TEMPORAL REASONING UNDER GENERALIZED FAIRNESS CONSTRAINTS[1]
(Extended Abstract)

E. Allen EMERSON and Chin-Laung LEI

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712
U.S.A.

1. Introduction

A variety of systems of temporal logic have been proposed for use in the design of correct concurrent programs (cf. [PN77], [PA78], [PR79], [EC80], [BMP81], [EC82], [OL82]). These systems offer differing modalities and notation. However, we can usually classify them into two groups: linear time logics (LTLs) and branching time logics (BTLs) (cf. [LA80], [EH83]). In an LTL, the temporal modalities are defined with respect to a single path which intuitively represents the future that the program will actually follow. Typical linear time operators include G ("always"), F ("sometime"), X ("nexttime"), and U ("until"). When LTLs are used to specify correctness properties of a concurrent program, there is an implicit universal quantification over all possible futures that could turn out to be the actual future. In contrast, branching time modalities are true or false of a state and are usually of the general form A ("for all futures") or E ("for some future") followed by some combination of linear time operators thus allowing explicit quantification over possible futures starting in the state. Some basic branching time operators are AFp ("for all futures, sometime p"), EFp ("for some future, sometime p"), and AGp ("for all futures, always p "). The first two properties (often called *inevitability* and *potentiality*, resp.) are *eventuality* properties which assert that something will eventually happen while the last property is an *invariance* property.

Two important characteristics of a temporal logic are (1) its expressive power (what correctness properties are (and are not) expressible in the logic), and (2) the complexity of the decision procedure for testing satisfiability (or equivalently, testing validity) in the logic. A great deal of attention has been devoted to classifying and comparing various logics in terms of these two characteristics. These matters are not only of theoretical interest, but are also of great practical significance as well. A logic that is very expressive is desirable since it may provide great generality and ease of use for specification in applications. At the same time, the

---

[1]Work supported in part by NSF Grant MCS8302878

complexity of the decision procedure is of concern in mechanical verification efforts. It is also of crucial importance in applications such as program synthesis from temporal logic specifications (cf. [EC82], [MW84]) where the decision procedure is used to construct a model of the specification formula from which a concurrent program is extracted. Since, as a general rule, the more expressive a logic the greater the complexity of its decision procedure, it is a challenge to get a logic with a satisfactory tradeoff between expressive power and cost of testing satisfiability.

As discussed in [EH83], there are certain advantages to doing temporal reasoning in a BTL. For example, in concurrent program synthesis applications the ability to assert the existence of alternative computation paths as allowed in a BTL (but not in an LTL) can be used to enforce lower bounds on nondeterminism so that the synthesized concurrent program exhibits an adequate degree of parallelism. On the other hand, a criticism often levelled at branching time logic is the inability to express correctness under fair scheduling assumptions (cf. [LA80], [LP84]). Indeed, the temporal logic (CTL) used in the branching time approach of [EC82] cannot express inevitability under fair scheduling assumptions. While another BTL, CTL*, does have a sufficiently rich syntax to allow expression of most any correctness property including fair inevitability, the best known deterministic time decision procedure is of triple exponential complexity (cf. [ES84]), and almost certainly unsuitable for use in applications. To have any hope of practical utility, we need a logic of exponential time complexity.

An inspection of the literature does not reveal any BTL sufficiently expressive to allow specification of correctness properties under fair scheduling constraints which also has an exponential time decision procedure We are thus left with the question: does there exist such a logic? We are now able to answer in the affirmative. We describe a BTL which we call *Fair Computation Tree Logic* (FCTL). FCTL generalizes the (ordinary) CTL as used in [EC82] and [CES83] by having all path quantifiers relativized to a fairness assumption Φ_0 specified by a boolean combination of the infinitary linear time temporal operators $\overset{\infty}{F}p$ (which abbreviates GFp and means "infinitely often p") and $\overset{\infty}{G}p$ (which abbreviates FGp and means "almost everywhere p"). Its basic modalities are thus of the form A_Φ ("for all fair paths") or E_Φ ("for some fair path") followed by a single linear time operator: F, G, X, or U. The infinitary operators of Φ_0 make it possible to express and reason under a wide variety of "practical" fairness assumptions from the literature including impartiality [LPS81], weak fairness ([LA80]), strong fairness ([LA80]), fair reachability of predicates ([QS83]), state fairness ([PN83]), generalized fairness ([FK84]), as well as the technical notion of "limited looping" fairness ([AB80]). We show that when Φ_0 is in a restricted canonical form $\bigwedge_{i=1}^{n} (\overset{\infty}{F}a_i \vee \overset{\infty}{G}b_i)$, FCTL has the *Small Model Property*: any satisfiable FCTL specification has a small (exponential size) finite model. Since most all "practical" types of fairness considered in the literature (including

all those listed above) can be directly specified using a Φ_0 in restricted canonical form, we immediately get a nondeterministic exponential time, and, hence, deterministic double exponential time decision procedure for a BTL which allows reasoning about many useful types of fairness. This is an exponential time improvement over the decision procedure for CTL*. We go on to show that when Φ_0 is further restricted to be of the form $\bigwedge_{i=1}^{n} \mathrm{F}a_i^{\infty}$, we can give a decision procedure for FCTL of *deterministic, single exponential time* complexity. Such a Φ_0 still suffices for expressing important types of fairness including impartiality, weak fairness, and unconditional fairness ([FK84]).

In broad outline, our decision procedure follows that for CTL (cf. [EC82], [EH82]). Given an FCTL specification, we show that we can collapse any (possibly infinite) model by identifying states which agree on the Fischer-Ladner ([FL79]) closure of the specification. The resulting quotient structure can be viewed as a small, "pseudo-model" which can be unwound into a small, genuine model of p. The unwinding is accomplished by detecting small subgraphs of the quotient structure which certify fulfillment of the eventualities and then splicing the fragments together appropriately. The decision procedure can be implemented by building a *tableau* which essentially contains the quotient structure as a subgraph. This method has been generalized in automata-theoretic terms in [VW84] and is somewhat similar to that used in [KO83] to decide a fragment of the propositional Mu-Calculus.

