

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

by

Mads Dam

LFCS Report Series

ECS-LFCS-92-217

LFCS

June 1992

Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ

Copyright © 1992, LFCS

**Copyright © 1992, Laboratory for Foundations of Computer Science
University of Edinburgh. All rights reserved.**

**Reproduction of all or part of this work
is permitted for educational or research use
on condition that this copyright notice is
included in any copy.**

CTL* and ECTL* as fragments of the modal μ -calculus¹

Mads Dam
Department of Computer Science
University of Edinburgh

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 (c.f. Gries [17], Clarke and Mishra [7], Browne et al [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 (Kozen [19]). This logic is obtained as an enrichment of a simple modal base logic, Hennessy-Milner logic (Hennessy, Milner [18]), by least and greatest fixpoints of formally monotone operators. The result is a very general branching-time temporal logic capable of expressing a wealth of properties related to for instance partial and total correctness, liveness, safety and fairness, that are of crucial importance in practical program verification (c.f. Bradfield, Stirling [2], Walker [31]). Indeed L_μ encompasses a great many well-known program logics such as PDL (Fischer, Ladner [15]), PDL- Δ (Streett [26]), linear-time temporal logic PTL (Gabbay et al [16]), CTL (Clarke, Emerson [5]), and CTL* (Emerson, Halpern [11]). This can be shown by providing constructive translations (c.f. Kozen [19], Emerson, Lei [13], Wolper [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” (c.f. Wolper [33]). Properties such as these are necessary in general for modular reasoning (Lichtenstein et al [20]). Despite this additional expressive power, L_μ is decidable in deterministic exponential time, and thus essentially is not harder than PDL (Emerson,

¹This work was supported by SERC Grant GR/F 32219.

Jutla [12], Fischer, Ladner [15]). Moreover, for closed formulas L_μ preserves the characterisation of bisimulation equivalence (Hennessy, Milner [18]) and thus provides a natural temporal logic for process calculi such as CCS (Stirling [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 (Cleaveland et al [9]) and used in several case-studies such as mutual exclusion algorithms (Walker [31]) and communication protocols (Bruns, Anderson [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. Most of the translations succeed very well in this: A first step towards a working understanding of L_μ is certainly to understand why the CTL-formula EFX is translated into $\mu Y.X \vee \diamond Y$.

In this respect CTL* is of particular interest. Beyond CTL, CTL* is capable of expressing properties such as $EGF\phi$ ("along some path ϕ holds infinitely often"), useful for dealing with fairness (Emerson, Halpern [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 (c.f. Emerson, Lei [13]). It involves 5 stages: The first and second stages builds a tableau and derives from it a deterministic Muller automaton (c.f. Emerson and Sistla [14]). Third stage derives from this automaton an equivalent ω -regular expression (c.f. McNaughton [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 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 fixpoints 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 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 (Wolper et al [34]), whereas PTL is capable of describing only the star-free ω -regular languages (Gabbay et al [16], Thomas [28]). Several expressively equivalent versions of extended CTL*, ECTL*, have been proposed (Vardi, Wolper [30], Clarke et al [6], Thomas [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 fixpoints 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, 5 and 6 describes 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 (1-letter) L_μ are built from propositional variables X, Y , boolean connectives \neg and \wedge , the modal nextstate quantifier \Diamond , and the least fixpoint 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 and R a binary transition relation on S . The semantics of the formula ϕ relative to T and a valuation $\mathcal{V} : X \mapsto B \subseteq S$ is the set $\|\phi\|\mathcal{V} \subseteq S$ defined as follows:

$$\begin{aligned} \|X\|\mathcal{V} &= \mathcal{V}(X) \\ \|\neg\phi\|\mathcal{V} &= \overline{\|\phi\|\mathcal{V}} \\ \|\phi \wedge \psi\|\mathcal{V} &= \|\phi\|\mathcal{V} \cap \|\psi\|\mathcal{V} \\ \|\Diamond\phi\|\mathcal{V} &= \{s \in S \mid \exists s' \in S. sRs' \text{ and } s' \in \|\phi\|\mathcal{V}\} \\ \|\mu X.\phi\|\mathcal{V} &= \bigcap \{B \subseteq S \mid \|\phi\|\mathcal{V}[X \mapsto B] \subseteq B\} \end{aligned}$$

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 fixpoints:

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

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

$$\begin{aligned} s \vdash \neg\neg\phi &\Rightarrow s \vdash \phi \\ s \vdash \phi_1 \wedge \phi_2 &\Rightarrow s \vdash \phi_i \text{ for } i = 1 \text{ and } i = 2 \\ s \vdash \phi_1 \vee \phi_2 &\Rightarrow s \vdash \phi_i \text{ for } i = 1 \text{ or } i = 2 \\ s \vdash \Box\phi &\Rightarrow s' \vdash \phi \text{ whenever } sRs' \\ s \vdash \Diamond\phi &\Rightarrow s' \vdash \phi \text{ for some } s' \text{ s.t. } sRs' \\ s \vdash \sigma X.\phi &\Rightarrow s \vdash \phi[\sigma X.\phi/X] \end{aligned}$$

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 fixpoints 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

$$s_1 \vdash \phi_1 \Rightarrow s_2 \vdash \phi_2 \Rightarrow \dots \Rightarrow s_n \vdash \phi_n$$

such that $n > 1$, $\phi_1 = \phi_n = \sigma X.\phi$, and $\sigma X.\phi$ is a subformula of ϕ_i for each $i : 1 \leq i \leq n$. Then \Rightarrow is *well-founded* if the regeneration relation is well-founded for each μ -formula. That is, there is no infinite path $sR^*s_0R^*s_1R^*s_2 \dots$ 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, Emerson [27]) *There is a well-founded choice relation \Rightarrow from $s \vdash \phi$ which agrees with \mathcal{V} iff $s \in \|\phi\|\mathcal{V}$. \square*

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 $A\phi \triangleq \neg E\neg\phi$. Other connectives are derived as usual. In particular $F\phi \triangleq \text{true}U\phi$ and $G\phi \triangleq \neg F\neg\phi$. Also we let $\phi_1\neg U\phi_2 \triangleq \neg(\phi_1U\phi_2)$. An *eventuality* is a formula of the form either $\phi U\psi$ or $O(\phi U\psi)$ (or $F\phi$ or $OF\phi$ if F is taken as primitive).

