Theoretical Computer Science 126 (1994) 77–96
Elsevier

CTL* and ECTL* as fragments of the modal μ-calculus*

Mads Dam

Swedish Institute of Computer Science, Box 1263, S-164 38 Kista, Sweden

Abstract

Dam, M., CTL* and ECTL* as fragments of the modal μ-calculus, Theoretical Computer Science 126 (1994) 77–96.

Direct embeddings of the full branching-time CTL* and its extension ECTL* into the modal μ-calculus are presented. The embeddings use tableaux as intermediate representations of formulas, and use extremal fixed points to characterise those paths through tableaux that satisfy an admissibility criterion, guaranteeing eventualities to be eventually satisfied. The version of ECTL* considered replaces the entire linear-time fragment of CTL* by Büchi automata on infinite strings. As a consequence the embedding of ECTL* turns out to be computable in linear time, while the embedding of CTL* is doubly exponential in the worst case.

1. Introduction

Due to its inherent combinatorial difficulties, concurrency is an area of computer science where formal and automated verification methods have proved themselves particularly valuable, for instance, in detecting errors difficult or impossible to find by informal reasoning alone (cf. [17, 7, 3]). For programming errors to be exposed by exhibiting mismatch to formal properties, those properties must correctly reflect the intent of the programmer. The correctness of this representation may be obvious for a few very simple theories, but unfortunately the need for greater expressive power often seems to call for a corresponding sacrifice in transparency.

A case in point is the modal μ-calculus L_μ [19]. This logic is obtained as an enrichment of a simple modal base logic, Hennessy–Milner logic [18], by least and greatest fixed points of formally monotone operators. The result is a very general branching-time temporal logic capable of expressing a wealth of properties related to,

Correspondence to: M. Dam, Swedish Institute of Computer Science, Box 1263, S-164 38 Kista, Sweden.

*This work was supported by SERC Grant GR/F 32219.

for instance, partial and total correctness, liveness, safety, and fairness, that are of crucial importance in practical program verification (c.f. [2,31]). Indeed L_μ encompasses a great many well-known program logics such as PDL [15], PDL-Δ [26], linear-time temporal logic PTL [16], CTL [5] and CTL* [11]. This can be shown by providing constructive translations (c.f. [19, 13, 32]).

In fact, the containment in all these cases is strict. A typical example of a property expressible in L_μ but not, for instance, in CTL* is a cyclic property such as "along any path, at all even moments ϕ holds, and at all odd moments ϕ may hold or not" (cf. [33]). Properties such as these are necessary in general for modular reasoning [20]. Despite this additional expressive power, L_μ is decidable in deterministic exponential time, and thus essentially is not harder than PDL [12, 15]. Moreover, for closed formulas, L_μ preserves the characterisation of bisimulation equivalence [18] and thus provides a natural temporal logic for process calculi such as CCS [24]. A model checker for checking L_μ-properties against finite-state (CCS) processes due to Stirling and Walker [25] has been implemented in the Edinburgh Concurrency Workbench [9] and used in several case studies such as mutual exclusion algorithms [31] and communication protocols [4].

Impeding the widespread practical use of L_μ, however, is its lack of transparency. Already at the second level of alternation, formulas can become highly unintelligible and alternation is indeed needed to express, for instance, fairness properties. Consequently, it is important to develop tools to aid users in manipulating, generating and understanding L_μ-formulas. Constructive translations such as those referred to above can be very useful for these purposes. They can be machine implemented to provide syntactical sugaring of L_μ-properties. Moreover, they can help also in understanding L_μ itself, provided they, and the results they produce, are sufficiently simple and intuitive. They mostly succeed very well in this: A first step towards a working understanding of L_μ is certainly to understand why the CTL-formula E*F*X ("along some path X holds eventually") is translated into the least fixed point formula $\mu Y.X \vee \Diamond Y$, where $\Diamond$ is the existential next-state quantifier.

In this respect CTL* is of particular interest. Beyond CTL, CTL* is capable of expressing properties such as EGFX ("along some path X holds infinitely often"), useful for dealing with fairness [11], and in general arbitrary nestings and boolean combinations of linear- and branching-time connectives for which the task of finding equivalent L_μ-formulations may present considerable difficulties. The previously only known translation of CTL* into L_μ is, however, rather indirect and not very transparent. It is obtained by composing Wolper's unpublished translation of CTL* into PDL-Δ [32] with the translation of PDL-Δ into L_μ (cf. [13]). It involves 5 stages: The first and second stages builds a tableau and derives from it a deterministic Muller automaton (cf. [14]). Third stage derives from this automaton an equivalent ω-regular expression (cf. [21]). As the fourth stage, a PDL-Δ formula is obtained, and finally this formula is translated into L_μ.

In this paper we present a relatively simple and much more direct algorithm for translating CTL* into L_μ. The idea is to represent a tableau directly as an equivalent L_μ formula. The notion of tableau used is fairly standard and closely related to those of e.g. Ben-Ari et al. [1] and Wolper [33]. Their role is to decompose formulas according to their structure, and to detect recursion in the natural way of terminating when a tableau node is repeated. The problem is to use the connectives of L_μ as an external means of characterising tableaux, and in particular to use least and greatest fixed points to classify loops. The solution involves an analysis of the admissible ways of "regenerating" nodes, using the terminology of Streett and Emerson [27].

An alternative way of allowing cyclic properties to be expressed is to include in CTL* not only the linear-time modalities of PTL, but also more generally all linear-time modalities expressible in Wolper's extended temporal logic, ETL [33]. ETL extends PTL by allowing arbitrary finite automata on infinite words as temporal operators. In this way the full power of the ω-regular languages is obtained [34], whereas PTL is capable of describing only the star-free ω-regular languages [16, 28]. Several expressively equivalent versions of extended CTL*, ECTL* have been proposed [30, 6, 29]. Here we follow the approach of Thomas, adding linear-time operators corresponding to Büchi automata on infinite words.

Also for ECTL* the translation involves the building of tableaux and the use of fixed points to classify loops. The construction turns out to be simpler, however, than for CTL* for two reasons: First there is no need to consider the explicit nesting of linear-time connectives present in CTL*, and secondly the use of Büchi automata allows attention to be restricted to automata with a single accepting state.

The remainder of the paper is organised as follows. In Sections 2 and 3, L_μ and CTL* are introduced, Sections 4–6 describe the translation from CTL*, and in Section 7 it is proved correct. In Sections 8 and 9 ECTL* is introduced and its translation described, and finally in Section 10 we discuss issues such as efficiency and related work.

2. The modal μ-calculus

Formulas ϕ, ψ, γ of L_μ are built from propositional variables X, Y, boolean connectives $\neg$ and $\wedge$, the modal next-state quantifier $\Diamond$, and the least fixed-point operator $\mu X.\phi$. The latter is subject to the formal monotonicity condition that all free occurrences of X in ϕ lie in the scope of an even number of negations. Other connectives are derived in the usual way and, in particular, $\Box\phi \triangleq \neg\Diamond\neg\phi$, $\nu X.\phi \triangleq \neg\mu X.\neg\phi[\neg X/X]$. We use σ as a metavariable ranging over $\{\mu, \nu\}$. Usually, indexed modalities $\langle a\rangle$ are considered instead of the unindexed $\Diamond$. For the purpose of embedding CTL* and ECTL*, however, one program letter suffices.

For the semantics fix a transition system $T=(S, R)$, where S is a set of states ranged over by s and R a binary transition relation on S. The semantics of the formula ϕ relative to T and a valuation $\mathscr{V}: X \mapsto B \subseteq S$ is the set $\|\phi\|\mathscr{V} \subseteq S$ defined as follows:

$$\|X\|\mathcal{V}=\mathcal{V}(X),$$

$$\|\neg\phi\|\mathcal{V}=S-\|\phi\|\mathcal{V},$$

$$\|\phi\wedge\psi\|\mathcal{V}=\|\phi\|\mathcal{V}\cap\|\psi\|\mathcal{V},$$

$$\|\Diamond\phi\|\mathcal{V}=\{s\in S\mid\exists s'\in S.\ sRs' \text{ and } s'\in\|\phi\|\mathcal{V}\},$$

$$\|\mu X.\phi\|\mathcal{V}=\bigcap\{B\subseteq S\mid \|\phi\|\mathcal{V}[X\mapsto B]\subseteq B\}.$$

Here $\mathcal{V}[X\mapsto B]$ is the usual update $\mathcal{V}'$ of $\mathcal{V}$ which agrees with $\mathcal{V}$ except that $\mathcal{V}'(X)=B$. The expected clauses are obtained for the derived connectives. In particular, for greatest fixed points,

$$\|\nu X.\phi\|\mathcal{V}=\bigcup\{B\subseteq S\mid B\subseteq \|\phi\|\mathcal{V}[X\mapsto B]\}.$$

Intuitively, least fixed points are used for eventualities and greatest fixed points for invariant properties. This intuition is brought out by the following characterisation of the relation of satisfaction $s\in\|\phi\|\mathcal{V}$ due to Streett and Emerson [27]. Closely related characterisations are due to Stirling and Walker [25], Bradfield and Stirling [2] and Cleaveland [8].

