Proof Systems for Hennessy–Milner Logic with Recursion

Kim G. Larsen
Aalborg University Centre
Denmark

Abstract

An extension of Hennessy–Milner Logic with recursion is presented. A recursively specified formula will have two standard interpretations: a *minimal* one and a *maximal* one. Minimal interpretations of formulas are useful for expressing *liveness* properties of processes, whereas maximal interpretations are useful for expressing *safety* properties. We present sound and complete proof systems for both interpretations for when a process satisfies a (possibly recursive) formula. The rules of the proof systems consist of an introduction rule for each possible structure of a formula and are intended to extend the work of Stirling and Winskel. Moreover the proof systems may be presented directly in PROLOG to yield a *decision procedure* for verifying when a finite-state process satisfies a specification given as a (possibly recursive) formula.

1 Motivation

A non-deterministic and concurrent program repeatedly interacts with its environment. For this reason the semantics cannot simply be a function from input to output. In the work by Milner [M80,M83] an alternative, operational semantics is proposed based on the idea of an observer attempting to experiment with the program. The author introduces the notion of observational equivalence in order to express when two programs are indistinguishable by all observers. In the original work by Milner [M80] observational equivalence was defined as the limit of a decreasing ω-sequence of approximating equivalences. Later, a more elegant definition of observational equivalence in terms of the notion of bisimulation was given by Park [P] and investigated by Milner [M83]. Further evidence for the naturalness of observational equivalence was then given by Hennessy and Milner in [HM], where they showed that observational equivalence can be characterized by a simple modal logic, refered to in this paper as the *Hennessy–Milner Logic*. Various extensions and variations of Hennessy–Milner Logic and an investigation of their relative strength are contained in [BR]. Also, it has later been shown that other equivalences are characterized by certain variations of Hennessy–Milner Logic [BT,GS84,M81] A survey of some of these results may be found in [Pn85].

The formulas or assertions of Hennessy–Milner Logic are intended to specify the desired properties of a non-deterministic or concurrent system. In order to support modular design and verification in this framework the problem of *compositionality* of modal assertions has been a major research topic: i.e to deduce that a composition of processes satisfies or implements a given modal assertion from the knowledge of its components satisfying certain other modal assertions. This problem has been successfully dealt with by Stirling [St83,St85A,St85B] and Winskel [W84,W85], who both give sound and complete compositional axiomatizations for the satisfiability relation between processes (expressed in various subsets of CCS [M80] and SCCS [M83]) and modal formulas.

Obviously, this quest for compositionality is an important one to solve if Hennessy–Milner Logic is going to be used in design and verification of large systems. So far, not even small systems have been dealt with using Hennessy–Milner Logic. However, equally important in this respect is the *expressiveness* of the logic: i.e. to what extent does the logic allow us to express "interesting" properties as formulas. Though, Hennessy–Milner Logic *is* expressive enough to distinguish between any inequivalent processes (the characterization result of Hennessy and Milner [HM]) it is clearly not expressive enough from a pragmatic point of view. In fact, a formula of Hennessy–Milner Logic is only capable of describing properties of a certain finite part of processes. As such, it is impossible — as a single formula — to describe properties of processes, which are to hold *invariantly* (i.e. throughout a possibly infinite lifetime) or *eventually* (i.e. at some unspecified point in time), properties which clearly are necessary judged by the whelm of applications in the framework of *Temporal Logic*.

Actually, by extending Hennessy–Milner Logic, with infinitary connectives (conjunction say), invariant and eventual properties are expressible. In fact, by conjuncting all formulas satisfied by a process, we obtain a property characteristic for that process and its equivalence class. However, we are also concerned with the problem of *automating the verification* of when a process meets a specification expressed as a formula of Hennessy–Milner Logic. For that reason, we must insist that all formulas of our logic are finitary expressions.