Formulas denote properties of infinite paths through a transition system $T = (S, R)$ with valuation \mathcal{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 i 'th 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 the following (c.f. Emerson and Halpern [11]):

$$\begin{aligned}
\rho \models_{\mathcal{V}} X &\text{ iff } \rho(i) \in \mathcal{V}(X) \\
\rho \models_{\mathcal{V}} \neg\phi &\text{ iff } \rho \not\models_{\mathcal{V}} \phi \\
\rho \models_{\mathcal{V}} \phi_1 \wedge \phi_2 &\text{ iff } \rho \models_{\mathcal{V}} \phi_1 \text{ and } \rho \models_{\mathcal{V}} \phi_2 \\
\rho \models_{\mathcal{V}} O\phi &\text{ iff } \rho^1 \models_{\mathcal{V}} \phi \\
\rho \models_{\mathcal{V}} \phi_1 U \phi_2 &\text{ iff } \exists i \in \omega \text{ such that } \rho^i \models_{\mathcal{V}} \phi_2 \text{ and } \forall j : 0 \leq j < i. \rho^j \models_{\mathcal{V}} \phi_1 \\
\rho \models_{\mathcal{V}} E\phi &\text{ iff } \exists \rho' \text{ such that } \rho(0) = \rho'(0) \text{ and } \rho' \models_{\mathcal{V}} \phi
\end{aligned}$$

Let $s \models_{\mathcal{V}} \phi$ iff for all paths ρ such that $\rho(0) = s$, $\rho \models_{\mathcal{V}} \phi$. We are particularly interested in formulas that depend only on the current state in the sense that $\rho \models_{\mathcal{V}} \phi$ iff $\rho(0) \models_{\mathcal{V}} \phi$. This is true, in particular, for boolean combinations of formulas that are either propositional variables or contains an outermost occurrence of the existential path-quantifier. Such formulas 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 (c.f. Wolper [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 the state formula ϕ is translated into the L_μ -formula ψ , and assume that the translation is correct in the sense that ϕ and ψ are semantically equivalent. That is, for all transition systems T , for all $s \in S_T$ and all valuations \mathcal{V} on T , $s \models_{\mathcal{V}} \phi$ iff $s \models \|\psi\|_{\mathcal{V}}$. Then the following four rules are validated:

$$\begin{array}{c}
X \rightsquigarrow X \quad \frac{\phi \rightsquigarrow \psi}{\neg\phi \rightsquigarrow \neg\psi} \quad \frac{\phi_1 \rightsquigarrow \psi_1 \quad \phi_2 \rightsquigarrow \psi_2}{\phi_1 \wedge \phi_2 \rightsquigarrow \psi_1 \wedge \psi_2} \\
\\
\frac{\phi_1 \rightsquigarrow \psi_1 \quad \dots \quad \phi_n \rightsquigarrow \psi_n \quad \phi \rightsquigarrow \psi}{\phi[\phi_1/X_1, \dots, \phi_n/X_n] \rightsquigarrow \psi[\psi_1/X_1, \dots, \psi_n/X_n]}
\end{array}$$

where $\gamma[\gamma_1/X_1, \dots, \gamma_n/X_n]$ 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 $E\phi$ where ϕ is a linear-time formula, i.e. ϕ does not contain occurrences of the existential path-quantifier E . 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 $E\Phi \triangleq E \wedge \Phi$ where Φ is a finite set of path-quantifier free formulas. 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 . To each rule is consequently associated one of these connectives, or possibly the identity operator I . The tableau rules are the following:

$$\begin{aligned}
I: & \frac{E(\Phi, \neg\neg\phi)}{E(\Phi, \phi)} & \wedge: & \frac{E(\Phi, X)}{E\Phi \quad X} & \wedge: & \frac{E(\Phi, \neg X)}{E\Phi \quad \neg X} \\
I: & \frac{E(\Phi, \phi \wedge \psi)}{E(\Phi, \phi, \psi)} & \vee: & \frac{E(\Phi, \phi \vee \psi)}{E(\Phi, \phi) \quad E(\Phi, \psi)} \\
\vee: & \frac{E(\Phi, \phi_1 U \phi_2)}{E(\Phi, \phi_2) \quad E(\Phi, \phi_1, O(\phi_1 U \phi_2))} \\
\vee: & \frac{E(\Phi, \phi_1 \neg U \phi_2)}{E(\Phi, \neg\phi_1, \neg\phi_2) \quad E(\Phi, \neg\phi_2, O(\phi_1 \neg U \phi_2))} \\
\diamond: & \frac{E(O\phi_1, \dots, O\phi_n)}{E(\phi_1, \dots, \phi_n)}
\end{aligned}$$

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

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

It is not difficult to see that tableaux are finite, and that all the tableau rules are forwards and backwards sound in the sense that if $\Omega: \frac{\phi}{\phi_1 \dots \phi_n}$ is an instance of any of the above rules, then $s \models_V \phi$ iff $s \models_V \Omega(\phi_1, \dots, \phi_n)$ where \diamond is interpreted as EO .