Relative to a transition system T, a *choice relation* is a minimal relation $\Rightarrow$ on *sequents* $s\vdash\phi$ such that

$$s\vdash\neg\neg\phi \;\Rightarrow\; s\vdash\phi,$$

$$s\vdash\phi_1\wedge\phi_2 \;\Rightarrow\; s\vdash\phi_i \quad \text{for } i=1 \text{ and } i=2,$$

$$s\vdash\phi_1\vee\phi_2 \;\Rightarrow\; s\vdash\phi_i \quad \text{for } i=1 \text{ or } i=2,$$

$$s\vdash\Box\phi \;\Rightarrow\; s'\vdash\phi \quad \text{whenever } sRs',$$

$$s\vdash\Diamond\phi \;\Rightarrow\; s'\vdash\phi \quad \text{for some } s' \text{ such that } sRs',$$

$$s\vdash\sigma X.\phi \;\Rightarrow\; s\vdash\phi[\sigma X.\phi/X].$$

Rooting $\Rightarrow$ at $s\vdash\phi$ restricts $\Rightarrow$ to $\{s'\vdash\phi'\mid s\vdash\phi\Rightarrow^* s'\vdash\phi'\}$. Let $\Rightarrow$ be rooted at $s\vdash\phi$. Then $\Rightarrow$ *agrees with* $\mathcal{V}$, if whenever $s\vdash\phi\Rightarrow^* s'\vdash(\neg)X$ then $s'\in\mathcal{V}(X)$ ($s'\notin\mathcal{V}(X)$). Regarding fixed points the crucial issue is to avoid having to regenerate μ-formulas infinitely often. A σ-formula $\sigma X.\phi$ is *regenerated from* s_1 *to* s_n if there is a derivation

$$s_1\vdash\phi_1 \;\Rightarrow\; s_2\vdash\phi_2 \;\Rightarrow\; \cdots \;\Rightarrow\; s_n\vdash\phi_n,$$

such that $n>1$, $\phi_1=\phi_n=\sigma X.\phi$, and $\sigma X.\phi$ is a subformula of ϕ_i for each i, $1\leqslant i\leqslant n$. Then $\Rightarrow$ is *well-founded* if the regeneration relation is well-founded for each μ-formula. That is, there is no infinite path $sR^*s_0R^*s_1R^*s_2\ldots$ and no μ-formula $\mu X.\phi$ which is regenerated from s_i to s_{i+1} for all $i\in\omega$.

Theorem 2.1 (Streett and Emerson [27]). *There is a well-founded choice relation* $\Rightarrow$ *from* $s\vdash\phi$ *which agrees with* $\mathcal{V}$ *iff* $s\in\|\phi\|\mathcal{V}$.

3. Computation tree logic, CTL*

Formulas of CTL* are built from propositional variables using boolean connectives $\neg$ and $\wedge$, the linear next-time and until-operators O and U, and the existential path-quantifier E. The dual of E is the universal path-quantifier defined by $\mathrm{A}\phi \triangleq \neg \mathrm{E} \neg \phi$. Other connectives are derived as usual. In particular, $\mathrm{F}\phi \triangleq \mathrm{true}\ \mathrm{U}\phi$ and $\mathrm{G}\phi \triangleq \neg \mathrm{F} \neg \phi$. Also we let $\phi_1 \neg \mathrm{U} \phi_2 \triangleq \neg(\phi_1 \mathrm{U} \phi_2)$. An *eventuality* is a formula of the form either $\phi \mathrm{U} \psi$ or $\mathrm{O}(\phi \mathrm{U} \psi)$ (or $\mathrm{F}\phi$ or $\mathrm{OF}\phi$ if F is taken as primitive).

Formulas denote properties of infinite paths through a transition system $T=(S, R)$ with valuation $\mathscr{V}$. Such a path is a mapping $\rho \in S^{\omega}$ such that for all $i \in \omega$, $\rho(i) R \rho(i+1)$. The ith suffix of ρ is the path ρ^i given by $\rho^i(j)=\rho(i+j)$ for all $j \in \omega$. The relation R is *total* if for all $s \in S$ there is some $s' \in S$ such that sRs'. The assumption of totality allows attention to be restricted to infinite paths. The semantics of formulas is given as follows (cf. [11]):

$$\rho \models_{\mathscr{V}} X \text{ iff } \rho(i) \in \mathscr{V}(X),$$

$$\rho \models_{\mathscr{V}} \neg \phi \text{ iff } \rho \not\models_{\mathscr{V}} \phi,$$

$$\rho \models_{\mathscr{V}} \phi_1 \wedge \phi_2 \text{ iff } \rho \models_{\mathscr{V}} \phi_1 \text{ and } \rho \models_{\mathscr{V}} \phi_2,$$

$$\rho \models_{\mathscr{V}} \mathrm{O}\phi \text{ iff } \rho^1 \models_{\mathscr{V}} \phi,$$

$$\rho \models_{\mathscr{V}} \phi_1 \mathrm{U} \phi_2 \text{ iff } \exists i \in \omega \text{ such that } \rho^i \models_{\mathscr{V}} \phi_2 \text{ and } \forall j, \text{ if } 0 \leqslant j < i \text{ then } \rho^j \models_{\mathscr{V}} \phi_1,$$

$$\rho \models_{\mathscr{V}} \mathrm{E}\phi \text{ iff } \exists \rho' \text{ such that } \rho(0)=\rho'(0) \text{ and } \rho' \models_{\mathscr{V}} \phi.$$

Let $s \models_{\mathscr{V}} \phi$ iff for all paths ρ such that $\rho(0)=s$, $\rho \models_{\mathscr{V}} \phi$. We are particularly interested in formulas that depend only on the current state in the sense that $\rho \models_{\mathscr{V}} \phi$ iff $\rho(0) \models_{\mathscr{V}} \phi$. This is true, in particular, for boolean combinations of formulas that are either propositional variables or contain an outermost occurrence of the existential path-quantifier. Formulas of this form are called *state formulas*.

4. Tableaux

The aim is to translate state formulas into semantically equivalent L_μ formulas. Substantial parts of this translation can be performed structurally (cf. [32]). Variables can be translated into themselves, and boolean combinations of state formulas can be translated into boolean combinations of their translations. Furthermore, substitutions of state formulas for variables can be preserved. More formally, let $\phi \rightsquigarrow \psi$ mean that ϕ is a state formula and ϕ can be translated into the L_μ-formula ψ. Moreover, assume that $\rightsquigarrow$ is correct in the sense that if $\phi \rightsquigarrow \psi$ then ϕ and ψ are semantically equivalent: for all transition systems T, for all $s \in S_T$ and all valuations $\mathscr{V}$ on T, $s \models_{\mathscr{V}} \phi$

iff $s \in \| \psi \| \mathscr{V}$. Then the following four rules are validated:

$$X \rightsquigarrow X \qquad \frac{\phi \rightsquigarrow \psi}{\neg \phi \rightsquigarrow \neg \psi} \qquad \frac{\phi_1 \rightsquigarrow \psi_1 \quad \phi_2 \rightsquigarrow \psi_2}{\phi_1 \wedge \phi_2 \rightsquigarrow \psi_1 \wedge \psi_2}$$

$$\frac{\phi_1 \rightsquigarrow \psi_1 \quad \cdots \quad \phi_n \rightsquigarrow \psi_n \quad \phi \rightsquigarrow \psi}{\phi[\phi_1/X_1, \ldots, \phi_n/X_n] \rightsquigarrow \psi[\psi_1/X_1, \ldots, \psi_n/X_n]}$$

where $\gamma[\gamma_1/X_1, \ldots, \gamma_n/X_n]$ denotes the simultaneous substitution of the γ_i for X_i in γ. By means of these rules, attention can be restricted to formulas of the form $\mathrm{E}\phi$ where ϕ is a linear-time formula, i.e. ϕ does not contain occurrences of the existential path-quantifier E. Call such formulas *basic*. For formulas of this form the translation uses tableaux as an intermediate representation. A tableau is a rooted, finite-directed graph which is generated by applying exactly one of a small set of rules to each node that is not a leaf. Leaves are propositional variables or their negations. Other nodes are formulas of the form $\mathrm{E}\Phi \triangleq \mathrm{E} \wedge \Phi$ where Φ is a finite set of path-quantifier free formulas. For simplicity of notation, we generally use a sequential notation for nodes, writing, for instance, $\mathrm{E}(\Phi, \phi)$ in place of $\mathrm{E}(\Phi \cup \{\phi\})$. The purpose of the tableaux rules is to analyse the state-related structure of formulas by means of the basic L_μ connectives $\wedge$, $\vee$ and $\diamond$. Rules consequently have the form

$$\Omega\colon \frac{\phi}{\phi_1 \ldots \phi_n}$$

where Ω is either $\wedge$, $\vee$, $\diamond$, or possibly I, and a rule instance of this form is forwards- and backwards-sound if $s \models_{\mathscr{V}} \phi$ iff $s \models_{\mathscr{V}} \Omega(\phi_1, \ldots, \phi_n)$ where $\diamond$ is interpreted as EO and I as the identity operator. The tableau rules are the following:

$$I\colon \frac{\mathrm{E}(\Phi, \neg\neg\phi)}{\mathrm{E}(\Phi, \phi)} \qquad \wedge\colon \frac{\mathrm{E}(\Phi, X)}{\mathrm{E}\Phi \quad X} \qquad \wedge\colon \frac{\mathrm{E}(\Phi, \neg X)}{\mathrm{E}\Phi \quad \neg X}$$

$$I\colon \frac{\mathrm{E}(\Phi, \phi \wedge \psi)}{\mathrm{E}(\Phi, \phi, \psi)} \qquad \vee\colon \frac{\mathrm{E}(\Phi, \phi \vee \psi)}{\mathrm{E}(\Phi, \phi) \quad \mathrm{E}(\Phi, \psi)}$$

$$\vee\colon \frac{\mathrm{E}(\Phi, \phi_1 \mathrm{U} \phi_2)}{\mathrm{E}(\Phi, \phi_2) \quad \mathrm{E}(\Phi, \phi_1, \mathrm{O}(\phi_1 \mathrm{U} \phi_2))}$$

$$\vee\colon \frac{\mathrm{E}(\Phi, \phi_1 \neg \mathrm{U} \phi_2)}{\mathrm{E}(\Phi, \neg\phi_1, \neg\phi_2) \quad \mathrm{E}(\Phi, \neg\phi_2, \mathrm{O}(\phi_1 \neg \mathrm{U} \phi_2))}$$

$$\diamond\colon \frac{\mathrm{E}(\mathrm{O}\phi_1, \ldots, \mathrm{O}\phi_n)}{\mathrm{E}(\phi_1, \ldots, \phi_n)}$$

For the operators F and G the following rules can be used instead of those derived from the "until"-based ones above:

$$I\colon \frac{\mathrm{E}(\Phi, \mathrm{G}\phi)}{\mathrm{E}(\Phi, \phi, \mathrm{OG}\phi)} \qquad \vee\colon \frac{\mathrm{E}(\Phi, F\phi)}{\mathrm{E}(\Phi, \phi) \quad \mathrm{E}(\Phi, \mathrm{OF}\phi)}$$

It is not difficult to see that tableaux are finite, and that all tableau rules are forwards- and backwards-sound.