However, there are significant technical differences between our approach here and that described above. We believe that these differences may account for our being able to handle branching time modalities in exponential time that no previous approach could. A crucial limitation of the previous approaches seems to be that the subgraphs they used to certify fulfillment of eventualities could not contain cycles. For example, for ordinary CTL the subtree certifying fulfillment in the original (say, without loss of generality, tree-like) model of the eventuality AFp collapses to a directed acyclic subgraph embedded (in a special way) in the quotient structure. Similarly, in [VW84] finite subtrees certify fulfillment of eventualities. In our approach however, the subgraphs do contain (disciplined) cycles: For example, the subgraph certifying fulfillment of $\mathrm{A}_{\Phi}\mathrm{Fp}$ will in general contain unfair cycles (corresponding to paths which violate the fairness specification Φ_0.) while the subgraph for $\mathrm{E}_{\Phi}\mathrm{Fp}$ will contain a fair cycle. Some additional, somewhat subtle, technical constraints are also placed on the subgraphs certifying fulfillment of eventualities. A further complicating factor is that $\mathrm{E}_{\Phi}\mathrm{Gp}$ and $\mathrm{E}_{\Phi}\mathrm{Xp}$ are essentially eventuality properties since we must check that there really exist fair paths.

We can gain some insight into the possible significance of our techniques by expressing correctness properties in the propositional Mu-Calculus (cf. [EC80], [DB80]). The propositional Mu-calculus provides a *least fixpoint* operator (μ) and a *greatest fixpoint* operator (ν) which

make it possible to give *fixpoint characterizations* of the branching time modalities. Intuitively, the Mu-Calculus makes it possible to characterize the modalities in terms of recursively defined tree-like patterns (and, thus, may itself be viewed as a BTL). For example, the ordinary CTL modality for potentiality $EFP = \mu Z.[P \vee EXZ]$, the least fixpoint of the functional $P \vee EXZ$ where Z is an atomic proposition variable (intuitively ranging over sets of states).[2] Similarly, ordinary inevitability $AFP = \mu Y.[P \vee AXY]$. The fixpoint characterizations for correctness properties under fair scheduling can be significantly more complicated however. For even a simple type of fairness, specified by, say, $\Phi_0 = \overset{\infty}{F}\neg P$, fair inevitability $A_{\Phi}FQ \equiv A[\overset{\infty}{F}\neg P \Rightarrow FQ] \equiv A[\overset{\infty}{G}P \vee FQ] = \mu Y.\nu Z.[AX((P \vee Y) \wedge Z) \vee Q]$. Hence while fair inevitability is expressible in the Mu-Calculus, it is not expressible in the *aconjunctive* μ-calculus shown to be decidable in deterministic exponential time in [KO83] (because Y is active in both conjuncts $AX(P \vee Y)$ and Z). We also note that alternating nestings of μ's and ν's in the above fixpoint characterization of fair inevitability lead to discontinuities so that transfinite ordinals are needed in well-foundedness arguments to prove fair inevitability (cf. [AO83], [LPS81], [GFMD81]). Moreover, all the eventuality properties that are handled in deterministic exponential time in the automata-theoretic approach of [VW84] are definable by Buchi automata on infinite trees (cf. [RA68]) whereas properties such as fair inevitability are not.

The limitation of both the [KO83] and [VW84] approaches can be viewed as an inability to cope with certain well-foundedness conditions corresponding to transfinite ordinals and for fulfillment of eventualities, a limitation reflected in the absence of cycles in the certifying subgraphs for eventualities. We thus believe that generalizations of our approach here might lead to deterministic exponential time decision procedures more expressive logics suitable for applications, and, in particular (i) for a larger fragment of the propositional Mu-Calculus than that of [KO83] and (ii) in a broader automata-theoretic framework than [VW84].

Finally, by way of comparison with [QS83], we remark that the main result of [QS83] is that (in effect) their "temporal logic for reasoning about fairness (FCL)" is translatable into (ordinary) CTL. (They are using a rather restricted form of fairness.) Since our FCTL subsumes CTL, it also subsumes their FCL.

The remainder of the paper is organized as follows: Section 2 presents important preliminary definitions including the characterization of the subgraphs certifying fulfillment of eventualities. Section 3 proves the small model theorem for FCTL while the decision procedure is given in section 4.

---

[2]PDL enthusiasts should think of EXZ as $<R>Z$ and AXZ as $[R]Z$ where R is the program letter corresponding to the underlying transition relation of the (Kripke) structure.

2. The Specification Language

A Fair Computation Tree Logic (FCTL) specification (p_0,Φ_0) consists of a functional assertion p_0 and an underlying fairness assumption Φ_0. The functional assertion p_0 is expressed in essentially CTL syntax with basic modalities of the form either A_Φ ("for all fair paths"), or E_Φ ("for some fair path") followed by one of the linear time operators Fp ("sometimes p"), Gp ("always p"), Xp ("nexttime p"), or [p U q] ("p holds until q becomes *true*"). We subscript the path quantifiers with the symbol Φ to emphasize that they range over paths meeting the fairness constraint Φ_0, and to syntactically distinguish FCTL from CTL. A fairness constraint Φ_0 is build up from atomic propositions, the infinitary linear time operators $\overset{\infty}{F}p$ ("infinitely often p") and $\overset{\infty}{G}p$ ("almost always p"), and boolean connectives. Note that p_0 is a state formula (*true* or *false* of states) whereas Φ_0 is a path formula (*true* or *false* of paths).

2.1. Syntax. Formally, the class of functional assertions in FCTL specifications is defined inductively as follows:

1. Any atomic proposition P is a functional assertion.
2. If p, q are functional assertions then so are $\neg p$, and $(p \wedge q)$.
3. If p, q are functional assertions then so are $A_\Phi Xp$, $E_\Phi Xp$, $A_\Phi(pUq)$, and $E_\Phi(pUq)$.

A propositional formula is one formed by rules 1, 2 above. A fairness constraint is then formed by the following rules:

4. If p, q are propositional formulae then $\overset{\infty}{F}p$, $\overset{\infty}{G}p$ are fairness constraints.
5. If p, q are fairness constraints then so are $\neg p$, and $(p \wedge q)$.