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 the notion of eventual satisfaction, and thus of admissibility, we analyse the way individual formulas are regenerated. Let $E\Phi_1 \rightarrow E\Phi_2$ if there is an edge from $E\Phi_1$ to $E\Phi_2$ in τ . Each transition $E\Phi_1 \rightarrow E\Phi_2$ determines the *generation relation* $\rightarrow_{E\Phi_1 \rightarrow E\Phi_2} \subseteq \Phi_1 \times \Phi_2$ in the following way where $\Delta(\Phi) = \{(\phi, \phi) \mid \phi \in \Phi\}$ is the diagonal relation on Φ :

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

$$\begin{aligned}
&\rightarrow_{E(\Phi,(-)X)\rightarrow E(\Phi)} = \Delta(\Phi) \\
&\rightarrow_{E(\Phi, \phi_1 \wedge \phi_2)\rightarrow E(\Phi, \phi_1, \phi_2)} = \{(\phi_1 \wedge \phi_2, \phi_1), (\phi_1 \wedge \phi_2, \phi_2)\} \cup \Delta(\Phi) \\
&\rightarrow_{E(\Phi, \phi_1 \vee \phi_2)\rightarrow E(\Phi, \phi_i)} = \{(\phi_1 \vee \phi_2, \phi_i)\} \cup \Delta(\Phi), i \in \{1, 2\} \\
&\rightarrow_{E(\Phi, \phi U \psi)\rightarrow E(\Phi, \psi)} = \{(\phi U \psi, \psi)\} \cup \Delta(\Phi) \\
&\rightarrow_{E(\Phi, \phi U \psi)\rightarrow E(\Phi, \phi, O(\phi U \psi))} = \{(\phi U \psi, \phi), (\phi U \psi, O(\phi U \psi))\} \cup \Delta(\Phi) \\
&\rightarrow_{E(\Phi, \phi \neg U \psi)\rightarrow E(\Phi, \neg \phi, \neg \psi)} = \{(\phi \neg U \psi, \neg \phi), (\phi \neg U \psi, \neg \psi)\} \cup \Delta(\Phi) \\
&\rightarrow_{E(\Phi, \phi \neg U \psi)\rightarrow E(\Phi, \neg \psi, O(\phi \neg U \psi))} = \{(\phi \neg U \psi, \neg \psi), (\phi \neg U \psi, O(\phi \neg U \psi))\} \cup \Delta(\Phi) \\
&\rightarrow_{E(O\phi_1, \dots, O\phi_n)\rightarrow E(\phi_1, \dots, \phi_n)} = \{(O\phi_i, \phi_i) \mid 1 \leq i \leq n\}
\end{aligned}$$

We usually abbreviate $\rightarrow_{E\Phi_1 \rightarrow E\Phi_2}$ by \rightarrow when the transition $E\Phi_1 \rightarrow E\Phi_2$ is understood from the context.

Example 5.1 Consider the transition $E(O(\phi U \psi), \phi U \psi) \rightarrow E(O(\phi U \psi), \psi)$. In this case $O(\phi U \psi) \rightarrow O(\phi U \psi)$ and $\phi U \psi \rightarrow \psi$, but *not* $\phi U \psi \rightarrow O(\phi U \psi)$, as relative to the given transition $O(\phi U \psi)$ is not generated by $\phi U \psi$.

We use Π for \rightarrow -paths and π for \rightarrow -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, $\pi(i-1) \rightarrow \pi(i)$ 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 $\pi(i) = \phi U \psi$ or $\pi(i) = F\psi$ then for some $j > i$, $\pi(j) = \psi$. Suppose that Π visits the node $E\Phi$ infinitely often. Consider a segment $\Pi(i_0, i_k) \triangleq \Pi(i_0), \dots, \Pi(i_k)$ of Π for which $\Pi(i_0) = \Pi(i_k) = E\Phi$ 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 $\pi(0) = \pi(i_k - i_0) = \phi$.

Example 5.2 Here and below we use a tree-like notation for tableaux, terminating the construction as soon as nodes are first repeated.

- (i) Fig. 1 shows a tableau for $E(XUY)$. The eventuality XUY is regenerated along the \rightarrow -path from the root to its repetition.
- (ii) Fig. 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 $E\Phi$ of fig. 3 expresses that both X and Y holds 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).

$$\vee \frac{Y}{\wedge \frac{X}{\diamond \frac{E(XUY)}{E(X, O(XUY))}}}}{\frac{E(XUY)}{E(O(XUY))}}}$$

Figure 1: Tableau for $E(XUY)$

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

Figure 2: Tableau for $E(OFX, OGOFX)$

$$\vee \frac{\diamond \frac{E\Phi}{E(GOFX, GOFY, FX, FY)}}{E(\Phi, FX, FY)}}{\frac{E(\Phi, X, FY)}{E\Phi^{(1)} \quad X \quad Y} \quad \frac{E(\Phi, X)}{E\Phi^{(2)} \quad X} \quad \vee \quad \frac{E(\Phi, Y)}{E\Phi^{(3)} \quad Y} \quad E\Phi^{(4)}}$$

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

The following characterisation of the admissible \rightarrow -paths is of central importance to the translation procedure. Let a *simple node* be any node $E\Phi$ 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 5.3 *Let Π be an infinite \rightarrow -path through a tableau τ , and let i_0, i_1, \dots be an infinite, strictly increasing sequence such that $\Pi(i_j)$ is the same simple node $E\Phi$ 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 \rightarrow -path $\pi \in \Pi$, a $k \in \omega$, and an eventuality $\phi = \phi_1 U \phi_2$, say, such that for all $k' \geq k$, either $\pi(k') = \phi$ or $\pi(k') = O\phi$. As $E\Phi$ is a simple node it follows that either for all j such that $i_j \geq k$, $\pi(i_j) = \phi$, or for all j with $i_j \geq k$, $\pi(i_j) = O\phi$. In any case either ϕ is regenerated along every segment $\Pi(i_j, i_{j+1})$ for which $i_j \geq k$, or else $O\phi$ is. \square