5. Admissible paths

A key task is to isolate those infinite paths through a given tableau τ that are admissible in the sense, intuitively, that eventualities are eventually also satisfied. To get at this notion of eventual satisfaction, and thus of admissibility, we analyse the way tableau rules decompose individual linear-time formulas. Let $\mathrm{E}\Phi_1 \rightarrow \mathrm{E}\Phi_2$ if there is an edge from $\mathrm{E}\Phi_1$ to $\mathrm{E}\Phi_2$ in τ. Each member of Φ_2 is determined, or *generated*, by at least one member of Φ_1. Consider, for instance, the transition $\mathrm{E}(\mathrm{O}(\phi \mathrm{U} \psi), \phi \mathrm{U} \psi) \rightarrow \mathrm{E}(\mathrm{O}(\phi \mathrm{U} \psi), \psi)$. Relative to this transition, $\mathrm{O}(\phi \mathrm{U} \psi)$ is generated by $\mathrm{O}(\phi \mathrm{U} \psi)$ and similarly ψ is generated by $\phi \mathrm{U} \psi$. On the other hand, it is *not* relative to this transition the case that $\mathrm{O}(\phi \mathrm{U} \psi)$ is generated by $\phi \mathrm{U} \psi$. It is the purpose of the relation $\rightarrowtriangle$ to formalise this notion of generation. Thus, each transition $\mathrm{E}\Phi_1 \rightarrow \mathrm{E}\Phi_2$ determines the *generation relation* $\rightarrowtriangle_{\mathrm{E}\Phi_1 \rightarrow \mathrm{E}\Phi_2} \subseteq \Phi_1 \times \Phi_2$ in the following way where $\Delta(\Phi) = \{(\phi, \phi) \mid \phi \in \Phi\}$ is the diagonal relation on Φ:

$$\rightarrowtriangle_{\mathrm{E}(\Phi, \neg\neg\phi) \rightarrow \mathrm{E}(\Phi, \phi)} = \{(\neg\neg\phi, \phi)\} \cup \Delta(\Phi),$$

$$\rightarrowtriangle_{\mathrm{E}(\Phi, (\neg)X) \rightarrow \mathrm{E}(\Phi)} = \Delta(\Phi),$$

$$\rightarrowtriangle_{\mathrm{E}(\Phi, \phi_1 \wedge \phi_2) \rightarrow \mathrm{E}(\Phi, \phi_1, \phi_2)} = \{(\phi_1 \wedge \phi_2, \phi_1), (\phi_1 \wedge \phi_2, \phi_2)\} \cup \Delta(\Phi),$$

$$\rightarrowtriangle_{\mathrm{E}(\Phi, \phi_1 \vee \phi_2) \rightarrow \mathrm{E}(\Phi, \phi_i)} = \{(\phi_1 \vee \phi_2, \phi_i)\} \cup \Delta(\Phi),\ i \in \{1, 2\},$$

$$\rightarrowtriangle_{\mathrm{E}(\Phi, \phi \mathrm{U} \psi) \rightarrow \mathrm{E}(\Phi, \psi)} = \{(\phi \mathrm{U} \psi, \psi)\} \cup \Delta(\Phi),$$

$$\rightarrowtriangle_{\mathrm{E}(\Phi, \phi \mathrm{U} \psi) \rightarrow \mathrm{E}(\Phi, \phi, \mathrm{O}(\phi \mathrm{U} \psi))} = \{(\phi \mathrm{U} \psi, \phi), (\phi \mathrm{U} \psi, \mathrm{O}(\phi \mathrm{U} \psi))\} \cup \Delta(\Phi),$$

$$\rightarrowtriangle_{\mathrm{E}(\Phi, \phi \neg \mathrm{U} \psi) \rightarrow \mathrm{E}(\Phi, \neg\phi, \neg\psi)} = \{(\phi \neg \mathrm{U} \psi, \neg\phi), (\phi \neg \mathrm{U} \psi, \neg\psi)\} \cup \Delta(\Phi),$$

$$\rightarrowtriangle_{\mathrm{E}(\Phi, \phi \neg \mathrm{U} \psi) \rightarrow \mathrm{E}(\Phi, \neg\psi, \mathrm{O}(\phi \neg \mathrm{U} \psi))} = \{(\phi \neg \mathrm{U} \psi, \neg\psi), (\phi \neg \mathrm{U} \psi, \mathrm{O}(\phi \neg \mathrm{U} \psi))\} \cup \Delta(\Phi),$$

$$\rightarrowtriangle_{\mathrm{E}(\mathrm{O}\phi_1, \ldots, \mathrm{O}\phi_n) \rightarrow \mathrm{E}(\phi_1, \ldots, \phi_n)} = \{(\mathrm{O}\phi_i, \phi_i) \mid 1 \leqslant i \leqslant n\}.$$

We usually abbreviate $\rightarrowtriangle_{\mathrm{E}\Phi_1 \rightarrow \mathrm{E}\Phi_2}$ by $\rightarrowtriangle$ when the transition $\mathrm{E}\Phi_1 \rightarrow \mathrm{E}\Phi_2$ is understood from the context. We use Π for $\rightarrow$-paths and π for $\rightarrowtriangle$-paths, and write $\pi \in \Pi$ if Π and π are of equal length, and for each $i > 0$ for which $\Pi(i)$ is defined, $\pi(i-1) \rightarrowtriangle \pi(i)$ relative to the transition $\Pi(i-1) \rightarrow \Pi(i)$. Then an infinite $\rightarrow$-path Π is *admissible*, if for each $\pi \in \Pi$, whenever $\pi(i) = \phi \mathrm{U} \psi$ or $\pi(i) = \mathrm{F}\psi$ then for some $j > i$, $\pi(j) = \psi$. Suppose that Π visits the node $\mathrm{E}\Phi$ infinitely often. Consider a segment $\Pi(i_0, i_k) \triangleq \Pi(i_0) \rightarrow \cdots \rightarrow \Pi(i_k)$ of Π for which $\Pi(i_0) = \Pi(i_k) = \mathrm{E}\Phi$ and let $\phi \in \Phi$. Then ϕ is *regenerated along* $\Pi(i_0, i_k)$ if there is some $\pi \in \Pi(i_0, i_k)$ such that $\pi(0) = \pi(i_k - i_0) = \phi$.

$$\vee\ \dfrac{E(X\,U\,Y)}{Y \quad \wedge\ \dfrac{E(X,O(X\,U\,Y))}{X \quad \Diamond\ \dfrac{E(O(X\,U\,Y))}{E(X\,U\,Y)}}}$$

Fig. 1. Tableau for E(XUY).

$$\vee\ \dfrac{\dfrac{\Diamond\ \dfrac{E(OFX,OGOFX)}{E(FX,GOFX)}}{E(FX,OFX,OGOFX)}}{\wedge\ \dfrac{E(X,OFX,OGOFX)}{X \quad E(OFX,OGOFX)^{(1)}} \quad E(OFX,OGOFX)^{(2)}}$$

Fig. 2. Tableau for E(OFX,OGOFX).

$$\Diamond\ \dfrac{\dfrac{\Diamond\ \dfrac{E\Phi}{E(GOFX,GOFY,FX,FY)}}{E(\Phi,FX,FY)}}{\vee\ \dfrac{E(\Phi,X,FY)}{\wedge\ \dfrac{E(\Phi,X,Y)}{E\Phi^{(1)} \quad X \quad Y} \quad \dfrac{E(\Phi,X)}{E\Phi^{2} \quad X}} \qquad \vee\ \dfrac{E(\Phi,FY)}{\wedge\ \dfrac{E(\Phi,Y)}{E\Phi^{(3)} \quad Y} \quad E\Phi^{(4)}}}$$

Fig. 3. Tableau with $\Phi=\{OGOFX, OGOFY, OFX, OFY\}$.

Example 5.1. Here and below we use a tree-like notation for tableaux, terminating the construction as soon as nodes are first repeated.

(i) Figure 1 shows a tableau for E(XUY). The eventuality XUY is regenerated along the $\rightarrow$-path from the root to its repetition.

(ii) Figure 2 shows a tableau for the formula $\phi=$E(OFX,OGOFX). This is semantically equivalent to the formula EGFX expressing the fairness-related property that X holds infinitely often along some path. No eventuality is regenerated along the $\rightarrow$-path from the root to its repetition labelled (1), whereas the eventuality OFX is regenerated along the $\rightarrow$-path to (2).

(iii) Similarly, the formula EΦ of Fig. 3 expresses that both X and Y hold infinitely often. In Fig. 3 no eventuality is regenerated from the root to its repetition labelled (1), OFY is regenerated to (2), OFX is regenerated to (3) and both are regenerated to (4).

The following characterisation of the admissible $\rightarrow$-paths is of central importance to the translation procedure. Let a *simple node* be any node EΦ with the property that whenever $\phi U\psi\in\Phi$ (F$\phi\in\Phi$) then O$(\phi U\psi)\notin\Phi$(OF$\phi\notin\Phi$). Clearly, any infinite $\rightarrow$-path Π will infinitely often visit the same simple node. Consider, for instance, the set of all nodes visited by Π to which the $\Diamond$-rule applies.