The other connectives can then be defined as abbreviations in the usual way: $p \vee q$ abbreviates $\neg(\neg p \wedge \neg q)$, $p \Rightarrow q$ abbreviates $\neg p \vee q$, $A_\Phi Fq$ abbreviates $A_\Phi[\mathit{true}\ U\ q]$, $E_\Phi Fq$ abbreviates $E_\Phi[\mathit{true}\ U\ q]$, $A_\Phi Gq$ abbreviates $\neg E_\Phi F \neg q$, $E_\Phi Gq$ abbreviates $\neg A_\Phi F \neg q$, $\Phi A_\Phi[p\ W\ q]$ abbreviates $\neg E_\Phi[p\ U\ q]$, $E_\Phi[p\ W\ q]$ abbreviates $\neg A_\Phi[p\ U\ q]$ etc.

Remark: Recall that CTL* is the full branching time logic in which the basic modalities have the form: A or E followed by an arbitrary combination (involving boolean connectives and nesting) of linear time operators F, G, X, and U. We could thus view the assertions of FCTL as a sublanguage of CTL* where, eg., the $A_\Phi Fp$ is an abbreviation for the CTL* formula $A[\Phi_0 \Rightarrow Fp]$. However, the corresponding CTL* formula might be rather unwieldy due to the need to repeatedly write down multiple copies of the actual fairness formula Φ_0.

2.2. Semantics. Let AP be the underlying set of atomic propositions. A *prestructure* M=(S, R, L) is a labelled transition graph, where

- S is a nonempty set of states.
- R is a binary relation on S which gives the possible transitions between states.
- L is a labeling which assigns to each state a set of formulae .

We say that prestructure M = (S,R,L) is a *subprestructure* of prestructure M' = (S',R',L') provided that $S \subseteq S'$, $R \subseteq R'$, and $L = L'|S$. We also say that M is *contained in* M'. The size of a prestructure M=(S, R, L), written |M|, is defined to be the number of states in S. A *structure* is a prestructure with R total. A *fullpath* x is an infinite sequence of states $(s_0, s_1, s_2, \ldots)$ such that $\forall i \geq 0[(s_i, s_{i+1})] \in R]$. We use x^i to denote the suffix of x beginning at state x_i, i.e. $x^i=(x_i, x_{i+1}, \ldots)$. We write $M, x \models \Phi_0$ to mean that fullpath x in prestructure M meets fairness constraint Φ_0. We define the $\models$ relation inductively in the usual way:

1. $M, x \models P$ iff $P \in L(s_0)$, for any atomic proposition P.
2. $M, x \models \neg p$ iff not$(M, x \models p)$
3. $M, x \models p \wedge q$ iff $M, x \models p$ and $M, x \models q$
4. $M, x \models \overset{\infty}{F} p$ iff there exists infinitely many $i \geq 0$ such that $M, x^i \models p$
5. $M, x \models \overset{\infty}{G} p$ iff $\exists i \geq 0[\forall j \geq i(M, x^j \models p)]$

We write $M, s \models E\Phi_0$ if there is a fullpath x starting at s such that $M, x \models \Phi_0$. We say that fullpath x is a fair path in prestructure M under fairness assumption Φ_0 if $M, x \models \Phi_0$ holds. A state is fair iff it lies on some fair path.

An FCTL specification (p_0,Φ_0) is interpreted with respect to a prestructure M. We write $M, s \models_{\Phi_0} p_0$ to mean that functional assertion p_0 is *true* at state s of prestructure M under fairness assumption Φ_0. We define $\models_{\Phi_0}$ inductively on the prestructure of the functional assertion p_0:

1. $M, s \models_{\Phi_0} P$ iff $P \in L(s)$, for any atomic proposition P.
2. $M, s \models_{\Phi_0} \neg p$ iff not$(M, s \models_{\Phi_0} p)$
3. $M, s \models_{\Phi_0} p \wedge q$ iff $M, s \models_{\Phi_0} p$ and $M, s \models_{\Phi_0} q$
4. $M, s_0 \models_{\Phi_0} E_\Phi Xp$ iff there exists a path $x=(s_0, s_1, s_2, \ldots)$ such that $M, x \models \Phi_0$, and $M, s_1 \models_{\Phi_0} p$
5. $M, s_0 \models_{\Phi_0} A_\Phi Xp$ iff for all paths $x=(s_0, s_1, s_2, \ldots)$ $[M, x \models \Phi_0 \Rightarrow M, s_1 \models_{\Phi_0} p]$
6. $M, s_0 \models_{\Phi_0} E_\Phi(p \text{ U } q)$ iff there exists a path $x=(s_0, s_1, s_2, \ldots)$ such that $M, x \models \Phi_0$ and $\exists j \geq 0[M, s_j \models_{\Phi_0} q \wedge \forall i<j(M, s_i \models_{\Phi_0} p)]$
7. $M, s_0 \models_{\Phi_0} A_\Phi(p \text{ U } q)$ iff for all paths $x=(s_0, s_1, s_2, \ldots)$, $M, x \models \Phi_0$ implies $\exists j \geq 0[M, s_j \models_{\Phi_0} q \wedge \forall i<j(M, s_i \models_{\Phi_0} p)]$

A model w.r.t. fairness assumption Φ_0 is a structure M=(S,R,L) such that for all $s \in S$, and for

every functional assertion p, $M,s \models_{\Phi_0} p$ iff $p \in L(s)$.

In the sequel we will assume that a functional assertional has been written in positive normal form where all negations have been pushed as deep as possible using duality and DeMorgan's laws. We use *not(p)* to indicate the formula, also in positive normal form, obtained from ¬p by driving the ¬ in as far as possible and eliminating double negations.

Remarks: (1) The restricted form of $\Phi_0 = \wedge_{i=1}^{n} \overset{\infty}{F} a_i$ suffices to handle unconditional fairness of [FK84], impartiality of [LPS81] as $\overset{n}{\underset{i=1}{\wedge}} (\overset{\infty}{F} executed_i)$, where $executed_i$ is a proposition which asserts that process i is being executed, and weak fairness of [LA80] (also known as justice in [LPS81]) as $\overset{n}{\underset{i=1}{\wedge}} (\overset{\infty}{G} enabled_i \Rightarrow \overset{\infty}{F} executed_i) \equiv \overset{n}{\underset{i=1}{\wedge}} (\overset{\infty}{F}(\neg enabled_i) \vee \overset{\infty}{F} executed_i) \equiv \overset{n}{\underset{i=1}{\wedge}} (\overset{\infty}{F}(\neg enabled_i \vee executed_i))$.

