omitting all other atomic propositions, and let L_ϕ be the corresponding first order language with unary predicates $P_1, \ldots, P_k$. We now regard $\mathcal{M}$ as a model for the restricted languages.

II.2.ii. For every state Γ from $\mathcal{M}$ and a natural number n we define a *modal description of depth n of* Γ inductively on n as follows:

$$d_0(\Gamma) = \downarrow_1 \mathbf{X}\neg \uparrow_1 \wedge \downarrow_1 \mathbf{G}\neg \uparrow_1 \wedge \hat{p}_1, \wedge \ldots \wedge \hat{p}_k,$$

where $\hat{p}_i = p_i$ if $\Gamma \in V(p_i)$, otherwise $\hat{p}_i = \neg p_i$;

$$\begin{aligned} d_{n+1}(\Gamma) = \; & \\ & d_n(\Gamma)\wedge \\ & \textstyle\bigwedge_{\Delta\neq\Gamma} \downarrow_1 \mathbf{E}(\neg \uparrow_1 \wedge d_n(\Delta))\wedge \\ & \textstyle\bigwedge_{\Gamma<\Delta} \mathbf{F}d_n(\Delta) \wedge \bigwedge_{\Gamma R\Delta} \mathbf{N}d_n(\Delta)\wedge \\ & \textstyle\bigwedge_{\Delta<\Gamma} \mathbf{P}d_n(\Delta) \wedge \bigwedge_{\Delta R\Gamma} \downarrow_1 \mathbf{E}(\mathbf{N} \uparrow_1 \wedge d_n(\Delta))\wedge \\ & \textstyle\mathbf{G}\left(\bigvee_{\Gamma<\Delta} d_n(\Delta)\right) \wedge \mathbf{X}(\bigvee_{\Gamma R\Delta} d_n(\Delta))\wedge \\ & \textstyle\mathbf{H}(\bigvee_{\Delta<\Gamma} d_n(\Delta))\wedge \downarrow_1 \mathbf{A}(\mathbf{N} \uparrow_1 \rightarrow \bigvee_{\Delta R\Gamma} d_n(\Delta))\wedge \\ & \textstyle\downarrow_1 \mathbf{A}(\neg \uparrow_1 \rightarrow \bigvee_{\Delta\neq\Gamma} d_n(\Delta)) \end{aligned}$$

Intuitively, $d_n(\Gamma)$ describes the part of the model $\mathcal{M}$ consisting of those states which can be reached from Γ within n steps (forward or backward) along the relations R and $<$.

Note that for every $n \in \mathcal{N}$ there are finitely many different formulae $d_n(\Gamma)$ for $\Gamma \in \mathcal{M}$. For $n = 0$ we denote them by $\delta_1, \ldots, \delta_j$.

Now, for every $n \in \mathcal{N}$ we denote

$$\Phi^n_{\mathcal{M}} = \bigwedge_{\Gamma \in \mathcal{M}} \mathbf{E} d_n(\Gamma) \wedge \mathbf{A} \left(\bigvee_{\Gamma \in \mathcal{M}} d_n(\Gamma) \right)$$

II.2.iii. A standard induction on n shows that $ST(\Phi^n_{\mathcal{M}})$ is equivalent to the formula $\phi^n_{\mathcal{M}}$ as introduced in the proof of Fraïssé's theorem in [EFT 94], p.253, and therefore the following lemma holds (see Th. 3.10, p.255 in [EFT 94]). (By $\equiv_n$ we denote n-equivalence of structures.)

Lemma 10 *For any L_ϕ-model $\mathcal{A}$ and $n \in \mathcal{N}$,*

$$\mathcal{A} \equiv_n \mathcal{M} \quad \textit{iff} \quad \mathcal{A} \Vdash ST(\Phi^n_{\mathcal{M}})$$

II.2.iv. Let $\{\Gamma_i\}_{i \in \mathcal{N}}$ be a maximal path in $\mathcal{M}$ without a terminal end. We add to the model a new state Γ^e in such a way that it is a terminal end for that path. In order to define the truth of $p_1, \ldots, p_k$ at Γ^e we construct chains of sets $D^0_i \supseteq D^1_i \supseteq \ldots$ for $i = 1, \ldots, j$ as follows.