Lemma 5.2. *Let Π be an infinite $\twoheadrightarrow$-path through a tableau τ, and let $i_0, i_1, \ldots$ be an infinite, strictly increasing sequence such that $\Pi(i_j)$ is the same simple node* $\mathrm{E}\Phi$ *for all $j \in \omega$. Then Π is admissible iff for each eventuality $\phi \in \Phi$ there are infinitely many $j \in \omega$ such that ϕ is not regenerated along the segment $\Pi(i_j, i_{j+1})$.*

Proof. The only-if direction is clear. For the if direction suppose that Π is inadmissible. Then there is an $\multimap$-path $\pi \in \Pi$, a $k \in \omega$, and an eventuality $\phi = \phi_1 \mathrm{U} \phi_2$, say, such that for all $k' \geqslant k$, either $\pi(k') = \phi$ or $\pi(k') = \mathrm{O}\phi$. As $\mathrm{E}\Phi$ is a simple node it follows that either for all j such that $i_j \geqslant k$, $\pi(i_j) = \phi$, or for all j with $i_j \geqslant k$, $\pi(i_j) = \mathrm{O}\phi$. In any case either ϕ is regenerated along every segment $\Pi(i_j, i_{j+1})$ for which $i_j \geqslant k$, or else $\mathrm{O}\phi$ is. □

The restriction to simple nodes in Lemma 5.2 is indeed necessary. To see this let Π_1 be the $\twoheadrightarrow$-path:

$$\begin{aligned}\mathrm{E}(\mathrm{OGF}X, \mathrm{F}X, \mathrm{OF}X, \mathrm{OGOF}X) &\twoheadrightarrow \mathrm{E}(\mathrm{OGF}X, \mathrm{OF}X, \mathrm{OGOF}X)\\ &\twoheadrightarrow \mathrm{E}(\mathrm{GF}X, \mathrm{F}X, \mathrm{GOF}X)\\ &\twoheadrightarrow \mathrm{E}(\mathrm{GF}X, \mathrm{OF}X, \mathrm{GOF}X)\\ &\twoheadrightarrow \mathrm{E}(\mathrm{OGF}X, \mathrm{F}X, \mathrm{OF}X, \mathrm{GOF}X)\\ &\twoheadrightarrow \mathrm{E}(\mathrm{OGF}X, \mathrm{F}X, \mathrm{OF}X, \mathrm{OGOF}X)\end{aligned}$$

and let Π_2 be the $\twoheadrightarrow$-path:

$$\begin{aligned}\mathrm{E}(\mathrm{OGF}X, \mathrm{F}X, \mathrm{OF}X, \mathrm{OGOF}X) &\twoheadrightarrow \mathrm{E}(\mathrm{OGF}X, \mathrm{OF}X, \mathrm{OGOF}X)\\ &\twoheadrightarrow \mathrm{E}(\mathrm{GF}X, \mathrm{F}X, \mathrm{GOF}X)\\ &\twoheadrightarrow \mathrm{E}(\mathrm{GF}X, \mathrm{F}X, \mathrm{OF}X, \mathrm{OGOF}X)\\ &\twoheadrightarrow \mathrm{E}(\mathrm{OGF}X, \mathrm{F}X, \mathrm{OF}X, \mathrm{OGOF}X).\end{aligned}$$

Notice that $\mathrm{F}X$ is not regenerated along Π_1 and that $\mathrm{OF}X$ is not regenerated along Π_2. Consider the infinite $\twoheadrightarrow$-path $\Pi = (\Pi_1 \cdot \Pi_2)^\omega$ obtained by alternating between Π_1 and Π_2 ad infinitum, and pick as the sequence $i_0, i_1, \ldots$ the largest sequence such that $\Pi(i_j) = \mathrm{E}(\mathrm{OGF}X, \mathrm{F}X, \mathrm{OF}X, \mathrm{OGOF}X)$. Clearly, Π is not admissible, but both eventualities $\mathrm{F}X$ and $\mathrm{OF}X$ fail to be regenerated along infinitely many segments $\Pi(i_j, i_{j+1})$.

6. Translating CTL*

Let then a tableau τ rooted in a formula $\phi_0 = \mathrm{E}\Phi_0$ be given. The translation of ϕ_0 is determined by the labelling of rule instances in τ, and least and greatest fixed points are used to characterise the admissible $\twoheadrightarrow$-paths through τ. The translation procedure passes each node ϕ at most twice in succession. The interesting case is when ϕ is a simple node. In this case a suitable context is built up in the first pass of ϕ, using the label information together with fixed-point quantifiers to bind propositional variables. For each simple node $\phi = \mathrm{E}\Phi$, we assume distinguished propositional variables X_Φ and $Y_{\psi,\Phi}$ when $\psi \in \Phi$ is an eventuality. These variables are used in the second pass, if it applies. Let $\phi = \mathrm{E}\Phi$. First, if there are no eventualities in Φ any infinite $\rightarrow$-path Π that visits ϕ infinitely often is admissible. A suitable scheme for the translation of ϕ in this case is consequently

$$\nu X_\Phi.\gamma(X_\Phi),$$

where ϕ in the second pass is translated into X_Φ. In general, however, by Lemma 5.2, each eventuality $\psi \in \Phi$ must infinitely often be prevented from being regenerated between subsequent visits to ϕ by Π. This suggests

$$\nu X_\Phi. \wedge_{\text{eventualities } \psi \in \Phi} \mu Y_{\psi,\Phi}.\gamma(X_\Phi, Y_{\psi,\Phi})$$

as a general scheme for the translation of ϕ, where in the second pass ϕ is translated into $Y_{\psi,\Phi}$ if ψ is regenerated along the path segment traversed since the first pass, and as X_Φ otherwise.

The translation algorithm is shown in Fig. 4. A node ϕ is translated into the L_μ-formula $tr(\phi, \Pi, \varepsilon)$, where Π is a path segment and ε is an eventuality selector: a mapping which given a set Φ chooses an eventuality $\varepsilon(\Phi) \in \Phi$, if one exists. The role of Π is to keep track of the path traversed so far, and ε is used to handle the conjunction over eventualities. We use $\Pi \rightarrow \phi$ to denote the $\rightarrow$-path obtained by appending ϕ to Π. Initially, Π is the empty $\rightarrow$-path ( ), and ε is an arbitrary eventuality selector ε_0. The

```
$tr(\phi, \Pi, \varepsilon) =$
cases $\phi$ of $X$: $X$ | $\neg X$: $\neg X$ | $\mathrm{E}\Phi$:
  if $\exists i : (i) = \phi$
  then if $\varepsilon(\Phi)$ is defined
    then if $\varepsilon(\Phi)$ is regenerated along segment $\Pi(i) \rightarrow \cdots \rightarrow \Pi(n) \rightarrow \phi$
      then $Y_{\varepsilon(\Phi),\Phi}$
      else $X_\Phi$
    else $X_\Phi$
  else let $\Omega : \dfrac{\phi}{\phi_1, \ldots, \phi_m}$ be the rule instance applied to $\phi$
  in if $\phi$ is not simple or $\phi$ is not reachable from any of the nodes $\phi_1, \ldots, \phi_m$
    then $\Omega(tr(\phi_1, \Pi \rightarrow \phi, \varepsilon), \ldots, tr(\phi_m, \Pi \rightarrow \phi, \varepsilon))$
    else if there are no eventualities in $\Phi$
      then $\nu X_\Phi.\Omega(tr(\phi_1, \Pi \rightarrow \phi, \varepsilon), \ldots, tr(\phi_m, \Pi \rightarrow \phi, \varepsilon))$
      else $\nu X_\Phi. \wedge_{\text{eventualities } \psi \in \Phi} \mu Y_{\psi,\Phi}.$
        $\Omega(tr(\phi_1, \Pi \rightarrow \phi, \varepsilon[\Phi \mapsto \psi]), \ldots, tr(\phi_m, \Pi \rightarrow \phi, \varepsilon[\Phi \mapsto \psi]))$
```

Fig. 4. CTL* translation algorithm tr.

translation algorithm works as follows. If $\phi = \mathrm{E}\Phi$ is a leaf the translation is trivial; otherwise, if $\exists i : \Pi(i) = \phi$ the procedure is in its second pass of ϕ. Then ϕ is translated into $Y_{\varepsilon(\Phi),\Phi}$ if $\varepsilon(\Phi)$ is defined and is regenerated along the segment traversed since ϕ was passed first, and ϕ is translated into X_Φ otherwise. Assume instead that the procedure is in its first pass. Three cases apply: If ϕ is not simple or ϕ is not reachable from any of its own children then no fixed points are needed. Otherwise, as we have explained, the translation depends on whether or not Φ contains any eventualities.

The translation relation $\rightsquigarrow$ is then completed by adding to the four rules of Section 4 the axiom $\phi_0 \rightsquigarrow \psi_0$ whenever ϕ_0 is a basic CTL* formula, a tableau τ rooted in ϕ_0 is given, and relative to τ, $\psi_0 = tr(\phi_0, (\;), \varepsilon_0)$.

Example 6.1. (i) From the tableau of Fig. 1 the expected translation is

$$\mathrm{E}(X \mathrm{U} Y) \rightsquigarrow \mu Y'.Y \vee (X \wedge \Diamond Y').$$

(ii) For the tableau of Fig. 2:

$$\mathrm{E}(\mathrm{OF}X, \mathrm{OGOF}X) \rightsquigarrow \nu X'.\mu Y'.\Diamond((X \wedge X') \vee Y').$$

The example serves to illustrate why the translation schema uses ν-μ-alternation. For comparison, two possible translations of the (equivalent) formula EG FX are $\nu X'.\mu Y'.(X \wedge \Diamond X') \vee \Diamond Y'$ and $\nu X'.(X \wedge \Diamond X') \vee \mu Y'.\Diamond((X \wedge X') \vee (X \wedge Y') \vee Y')$.

(iii) Similarly, the tableau of Fig. 3 illustrates why the conjunction of μ-formulas is in general necessary. It is translated thus:

$$\mathrm{E}\Phi \rightsquigarrow \nu X'.(\mu Y'.\Diamond(X' \wedge X \wedge Y) \vee (Y' \wedge X) \vee (X' \wedge Y) \vee Y')$$
$$\wedge (\mu Y'.\Diamond(X' \wedge X \wedge Y) \vee (X' \wedge X) \vee (Y' \wedge Y) \vee Y').$$

The translation algorithm can be optimised in several respects. For instance, we avoid generally introducing variables that are never actually used. The topic of optimisations is returned to in the concluding section.