(2) It is routine to extend the semantics of FCTL to be interpreted over PDL-like (cf. [FL79]) structures $M = (S,A_1,...A_k,L)$ in order to formalize the distinction between a process being enabled and being executed. Alternatively, we can encode the extended semantics into the present semantics as is done in [PN77], [QS83], or [CES83].

(3) The reader wishing additional information on FCTL, its generalizations and range of applicability, is referred to the companion papers [EL85a] and [EL85b] which consider the - technically much simpler - problem of model checking.

3. Preliminaries.

Remark: To simplify the exposition, we will just deal with $A_\Phi Fq$ and $E_\Phi Fq$; the generalization to $A_\Phi[p\ U\ q]$ and $E_\Phi[p\ U\ q]$ is routine and left to the full paper.

In the next section, we will prove the Small Model Theorem for FCTL as follows: Suppose M is a model of (p_0,Φ_0). Identify those states which agree on the *closure* of (p_0,Φ_0). The resulting *quotient structure* M' is a *Pseudo-Hintikka Structure* which can be unwound into a small model of (p_0,Φ_0). More precisely, for each state s of M' and for each eventuality p in the label of s, we show that there is a "fragment" contained in M' which certifies that p is fulfilled at s. These fulfilling fragments can then be spliced together in such a way as to get a small model of (p_0,Φ_0). We now give some terminology to the make the above precise. (We suggest that the reader go over this section quickly on first reading and then refer to it in detail as needed while reading the subsequent sections.)

3.1. Quotient Construction. We define the *Extended Fischer-Ladner Closure* (cf. [FL79]} of an FCTL specification (p_0,Φ_0), written $EFL(p_0,\Phi_0)$ to be the least set such that,

EFL1. $p_0 \in EFL(p_0,\Phi_0)$
EFL2. $p \in EFL(p_0,\Phi_0) \Rightarrow not(p) \in EFL(p_0,\Phi_0)$
EFL3. $p \wedge q \in EFL(p_0,\Phi_0) \Rightarrow p, q \in EFL(p_0,\Phi_0)$
EFL4. $p \vee q \in EFL(p_0,\Phi_0) \Rightarrow p, q \in EFL(p_0,\Phi_0)$
EFL5. $E_\Phi Xp \in EFL(p_0,\Phi_0) \Rightarrow p \in EFL(p_0,\Phi_0)$
EFL6. $A_\Phi Xp \in EFL(p_0,\Phi_0) \Rightarrow p \in EFL(p_0,\Phi_0)$
EFL7. $E_\Phi Fp \in EFL(p_0,\Phi_0) \Rightarrow p, E_\Phi XE_\Phi Fp \in EFL(p_0,\Phi_0)$
EFL8. $A_\Phi Fp \in EFL(p_0,\Phi_0) \Rightarrow p, A_\Phi XA_\Phi Fp \in EFL(p_0,\Phi_0)$
EFL9. $E_\Phi Gq \in EFL(p_0,\Phi_0) \Rightarrow q, E_\Phi XE_\Phi Gq \in EFL(p_0,\Phi_0)$
EFL10. $A_\Phi Gq \in EFL(p_0,\Phi_0) \Rightarrow q, A_\Phi XA_\Phi Gq \in EFL(p_0,\Phi_0)$
EFL11. All (atomic) propositional subformulae of Φ_0 are in $EFL(p_0,\Phi_0)$.
EFL12. $E_\Phi X\mathit{true} \in EFL(p_0,\Phi_0)$

Note that $|EFL(p_0,\Phi_0)| = O(|p_0| + |\Phi_0|)$.

A subset s of $EFL(p_0,\Phi_0)$ is said to be *maximal* if for every formula $q \in EFL(p_0,\Phi_0)$, either $q \in s$ or $not(q) \in s$. The *quotient structure* of a prestructure $M=(S, R, L)$ w.r.t. (Φ_0,p_0), written $M/EFL(p_0,\Phi_0)$, is defined as the structure $M'=(S', R', L')$ such that $S'=\{[s]: [s] = \{t\in S: L(t) \cap EFL(p_0,\Phi_0) = L(s) \cap EFL(p_0,\Phi_0)\}\}$, $R'=\{([s],[t]): (s,t)\in R\}$, and $L'([s]) = L(s)\cap EFL(p_0,\Phi_0)$. Intuitively, it is the structure obtained from M by identifying and collapsing together all states that have the same closure. [s] is the equivalence class of states t in M satisfying the same formulae in the closure as s.

3.2. Hintikka Structure. A Hintikka structure for an FCTL specification (p_0,Φ_0) is a structure $M=(S,R,L)$ such that $p_o \in L(s)$ for some $s \in S$ which satisfies the following constraints:

H1. $E_\Phi X\mathit{true} \in L(s)$ iff $\exists$fullpath $x[M,x \models \Phi_0]$
H2. $E_\Phi X\mathit{true} \in L(s)$ or $A_\Phi X\mathit{false} \in L(s)$
H3. $not(p) \in L(s) \Rightarrow p \notin L(s)$
H4. $p \wedge q \in L(s) \Rightarrow p,q \in L(s)$
H5. $p \vee q \in L(s) \Rightarrow p \in L(s)$ or $q \in L(s)$
H6. $E_\Phi Fq \in L(s) \Rightarrow E_\Phi X\mathit{true} \in L(s)$ and $[q \in L(s)$ or $E_\Phi XE_\Phi Fq \in L(s)]$
H7. $A_\Phi Fq \in L(s) \Rightarrow A_\Phi X\mathit{false} \in L(s)$ or $q \in L(s)$ or $A_\Phi XA_\Phi Fq \in L(s)$
H8. $E_\Phi Gq \in L(s) \Rightarrow q, E_\Phi X\mathit{true}, E_\Phi XE_\Phi Gq \in L(s)$
H9. $A_\Phi Gq \in L(s) \Rightarrow A_\Phi X\mathit{false} \in L(s)$ or $q, A_\Phi XA_\Phi Gq \in L(s)$
H10. $E_\Phi Xp \in L(s) \Rightarrow E_\Phi X\mathit{true} \in L(s) \wedge \exists(s,t) \in R[p, E_\Phi X\mathit{true} \in L(t)]$
H11. $A_\Phi Xp \in L(s) \Rightarrow \forall(s,t) \in R[p \in L(t) \vee A_\Phi X\mathit{false} \in L(t)]$
H12. $E_\Phi Fq \in L(s) \Rightarrow$ for some fair path x starting at s and some state t on x, $q \in L(t)$.
H13. $A_\Phi Fq \in L(s) \Rightarrow$ for all fair paths x starting at s and some state t on x, $q \in L(t)$
H14. $E_\Phi Gq \in L(s) \Rightarrow$ there is a fair path x starting at s such that for all t on x, $q \in L(t)$.