In order to gain expressiveness without introducing infinitary connectives, we extend Hennessy–Milner Logic with *recursion* in section 2. A recursively specified formula will have two *standard interpretation*: a *minimal* one and a *maximal* one. It will be indicated in section 3 that the minimal interpretation is especially well–suited for expressing *liveness properties*, whereas the maximal interpretation is particularly useful for expressing *safety properties*. In fact, all the standard operators from Branching Time Temporal Logic [BMP] are derivable.

In sections 4, 5 and 6 we offer sound and complete proof systems for the satisfiability relation between processes and (possibly recursive) formulas, both for the minimal as well as for the maximal interpretation of recursion. The proof systems follow the spirit of those of Stirling [St83,St85A,St85B] and Winskel [W84,W85] except that no structure on the processes is assumed. The proof systems are therefore not compositional on the processes. Instead we concentrate on the new recursive structure on formulas. However, since no assumptions as to the structure of processes is made, our proof rules for recursive formulas may be added directly to the proof systems of Stirling and Winskel.

Moreover, the proof systems presented in section 6 may be represented directly in PROLOG to yield a *decision procedure* for verifying when a finite–state process satisfies a specification given as a (possibly recursive) formula of Hennessy–Milner Logic. Our procedure offers an alternative to the procedure of Clarke, Emerson and Sistla in [CES], which verifies that a finite–state process meets a specification expressed in a Propositional Branching Time Temporal Logic.

The extension of Hennessy–Milner Logic with recursion is not a new idea. The resulting logic is closely related to the μ–calculus [K], and indeed, similar extensions of Hennessy–Milner Logic *with* recursion has been studied by Graf and Sifakis [GS85,GS86]. However, the proof systems we present are new, and — we believe — interesting because they make precise in a logical sense the difference between minimal and maximal interpretations of recursive formulas. Also, the computational application of the proof systems seems to be of potential use for future application of Hennessy–Milner Logic in practical verifications of concurrent systems.

In the work of Graf and Sifakis there is no direct axiomatization of the satisfiability relation between processes and formulas. Instead, they are concerned with finding logics "adequate" with respect to a given equivalence on processes. In order for a logic to be adequate, formulas must represent unions of equivalence classes and each equivalence class must be represented by a formula. The verification of when a process p satisfies a formula F is then reduced to a verification of the *validity* of the formula $|p| \Rightarrow F$, where $|p|$ is the formula representing the equivalence class of p. Thus, in the work of Graf and Sifakis the problem of satisfiability of a formula with respect to a given process is reduced to the problem of validity of formulas. In [GS85] they present a *sound* deductive system for the validity of formulas, buth no completeness result has been stated. Also, from a computational point of view, the reduction to validity of formulas is not particular helpfull, since this problem is easily shown to be at least coNP–hard.

2 Hennessy-Milner Logic with Recursion

A non-deterministic and concurrent program repeatedly interacts with its environment. For this reason, the input/output function semantics — which has been successfully applied to sequential programs — is no longer adequate. Instead an operational semantics in terms of a *labelled transition system* [Pl] is given. Let Pr be a set of processes and Act a set of actions which the processes may perform. The dynamic changes of processes as they perform actions are given by a derivation relation $\longrightarrow \subseteq Pr \times Act \times Pr$. For $(p,a,q) \in \longrightarrow$ we shall normally write $p \xrightarrow{a} q$ which may be interpreted: "the process p can perform the action a and in doing so become the process q". The triple $\mathbf{P} = (Pr, Act, \longrightarrow)$ constitutes the transition system of processes. Often we shall restrict ourselves to transition systems with a kind of bounded non-determinism; namely that of image-finiteness. The transition system $\mathbf{P} = (Pr, Act, \longrightarrow)$ is said to be *image-finite* in case the set $\{q \mid p \xrightarrow{a} q\}$ is finite for all processes p and actions a.

Hennessy–Milner Logic [HM] — originally proposed as an alternative characterization of bisimulation equivalence [P,M83] — provides a language for specifying modal properties of processes. Following [St83,St85B] we use a slightly different but equivalent modal logic without an explicit negation operator.