Theorem 6.2. *For any CTL* formula* ϕ, *if* $\phi \rightsquigarrow \psi$ *then* $\{s \mid s \models_{\mathscr{V}} \phi\} = \|\psi\|\mathscr{V}$.

Proof. See Section 7. □

7. Proof of Theorem 6.2

Let τ be a tableau rooted in $\phi_0 = \mathrm{E}\Phi_0$ and let $\phi_0 \rightsquigarrow \psi_0$ relative to τ. For simplicity, assume ϕ_0 to be in positive form, i.e. with $\wedge, \vee, \mathrm{U}, \neg\mathrm{U}$ primitive and negations applied only to propositional variables. Similarly, we also assume ψ_0 to be in a positive form. The proof is split into two parts:

(i) If $s_0 \models_{\mathscr{V}} \phi_0$ then there is a successful choice relation rooted in $s_0 \vdash \psi_0$, i.e. one which is well-founded and agrees with $\mathscr{V}$.

(ii) If $s_0 \not\models_{\mathscr{V}} \phi_0$ then there is a successful choice relation rooted in $s_0 \vdash \neg\psi_0$.

This is sufficient by Theorem 2.1. Both (i) and (ii) are proved by first building a choice relation $\Rightarrow$ agreeing with $\mathscr{V}$, and then showing $\Rightarrow$ to be well-founded.

Note first that ψ_0 determines a finite set $\mathrm{BV}(\psi_0)=\{X_1,\ldots,X_n\}$ of propositional variables bound in ψ_0. Moreover, each $X\in\mathrm{BV}(\psi_0)$ is bound exactly once. Let σX denote the subformula of ψ_0 binding X. Given any subformula ψ of ψ_0, $\mathrm{unf}(\psi)$ is the formula resulting from recursively replacing each $X\in\mathrm{BV}(\psi_0)$ free in ψ by σX:

$$\mathrm{unf}(\psi)\triangleq\psi[\mathrm{unf}(\sigma X_1)/X_1]\cdots[\mathrm{unf}(\sigma X_n)/X_n].$$

The construction of $\Rightarrow$ proceeds in stages, and is guided by the tableau and the model. Initially, we are given the tableau root node ϕ_0 and the sequent $s_0\vdash(\neg)\psi_0$. At the completion of stage k, having reached tableau node $\phi=\mathrm{E}\Phi$ and sequent $s\vdash(\neg)\psi$ the following properties are maintained invariant:

(a) $s\models_{\mathscr{V}}(\neg)\phi$, and

(b) for some ε and Π, $\psi=\mathrm{unf}(tr(\phi,\Pi,\varepsilon))$.

Note that, depending on Π, $tr(\phi,\Pi,\varepsilon)$ is either the variable X_Φ or the variable $Y_{\varepsilon(\Phi),\Phi}$, or else $tr(\phi,\Pi,\varepsilon)=\sigma X_\Phi$. We show how to complete stage $k+1$. For leaves the construction of $\Rightarrow$ is complete. Assume then that Ω labels the rule instance

$$\frac{\phi}{\phi_1,\ldots,\phi_m},$$

ϕ is simple, that ϕ is reachable from one of the nodes $\phi_1,\ldots,\phi_m$, and that Φ contains an eventuality. The proofs where one of these assumptions fail are easy special cases. In completing stage $k+1$ the important issue is how to resolve choices, and this is dependent on whether we are proving (i) or (ii).

(i) Here the problem cases are when $\Omega=\vee$ or $\Omega=\Diamond$. From the assumption that $s\models_{\mathscr{V}}\phi$ we know the existence of a path which validates each member of Φ. The procedure builds this path by indexing eventualities. An index ξ of ϕ assigns a natural number to the *top-level eventualities* of ϕ: the subformulas of ϕ of the form $\phi_1\mathrm{U}\phi_2$ that do not occur within the scope of $\neg\mathrm{U}$ in ϕ. The predecessor of ξ is the index $\mathrm{pred}(\xi)$ defined by $\mathrm{pred}(\xi)(\phi_1\mathrm{U}\phi_2)=\xi(\phi_1\mathrm{U}\phi_2)-1$ when $\xi(\phi_1\mathrm{U}\phi_2)>0$ and $\mathrm{pred}(\xi)(\phi_1\mathrm{U}\phi_2)=0$ otherwise. Then $\phi[\xi]$ indexes each member of Φ in the following way:

$$((\neg)X)[\xi]=(\neg)X,\qquad(\phi_1\wedge\phi_2)[\xi]=\phi_1[\xi]\wedge\phi_2[\xi],$$

$$(\phi_1\vee\phi_2)[\xi]=\phi_1[\xi]\vee\phi_2[\xi],$$

$$(\mathrm{O}\phi_1)[\xi]=\mathrm{O}(\phi_1[\mathrm{pred}(\xi)]),\qquad(\phi_1\neg\mathrm{U}\phi_2)[\xi]=\phi_1\neg\mathrm{U}\phi_2,$$

$$(\phi_1\mathrm{U}\phi_2)[\xi]=\phi_1\mathrm{U}^{\xi(\phi_1\mathrm{U}\phi_2)}\phi_2,$$

where $\phi_1\mathrm{U}^i\phi_2$ is the obvious approximation, i.e.

$$\phi_1\mathrm{U}^0\phi_2=\phi_2,$$

$$\phi_1\mathrm{U}^{n+1}\phi_2=\phi_2\vee(\phi_1\wedge\mathrm{O}(\phi_1\mathrm{U}^n\phi_2)).$$

It is clear that if $s\models_{\mathscr{V}}\phi$ then there is some index ξ appropriate for ϕ at s – i.e. such that $s\models_{\mathscr{V}}\phi[\xi]$.

The construction of $\Rightarrow$ now proceeds as follows. First the rules for fixed points (and, if necessary, conjunction) are applied to $s \vdash \psi$ until a sequent of the form $s \vdash \Omega(\phi_1, \ldots, \psi_m)$ is reached. The construction now depends on the tableau rule applied.

(1) Double negation: $\Phi = (\Phi', \neg\neg\phi')$. Then $s \models_{\mathscr{T}} \mathrm{E}(\Phi', \phi')[\xi]$. Moreover, $\Omega = I$ (so $m = 1$) and we proceed from the sequent $s \vdash \psi_1$.

(2) (Negated) propositional variable: $\Phi = (\Phi', (\neg)X)$. Then $s \models_{\mathscr{T}} \mathrm{E}\Phi'[\xi]$, $\Omega = \wedge$, $\psi_2 = (\neg)X$, and we proceed from $s \vdash \psi_1$.

(3) Conjunction: $\Phi = (\Phi', \phi'_1 \wedge \phi'_2)$. Then $s \models_{\mathscr{T}} \mathrm{E}(\Phi', \phi'_1, \phi'_2)[\xi]$, $\Omega = I$ and we proceed from $s \vdash \psi_1$.

(4) Disjunction: $\Phi = (\Phi', \phi'_1 \vee \phi'_2)$. Choose $i \in \{1,2\}$ such that $s \models_{\mathscr{T}} \mathrm{E}(\Phi', \phi'_i)[\xi']$ where ξ' is ξ restricted to the top-level eventualities of Φ', ϕ'_i. Also $\Omega = \vee$, and we proceed from $s \vdash \psi_i$.

(5) Until: $\Phi = (\Phi', \phi'_1 \mathrm{U} \phi'_2)$. We know that either $s \models_{\mathscr{T}} \mathrm{E}(\Phi', \phi'_2)[\xi']$ or else $s \models_{\mathscr{T}} \mathrm{E}(\Phi', \phi'_1, \mathrm{O}(\phi'_1 \mathrm{U} \phi'_2))[\xi']$, where in either case ξ' is the appropriate restriction of ξ. Also $\Omega = \vee$, and we proceed from $s \vdash \psi_1$ if the first case applies and from $s \vdash \psi_2$ if the second does.

(6) "Not until": $\Phi = (\Phi', \phi'_1 \neg \mathrm{U} \phi'_2)$. Either $s \models_{\mathscr{T}} \mathrm{E}(\Phi', \neg\phi'_1, \neg\phi'_2)[\xi']$ or $s \models_{\mathscr{T}} \mathrm{E}(\Phi', \neg\phi'_2, \mathrm{O}(\phi'_1 \neg \mathrm{U} \phi'_2))[\xi']$ where in either case ξ' is an index agreeing with ξ on their common top-level eventualities. Again $\Omega = \vee$ and the choice of ψ_i is guided as in (4).

(7) Next-time: $\Phi = (\mathrm{O}\phi'_1, \ldots, \mathrm{O}\phi'_n)$. Then there is some s' such that sRs' and $s \models_{\mathscr{T}} \mathrm{E}(\phi'_1, \ldots, \phi'_n)[\mathrm{pred}(\xi)]$. Moreover $\Omega = \Diamond$ and we proceed from $s' \vdash \psi_1$.

We have thus verified that the invariants (a) and (b) above are maintained, taking indexing into account.

(ii) In this case the problem is to deal with the conjunction over eventualities. With each variable $X_{\Phi'}$, bound in ψ_0 is associated a *scheduler*, $f_{\Phi'}$, which picks out an eventuality $ev(f_{\Phi'})$ in Φ' in a round-robin fashion. If there are no eventualities in Φ' the scheduler is never applied. Assume first that $tr(\phi, \Pi, \varepsilon)$ is identical to either σX_Φ or X_Φ. Note that in this case $\neg\psi$ is a formula of the form

$$\mu X_\Phi . \vee_{\text{eventualities } \phi' \in \Phi} \nu Y_{\phi', \Phi} . \neg\Omega(\cdots) \tag{1}$$

The sequence of actions is as follows. The scheduler is updated, the fixed-point unfolding rule is applied to $\neg\psi$, the scheduled disjunct

$$\neg\psi' = \nu Y_{ev(f_\Phi), \Phi} . \neg(\cdots) \tag{2}$$

is chosen, and the fixed-point unfolding rule is applied to $\neg\psi'$. If, on the other hand, $tr(\phi, \Pi, \varepsilon) = Y_{\varepsilon(\Phi), \Phi}$ then $\neg\psi$ has the form (2) already and we merely apply the fixed-point unfolding rule to $\neg\psi$. In either case a formula of the form $\neg\Omega(\psi_1, \ldots, \psi_m)$ results. Stage $k+1$ is now completed in a fashion very similar to the corresponding construction in (i). It is in fact simpler as the only points involving choice is when $\Omega = \wedge$, and this is the case only when one conjunct is a (possibly negated) propositional variable which is chosen whenever possible.