Remarks: A Hintikka structure M is a structure with certain syntactic constraints on its labelling which guarantee that if formula p appears in L(s) for state s, then p is actually true at s in M. (Hence, if M is a Hintikka structure for (p_0,Φ_0) it defines a model of (p_0,Φ_0); the

converse is also true.) Rules H1-H2 guarantee that fair paths are handled correctly, rules H4-10 guarantee propositional consistency, rules H10-11 guarantee local consistency, while rules H1,12-14 guarantee that eventualities are fulfilled. Note that $A_{\Phi}Gq$ is automatically handled by the local consistency rule H9.

Proposition 3.2.1: An FCTL specification (p_0, Φ_0) is satisfiable iff there exists a structure M'=(S',R',L'), a state s' $\in$ S', such that M', s' $\models_{\Phi_0}$ p_0, and either (1) $\forall t \in S'$[M', t $\models_{\Phi_0}$ $E_{\Phi}X$*true*] or (2) $\forall t \in S'$[M', t $\models_{\Phi_0}$ $A_{\Phi}X$*false*].

proof: It suffices to show the only-if part since the if-part is trivial. Assume that (p_0, Φ_0) is satisfiable. Then there exists a structure M=(S,R,L) and a state s $\in$ S such that M, s $\models_{\Phi_0}$ p_0.

Case 1: M, s $\models_{\Phi_0}$ $E_{\Phi}X$*true*. Let S'=$\{t \in S$: M, t $\models_{\Phi_0}$ $E_{\Phi}X$*true*$\}$, R' = R|S'$\times$S', and L' = L|S'. It is easily argued by induction on formula length that $\forall$ subformula p' of p_0 $\forall t \in S'$[M, t $\models_{\Phi_0}$ p' iff M', t $\models_{\Phi_0}$ p']. In particular, M, s $\models_{\Phi_0}$ p_0 iff M' s $\models_{\Phi_0}$ p_0. Moreover, condition (1) holds on structure M'.

Case 2: M, s $\models_{\Phi_0}$ $A_{\Phi}X$*false*. Since $A_{\Phi}X$*false* $\equiv A[\Phi_0 \Rightarrow X$*false* $\equiv A[\neg\Phi_0]$, it is clear that Φ_0 $\not\equiv$*true* (otherwise, M, s $\models_{\Phi_0}$ *false* -- which is impossible). Hence we can find in M an unfair path starting at s. More precisely, take S' = $\{t \in S$: M, t $\models_{\Phi_0}$ $A_{\Phi}X$*false* and t is reachable from s$\}$, R' = R|S'$\times$S', and L' = L|S'. It is quite obvious that R' is total and every path in M' is unfair. . Again, it can be proved easily by induction on formula length that $\forall$ subformula p' of p_0 [M, s $\models_{\Phi_0}$ p' iff M', s $\models_{\Phi_0}$ p'].

Note that for any state s in a structure, if A_{Φ}*false* holds at s then every formula of the form $E_{\Phi}p$ is false at s and every formula of the form $A_{\Phi}p$ is true at s. Let prop(p) denote the formula obtained from p by substituting *true* (*false*) for each subformula $A_{\Phi}p'$ ($E_{\Phi}p'$) of p.

Corollary 3.2.2: Let (p_0, Φ_0) be an FCTL specification, M = (S,R,L) be a structure and s be a state in S. If M, s $\models_{\Phi_0}$ $A_{\Phi}X$*false* then M, s $\models_{\Phi_0}$ p_0 iff M, s $\models_{\Phi_0}$ prop(p_0).

Note that prop(p_0) is a pure propositional formula whose truth value at a state s in a structure M = (S,R,L) depends on the labelling function L only and has nothing to do with the fairness assumption Φ_0.

Thus, to test satisfiability of (p_0,Φ_0), it suffices to check whether (1) (p_0,Φ_0) is satisfiable in a structure all of whose states satisfy $E_{\Phi}X$*true* or (2) the propositional formula prop(p_0) is satisfiable. In what follows, we will concentrate on how to check condition (1). We therefore assume that every state in a structure satisfies $E_{\Phi}X$*true*.

3.3. Fragments. We say that the A-formula $A_\Phi Fq$ is *not violated* in M provided that $\forall s \in S[E_\Phi X\mathit{true}, A_\Phi Fq \in L(s) \Rightarrow$ for all fair fullpaths x starting at s, q appears (in the label of some node) on x]. An A-formula $A_\Phi Fq$ is *fulfilled* in M provided that $\forall s \in S[E_\Phi X\mathit{true}, A_\Phi Fq \in L(s)$ $\Rightarrow$ q appears in the label of some node along every path starting at s which is infinite and fair or finite and has $E_\Phi X\mathit{true}$ at the last node]

An E-formula $E_\Phi Fq$ appearing in the label of a node s of M is said to be *fulfilled* at s if there exists a fair fullpath x in M starting at s such that q appears in the label of some node on x; we say that $E_\Phi Fq$ is *pending* at s if it is not fulfilled at s but there is a finite maximal path y in M starting at s, such that either $E_\Phi Fq$ appears everywhere on the path, or q appears on y and $E_\Phi X\mathit{true}$ appears at the last node of y.

An E-formula $E_\Phi Gq$ appearing in the label of an node s of M is said to be *fulfilled* at s if there exists a fair fullpath x in M starting at s such that q appears in the label of every node on x; we say that $E_\Phi Gq$ is *pending* at s if it is not fulfilled at s but there is a finite maximal path y in M starting at s, such that $E_\Phi Gq$ and $E_\Phi X\mathit{true}$ appear everywhere on y.