The restriction to simple nodes in Lemma 5.3 is indeed necessary. To see this let Π_1 be the \rightarrow -path

$$\begin{aligned} E(OGFX, FX, OFX, OGOFX) &\rightarrow E(OGFX, OFX, OGOFX) \\ &\rightarrow E(GFX, FX, GOFX) \\ &\rightarrow E(GFX, OFX, GOFX) \\ &\rightarrow E(OGFX, FX, OFX, GOFX) \\ &\rightarrow E(OGFX, FX, OFX, OGOFX) \end{aligned}$$

and let Π_2 be the \rightarrow -path

$$\begin{aligned} E(OGFX, FX, OFX, OGOFX) &\rightarrow E(OGFX, OFX, OGOFX) \\ &\rightarrow E(GFX, FX, GOFX) \\ &\rightarrow E(GFX, FX, OFX, OGOFX) \\ &\rightarrow E(OGFX, FX, OFX, OGOFX) \end{aligned}$$

Notice that FX is not regenerated along Π_1 , and that OFX is not regenerated along Π_2 . Consider the infinite \rightarrow -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, \dots the largest sequence such that $\Pi(i_j) = E(OGFX, FX, OFX, OGOFX)$. Clearly Π is not admissible, but both eventualities FX and OFX fails to be regenerated along infinitely many segments $\Pi(i_j, i_{j+1})$.

6 Translating CTL*

The structure of the translation is determined by the labelling of rule instances, and least or greatest fixpoints are used to characterise the admissible \rightarrow -paths.

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 fixpoint quantifiers to bind propositional variables. These variables are then used in the second pass, if it applies. Let $\phi = E\Phi$. 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

$$\nu X_\Phi.\psi(X_\Phi)$$

where ϕ in the second pass is translated into X_Φ . In general, however, by Lemma 5.3, each eventuality $\phi \in \Phi$ must infinitely often be prevented from being regenerated between subsequent visits to ϕ by Π . This suggests

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

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

To make the notion of pass and segment traversal precise the translation of a node ϕ is really relativised to a path segment $\Pi = \phi_0, \dots, \phi_n$ for which $\phi_n \rightarrow \phi$. We write $\Pi \vdash \phi$ for the relativisation of ϕ to Π . In addition as the translation progresses an environment ε is carried along which when $\phi = E\Phi$ choses an eventuality $\varepsilon(\Phi) \in \Phi$, provided one exists, intended to be significant when considering the regeneration of eventualities as above. Relative to Π and ε the node ϕ is translated into $tr(\Pi \vdash \phi)\varepsilon$. When the segment Π is understood from the context we often abbreviate $tr(\Pi \vdash \phi)\varepsilon$ as $tr(\phi)\varepsilon$. For leaves the translation is trivial:

$$tr(\Pi \vdash (\neg)X)\varepsilon = (\neg)X. \quad (1)$$

Let then $\phi = E\Phi$. Suppose first that $\Pi(i) = \phi$ for some $i : 1 \leq i \leq n$ so that ϕ is currently being passed the second time. If either $\varepsilon(\Phi)$ is undefined (because there are no eventualities in Φ), or $\varepsilon(\Phi)$ is not regenerated along the segment $\Pi(i), \dots, \Pi(n)$, ϕ let

$$tr(\Pi \vdash \phi)\varepsilon = X_\Phi \quad (2)$$

and otherwise let

$$tr(\Pi \vdash \phi)\varepsilon = Y_{\varepsilon(\Phi), \Phi}. \quad (3)$$

Assume then that $\Pi(i) \neq \phi$ for all $i : 1 \leq i \leq n$, and let the rule instance applied to ϕ be $\Omega : \frac{\phi}{\phi_1, \dots, \phi_m}$. First, if ϕ is not a simple node or ϕ is not reachable from any of the nodes ϕ_1, \dots, ϕ_m let

$$tr(\Pi \vdash \phi)\varepsilon = \Omega(tr(\Pi, \phi \vdash \phi_1)\varepsilon, \dots, tr(\Pi, \phi \vdash \phi_m)\varepsilon). \quad (4)$$

Secondly, if there are no eventualities in Φ let

$$tr(\Pi \vdash \phi)\varepsilon = \nu X_\Phi.\Omega(tr(\Pi, \phi \vdash \phi_1)\varepsilon, \dots, tr(\Pi, \phi \vdash \phi_m)\varepsilon). \quad (5)$$

Finally, in any other case,

$$\begin{aligned} \text{tr}(\Pi \vdash \phi)\varepsilon &= \nu X_{\Phi}. \bigwedge_{\text{eventualities } \psi \in \Phi} \\ &\mu Y_{\psi, \Phi}. \Omega(\text{tr}(\Pi, \phi \vdash \phi_1)\varepsilon[\Phi \mapsto \psi], \dots, \text{tr}(\Pi, \phi \vdash \phi_m)\varepsilon[\Phi \mapsto \psi]). \end{aligned} \quad (6)$$

For the root node ϕ_0 let then $\phi_0 \rightsquigarrow \psi_0$ if for some tableau rooted in ϕ_0 , $\psi_0 = \text{tr}(\phi_0)()$ where $()$ is the empty environment, and ϕ_0 is relativised to the empty path segment.

Example 6.1 (i) From the tableau of fig. 1 the expected translation obtains:

$$E(XUY) \rightsquigarrow \mu Y'. Y \vee (X \wedge \diamond Y').$$

(ii) For the tableau of fig. 2:

$$E(OFX, OGOFX) \rightsquigarrow \nu X'. \mu Y'. \diamond((X \wedge X') \vee Y').$$

The example serves to illustrate why the translation schema uses ν - μ -alternation. For comparison two possible translations of the (equivalent) formula $EGFX$ 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 of (6) is in general necessary. It is translated thus:

$$\begin{aligned} E\Phi &\rightsquigarrow \nu X'. (\mu Y'. \diamond(X' \wedge X \wedge Y) \vee (Y' \wedge X) \vee (X' \wedge Y) \vee Y') \\ &\quad \wedge (\mu Y'. \diamond(X' \wedge X \wedge Y) \vee (X' \wedge X) \vee (Y' \wedge Y) \vee Y'). \end{aligned}$$

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

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

PROOF: See section 7. □

7 Proof of Theorem 6.2

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

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

(ii) If $s_0 \not\models_{\mathcal{V}} \phi_0$ 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 \mathcal{V} , and then showing \Rightarrow to be well-founded.