Let, for every $n \in \mathcal{N}$, $\gamma^n_1, \ldots, \gamma^n_{j_n}$ be all formulae $d_n(\Gamma_i)$, $i \in \mathcal{N}$ and $\psi_n = (\gamma^n_1 \vee \ldots \vee \gamma^n_{j_n})$. Then $\mathbf{H}\psi_n$ is true at every $\Gamma_i, i \in I$, hence $\mathbf{N}^m \mathbf{H}\psi_n$ are true at every Γ_i for all $m \in \mathcal{N}$. Let $q \in \mathcal{N}$ be large enough so that for every $r \geq q$, $\theta_n = \mathbf{P}\gamma^n_1 \wedge \ldots \wedge \mathbf{P}\gamma^n_{j_n} \in \Gamma_r$. By the truth lemma and the rule PATH, $\mathbf{F}(\mathbf{G}\bot \wedge \mathbf{H}\psi_n)$ is true at Γ_r, hence there is a terminal end Δ such that $\Gamma_r < \Delta$ and $\mathbf{H}\psi_n \wedge \mathbf{P}\theta_n$, hence $\mathbf{H}\psi_n \wedge \theta_n$ is true at Δ. Since Δ is not a terminal end for $\{\Gamma_i\}_{i \in \mathcal{N}}$, for large enough $r_1 > r$, $\Gamma_{r_1} \not< \Delta$. Repeating the same argument for Γ_{r_1} we find a terminal end Δ_1 such that $\Gamma_{r_1} < \Delta_1$ and $\mathbf{H}\psi_n \wedge \theta_n$ is true at Δ_1, etc.

Now, for every $i = 1, \ldots, j$ we define D^n_i to be the set of those (infinitely many) terminal ends Δ in $\mathcal{M}$ such that $\delta_i \wedge \mathbf{H}\psi_n \wedge \theta_n$ is true at Δ.

Note that for every n:

i) at least one D^n_i is infinite, and

ii) $D^{n+1}_i \subseteq D^n_i$ because ψ_{n+1} implies ψ_n and θ_{n+1} implies θ_n since (by an easy induction on n) for any Δ, $d_{n+1}(\Delta)$ implies $d_n(\Delta)$.

Therefore, there is an index i such that D^n_i is infinite for every $n \in \mathcal{N}$.

We then extend the valuation V of the atomic propositions $p_1, \ldots, p_k$ to Γ^e according to δ_i.

II.2.v. Let $\mathcal{M}^e = \langle S^e, R, <^e, V^e \rangle$ be the extension of the model $\mathcal{M}$ by adding terminal ends, as described above, to all maximal paths which do not have them, i.e.:

- S^e extends S by adding all newly constructed terminal ends;
- $<^e$ extends $<$ with $\Gamma_i <^e \Gamma^e$ for each maximal path $\{\Gamma_i\}_{i\in I}$ with a newly constructed terminal end Γ^e;
- V^e extends V as described above.

Note that for every $n \in \mathcal{N}$ and $\Gamma \in \mathcal{M}$, $d_n(\Gamma)$ in $\mathcal{M}^e$ is the same as $d_n(\Gamma)$ in $\mathcal{M}$ (straightforward induction on n) and $d_n(\Gamma^e) = d_n(\Delta)$ for every $\Delta \in D_i^n(\Gamma^e)$. Therefore $\Phi^n_{\mathcal{M}} = \Phi^n_{\mathcal{M}^e}$.

II.2.vi. We now prove that $\mathcal{M}^e$ is elementarily equivalent in L_ϕ to $\mathcal{M}$, and therefore $\mathcal{M}^e \Vdash \exists x ST(\phi)$. By lemma 10 it is sufficient to show that $\mathcal{M}^e \Vdash ST(\Phi^n_{\mathcal{M}})$, or equivalently, $\mathcal{M}^e \models \Phi^n_{\mathcal{M}}$, which follows from the fact that $\Phi^n_{\mathcal{M}} = \Phi^n_{\mathcal{M}^e}$.

This completes the proof of the main theorem. ■

Theorem 11 *The logic CTL_{rp} is decidable.*

Proof: This follows from the more general result in [GuS 85] about decidability of the monadic second order theory of trees with path quantifiers. ■

We can now extend Th 2 as follows.

Theorem 12 *For any state formula ϕ in the language of CTL^* the following are equivalent:*
(i) ϕ is CTL^*-valid.
(ii) ϕ is valid in the class of finitely branching computation trees.
(iii) $(F\top \to \tau(\phi))$ is valid in the class $\mathcal{ECT}$.
(iv) $(F\top \to \tau(\phi))$ is valid in the class $\mathcal{FECT}$.
(v) $(F\top \to \tau(\phi))$ is a theorem of CTL_{rp}.