If $E_\Phi g$ (g is either Fq or Gq) is either fulfilled or pending at a state s, we say that $E_\Phi g$ is *not violated* at s. We say that $E_\Phi g$ is *not violated in a prestructure* M=(S,R,L) provided that $E_\Phi g$ is not violated at any state of S.

We define *fulfillment* and *violation* for an E-formula $E_\Phi Xp$ similarly.

A node in a prestructure M is said to be a *frontier* node provided it does not have any successor in M. On the contrary, nodes with successor(s) are *interior* nodes.

A *fragment* (rooted) at state s is a prestructure which satisfies the following 5 conditions:

1. All nodes are reachable from s.
2. No A-formulae are violated.
3. No E-formulae are violated.
4. All the interior nodes of the prestructure satisfy H2-H11.
5. All the frontier nodes of the prestructure satisfies H2-H9.

Note that all eventualities appearing in the root of a fragment are either fulfilled in the fragment or propagated to the frontier nodes in order to be fulfilled later.

In the following definitions for A/E/full-fragments, we only consider nodes including in their label $E_\Phi X\mathit{true}$ because the A/E/full-fragment for node s containing $A_\Phi X\mathit{false}$ can be defined to be to be an unfair path starting at s with $A_\Phi X\mathit{false}$ in all its nodes.

An *E-fragment at a state s for an E-formula* $E_\Phi g$ (g = Fq, Gq, or Xq) is a fragment such that $E_\Phi g$ is fulfilled at s.

An *A-fragment at a state s for an A-formula* $A_\Phi Fq$, is a fragment M=(S,R,L) which satisfies the following conditions:

1. There is no fair path in M.
2. For all exterior nodes u of M, q L(u).

3. Each interior node containing $E_\Phi X\mathit{true}$ can reach some frontier node.

Note that in such a fragment $A_\Phi Fq$ is fulfilled.

A *full fragment* for a state s is a prestructure M such that every eventuality $A_\Phi Fq$, $E_\Phi Fq$, $E_\Phi Gq$, $E_\Phi Xq$ in s is fulfilled in M. Note that every formula that appears in the label of some node of a full fragment will not be violated.

A *pseudo-fair component (PFC)* C of structure M is a strongly connected subprestructure of M such that

1. C contains a fair fullpath.
2. Every fullpath in C which goes through each node of C infinitely often is fair.
3. If any $A_\Phi Fq$ appears in (the label of some node in) C then q also appears in (the label of some, in general, different node in) C.
4. $E_\Phi X\mathit{true}$

3.4. Pseudo Hintikka Structure. A Pseudo Hintikka Structure (PHS) for an FCTL specification (p_0,Φ_0) is a structure M=(S,R,L) with $p_0 \in L(s)$ for some $s \in S$ which satisfies the consistency requirements H2-H12 of Hintikka Structure and the following conditions (which certify "pseudo-fulfillment of the eventualities"):

H1'. $E_\Phi X\mathit{true} \in L(s) \Rightarrow \exists$ a path starting at s leading to a PFC in M.
H12'. $E_\Phi Fq \in L(s) \Rightarrow \exists$ in M a path starting at s leading to a state t such that $p \in L(t)$ and a PFC is reachable from t.
H13'. $A_\Phi Fq \in L(s) \Rightarrow \exists$ an A-fragment for s w.r.t. $A_\Phi Fq$ that is contained in M.
H14'. $E_\Phi Gq \in L(s) \Rightarrow \exists$ in M a finite path from s leading to a PFC such that all nodes on the finite path and in the PFC contain p in their label.

4. Small Model Theorem for FCTL

4.1. Theorem. Suppose (p_0,Φ_0) is an FCTL specification with Φ_0 of the form $\bigwedge_{i=1}^{k}(\overset{\infty}{F}a_i \vee \overset{\infty}{G}b_i)$, and $|EFL(p_0,\Phi_0)|=n$. Then the following are equivalent:

(1) (p_0,Φ_0) is satisfiable.
(2) There is a PHS for (p_0,Φ_0) of size $\leq O(2^n)$.
(3) There is a Hintikka structure for (p_0,Φ_0) of size $\leq O(n\cdot 2^{4n})$.
(4) (p_0,Φ_0) has a model of size $\leq O(n\cdot 2^{4n})$.

proof. (1) $\Rightarrow$ (2): In the sequel, let M=(S,R,L) be a model for (p_0,Φ_0), where $\Phi_0=\wedge_{i=1}^{n}(\overset{\infty}{F}a_i \vee \overset{\infty}{G}b_i)$, and $|EFL(p_0,\Phi_0)|=n$. We can assume w.l.o.g. M is a model such that, for each s, L(s) is exactly all the formula in $EFL(p_0,\Phi_0)$ that are true at s in M Let M'=(S',R',L') be the quotient structure of M w.r.t. $EFL(p_0,\Phi_0)$. In order to establish that M' is a PHS for

(p_0,Φ_0) we will prove a series of lemmas asserting that each eventuality is "pseudo-fulfilled" in M'.

4.1.1. Lemma. If $A_\Phi Fq \in L'([s])$ then there is an A-fragment N at [s] for AFq contained in M' (such that all frontier nodes of N contain q in their label).

proof sketch. Since $A_\Phi Fq \in L'([s]) = L(s)$ it follows that there is a fragment N_1 in M rooted at s which certifies fulfillment of $A_\Phi Fq$ at s (just follow every path out from s as long as q has not occurred.) Since some nodes may have (infinitely) many successors, for each node t in N_1 and each formula $E_\Phi Xp$ in t choose a successor u of t such that $p \in L(t)$. Delete all successor nodes that are not so chosen. Each node of the resulting structure, $N_2=(S_2,R_2,L')$, has at most n successors. Moreover, N_2 is still a fragment for AFq. If N_2 is finite then we are done; otherwise, we can now give a Ramsey-type argument collapsing N_2 into a finite fragment N_3. We can then eliminate nodes with duplicate labels to get a fragment N_4 contained in M'. Details are given in the full paper. □

4.1.2. Lemma. If $E_\Phi Gp \in L'([s])$ then there exists in M' a finite path from [s] leading to a PFC such that all nodes on the finite path and in the PFC contain p in their label.