Definition 2.1 *Let $\mathcal{M}$ be the smallest set of formulas built up according to the following abstract syntax:*

$$F ::= \mathrm{tt} \mid \mathrm{ff} \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F$$

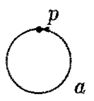
The meaning of a formula F of $\mathcal{M}$ is given by defining, inductively on the structure of F, the set of processes $[\![F]\!]$ which satisfies F:

Definition 2.2 *Define $[\![F]\!] \subseteq Pr$ for $F \in \mathcal{M}$ inductively as:*

1. $[\![\mathrm{tt}]\!] = Pr$
2. $[\![\mathrm{ff}]\!] = \emptyset$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$
4. $[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$
5. $[\![\langle a \rangle F]\!] = \langle \cdot a \cdot \rangle [\![F]\!]$
6. $[\![[a]F]\!] = [\cdot a \cdot][\![F]\!]$

where for $S \subseteq Pr$: $\langle \cdot a \cdot \rangle S = \{p \in Pr \mid \exists p'.p \xrightarrow{a} p' \wedge p' \in S\}$ and $[\cdot a \cdot]S = \{p \in Pr \mid \forall p'.p \xrightarrow{a} p' \Rightarrow p' \in S\}$. Following the notation of [W84] *we write $\models p : F$ in case $p \in [\![F]\!]$, where p is a process and F is a formula.*

It is clear from the above definition, that any single formula of $\mathcal{M}$ only describes a certain finite part of processes. In order to capture properties for the full infinite behaviour we need infinite sets of formulas from $\mathcal{M}$. As an example consider the following simple transition system:



A property characteristic for p is that it always, i.e. at any moment in its infinite life, is capable of performing an a action. The only way to describe this in HML is as an infinite set of formulas: $\{\langle a\rangle\mathrm{tt}, [a]\langle a\rangle\mathrm{tt}, [a][a]\langle a\rangle\mathrm{tt}, \ldots\}$. Obviously, this is rather unsatisfactory and we would like an extension of $\mathcal{M}$ in which this and similar properties are expressible as single formulas. The solution is rather obvious: simply add *recursion* to HML. Then the property of p above may be specified recursively as a property X satisfying the following equation:

$$X \equiv \langle a\rangle\mathrm{tt} \wedge [a]X$$

where $F \equiv G$ if and only if F and G are satisfied by exactly the same processes of the transition system $\mathbf{P}$ (Thus, to be precise, we ought to write $F \equiv_{\mathbf{P}} G$ indicating in which process system $\mathbf{P}$, F and G take their semantics).

Clearly, $X = \mathrm{ff}$ is a solution to the above equation (since no process can satisfy both $\langle a\rangle\mathrm{tt}$ and $[a]\mathrm{ff}$). However, $p \not\models \mathrm{ff}$, so obviously this is not the solution we are looking for. It turns out that by taking the maximal solution to the equation we obtain the desired property. In contrast, $X = \mathrm{ff}$ is the minimal solution. It turns out that maximal solutions are useful for describing *invariant or safety* properties, whereas minimal solutions

are useful for describing *liveness* properties. We now extend $\mathcal{M}$ with a set of *identifiers* for recursion:

Definition 2.3 *Let* Id *be a set of identifiers. Then the set of formulas over* Id, $\mathcal{M}_{\mathrm{Id}}$, *is the smallest set of formulas built up according to the following abstract syntax:*

$$F ::= \mathrm{tt} \mid \mathrm{ff} \mid F \wedge G \mid F \vee G \mid \langle a\rangle F \mid [a]F \mid X$$

where $X \in \mathrm{Id}$.