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

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

The construction of \Rightarrow proceed 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 $\phi = E\Phi$ and sequent $s \vdash (\neg)\psi$ the following properties are maintained invariant:

- (a) $s \models_{\mathcal{V}} (\neg)\phi$, and
- (b) for some ε and Π , $\psi = \text{unf}(tr(\Pi \vdash \phi)\varepsilon)$.

Note that, depending on Π , $tr(\Pi \vdash \phi)\varepsilon$ is either the variable X_{Φ} , or the variable $Y_{\varepsilon(\phi), \Phi}$, or else $tr(\Pi \vdash \phi)\varepsilon = \sigma X_{\Phi}$. We show how to complete stage $k+1$. For leaves the construction of \Rightarrow is complete. Assume then that Ω labels the rule instance $\frac{\phi}{\phi_1, \dots, \phi_m}$, that ϕ is simple, that ϕ is reachable from one of the nodes ϕ_1, \dots, ϕ_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 $s \models_{\mathcal{V}} \phi$ 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 *toplevel eventualities* of ϕ : the subformulas of ϕ of the form $\phi_1 U \phi_2$ that do not occur within the scope of $\neg U$ in ϕ . The predecessor of ξ is the index $\text{pred}(\xi)$ defined by $\text{pred}(\xi)(\phi_1 U \phi_2) = \xi(\phi_1 U \phi_2) - 1$ when $\xi(\phi_1 U \phi_2) > 0$ and $\text{pred}(\xi)(\phi_1 U \phi_2) = 0$ otherwise. Then $\phi[\xi]$ indexes each member of Φ in the following way:

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

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

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

$$\begin{aligned} \phi_1 U^0 \phi_2 &= \phi_2 \\ \phi_1 U^{n+1} \phi_2 &= \phi_2 \vee (\phi_1 \wedge O(\phi_1 U^n \phi_2)). \end{aligned}$$

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

The construction of \Rightarrow now proceeds as follows: First the rules for fixpoints (and, if necessary, conjunction) are applied to $s \vdash \psi$ until a sequent of the form $s \vdash \Omega(\psi_1, \dots, \psi_m)$ is reached. The construction now depends on the tableau rule applied.

- (1) Double negation: $\Phi = \Phi', \neg\neg\phi'$. Then $s \models_{\nu} E(\Phi', \phi')[\xi]$. 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 $s \models_{\nu} E\Phi'[\xi]$, $\Omega = \wedge$, $\psi_2 = (\neg)X$, and we proceed from $s \vdash \psi_1$.
- (3) Conjunction: $\Phi = \Phi', \phi'_1 \wedge \phi'_2$. Then $s \models_{\nu} E(\Phi', \phi'_1, \phi'_2)[\xi]$, $\Omega = I$ and we proceed from $s \vdash \psi_1$.
- (4) Disjunction: $\Phi = \Phi', \phi'_1 \vee \phi'_2$. Here $s \models_{\nu} E(\Phi', \phi'_i)[\xi']$ for either $i = 1$ or $i = 2$, where ξ' is ξ restricted to the toplevel eventualities of Φ', ϕ'_i . Also $\Omega = \vee$, and we proceed from $s \vdash \psi_i$.
- (5) Until: $\Phi = \Phi', \phi'_1 U \phi'_2$. We know that either $s \models_{\nu} E(\Phi', \phi'_2)[\xi']$, or else $s \models_{\nu} E(\Phi', \phi'_1, O(\phi'_1 U \phi'_2))[\xi']$, 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”: $\Phi = \Phi', \phi'_1 \neg U \phi'_2$. Either $s \models_{\nu} E(\Phi', \neg\phi'_1, \neg\phi'_2)[\xi']$ or $s \models_{\nu} E(\Phi', \neg\phi'_2, O(\phi'_1 \neg U \phi'_2))[\xi']$ where in either case ξ' is an index agreeing with ξ on their common toplevel eventualities. Again $\Omega = \vee$ and the choice of ψ_i is guided as in (6).
- (7) Nexttime: $\Phi = O\phi'_1, \dots, O\phi'_n$. Then there is some s' such that $s R s'$ and $s \models_{\nu} E(\phi'_1, \dots, \phi'_n)[pred(\xi)]$. 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 in equation (6). 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 $tr(\Pi \vdash \phi)\varepsilon$ is identical to either $\sigma X_{\Phi'}$ or $X_{\Phi'}$. Note that in this case $\neg\psi$ is a formula of the form

$$\mu X_{\Phi'} \cdot \vee_{\text{eventualities } \phi' \in \Phi'} \nu Y_{\phi', \Phi'} \cdot \neg\Omega(\dots). \quad (7)$$

The sequence of actions is as follows: The scheduler is updated, the fixpoint unfolding rule is applied to $\neg\psi$, the scheduled disjunct

$$\neg\psi' = \nu Y_{ev(f_{\Phi'}), \Phi'} \cdot \neg\Omega(\dots) \quad (8)$$

is chosen, and the fixpoint unfolding rule is applied to $\neg\psi'$. If on the other hand $tr(\Pi \vdash \phi)\varepsilon = Y_{\varepsilon(\Phi'), \Phi}$ then $\neg\psi$ has the form (8) already and we merely apply the

fixpoint unfolding rule to $\neg\psi$. In either case a formula of the form $\neg\Omega(\psi_1, \dots, \psi_m)$ 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 \mathcal{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, \dots 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 ϕ_0, ϕ_1, \dots through τ . Assume that each ϕ_i has the form $E\Phi_i$, and that the suffix of ρ corresponding to the \rightarrow -path $\phi_i, \phi_{i+1}, \dots$ is ρ_i .