proof sketch. Since $E_\Phi Gp \in L'([s]) = L(s)$ and M is an actual model of (p_0,Φ_0), $E_\Phi Gp$ is true at s in M. Thus, there is a fair fullpath $x = (s=s_0,s_1,s_2,...)$ starting at s in M each of whose states contains $E_\Phi Gp$ and p in its label. Consider the "image" of x in M': $([s_0],[s_1],[s_2],...)$. It consists of a finite prefix (those $[s_i]$ which appear only finitely often) followed by an infinite suffix (those $[s_i]$ which appear infinitely often). The nodes of M' appearing in the infinite suffix define a strongly connected subprestructure of M' which is a PFC and the finite prefix defines a finite path leading to it. Since $p \in L(s_i) = L([s_i])$ for each s_i on x, we thus have the desired finite path leading to a PFC. □

Arguments similar to that for Lemma 4.1.2 can be used to prove the following two lemmas:

4.1.3. Lemma. If $E_\Phi X\mathit{true} \in L'([s])$ then there is a path from [s] leading to a PFC contained in M'. □

4.1.4. Lemma. If $E_\Phi Fp \in L'([s])$ then there exists in M' a path starting at [s] leading to a state [t] such that $p \in L'([t])$ and a PFC contained in M' is reachable from [t]. □

Thus M' is the desired PHS for (p_0,Φ_0) of size $\leq O(2^n)$, and we have established that (1) $\Rightarrow$ (2).

(2) $\Rightarrow$ (3): We must show how to unwind a PHS M = (S,R,L) for (p_0,Φ_0) of size $O(2^n)$ into an HS for (p_0,Φ_0) of size $O(n\cdot 2^{4n})$.

4.1.5. Lemma. If $E_\phi Gp \in L(s)$, then we can construct an E-fragment for $E_\phi Gp$ at s of size $O(2^{2n})$.

proof sketch. Since M is a PHS, there exists in M a (shortest) finite path from s leading to a PFC; all of the nodes appearing therein have p in their label. Unwind the finite path into a finite simple path and the PFC into a finite simple cycle duplicating nodes as needed. (This can cause a quadratic blowup.) Each node u' in the resulting graph is a copy of some node u in M. For each such u' and for each $E_\phi Xq \in L(u)$, add as a successor to u' a copy v' of some successor in M of u such that $q \in L(v)$. This final graph is the desired E-fragment for $E_\phi Gp$ at (a copy of) s. □

We can similarly establish the following two lemmas:

4.1.6. Lemma. If $E_\phi Fp \in L(s)$, then we can construct an E-fragment for $E_\phi Fp$ at s of size $O(2^{2n})$. □

4.1.7. Lemma. If $E_\phi Xp \in L(s)$, then we can construct an E-fragment for $E_\phi Xp$ at s of size $O(2^{2n})$. □

We then splice together the E-fragments and A-fragments to get a full fragment at each state s of size $O(n \cdot 2^{3n})$. These full fragments are then spliced together to get a Hintikka Structure for (p_0,Φ_0) of size $O(n \cdot 2^{4n})$. The details are similar to those in [EC82] and [EH82].

The proofs that (3) $\Rightarrow$ (4) and (4) $\Rightarrow$ (1) are immediate. □

5. Decision Procedure

It follows from the Small Model Theorem that satisfiability for FCTL specifications (p_0,Φ_0) with Φ_0 in canonical form is decidable in nondeterministic exponential time: just guess a small structure M and check that it is a model. This model checking can be done in time polynomial in the size of M and (p_0,Φ_0). (Actually, the model checking can be done in linear time; see [EL84a], [EL84b] for details.) However, when Φ_0 is further restricted we can do much better:

5.1. Theorem. There is an algorithm for deciding whether a given FCTL specification (p_0,Φ_0) with $\Phi_0 = \bigwedge_{i=1}^{k} \overset{\infty}{F} a_i$ is satisfiable which runs in deterministic time 2^{cn} for some constant $c>0$, where $n=|EFL(p_0,\Phi_0)|$.

Proof sketch. We define a "tableau" $M=(S,R,L)$ where $S=\{s: s \in EFL(p_0,\Phi_0)$ and s is maximal$\}$, $L(s)=s$, and R is a binary relation on S consisting of all pairs $(s,t) \in S \times S$ *except* those for which $A_\phi Xp \in s$, $E_\phi X\mathit{true} \in t$, and $p \notin t$ or (ii) $E_\phi X\mathit{true} \notin s$ and $E_\phi X\mathit{true} \in t$.

We then repeatedly scan through the tableau checking for states that have no R-successor or violate one of the PHS rules H1', H2-H11, H12', H13', H14'. Any such state detected is deleted along with all incident arcs. This continues until no more nodes can be deleted.

Let M'=(S',R',L') be the (stablized) tableau upon termination. If S' is nonempty then M' is a PHS. So if $p_0 \in s$ for some $s \in S'$, (p_0,Φ_0) is satisfiable by the small model theorem of FCTL. Conversely, if (p_0,Φ_0) is satisfiable by a model N, then the nodes in the quotient structure of N (which trivially are a subset of S) will not be eliminated. Hence, for some node $s \in S'$, we have $p_0 \in s$.

Checking that rules H2-H11 are not violated is routine. But, it remains to show how to check H1', H12'-H14'. To handle rules H1', H12' and H14' we must be able to compute PFCs. To compute PFCs, we first compute strongly connected components (SCCs) of the tableau. An SCC will be a PFC iff each a_i appears in some node of the SCC, $E_\Phi X\mathit{true}$ appears in every node of the SCC, and for every $A_\Phi Fg \in EFL(p_0,\Phi_0)$, if $A_\Phi Fg$ appears somewhere in the SCC then so does g. This method allows us to detect the nodes which can reach a PFC making it routine to check H1', H12'. To check H14', for $E_\Phi Gq$, delete all nodes not containing q in the label. Then compute PFCs in the resulting subprestructure and check that there is a path from s to a PFC in the subprestructure.