Syntactically the meaning of the identifiers is specified, often recursively, by a *declaration* assigning a formula (the body) of $\mathcal{M}_{\mathrm{Id}}$ to each identifier:

Definition 2.4 *A* declaration $\mathcal{D}$ *is a function* $\mathrm{Id} \longrightarrow \mathcal{M}_{\mathrm{Id}}$. *For* $X \in \mathrm{Id}$ *we shall simply write* F_X *for* $\mathcal{D}(X)$ *when* $\mathcal{D}$ *is understood.*

In all that follows we shall assume a fixed set of identifiers Id and a fixed declaration $\mathcal{D}$. Semantically, the meaning of a formula is defined relative to an *assignment*, $\sigma : \mathrm{Id} \longrightarrow 2^{Pr}$, assigning meaning, i.e. a set of processes, to each identifier. We may then define the meaning of a formula F over $\mathcal{M}_{\mathrm{Id}}$ *relative* to an assignment σ as a set of processes $[\![F]\!]\sigma$ satisfying the following extension of definition 2.2:

Definition 2.5 *Define for* $F \in \mathcal{M}_{\mathrm{Id}}$ *and* $\sigma \in \mathrm{Id} \longrightarrow 2^{Pr}$ *the set of processes* $[\![F]\!]\sigma$ *inductively as follows:*

1. $[\![\mathrm{tt}]\!]\sigma = Pr$
2. $[\![\mathrm{ff}]\!]\sigma = \emptyset$
3. $[\![F \wedge G]\!]\sigma = [\![F]\!]\sigma \cap [\![G]\!]\sigma$
4. $[\![F \vee G]\!]\sigma = [\![F]\!]\sigma \cup [\![G]\!]\sigma$
5. $[\![\langle a\rangle F]\!]\sigma = \langle\cdot a\cdot\rangle([\![F]\!]\sigma)$
6. $[\![[a]F]\!]\sigma = [\cdot a\cdot]([\![F]\!]\sigma)$
7. $[\![X]\!]\sigma = \sigma(X)$

We shall write $\models_\sigma p : F$ *in case* $p \in [\![F]\!]\sigma$, *where* p *is a process,* F *is a formula and* σ *is an assignment.*

We call $p : F$ a *correctness assertion* following [W84]. So $\models_\sigma p : F$ means that the correctness assertion $p : F$ is valid under the assignment σ. We extend $\models_\sigma$ to sets of correctness assertions Γ in the obvious way: i.e. $\models_\sigma \Gamma$ iff $\models_\sigma q : G$ whenever $q : G \in \Gamma$.

For a given specified declaration $\mathcal{D}$, we are searching for assignments σ under which the semantics of any identifier, $[\![X]\!]\sigma$, equals the semantics of the corresponding body in $\mathcal{D}$, $[\![F_X]\!]\sigma$. Such assignments can be seen as solutions to the recursive declaration $\mathcal{D}$ and are called *models* for $\mathcal{D}$. We are especially interested in maximal and minimal models, and we shall use those for generating the two standard semantics of formulas over $\mathcal{M}_{\mathrm{Id}}$. Models are defined through the following notions of *post-models* and *pre-models*:

Definition 2.6 *A* pre-model *(for $\mathcal{D}$) is an assignment σ such that $[\![F_X]\!]\sigma \subseteq [\![X]\!]\sigma$ holds for any identifier X. Similarly, a* post-model *(for $\mathcal{D}$) is an assignment σ such that $[\![X]\!]\sigma \subseteq [\![F_X]\!]\sigma$ holds for any identifier X. A* model *(for $\mathcal{D}$) is an assignment which is both a pre- and post-model.*

We shall adopt the following convenient notation for assignments:

Notation 2.7 *Let $\sigma_1, \sigma_2, \sigma_i$, $i \in I$, be assignments. Then:*

- *$\sigma_1 \subseteq \sigma_2$ iff $\sigma_1(X) \subseteq \sigma_2(X)$ for all identifiers X.*
- *$\bigcap_{i \in I} \sigma_i$ is the assignment given by: $(\bigcap_{i \in I} \sigma_i)(X) = \bigcap_{i \in I}(\sigma_i(X))$.*
- *$\bigcup_{i \in I} \sigma_i$ is the assignment given by: $(\bigcup_{i \in I} \sigma_i)(X) = \bigcup_{i \in I}(\sigma_i(X))$.*

From this it is rather obvious that the set of assignments constitutes a *complete lattice* under $\subseteq$ with $\bigcap$ as greatest lower bound and $\bigcup$ as least upper bound.