(i). In this case we find some $\phi = E\Phi$ such that $\mu X.\psi$ is of the form $\text{unf}(tr(\Pi \vdash \phi)\varepsilon)$ where $tr(\Pi \vdash \phi)\varepsilon = Y_{\varepsilon(\Phi), \Phi}$. 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 $\rho_i \models_{\mathcal{V}} \phi'_i[\xi_i]$.

- (1) $\phi'_i = (\neg)X$. Here $\rho_i(0) \in \mathcal{V}(X)$ ($\rho_i(0) \notin \mathcal{V}(X)$) as \Rightarrow agrees with \mathcal{V} .
- (2) $\phi'_i = \neg\neg\phi_{i,1}$. By the induction hypothesis $\rho_i \models_{\mathcal{V}} \phi_{i,1}[\xi_i]$ so also $\rho_i \models_{\mathcal{V}} \phi'_i[\xi_i]$.
- (3) $\phi'_i = \phi_{i,1} \wedge \phi_{i,2}$. By the induction hypothesis, $\rho_i \models_{\mathcal{V}} \phi_{i,j}[\xi_i]$ for both $j = 1$ and $j = 2$, so $\rho_i \models_{\mathcal{V}} \phi'_i[\xi_i]$.
- (4) $\phi'_i = \phi_{i,1} \vee \phi_{i,2}$. By the induction hypothesis, $\rho_i \models_{\mathcal{V}} \phi_{i,j}[\xi_i]$ for either $j = 1$ or $j = 2$, so $\rho_i \models_{\mathcal{V}} \phi'_i[\xi_i]$.
- (5) $\phi'_i = \phi_{i,1}U\phi_{i,2}$. By the construction of \Rightarrow we obtain a smallest $i' \geq i$ such that $\phi_{i,2} \in \Phi_{i'}$ whence by the induction hypothesis, $\rho_{i'} \models_{\mathcal{V}} \phi_{i,2}[\xi_{i'}]$. Moreover for all $i'' : i \leq i'' < i'$, $\phi_{i,1} \in \Phi_{i''}$, so $\rho_{i''} \models_{\mathcal{V}} \phi_{i,1}[\xi_{i''}]$, so indeed $\rho_i \models_{\mathcal{V}} \phi'_i[\xi_i]$.
- (6) $\phi'_i = \phi_{i,1}\neg U\phi_{i,2}$. We either obtain a smallest $i' \geq i$ such that $\neg\phi_{i,1}, \neg\phi_{i,2} \in \Phi_{i'}$, so $\rho_{i'} \models_{\mathcal{V}} \neg\phi_{i,j}[\xi_{i'}]$ for both $j = 1$ and $j = 2$. Moreover for all $i'' : i \leq i'' < i'$, $\neg\phi_{i,2} \in \Phi_{i''}$, so $\rho_{i''} \models_{\mathcal{V}} \neg\phi_{i,2}[\xi_{i''}]$, so indeed in this case $\rho_i \models_{\mathcal{V}} \phi'_i[\xi_i]$. Alternatively for all $i' \geq i$ $\neg\phi_{i,2} \in \Phi_{i'}$, and using the induction hypothesis we can conclude that $\rho_i \models_{\mathcal{V}} \phi'_i[\xi_i]$.
- (7) $\phi'_i = O\phi_{i,1}$. By the induction hypothesis, $\rho_{i+1} = \rho_i^1 \models \phi_{i,1}[\xi_{i+1}]$ and $\xi_{i+1} = \text{pred}(\xi_i)$. Hence $\rho_i \models \phi'_i[\xi_i]$.

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 \rightarrow -path π relative to the given infinite

Π -path through τ which from some point onwards always has one of the forms $\phi_1 U \phi_2$ or $O(\phi_1 U \phi_2)$ for fixed ϕ_1, ϕ_2 . But this contradicts the indexing strategy.

(ii). Here $\mu X.\psi$ has the form $\neg \text{unf}(tr(\Pi \vdash \phi)\varepsilon)$ where either $tr(\Pi \vdash \phi)\varepsilon = X_\Phi$ or $tr(\Pi \vdash \phi)\varepsilon = \sigma X_\Phi$. Completing the proof, we show by induction on the size of formulas that $\rho_i \models_{\mathcal{V}} \phi'_i$ whenever $\phi'_i \in \Phi_i$, contradicting the assumption that $\rho_i(0) \models \neg \phi_i$. The only slightly difficult case is for $\phi'_i = \phi_{i,1} U \phi_{i,2}$. As $\mu X.\psi$ has the form (7) it cannot for all $i' \geq i$ be the case that $\phi_{i,2} \notin \Phi_{i'}$. For otherwise, by Lemma 5.3, for all $i' \geq i$ such that $\Phi_{i'} = \Phi_i$, the eventuality ϕ'_i would be regenerated along the \rightarrow -derivation from $E\Phi_i$ to $E\Phi_{i'}$. Moreover the scheduling mechanism ensures that a disjunct of the form

$$\nu Y_{\gamma, \phi_i}. \neg \Omega(\dots)$$

is eventually visited for some $i' \geq i$ 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 $i'' : i \leq i'' < i'$, $\phi_{i,1} \in \Phi_{i''}$. But then by the induction hypothesis, $\rho_{i'} \models_{\mathcal{V}} \phi_{i,2}$ and $\rho_{i''} \models_{\mathcal{V}} \phi_{i,1}$ whenever $i \leq i'' < i'$, i.e. $\rho_i \models_{\mathcal{V}} \phi_i$.