To check the fulfillment for an A-formulae $A_\Phi Fq$, we only have to consider nodes with $E_\Phi X\mathit{true}$ appearing on their labels because nodes with $A_\Phi X\mathit{false}$ satisfy all A-formulae. To check H13' (for $A_\Phi Fq$), we proceed as follows:

A. for all $t \in S$ do if $q \in L(t)$ then mark t GREEN;

B. for each conjunct $\overset{\infty}{F}a_i$ do the following:

B.1. for each $t \in S$ do if t has enough GREEN successors then mark t GREEN;

B.2. repeat

for all unmarked nodes $t \in S$ such that $\neg a_i \in L(t)$ do
if t has enough successors u such that
either $\neg a_i \in L(u)$ or u is marked (GREEN or BLUE)
then mark t BLUE
until no more node can be marked;

B.3. repeat

for all BLUE nodes $t \in S$ do
if t does not has enough marked successors
or t cannot reach a GREEN node via a BLUE path
then erase the marker of t
until no more makers can be erased;

B.4. Change BLUE markers to GREEN.

C. Repeat step B until no more changes can be made.

Claim: A state s ∈ S is marked GREEN by the above algorithm iff s satisfies H13'.

The proof of the claim is given in the full paper.

Since all the above processing can be implemented in time polynomial in the size of the tableau, the algorithm has the desired 2^{cn} complexity. □

6. References

[AB80] Abrahamson, K., *Decidability and Expressiveness of Logics of Processes*, PhD Thesis, University of Washington, 1980.

[AO83] Apt, K. R., Olderog, E. R. *Proof Rules and Translations Dealing with Fairness*, Science of Computer Programming 3 (1983), pp. 65-100.

[BMP81] Ben-Ari, M., Manna, Z., and Pnueli, A., *The Temporal Logic of Branching Time*, 8th Annual ACM Symp. on Principles of Programming Languages, 1981.

[CES83] Clarke, E. M., Emerson, E. A., and Sistla, A. P., *Automatic Verification of Finite State Concurrent System Using Temporal Logic*, 10th Annual ACM 10th Annual ACM Symp. on Principles of Programming Languages, 1983.

[DB80] DeBakker, J. W., *Mathematical Theory of Program Correctness* (Prentice-Hall, Englewood Cliffs, NJ, 1980).

[EC80] Emerson, E. A., and Clarke, E. M., *Characterizing Correctness Properties of Parallel Programs Using Fixpoints*, Proc. ICALP 80, LNCS Vol. 85, Springer Verlag, 1980, pp. 169-181.

[EC82] Emerson, E. A., and Clarke, E. M., *Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons*, Tech. Report TR-208, Univ. of Texas, 1982.

[EH82] Emerson, E. A., and Halpern, J. Y., *Decision Procedures and Expressiveness in the Temporal Logic of Branching Time*, 14th Annual ACM Symp. on Theory of Computing, 1982.

[EH83] Emerson, E. A., and Halpern, J. Y., *"Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time*, 14th Annual ACM Symp. on Theory of Computing, 1982.

[EL85a] Emerson, E. A., and Lei, C. L., *Temporal Model Checking Under Generalized Fairness Constraints*, to be presented at the 18th Annual Hawaii International Conference on System Sciences.

[EL85b] Emerson, E. A., and Lei, C. L., *Modalities for Model Checking: Branching Time Strikes Back*, to be presented at the 12th Annual ACM Symposium on Principles of Programming Languages.

[ES84] Emerson, E. A., and Sistla, A. P., *Deciding Branching Time Logic*, 16 Annual ACM Symp. on Theory of Computing, 1984.

[FK84] Francez, N., and Kozen, D., *Generalized Fair Termination*, 11th Annual ACM Symp. on Principles of Programming Languages, 1984, pp. 46-53.

[FL79] Fischer, M. J., and Ladner, R. E, *Propositional Dynamic Logic of Regular Programs*, JCSS vol. 18, pp194-211, 1979.}

[GFMD81] Grimberg, O., Francez, N., Makowsky, J. A., and deRoever, W. P., *A proof rule for fair termination of guarded commands*, Proc. International Symp. on Algorithmic Languages (North-Holland, Amsterdam, 1981).

[KO83] Kozen, D., *Results on the Propositional Mu-calculus*, Theoretical Computer Science, pp. 333-354, December 83.

[LA80] Lamport, L., *Sometimes is Sometimes "Not Never" - on the temporal logic of programs*, 7th Annual ACM Symp. on Principles of Programming Languages, 1980, pp. 174-185.

[LPS81] Lehmann. D., Pnueli, A., and Stavi, J., *Impartiality, Justice and Fairness: The Ethics of Concurrent Termination*, ICALP 1981, LNCS Vol. 115, pp 264-277.

[LP84] Lichtenstein, O. and Pnueli, A., *Checking that Finite State Concurrent Programs Satisfy their Linear Specification*, unpublished manuscript, July 84, (to appear in POPL85.)

[MP79] Manna, Z., and Pnueli, A., *The modal logic of programs*, Proc. 6th Int. Colloquium on Automata, Languages, and Programming, Springer-Verlag Lecture Notes in Computer Science #71, pp. 385-410, 1979.

[MW84] Manna, Z., and Wolper, P., *Synthesis of Communicating Processes from Temporal Logic Specifications*, TOPLAS, Vol. 6, #1, pp. 68-93.

[OL82] Owicki, S. S., and Lamport, L., *Proving Liveness Properties of Concurrent Programs*, ACM Trans. on Programming Languages and Syst., Vol. 4, No. 3, July 1982, pp. 455-495.

[PA78] Parikh, R., *A Decidability Result for a Second Order Process Logic*, 17th Annual Symp. on Foundations of Computer Science, 1978.

[PN77] Pnueli, A., *The Temporal Logic of Programs*, 19th annual Symp. on Foundations of Computer Science, 1977.

[PN83] Pnueli, A., *On The Extremely Fair Termination of Probabilistic Algorithms*, 15 Annual ACM Symp. on Theory of Computing, 1983, 278-290.

[PR79] Pratt, V., *Process Logic*, 6th Annual ACM Symposium on Programming Languages, 1979.

[QS83] Queille, J. P., and Sifaki, J., *Fairness and Related Properties in Transition Systems*, Research Report #292, IMAG, Grenoble, 1982.

[RA68] Rabin, M. O., *Weakly Definable Relations and Special Automata*, in Mathematical Logic and Foundations of Set Theory, Y. Bar-Hillel, editor, North-Holland, Amsterdam, 1968, pp. 1-23.

[VW84] Vardi, M. and Wolper, P., *Automata Theoretic Techniques for Modal Logics of Programs*, pp. 446-455, STOC84.