The declaration $\mathcal{D}$ induces a transformation $[\![\mathcal{D}]\!]$ on assignments in the following way: the transformed assignment $[\![\mathcal{D}]\!]\sigma$ of σ is the function given by:

$$[\![\mathcal{D}]\!]\sigma \,:\, X \longmapsto [\![F_X]\!]\sigma$$

for all identifiers X. With this definition it is clear that an assignment σ is a post-model (pre-model) iff $\sigma \subseteq [\![\mathcal{D}]\!]\sigma$ ($[\![\mathcal{D}]\!]\sigma \subseteq \sigma$). Thus, a model is simply a fixed-point of $[\![\mathcal{D}]\!]$. Because $\mathcal{M}_{\mathrm{Id}}$ does not contain an explicit negation operator, $[\![F]\!]\sigma$ is monotonic in σ for all formulas F. This will ensure the existence of a minimal and maximal model. Moreover, provided the process system $\mathbf{P}$ is image-finite, $[\![F]\!]\sigma$ is both continuous and anti-continuous in σ:

Lemma 2.8 *Let F be a formula and σ_1, σ_2 assignments. Then $[\![F]\!]\sigma_1 \subseteq [\![F]\!]\sigma_2$ whenever $\sigma_1 \subseteq \sigma_2$. If $\mathbf{P}$ is image-finite and $\sigma_1 \subseteq \sigma_2 \subseteq \sigma_3 \subseteq \ldots$ is an increasing chain of assignments, then $[\![F]\!](\bigcup_i \sigma_i) = \bigcup_i([\![F]\!]\sigma_i)$. Also, if $\mathbf{P}$ is image-finite and $\sigma_1 \supseteq \sigma_2 \supseteq \sigma_3 \supseteq \ldots$ is a decreasing chain of assignments, then $[\![F]\!](\bigcap_i \sigma_i) = \bigcap_i([\![F]\!]\sigma_i)$.*

From this it follows directly that the transformation of assignments $[\![\mathcal{D}]\!]$ is also a monotonic and, when $\mathbf{P}$ is image-finite, a continuous as well as anti-continuous operation on the complete lattice of assignments. Moreover, monotonicity ensures the following important closure properties of pre- and post-models:

Lemma 2.9 *Let σ_i, $i \in I$, be a family of pre-models (post-models). Then $\bigcap_i \sigma_i$ ($\bigcup_i \sigma_i$) is also a pre-model (post-model).*

We may now state the following main-theorem which ensures the existence of a maximal and minimal model (post–model / pre–model). The theorem is a simple application of the Knaster–Tarski fixed–point theorem for monotonic functions on complete lattices [T].

Theorem 2.10 *There exist a maximal post–model, $\sigma_{\max}$, and a minimal pre–model, $\sigma_{\min}$, given by the following:*

$$\sigma_{\max} = \bigcup\{\sigma \mid \sigma \subseteq [\![\mathcal{D}]\!]\sigma\}$$
$$\sigma_{\min} = \bigcap\{\sigma \mid [\![\mathcal{D}]\!]\sigma \subseteq \sigma\}$$

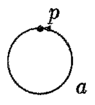
Moreover, $\sigma_{\max}$ and $\sigma_{\min}$ are both models. When **P** *is image–finite, $\sigma_{\max}$ and $\sigma_{\min}$ may alternatively be defined as greatest lower bound resp. least upper bound of the following chains:*

$$\sigma_{Pr} \supseteq [\![\mathcal{D}]\!]\sigma_{Pr} \supseteq [\![\mathcal{D}]\!]^2\sigma_{Pr} \supseteq \ldots$$
$$\sigma_{\emptyset} \subseteq [\![\mathcal{D}]\!]\sigma_{\emptyset} \subseteq [\![\mathcal{D}]\!]^2\sigma_{\emptyset} \subseteq \ldots$$

where $\sigma_{Pr}(X) = Pr$, and $\sigma_{\emptyset}(X) = \emptyset$ for all identifiers X.