8 Extended Computation Tree Logic, ECTL*

Several natural extensions of CTL* merit consideration. One direction of pragmatic interest is to add nexttime operators indexed by labels or sets of labels as in Bradfield, Stirling [2]. This direction is pursued in Dam [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 $\mathcal{A} = (Q, q_0, \{\xrightarrow{a}\}_{a \in \Sigma}, F)$ 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 $\mathcal{A}(q_0)$ to emphasize the initial state q_0 . A *run* of \mathcal{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 \mathcal{A} *accepts* α , if there is a run r of \mathcal{A} on α and a $q \in F$ s.t. $r(i) = q$ for infinitely many i , and the language *recognised* by \mathcal{A} is $\mathcal{L}(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}$.

Formulas of ECTL* are inductively defined as usual and built from propositional variables using boolean connectives and containing the formula $E(\mathcal{A})$ for each Büchi automaton \mathcal{A} over an alphabet $2^{\{\phi_1, \dots, \phi_n\}}$ 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:

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

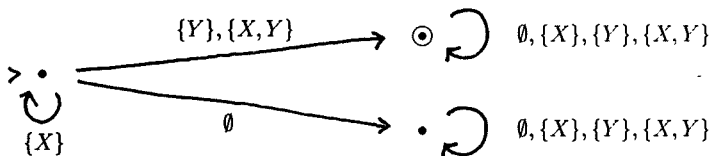


Figure 4: Büchi automaton \mathcal{A}_1 for XUY .

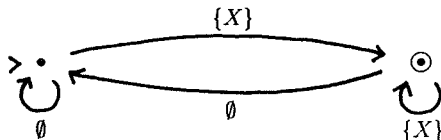


Figure 5: Büchi automaton \mathcal{A}_2 for GFX

$$s \models_{\nu} \neg\phi \text{ iff } s \not\models_{\nu} \phi$$

$$s \models_{\nu} \phi_1 \wedge \phi_2 \text{ iff } s \models_{\nu} \phi_1 \text{ and } s \models_{\nu} \phi_2$$

Let then \mathcal{A} be a Büchi automaton over the alphabet $2^{\{\phi_1, \dots, \phi_n\}}$. The intuition is that $E(\mathcal{A})$ is true of a state s just in case there is an infinite path ρ originating in s s.t. the ω -word over $2^{\{\phi_1, \dots, \phi_n\}}$ that encodes the holding of ϕ_1, \dots, ϕ_n along ρ is in the language recognised by \mathcal{A} . More precisely let the word $\alpha(\rho)$ be determined by

$$\alpha(\rho)(i) = \{\phi_j \mid 1 \leq j \leq n, \rho(i) \models_{\nu} \phi_j\}$$

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

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

Example 8.1 The ECTL* formula $E(\mathcal{A}_1)$ where \mathcal{A}_1 is the Büchi automaton of fig. 4 expresses the CTL* property $E(XUY)$. Similarly the ECTL* formula $E(\mathcal{A}_2)$ expresses $EGFX$ (infinitely often X) where \mathcal{A}_2 is the automaton of fig. 5. \square

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 that in particular the standard derivation of the universal path quantifier require complementation of Büchi automata. In this respect the present account sacrifice clarity to some extent. This can be remedied by using for instance deterministic Muller automata as in (Clarke et al [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 $E(\mathcal{A})$ in a standard form where \mathcal{A} is an automaton over sets of propositional variables only, and for which F is a singleton set. For let $F = \{q_1, \dots, q_m\}$ and let each \mathcal{A}_i be obtained from \mathcal{A} by replacing F by $\{q_i\}$. Then in addition to the four rules of section 4 the following rule is validated

$$\frac{E(\mathcal{A}_1) \rightsquigarrow_e \psi_1 \quad \dots \quad E(\mathcal{A}_m) \rightsquigarrow_e \psi_m}{E(\mathcal{A}) \rightsquigarrow_e \psi_1 \vee \dots \vee \psi_m}$$

for the ECTL* correlate \rightsquigarrow_e of \rightsquigarrow .

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

$$\Omega: \frac{E(\mathcal{A}(q))}{E(\mathcal{A}(q_{1,1})) \dots E(\mathcal{A}(q_{1,k_1})) \dots E(\mathcal{A}(q_{m,1})) \dots E(\mathcal{A}(q_{m,k_m}))}$$

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

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

As we only consider tableau constructed from nodes in standard form, there is at most one node $E(\mathcal{A}(q))$ 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 can be done by resetting the “visited-table” whenever the accepting state is first encountered.

For the definition of the translation function tr_e assume for each state q a unique variable X_q . The environment $\varepsilon \subseteq Q_{\mathcal{A}}$ keeps track of visited states. First, if $q \in \varepsilon$ then $tr_e(E(\mathcal{A}(q)))\varepsilon = X_q$. If not, let Ω label the rule instance $\frac{E(\mathcal{A}(q))}{E(\mathcal{A}(q_1)) \dots E(\mathcal{A}(q_k))}$. The following schema applies:

$$tr_e(E(\mathcal{A}(q)))\varepsilon = \sigma X_q \cdot \Omega(tr_e(E(\mathcal{A}(q_1)))\varepsilon', \dots, tr_e(E(\mathcal{A}(q_k)))\varepsilon'). \quad (9)$$

There are two cases: If q is the accepting state then $\sigma = \nu$ and $\varepsilon' = \{q\}$. That is, all previously encountered states are set to “unvisited”. If on the other hand q is not accepting the obvious strategy applies: $\sigma = \mu$ and $\varepsilon' = \varepsilon \cup \{q\}$. As before we then let $\phi \rightsquigarrow_e \psi$ whenever $\psi = tr_e(\phi)\emptyset$.

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

PROOF: Let a tableau τ for $\phi_0 = E(\mathcal{A}(q_0))$ be given where $\mathcal{A}(q_0)$ is in standard form. Let $\psi_0 = tr_e(\phi_0)\emptyset$. Note that due to the way environments are reset when the accepting state q is first visited by the translation procedure the only ν -subformula of ψ_0 is the formula $tr_e(E(\mathcal{A}(q)))\emptyset$. By Theorem 2.1 it suffices to show that $s_0 \models_{\mathcal{V}} \phi_0$ iff there is a well-founded choice relation \Rightarrow from $s_0 \vdash \psi_0$ which agrees with \mathcal{V} .