We have thus given strategies for building choice relations $\Rightarrow$ that agree with $\mathscr{V}$. It remains to show that any relation $\Rightarrow$ built using these strategies is well-founded. Assume for a contradiction that it is not. Note that ψ_0 is *well-guarded* in the sense that whenever $\sigma X.\psi$ is a subformula of ψ_0 then each occurrence of X in ψ is in the scope of a modal operator. It then follows that there is an infinite path $\rho \in S^{\omega}$, a μ-formula $\mu X.\psi$, and an infinite, strictly increasing sequence $i_0, i_1, \ldots$ such that for all $j \in \omega$, $\mu X.\psi$ is regenerated from $\rho(i_j)$ to $\rho(i_{j+1})$. The path ρ corresponds to an infinite $\rightarrow$-path $\phi_0 \rightarrow \phi_1 \rightarrow \cdots$ through τ. Assume that each ϕ_i has the form $\mathrm{E}\Phi_i$, and that the suffix of ρ corresponding to the $\rightarrow$-path $\phi_i \rightarrow \phi_{i+1} \rightarrow \cdots$ is ρ_i. Again the proof splits according to whether we are proving (i) or (ii).

(i) In this case we find some $\phi = \mathrm{E}\Phi$ such that $\mu X.\psi$ is of the form $\mathrm{unf}(tr(\phi, \Pi, \varepsilon))$ where $tr(\phi, \Pi, \varepsilon) = Y_{\varepsilon(\Phi), \Phi}$. Let ξ_i be the index assigned to the tableau node ϕ_i during the choice relation construction. An easy induction on the structure of $\phi_i' \in \Phi_i$ then shows that $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$.

(1) $\phi_i' = (\neg)X$. Here $\rho_i(0) \in \mathscr{V}(X)$ $(\rho_i(0) \notin \mathscr{V}(X))$ as $\Rightarrow$ agrees with $\mathscr{V}$.

(2) $\phi_i' = \neg\neg\phi_{i,1}$. By the induction hypothesis, $\rho_i \models_{\mathscr{V}} \phi_{i,1}[\xi_i]$ so also $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$.

(3) $\phi_i' = \phi_{i,1} \wedge \phi_{i,2}$. By the induction hypothesis, $\rho_i \models_{\mathscr{V}} \phi_{i,j}[\xi_i]$ for both $j=1$ and $j=2$, so $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$.

(4) $\phi_i' = \phi_{i,1} \vee \phi_{i,2}$. By the induction hypothesis, $\rho_i \models_{\mathscr{V}} \phi_{i,j}[\xi_i]$ for either $j=1$ or $j=2$, so $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$.

(5) $\phi_i' = \phi_{i,1} \mathrm{U} \phi_{i,2}$. By the construction of $\Rightarrow$, we obtain a smallest $i' \geqslant i$ such that $\phi_{i,2} \in \Phi_{i'}$ whence by the induction hypothesis, $\rho_{i'} \models_{\mathscr{V}} \phi_{i,2}[\xi_{i'}]$. Moreover, for all i'', $i \leqslant i'' < i'$, $\phi_{i,1} \in \Phi_{i''}$, so $\rho_{i''} \models_{\mathscr{V}} \phi_{i,1}[\xi_{i''}]$, so indeed $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$.

(6) $\phi_i' = \phi_{i,1} \neg\mathrm{U} \phi_{i,2}$. We either obtain a smallest $i' \geqslant i$ such that $\neg\phi_{i,1}, \neg\phi_{i,2} \in \Phi_{i'}$, so $\rho_{i'} \models_{\mathscr{V}} \neg\phi_{i,j}[\xi_{i'}]$ for both $j=1$ and $j=2$. Moreover, for all i'', $i \leqslant i'' < i'$, $\neg\phi_{i,2} \in \Phi_{i''}$, so $\rho_{i''} \models_{\mathscr{V}} \neg\phi_{i,2}[\xi_{i''}]$, so indeed in this case $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$. Alternatively, for all $i' \geqslant i \neg\phi_{i,2} \Phi_{i'}$, and using the induction hypothesis we can conclude that $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$.

(7) $\phi_i' = \mathrm{O}\phi_{i,1}$. By the induction hypothesis, $\rho_{i+1} = \rho_i^1 \models_{\mathscr{V}} \phi_{i,1}[\xi_{i+1}]$ and $\xi_{i+1} = \mathrm{pred}(\xi_i)$. Hence $\rho_i \models_{\mathscr{V}} \phi_i'[\xi_i]$.

But we are then almost done, for the μ-formula $\mu X.\psi$ is regenerated infinitely often along ρ only if we find some infinite $\rightarrowtriangle$-path π relative to the given infinite Π-path through τ which from some point onwards always has one of the forms $\phi_1 \mathrm{U} \phi_2$ or $\mathrm{O}(\phi_1 \mathrm{U} \phi_2)$ for fixed ϕ_1, ϕ_2. But this contradicts the indexing strategy.

(ii) Here $\mu X.\psi$ has the form $\neg\mathrm{unf}(tr(\phi, \Pi, \varepsilon))$ where either $tr(\phi, \Pi, \varepsilon) = X_\Phi$ or $tr(\phi, \Pi, \varepsilon) = \sigma X_\Phi$. Completing the proof, we show by induction on the size of formulas that $\rho_i \models_{\mathscr{V}} \phi_i'$ whenever $\phi_i' \in \Phi_i$, contradicting the assumption that $\rho_i(0) \models_{\mathscr{V}} \neg\phi_i$. The only slightly difficult case is for $\phi_i' = \phi_{i,1} \mathrm{U} \phi_{i,2}$. As $\mu X.\psi$ has the form (1) it cannot for all $i' \geqslant i$ be the case that $\phi_{i,2} \notin \Phi_{i'}$. For otherwise, by Lemma 5.2, for all $i' \geqslant i$ such that $\Phi_{i'} = \Phi_i$, the eventuality ϕ_i' would be regenerated along the $\rightarrow$-derivation from $\mathrm{E}\Phi_i$ to $\mathrm{E}\Phi_{i'}$. Moreover, the scheduling mechanism ensures that a disjunct of the form

$$\nu Y_{\gamma, \Phi_i}.\neg\Omega(\cdots)$$

is eventually visited for some $i' \geqslant i$ such that from the point i' onwards $\mu X.\phi$ can never be regenerated. In particular, γ may be ϕ'_i. Thus, we can find a minimal i' such that $\phi_{i,2} \in \Phi_{i'}$. By minimality of i' for all i'', $i \leqslant i'' < i'$, $\phi_{i,1} \in \Phi_{i''}$. But then by the induction hypothesis, $\rho_{i'} \models_{\mathscr{V}} \phi_{i,2}$ and $\rho_{i''} \models_{\mathscr{V}} \phi_{i,1}$ whenever $i \leqslant i'' < i'$, i.e. $\rho_i \models_{\mathscr{V}} \phi_i$.

8. Extended computation tree logic, ECTL*

Several natural extensions of CTL* merit consideration. One direction of pragmatic interest is to add next-time operators indexed by labels or sets of labels as in [2]. This direction is pursued in [10]. Another direction of more fundamental interest is to extend the linear-time fragment of CTL* such as to give it the full power of the ω-regular languages. Here we follow the approach of Thomas [29] by adding temporal operators corresponding to Büchi automata on infinite words.

A *Büchi automaton* over the finite alphabet Σ is a nondeterministic finite automaton $\mathscr{A} = (Q, q_0, \{\xrightarrow{a}\}_{a \in \Sigma}, F)$ with Q the finite set of states, $q_0 \in Q$ the initial state, $\xrightarrow{a} \subseteq Q \times Q$ the transition relation for each $a \in \Sigma$, and $F \subseteq Q$ the set of final states. We sometimes write $\mathscr{A}(q_0)$ to emphasize the initial state q_0. A *run* of $\mathscr{A}$ on the ω-word $\alpha \in \Sigma^\omega$ is an ω-word $r \in Q^\omega$ with the property that $r(0) = q_0$ and $r(i) \xrightarrow{\alpha(i)} r(i+1)$ for each $i \in \omega$. Then $\mathscr{A}$ *accepts* α, if there is a run r of $\mathscr{A}$ on α and a $q \in F$ s.t. $r(i) = q$ for infinitely many i, and the language *recognised* by $\mathscr{A}$ is $\mathscr{L}(\mathscr{A}) = \{\alpha \in \Sigma^\omega \mid \mathscr{A} \text{ accepts } \alpha\}$.