4 Some concluding remarks and open problems.

As mentioned in the introduction, the logic CTL_{rp} can be considered as an alternative to the μ-calculus strongly expressive logical system in which CTL^* can be embedded, and for which an explicit complete axiomatization is provided. The μ-calculus is well-studied and known to have many virtues which make it a very interesting and attractive logical system: elegant axiomatization, decidability, well-developed model-checking systems etc. On the other hand, the idea and technicalities of reference pointers on which the temporal logic proposed here is based are still little known. It is therefore difficult to offer an objective comparison at this stage, yet a few remarks can be made. The two languages, although both quite expressive, are formally incomparable in their expressiveness (which is not yet known precisely for either language) and quite different in style. While the semantics of μ-calculus is mathematically quite clear, it is a rather non-trivial task to determine explicitly the semantic meaning of a formula from that language. In that respect, the language of CTL_{rp} seems easier to use as it comes closer to the style of first-order logic. That feature, however, comes as a trade-off for the elegance and succinctness of the expression. As for the axiomatization, the one proposed here being a typical Hilbert-style deductive system is of a little practical use and the logics with reference pointers are awaiting the development of efficient proof systems, although a significant step forward is made in [Bl 99]. Until that time the usefulness of the CTL_{rp} will admittedly remain mainly theoretical. Finally, unlike μ-calculus, the complexity of the latter logic has not been studied yet, though some related results for logics with reference pointers are known (see [BT 99]).

Now, some more questions and directions for further research.

Due to the decidability of CTL_{rp}, the infinitary rule PATH can be replaced by a recursive set of axioms. It is an open question if it can be eliminated at the expense of adding finitely many new axioms.

An important question is whether the axiomatic system for CTL_{rp} and its completeness proof can be "translated backwards" into CTL^* and thus provide a solution of the long-standing problem for an explicit axiomatization of that logic.

The expressiveness of the TL_{rp} still awaits precise characterization, though some related results are included in [Gor 96] and [BT 99]. A related question is if there is a natural notion of bisimulation which would correspond to TL_{rp}-equivalence of models.

Finally, some topics for further work, related to the practical utilization of temporal logics with reference pointers for specification, analysis and verification of programs are:

- investigation of practically important properties of computation trees which are expressible in these logics but not in CTL^*.
- construction of efficient deductive systems for CTL_{rp}, in particular semantic tableaux, and development of efficient decision procedures and systems for automated deduction for that logic.
- development and implementation of real systems for specification and verification of programs based on temporal logics with reference pointers.

5 Acknowledgements

This work was supported by research grant GUN 2034353 of the Foundation for Research Development of South Africa. The paper has benefited from the referees' suggestions and critical comments.

References

[BRV95] Backofen R., J. Rogers, & K. Vijay-Shanker, A First-order Axiomatization of the Theory of Finite Trees, *Journal of Logic, Language and Information*, **4**, 1995, 5-39.

[BMP 81] Ben-Ari M., Z. Manna, & A. Pnueli, The Temporal Logic of Branching Time, *Proc. 8th ACM Symp. on Princ. of Prog. Lang.* Wiliiamsburg, VA, 1981, 164-176; also in: *Acta Informatica*, 20(3), 1983, 207-226.

[Ben 91] van Benthem J.F.A.K., *The Logic of Time*, Reidel, Dordrecht, 2nd ed., 1991.

[BS 95] Blackburn P. & J. Seligman, Hybrid Languages, *Journal of Logic, Language and Information*, **4**, 1995, 251-272.

[BT 98] Blackburn, Patrick and Miroslava Tzakova. Hybrid Completeness, Logic Journal of the IJPL, 4 (1998), 625-650.

[BT 99] Blackburn, Patrick and Miroslava Tzakova. Hybrid Languages and Temporal logic, J. of Logic, Language and Information, to appear.

[Bl 99] Blackburn, Patrick. Internalizing Labelled Deduction, to appear.

[Bur 84] Burgess J., Basic Tense Logic, in: *Handbook of Philosophical Logic*, D. Gabbay and F. Guenthner (eds.), Reidel, Dordrecht, vol.II, 1984, 89-133.

[CE 81] Clarke E.M. & E.A. Emerson, Design and Synthesis of Synchronization Skeletons using Branching Time Temporal Logic, in: *Proc. Workshop on Logics of Programs*, LNCS 131, Springer-Verlag, 1981, 52-71.

[Dam 94] Dam M., CTL* and ECTL* as Fragments of the modal mu-calculus, *Theoretical Computer Science*, 126 (1994), pp77-96.

[EFT 94] Ebbinghaus H.-D., J. Flum, W. Thomas, *Mathematical Logic*, Springer-Verlag, 2nd ed., 1994.