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

$$\Delta = s_0 \vdash \psi_0 \Rightarrow s_1 \vdash \psi_1 \Rightarrow \dots$$

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 , $\psi_i = tr_e(E(\mathcal{A}(q)))\emptyset$, and the result follows. Conversely assume that r visits q infinitely often. Then there is an infinite, strictly increasing sequence i_0, i_1, \dots such that for all $j \in \omega$ is $\psi_{i_j} = tr_e(E(\mathcal{A}(q)))\emptyset$. Moreover ψ_{i_0} is a subformula of each ψ_k for all $k \geq i_0$. 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 $\mu X.\phi = tr_e(E(\mathcal{A}(q)))\emptyset$, a contradiction. It follows that \Rightarrow is well-founded. \square

Example 9.2 The formula $E(\mathcal{A}_2)$ of example 8.1 is translated into the formula

$$\mu X'.(-X \wedge \diamond X') \vee (X \wedge \diamond \phi)$$

where ϕ is determined in the following way:

$$\begin{aligned} \phi &= \nu Y'.(-X \wedge \diamond \psi) \vee (X \wedge \diamond Y') \\ \psi &= \mu X'.(-X \wedge \diamond X') \vee (X \wedge \diamond Y') \end{aligned}$$

Note that without resetting the “visited-table” ϕ becomes instead

$$\phi = \nu Y'.(-X \wedge \diamond X') \vee (X \wedge \diamond Y')$$

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

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 (c.f. Thomas [29]), and Niwinsky [22] shows S2S to be expressively equivalent to a μ -calculus with, in effect, a left and right nexttime 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 fixpoint nodes can easily be found: Suppose Φ contains a formula ϕ of one of the forms $\phi_1 U \phi_2$ or $O(\phi_1 U \phi_2)$, and ϕ is not a subformula of any other $\phi' \in \Phi$. Then ϕ is regenerated along any \rightarrow -path $E\Phi_0 \rightarrow \dots \rightarrow E\Phi_n$ 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 fixpoints 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 (c.f. Park [23]). If the restriction concerning E is lifted the result, the full branching-time μ -calculus (Stirling [24]), is at least as expressive as L_μ . It is open whether the containment is strict.

Acknowledgements: 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 Informatica* **20** (1983) 207–226.
- [2] J. C. Bradfield and C. P. Stirling. "Verifying temporal properties of processes," *Lecture Notes in Computer Science* **458** (Springer,1990) pp. 115–125. To appear in *Theoretical Computer Science*.

- [3] M. C. Browne, E. M. Clarke, D. L. Dill and B. Mishra. "Automatic verification of sequential circuits using temporal logic," *IEEE Transactions on Computers* **C-35** (1986).
- [4] G. Bruns and S. Anderson. "The formalization and analysis of a communications protocol," Tech. rep. 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* **131** (1981) pp. 52-71.
- [6] E. M. Clarke, O. Grumberg 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* **164** (1983) 101-115.
- [8] R. Cleaveland. "Tableau-based model checking in the propositional mu-calculus." *Acta Informatica* **27** (1990) 725-747.
- [9] R. Cleaveland, J. Parrow and B. Steffen. "A semantics based verification tool for finite state systems," *Proc. 9th IFIP Symp. on Protocol Specification, Testing, and Verification* North-Holland, 1989.
- [10] M. Dam. "Translating CTL* into the modal μ -calculus," Tech. rep. 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." *Journal of the ACM* **33** (1986) 151-178.
- [12] E. A. Emerson and C. S. Jutla. "The complexity of tree automata and logics of programs," *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," *Information and Control* **61** (1984) pp. 175-201.
- [15] M.J. Fischer and R.E. Ladner. "Propositional dynamic logic of regular programs," *Journal of Computer and System Science* **18** pp. 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," *Communications of the ACM* **20** (1977) pp. 921–930.
- [18] M. Hennessy and R. Milner. "Algebraic laws for nondeterminism and concurrency," *Journal of the ACM* **32** (1985) 137–162.
- [19] D. Kozen. "Results on the propositional μ -calculus," *Theoretical Computer Science* **27** (North-Holland, 1983) 333–354.
- [20] O. Lichtenstein, A. Pnueli and L. Zuck. "The glory of the past," *Lecture Notes in Computer Science* **193** (1985) 97–107.
- [21] R. McNaughton. "Testing and generating infinite sequences by a finite automaton," *Information 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* **104** (1981) 167–183.
- [24] C. P. Stirling. "Modal and temporal logics," to appear in: *Handbook of Logic in Computer Science* (S. Abramsky, D. Gabbay, T. Maibaum, eds.) Oxford University Press (1991).
- [25] C. P. Stirling and D. J. Walker. "Local model checking in the modal μ -calculus," *Theoretical Computer Science* **89** (1991) 161–177.
- [26] R. S. Streett. "Propositional dynamic logic of looping and converse is elementarily decidable," *Information and Control* **54** (Academic Press, 1982) pp. 121–141.
- [27] R.S. Streett and E.A. Emerson. "An automata theoretic decision procedure for the propositional μ -calculus," *Information and Computation* **81** (1989) 249–264.
- [28] W. Thomas. "Star-free regular sets of ω -sequences," *Information and Control* **42** (1979) 148–156.
- [29] W. Thomas. "Computation tree logic and regular ω -languages," *Lecture Notes in Computer Science* **354** (1988) 690–713.
- [30] M. Y. Vardi and P. Wolper. "Yet another process logic," *Lecture Notes in Computer Science* **164** (1984) 501–512.
- [31] D. J. Walker. "Automated analysis of mutual exclusion algorithms using CCS," Tech. rep. 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," *Information 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.