Formulas of ECTL* are inductively defined as usual and built from propositional variables using boolean connectives and containing the formula $\mathrm{E}(\mathscr{A})$ for each Büchi automaton $\mathscr{A}$ over an alphabet $2^{\{\phi_1, \ldots, \phi_n\}}$ where the ϕ_i are ECTL* formulas. Note that all formulas of ECTL* are state-formulas; linear-time dependencies are accounted for by automata. Also the semantics is inductively defined. The clauses for variables and boolean connectives are the usual ones:

$$s \models_{\mathscr{V}} X \text{ iff } s \in \mathscr{V}(X),$$

$$s \models_{\mathscr{V}} \neg\phi \text{ iff } s \not\models_{\mathscr{V}} \phi,$$

$$s \models_{\mathscr{V}} \phi_1 \wedge \phi_2 \text{ iff } s \models_{\mathscr{V}} \phi_1 \text{ and } s \models_{\mathscr{V}} \phi_2.$$

Then let $\mathscr{A}$ be a Büchi automaton over the alphabet $2^{\{\phi_1, \ldots, \phi_n\}}$. The intuition is that $\mathrm{E}(\mathscr{A})$ is true of a state s just in case there is an infinite path ρ originating in s such that the ω-word over $2^{\{\phi_1, \ldots, \phi_n\}}$ that encodes the satisfaction of $\phi_1, \ldots, \phi_n$ along ρ is in the language recognised by $\mathscr{A}$. More precisely, let the word $\alpha(\rho)$ be determined by

$$\alpha(\rho)(i) = \{\phi_j \mid 1 \leqslant j \leqslant n,\ \rho(i) \models_{\mathscr{V}} \phi_j\}.$$

The satisfaction clause for $\mathrm{E}(\mathscr{A})$ is then the following:

$$s \models_{\mathscr{V}} \mathrm{E}(\mathscr{A}) \text{ iff } \exists\rho.\ \rho(0) = s,\ \alpha(\rho) \in \mathscr{L}(\mathscr{A}).$$

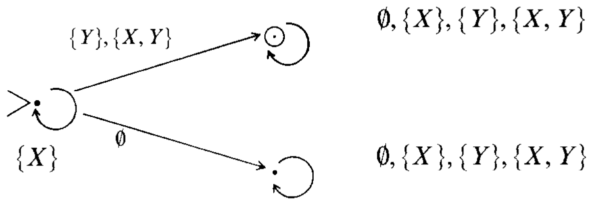
$\{Y\},\{X,Y\}$
$\emptyset,\{X\},\{Y\},\{X,Y\}$
$\{X\}$
$\emptyset$
$\emptyset,\{X\},\{Y\},\{X,Y\}$

Fig. 5. Büchi automaton $\mathscr{A}_1$ for $X\mathrm{U}Y$.

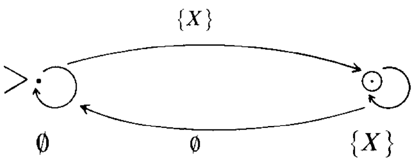

Fig. 6. Büchi automaton $\mathscr{A}_2$ for $\mathrm{GF}X$.

Example 8.1. The ECTL* formula $\mathrm{E}(\mathscr{A}_1)$ where $\mathscr{A}_1$ is the Büchi automaton of Fig. 5 expresses the CTL* property $\mathrm{E}(X\mathrm{U}Y)$. Similarly, the ECTL* formula $\mathrm{E}(\mathscr{A}_2)$ expresses $\mathrm{EGF}X$ (infinitely often X) where $\mathscr{A}_2$ is the automaton of Fig. 6.

The closure properties of languages recognised by Büchi automata provide mechanisms for deriving various linear-time connectives. Complementation, for instance, is derived by complementing the Büchi automaton concerned. In this way it is not too difficult to see that any CTL* formula can be written as an equivalent ECTL* formula.

Note in particular that the definition of the universal path-quantifier in terms of the existential one in the case of ECTL* requires complementation of Büchi automata. In this respect the present account sacrifices clarity to some extent. This can be remedied by using, for instance, deterministic Muller automata as in [6] at a cost, however, of a vastly more complicated translation procedure.

9. Translating ECTL*

The use of Büchi automata together with the existential path-quantifier allows attention to be restricted to formulas $\mathrm{E}(\mathscr{A})$ in a standard form, where $\mathscr{A}$ is an automaton over sets of propositional variables only, and for which the set F of final states is a singleton. For let $F=\{q_1,\ldots,q_m\}$ and let each $\mathscr{A}_i$ be obtained from $\mathscr{A}$ by replacing F by $\{q_i\}$. If $\rightsquigarrow_e$ is the ECTL* correlate of $\rightsquigarrow$ then in addition to the four rules of Section 4 the following rule is validated:

$$\frac{\mathrm{E}(\mathscr{A}_1)\rightsquigarrow_e\psi_1 \quad \cdots \quad \mathrm{E}(\mathscr{A}_m)\rightsquigarrow_e\psi_m}{\mathrm{E}(\mathscr{A})\rightsquigarrow_e\psi_1 \vee \cdots \vee \psi_m} \tag{3}$$

Tableaux are constructed using the following single rule. Let $\mathrm{E}(\mathscr{A}(q))$ be in standard form where the alphabet of $\mathscr{A}(q)$ is $\Sigma = 2^{\{Y_1, \ldots, Y_n\}}$. Let $a_1, \ldots, a_m$ list the members of Σ and for each i such that $1 \leqslant i \leqslant m$ let $q_{i,1}, \ldots, q_{i,k_i}$ list the q' such that $q \xrightarrow{a_i} q'$. Moreover, for each $a \in \Sigma$ let $\bar{a}$ abbreviate the formula $\bigwedge a \wedge \bigwedge \{\neg Y \mid Y \notin a\}$. The rule is then the following:

$$\Omega\colon \frac{\mathrm{E}(\mathscr{A}(q))}{\mathrm{E}(\mathscr{A}(q_{1,1}))\cdots\mathrm{E}(\mathscr{A}(q_{1,k_1}))\cdots\mathrm{E}(\mathscr{A}(q_{m,1}))\cdots\mathrm{E}(\mathscr{A}(q_{m,k_m}))}$$

where Ω is the operator that takes $X_{1,1}, \ldots, X_{1,k_1}, \ldots, X_{m,1}, \ldots, X_{m,k_m}$ into

$$(\bar{a}_1 \wedge \Diamond X_{1,1}) \vee \cdots \vee (\bar{a}_1 \wedge \Diamond X_{1,k_1}) \vee \cdots \vee (\bar{a}_m \wedge \Diamond X_{m,1}) \vee \cdots \vee (\bar{a}_m \wedge \Diamond X_{m,k_m}).$$

As we only consider tableau constructed from nodes in standard form, there is at most one node $\mathrm{E}(\mathscr{A}(q))$ for which q is an accepting state. The intention is to translate this node as a ν-formula and all other internal nodes as μ-formulas. For this to work, however, we must ensure that all possible ways of "regenerating" q are captured. This is done by carrying along a set $V \subseteq Q_{\mathscr{A}}$ indicating the states that have been visited so far, and then resetting this "visited-table" whenever the accepting state is first encountered. This is the idea of the translation algorithm tr_e shown in Fig. 7. The algorithm assumes for each state q a unique variable X_q. A node ϕ is translated into the L_μ-formula $tr_e(\phi, V)$ where $V \subseteq Q_{\mathscr{A}}$ is the visited-table, initially empty. The translation relation $\rightsquigarrow_e$ is then completed by adding to the four rules of Section 4 and the rule (3) above the axiom $\phi_0 \rightsquigarrow_e \psi_0$ whenever ϕ_0 is in standard form and $\psi_0 = tr_e(\phi_0, \emptyset)$.

Theorem 9.1. *For any ECTL* formula ϕ, if $\phi \rightsquigarrow_e \psi$ then $\{s \mid s \models_{\mathscr{V}} \phi\} = \|\psi\| \mathscr{V}$.*

Proof. Let a tableau τ for $\phi_0 = \mathrm{E}(\mathscr{A}(q_0))$ be given where $\mathscr{A}(q_0)$ is in standard form. Let $\psi_0 = tr_e(\phi_0, \emptyset)$. Note that due to the way the visited-table V is reset when the accepting state q is first visited by the translation procedure the only ν-subformula of

```
$tr_e(\mathrm{E}(\mathscr{A}(q)), V) =$
if $q \in V$
then $X_q$
else let $\Omega\colon \frac{\mathrm{E}(\mathscr{A}(q))}{\mathrm{E}(\mathscr{A}(q_1))\cdots\mathrm{E}(\mathscr{A}(q_k))}$ be the rule instance applied to $\mathrm{E}(\mathscr{A}(q))$
   in if $q$ is accepting
      then $\nu X_q.\Omega(tr_e(\mathrm{E}(\mathscr{A}(q_1)), \{q\}), \ldots, tr_e(\mathrm{E}(\mathscr{A}(q_k)), \{q\}))$
      else $\mu X_q.\Omega(tr_e(\mathrm{E}(\mathscr{A}(q_1)), V \cup \{q\}), \ldots, tr_e(\mathrm{E}(\mathscr{A}(q_k)), V \cup \{q\}))$
```

Fig. 7. ECTL* translation algorithm tr_e.

ψ_0 is the formula $tr_e(\mathrm{E}(\mathscr{A}(q)), \emptyset)$. By Theorem 2.1 it suffices to show that $s_0 \models_{\mathscr{V}} \phi_0$ iff there is a well-founded choice relation $\Rightarrow$ from $s_0 \vdash \psi_0$ which agrees with $\mathscr{V}$.