Since any model is also a post–model and a pre–model, it follows that $\sigma_{\max}$ is the maximal model, and $\sigma_{\min}$ is the minimal model. We shall write $\models_{\max} p{:}\,F$ resp. $\models_{\min} p{:}\,F$ whenever $\models_{\sigma_{\max}} p{:}\,F$ resp. $\models_{\sigma_{\min}} p{:}\,F$, where p is a process and F is a formula. $\models_{\max}$ and $\models_{\min}$ are extended to sets of correctness assertions in the obvious ways.

Example 2.11 *Let* **P** *be given by the following diagram:*



We want to prove that the process p is always capable of doing an a-action. We shall represent a declaration $\mathcal{D}$ over a finite set of identifiers Id $= \{X_1, \ldots, X_n\}$ *as a system of n simultaneous recursive equations of the form:*

$$\begin{array}{ccc} X_1 & \Leftrightarrow & \mathcal{D}(X_1) \\ \vdots & \vdots & \vdots \\ X_n & \Leftrightarrow & D(X_n) \end{array}$$

where either "$\Leftarrow$" or "$\Rightarrow$" is to replace all occurrences of "$\Leftrightarrow$" (thus, no mixture of "$\Leftarrow$" and "$\Rightarrow$" is allowed). Following the convention introduced in [P], *the use of "$\Leftarrow$" will indicate that the minimal model for $\mathcal{D}$ is the intended one, whereas "$\Rightarrow$" specifies the maximal model as the intended one. Now consider the following recursive definition of X:*

$$X \Leftrightarrow \langle a\rangle \mathrm{tt} \wedge [a]X$$

where "$\Leftrightarrow$" is to be replaced by either "$\Leftarrow$" or "$\Rightarrow$". Obviously **P** *is image-finite, so we may apply the constructions from theorem 2.10 to obtain $\sigma_{\min}$ and $\sigma_{\max}$:*

$$\sigma^0_{\min}(X) = \emptyset$$
$$\sigma^1_{\min}(X) = \{p\} \cap [\cdot a \cdot]\emptyset \;=\; \{p\} \cap \emptyset \;=\; \emptyset$$

Thus, $\sigma_{\min} = \sigma_\emptyset$ and $\not\models_{\min} p: X$.

$$\sigma^0_{\max}(X) = \{p\}$$
$$\sigma^1_{\max}(X) = \{p\} \cap [\cdot a \cdot]\{p\} \;=\; \{p\} \cap \{p\} \;=\; \{p\}$$

Hence $\sigma_{\max} = \sigma_{Pr}$ and $\models_{\max} p: X$

3 Derived Temporal Operators

The property X studied in example 2.11 is an attempt of expressing an invariant property in Hennessy–Milner Logic with recursion. However, there is a general recursive scheme to be used for expressing such invariant properties. In fact — as is also partially pointed out in [GS85] — all the modalities of standard Branching Time Temporal Logic [BMP] may be expressed in our logic, underlining its expressiveness. For convenience we assume that the set of actions, *Act*, is a finite set $\{a_1, \ldots, a_n\}$. We then write $[Act]F$ respectively $\langle Act\rangle F$ as an abbreviation for the formula $[a_1]F \wedge \ldots \wedge [a_n]F$ respectively $\langle a_1\rangle F \vee \ldots \vee \langle a_n\rangle F$. This finiteness restriction on *Act* may be removed by using as basis of our extension, a variant of HML, where [] and $\langle\rangle$ are parameterized with sets of actions. However, in order to simplify the soundness and completeness proofs of the proof systems to follow, we have decided to stay with the simpler version of HML.