[Eme 90] Emerson, E.A., *Temporal and Modal Logic*, in: *Handbook of Theoretical Computer Science*, vol. B, J. van Leeuwen (ed.), Elsevier, 1990, pp. 995-1072.

[EmH 83] Emerson, E.A., & J.Y. Halpern, "Sometimes" and "Not Never" revisited: on Branching versus Linear Time Temporal Logic, in:*Proc. of the 10th Annual Symp. on Principles of Programming Languages*, 1983, 127-140; also in *Journal of the ACM*, **33**(1), 151-178.

[EmS 84] Emerson, E.A., & A.P. Sistla, Deciding full Branching Time Logic, Information and Control, 61(3), 1984, 175-201.

[ES 89] Emerson E.A. & J. Srinivasan, Branching Time Temporal Logic, in: *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, J de Bakker, W.-P. de Roever, G. Rosenberg (eds.), LNCS 354, Springer-Verlag, 1989, 123-172.

[Ga 81] Gabbay, D., An Irreflexivity Lemma with Applications to Axiomatizations of Conditions on Tense Frames. in: *Aspects of Philosophical Logic*, U. Monnich (ed.). Reidel, Dordrecht, 1981, 67-89.

[GaG 93] Gargov G. & V. Goranko, Modal Logic with Names, *Journal of Philosophical Logic*, **22**(6), 1993, 607-636.

[Gol 82] Goldblatt R.I., *Axiomatizing the Logic of Computer Programming*, Springer LNCS **130**, 1982.

[Gol 87] Goldblatt R.I., *Logic of Time and Computation*, CSLI Lecture Notes, No. 7., 1987.

[Gor 94] Goranko V., *Temporal Logic with Reference pointers*, in: *Temporal Logic*, D. Gabbay, H.-J. Ohlbach (eds.), Lecture Notes in Artificial Intelligence 827, Springer-Verlag, 1994, pp. 133-148.

[Gor 96] Goranko, V., Hierarchies of Modal and Temporal Logics with Reference Pointers, *Journal of Logic, Language and Information*, **5**, 1996, 1-24.

[Gor 99] Goranko, V., Trees and Finite Branching, 1999, submitted.

[GuS 85] Gurevich Y. & S. Shelah, The Decision Problem for Branching Time Logic, *Journal of Symbolic Logic*, **50**(3), 1985, 668-681.

[MaP 89] Manna Z., & A. Pnueli, The Anchored Version of the Temporal Framework, in: *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, J. de Bakker, W.-P. de Roever, G. Rosenberg (eds.), LNCS 354, Springer-Verlag, 1989, 201-284.

[PaT 91] Passy S. & T. Tinchev, An Essay in Combinatory Dynamic Logic, *Information and Computation*, **93**(2), 1991, 263-332.

[Pen 95] Penczek W., Branching Time and Partial Order in Temporal Logics, in: *Time and Logic: a Computational Approach*, Univ. College of London, 1995, 179-228.

[Pnu 77] Pnueli A., The Temporal Logic of Programs, in: *Proc. 18th Ann. IEEE Symp. on Foundations of Computer Science* (1977), 46-57.

[Pri 67] Prior A., *Past, Present, and Future,* Clarendon Press, Oxford, 1967.

[Sti 92] Stirling C., Modal and Temporal Logics, in: *Handbook of Logic in Computer Science, vol. 2: Computational Structures*, S. Abramski, D. Gabbay, T. Maibaum (eds.), Clarendon Press, Oxford, 1992, 478-563.

[Tho 88] Thomas W., Computation Tree Logic and Regular ω-languages, in: *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, J de Bakker, W.-P. de Roever, G. Rosenberg (eds.), LNCS 354, Springer-Verlag, 1989, 690-713.

[Ven 93] Venema Y., Derivation Rules as Anti-axioms in Modal Logic, *Journal of Symbolic Logic,* **58**(3), 1993, 1003-1034.

[Wol 95] Wolper P., On the Relation of Programs and Computations to Models of Temporal Logic, in: *Time and Logic: a Computational Approach*, Univ. College of London, 1995, 131-178.

[ZaC 93] Zanardo A., J. Carmo, Ockhamist Computational Logic: Past-Sensitive Necessitation in CTL*, *J. of Logic and Computation*, **3** (3), 1993, 249-268.

[Zan 96] Zanardo A., Branching-time Logic with Quantification Over Branches: The Point of View of Modal Logic, *J. of Symbolic Logic*, **61** (1), 1996, 1-39.