Corresponding to any choice relation $\Rightarrow$ which agrees with $\mathscr{V}$ there is an infinite path ρ from s_0 and a run r of $\mathscr{A}(q_0)$ on $\alpha(\rho)$. Conversely, any run r of $\mathscr{A}(q_0)$ on $\alpha(\rho)$ determines a choice relation $\Rightarrow$ which agrees with $\mathscr{V}$. Fix then a choice relation $\Rightarrow$ which agrees with $\mathscr{V}$. Note that there is exactly one infinite derivation of $\Rightarrow$, say,

$$\Delta \;=\; s_0 \vdash \psi_0 \Rightarrow s_1 \vdash \psi_1 \Rightarrow \cdots$$

originating in $s_0 \vdash \psi_0$. The proof is complete when it is shown that the run r corresponding to Δ visits q infinitely often iff $\Rightarrow$ is well-founded. Assume the latter. Then for infinitely many i, $\psi_i = tr_e(\mathrm{E}(\mathscr{A}(q)), \emptyset)$, and the result follows. Conversely, assume that r visits q infinitely often. Then there is an infinite, strictly increasing sequence $i_0, i_1, \ldots$ such that for all $j \in \omega$ is $\psi_{i_j} = tr_e(\mathrm{E}(\mathscr{A}(q)), \emptyset)$. Moreover ψ_{i_0} is a subformula of each ψ_k for all $k \geqslant i_0$. Hence, there can be no μ-formula $\mu X.\phi$ for which $\psi_k = \mu X.\phi$ for infinitely many k and such that $\mu X.\phi$ is a subformula of ψ_k for almost all k. For then $\mu X.\phi = tr_e(\mathrm{E}(\mathscr{A}(q)), \emptyset)$, a contradiction. It follows that $\Rightarrow$ is well-founded. □

Example 9.2. The formula $\mathrm{E}(\mathscr{A}_2)$ of Example 8.1 is translated into the formula

$$\mu X'.(\neg X \wedge \Diamond X') \vee (X \wedge \Diamond \phi),$$

where ϕ is determined in the following way:

$$\phi = \nu Y'.(\neg X \wedge \Diamond \psi) \vee (X \wedge \Diamond Y'),$$

$$\psi = \mu X'.(\neg X \wedge \Diamond X') \vee (X \wedge \Diamond Y').$$

Note that without resetting the "visited-table" ϕ becomes instead

$$\phi = \nu Y'.(\neg X \wedge \Diamond X') \vee (X \wedge \Diamond Y')$$

equivalent to EFGX and inequivalent to $\mathrm{E}(\mathscr{A}_2)$.

10. Concluding remarks

As the "standard translation", our translation from CTL* is doubly exponential in the length of the input formula, and the translation from ECTL* is singly exponential. The difference is accounted for by the exponential cost of representing CTL* in ECTL*. This suggests an alternative, also doubly exponential, translation from CTL* in three stages that first translates tableaux into ECTL* using, for instance, Emerson and Sistla's construction [14], and then translates ECTL* into L_μ. This translation shares, however, with the standard translation the pragmatic but nonetheless important problem that their results are not very intuitive or readable, and indeed best thought of as L_μ encodings of Büchi automata.

The existence of a translation of ECTL* into L_μ is not very surprising. It is well-known that ECTL* is strictly less expressive than Rabin's S2S (cf. [29]), and Niwinsky [22] shows S2S to be expressively equivalent to a μ-calculus with, in effect, a left and right next-time operator. This does not, however, give an embedding into L_μ itself. The standard translation, for instance, is easily modified for this purpose, by translating Büchi automata into L_μ via PDL-Δ. Our ECTL* translation can be considered a direct version of this algorithm. Note that from the existence of a translation into PDL-Δ and Niwinsky's result that PDL-Δ is strictly less expressive than L_μ, it follows that also ECTL* is strictly less expressive than L_μ.

Although the CTL*-translation of Section 6 is doubly exponential in the worst case we hope that its complexity will nonetheless turn out to be manageable in many practical situations. Many optimisations are possible to support this hope. By suitably encoding the conjuncts involved in the translation of internal nodes translated formulas can be represented in size $\mathcal{O}(n2^n)$. Note also that sufficient syntactic criteria for classifying internal nodes as least or greatest fixed-point nodes can easily be found: Suppose Φ contains a formula ϕ of one of the forms $\phi_1 \mathrm{U} \phi_2$ or $\mathrm{O}(\phi_1 \mathrm{U} \phi_2)$, and ϕ is not a subformula of any other $\phi' \in \Phi$. Then ϕ is regenerated along any $\rightarrow$-path $\mathrm{E}\Phi_0 \rightarrow \cdots \rightarrow \mathrm{E}\Phi_n$ for which $\Phi_0 = \Phi_n = \Phi$. We expect this in particular to cover a large number of applications, and where it applies the complexity can be cut to a single exponential.

An alternative to the use of automata in the syntax of ECTL* is to use a μ-calculus with basic modalities O and E. If fixed points are restricted such as to allow the formation of $\sigma X.\phi$ only when X does not occur within the scope of E in ϕ yet another version of ECTL* results. This follows by the equivalence of the ω-regular languages with the linear-time μ-calculus (cf. [23]). If the restriction concerning E is lifted the result, the full branching-time μ-calculus [24], is at least as expressive as L_μ. It is open whether the containment is strict.

Acknowledgment

Thanks to Colin Stirling for many valuable discussions and comments.

References

[1] M. Ben-Ari, A. Pnueli and Z. Manna, The temporal logic of branching time, *Acta Inform.* **20** (1983) 207–226.

[2] J.C. Bradfield and C.P. Stirling, Verifying temporal properties of processes, Lecture Notes in Computer Science, Vol. 458 (Springer, Berlin, 1990) 115–125; *Theoret. Comput. Sci.* **96** (1992) 157–174.

[3] M.C. Browne, E.M. Clarke, D.L. Dill and B. Mishra, Automatic verification of sequential circuits using temporal logic, *IEEE Trans. Comput.* **C-35** (1986).

[4] G. Bruns and S. Anderson, The formalization and analysis of a communications protocol, Tech. Report ECS-LFCS-91-137, University of Edinburgh, 1991.

[5] E.M. Clarke and E.A. Emerson, Design and synthesis of synchronisation skeletons using branching time temporal logic, Lecture Notes in Computer Science, Vol. 131 (Springer, Berlin, 1981) 52–71.

[6] E.M. Clarke, O. Grümberg and R.P. Kurshan, A synthesis of two approaches for verifying finite state concurrent systems, Manuscript, Carnegie-Mellon University, 1987.

[7] E.M. Clarke and B. Mishra, Automatic verification of asynchronous circuits, Lecture Notes in Computer Science, Vol. 164 (Springer, Berlin, 1983) 101–115.

[8] R. Cleaveland, Tableau-based model checking in the propositional mu-calculus, *Acta Inform.* **27** (1990) 725–747.

[9] R. Cleaveland, J. Parrow and B. Steffen, A semantics based verification tool for finite state systems, in: *Proc. 9th IFIP Symp. on Protocol Specification, Testing, and Verification* (North-Holland, Amsterdam, 1989).

[10] M. Dam, Translating CTL* into the modal μ-calculus, Tech. Report ECS-LFCS-90-123, University of Edinburgh, 1990.

[11] E.A. Emerson and J. Halpern "Sometimes" and "not never" revisited: on branching versus linear time, *J. ACM* **33** (1986) 151–178.

[12] E.A. Emerson and C.S. Jutla, The complexity of tree automata and logics of programs, in: *Proc. 29th Symp. on Foundations of Computer Science* (1988) 328–337.

[13] E.A. Emerson and C. Lei, Efficient model checking in fragments of the propositional mu-calculus, in: *Proc. 1st Ann. Symp. on Logic in Computer Science* (1986) 267–278.

[14] E.A. Emerson and A.P. Sistla, Deciding full branching time logic, *Inform. and Control* **61** (1984) 175–201.

[15] M.J. Fischer and R.E. Ladner, Propositional dynamic logic of regular programs, *J. Comput. System Sci.* **18** (1979) 194–211.

[16] D. Gabbay, A. Pnueli, S. Shelah and J. Stavi, On the temporal analysis of fairness, *Proc. 7th ACM Symp. on Principles of Programming Languages* (1980) 163–173.

[17] D. Gries, An exercise in proving parallel programs correct, *Commun. ACM* **20** (1977) 921–930.

[18] M. Hennessy and R. Milner, Algebraic laws for nondeterminism and concurrency, *J. ACM* **32** (1985) 137–162.

[19] D. Kozen, Results on the propositional μ-calculus, *Theoret. Comput. Sci.* **27** (1983) 333–354.

[20] O. Lichtenstein, A. Pnueli and L. Zuck, The glory of the past, Lecture Notes in Computer Science, Vol. 193 (1985) 97–107.

[21] R. McNaughton, Testing and generating infinite sequences by a finite automaton, *Inform. and Control* **9** (1966) 521–530.

[22] D. Niwinsky, Fixed points vs. infinite generation, in: *Proc. 3rd Ann. Symp. on Logic in Computer Science* (1988) 402–409.

[23] D. Park, Concurrency and automata on infinite sequences, Lecture Notes in Computer Science, Vol. 104 (Springer, Berlin, 1981) 167–183.

[24] C.P. Stirling, Modal and temporal logics, in: S. Abramsky, D. Gabbay and T. Maibaum, eds., *Handbook of Logic in Computer Science* (Oxford Univ. Press, Oxford, 1991).

[25] C.P. Stirling, and D.J. Walker, Local model checking in the modal mu-calculus, *Theoret. Comput. Sci.* **89** (1991) 161–177.

[26] R.S. Streett, Propositional dynamic logic of looping and converse is elementarily decidable, *Inform. and Control* **54** (1982) 121–141.

[27] R.S. Streett and E.A. Emerson, An automata theoretic decision procedure for the propositional mu-calculus, *Inform. and Comput.* **81** (1989) 249–264.

[28] W. Thomas, Star-free regular sets of ω-sequences, *Inform. and Control* **42** (1979) 148–156.

[29] W. Thomas, Computation tree logic and regular ω-languages, Lecture Notes in Computer Science, Vol. 354 (Springer, Berlin, 1988) 690–713.

[30] M.Y. Vardi and P. Wolper, Yet another process logic, Lecture Notes in Computer Science, Vol. 164 (Springer, Berlin, 1984) 501–512.

[31] D.J. Walker, Automated analysis of mutual exclusion algorithms using CCS, Tech. Report ECS-LFCS-89-91, University of Edinburgh, 1989.

[32] P. Wolper, A translation from full branching time temporal logic to one letter propositional dynamic logic with looping, unpublished manuscript.

[33] P. Wolper, Temporal logic can be more expressive, *Inform. and Control* **56** (1983) 72–99.

[34] P. Wolper, M.Y. Vardi and A.P. Sistla, Reasoning about infinite computation paths, in: *Proc. 24th IEEE Symp. on Foundations of Computer Science* (1983) 185–194.