Now, let F be a given formula. Then the following recursive equations specify the four modalities of Branching Time Temporal Logic presented in [BMP]:

1. $I_F \Rightarrow F \wedge [Act]I_F$
2. $S_F \Rightarrow F \wedge ([Act]\mathrm{ff} \vee \langle Act\rangle S_F)$
3. $P_F \Leftarrow F \vee \langle Act\rangle P_F$
4. $E_F \Leftarrow F \vee (\langle Act\rangle tt \wedge [Act]E_F)$

The above properties — denoted $\forall\mathrm{G}\, F$, $\exists\mathrm{G}\, F$, $\exists\mathrm{F}\, F$ and $\forall\mathrm{F}\, F$ in [BMP] — may informally be explained as follows: I_F is satisfied by a process, if any derivation of the process enjoys the property F. Hence I_F expresses the *invariance* of F *under all computations* (a computation being a maximal derivation sequence). S_F is satisfied by a process, if the process has a computation — finite or infinite — on which all processes enjoys F. Thus, S_F expresses the *invariance* of F *under some computation.* A process satisfies P_F if F holds for some derivative of it; i.e. it is *possible* that F will hold at some point in the future. Finally, a process satisfies E_F if F holds *inevitable* or *eventually*; i.e. if for all computations of the process F holds at some point.

Not only are the operators from [BMP] expressible in the extended logic — so too are the more complex and general *weak* and *strong until operators.* For F and G being given formulas we make the following recursive definitions:

5. $U^s_{F,G} \Leftarrow G \vee (F \wedge \langle Act\rangle \mathrm{tt} \wedge [Act]U^s_{F,G})$
6. $U^w_{F,G} \Rightarrow G \vee (F \wedge [Act]U^w_{F,G})$

$$\text{tt} \qquad \vdash_{\min} p\colon \text{tt}$$

$$\wedge \qquad \frac{\vdash_{\min} p\colon F \quad \vdash_{\min} p\colon G}{\vdash_{\min} p\colon F \wedge G}$$

$$\vee \qquad \frac{\vdash_{\min} p\colon F}{\vdash_{\min} p\colon F \vee G} \qquad \frac{\vdash_{\min} p\colon G}{\vdash_{\min} p\colon F \vee G}$$

$$\langle\rangle \qquad \frac{\vdash_{\min} p'\colon F}{\vdash_{\min} p\colon \langle a\rangle F} \quad p \xrightarrow{a} p'$$

$$[] \qquad \frac{\vdash_{\min} p_1\colon F \ldots\ldots \vdash_{\min} p_n\colon F}{\vdash_{\min} p\colon [a]F} \quad \{p_1,\ldots,p_n\} = \{p' \mid p \xrightarrow{a} p'\}$$

$$Rec \qquad \frac{\vdash_{\min} p\colon F_X}{\vdash_{\min} p\colon X} \quad F_X = \mathcal{D}(X)$$

Figure 1: The proof system **Min**

From these definitions it is clear that $U^s_{F,G}$ and $U^w_{F,G}$ are true generalizations of E_F and I_F; endeed the following equivalences hold: $U^s_{\text{tt},G} \equiv E_G$ and $U^w_{F,\text{ff}} \equiv I_F$.

Informally, both $U^s_{F,G}$ and $U^w_{F,G}$ are satisfied by a process if for any of its computations F holds until G does. In $U^w_{F,G}$ however, F is allowed to hold all through a computation wihtout G ever being achieved. This is in contrast to $U^s_{F,G}$, where G must be guaranteed to hold eventually in all computations of the process. Thus, we expect $U^s_{F,G}$ to be a stronger property that $U^w_{F,G}$ — hence the terminology.

4 The Proof System Min

In this section we present a proof system for Hennessy-Milner Logic with recursion which is complete under the *minimal interpretation.* The proof system is compositional on the structure of formulas whereas the processes — unlike the proof systems of Stirling [St83,St85A,St85B] and Winskel [W84,W85] — are assumed to be structureless elements of a general image–finite transition system. This assumption has been made in order to focus solely on the recursive structure which has been added to the formulas.

The proof system we offer consists of an introduction rule for each possible structure of a formula. The introduction rule for recursion is the obvious one: in order to prove $\models p\colon X$, where X is an identifier, simply prove $\models p\colon F_X$, where $F_X = \mathcal{D}(X)$.

Definition 4.1 (The Min Proof System)
Let **P** *be an image–finite process system. Let X be an identifier, $p, p', p_1, p_2, \ldots$ processes and $F, G, \ldots$ formulas over* Id. *Then $\vdash_{\min}$ is the smallest set of correctness assertions satisfying the rules of figure 1, where $\vdash_{\min} p\colon F$ means $p\colon F \in \vdash_{\min}$*

The proof system **Min** is sound and complete as stated in the following theorem:

Theorem 4.2 (Soundness and Completeness)
Let **P** *be an image–finite process system, p be a process and F be a formula. Then $\vdash_{\text{min}} p{:}\, F$ if and only if $\models_{\text{min}} p{:}\, F$.*

PROOF: For the "$\Rightarrow$" direction, we show that $\models_{\text{min}}$ is closed under the rules for $\vdash_{\text{min}}$. Since σ_{min} is a premodel we know that $[\![F_X]\!]\sigma_{\text{min}} \subseteq [\![X]\!]\sigma_{\text{min}}$ for any identifier X. Thus, if $\models_{\text{min}} p{:}\, F_X$ then clearly also $\models_{\text{min}} p{:}\, X$. Hence, *Rec* preserves $\models_{\text{min}}$. From definition 2.5 it is clear that $\models_\sigma$ for any assignment σ is preserved by the remaining rules $\text{tt}, \wedge, \vee, \langle\rangle$. For the "$\Leftarrow$" direction, consider the following assignment $\sigma_\vdash$ induced by $\vdash_{\text{min}}$: $\sigma_\vdash(X) = \{p \mid \vdash_{\text{min}} p{:}\, X\}$. We show that $\sigma_\vdash$ is a premodel and that $\models_{\sigma_\vdash} p{:}\, F$ iff $\vdash_{\text{min}} p{:}\, F$. Since σ_{min} is the smallest premodel the direction will follow. □

Let us now demonstrate the use of the proof system through an example:

Example 4.3 *We want to prove that the process p described by the following diagram may* possible *deadlock:*

$$\begin{tikzcd}
p \arrow[r, bend left, "a"] & q \arrow[l, bend left, "a"] \arrow[r, "a"] & r
\end{tikzcd}$$

Following the schemes presented in section 3, the property of possible deadlock may be represented by the following recursive equation:

$$X \Leftarrow [a]\text{ff} \vee \langle a\rangle X$$

Let us now prove that the process p enjoys the property X (under the minimal interpretation) using the proof system **Min***:*

$$
{\scriptstyle Rec}\,\dfrac{{\scriptstyle \vee}\,\dfrac{{\scriptstyle \langle\rangle}\,\dfrac{{\scriptstyle Rec}\,\dfrac{{\scriptstyle \vee}\,\dfrac{{\scriptstyle \langle\rangle *}\,\dfrac{{\scriptstyle Rec}\,\dfrac{{\scriptstyle \vee}\,\dfrac{{\scriptstyle []}\,\dfrac{-}{\vdash_{\text{min}} r{:}\,[a]\text{ff}}}{\vdash_{\text{min}} r{:}\,[a]\text{ff} \vee \langle a\rangle X}}{\vdash_{\text{min}} r{:}\,X}}{\vdash_{\text{min}} q{:}\,\langle a\rangle X}}{\vdash_{\text{min}} q{:}\,[a]\text{ff} \vee \langle a\rangle X}}{\vdash_{\text{min}} q{:}\,X}}{\vdash_{\text{min}} p{:}\,\langle a\rangle X}}{\vdash_{\text{min}} p{:}\,[a]\text{ff} \vee \langle a\rangle X}}{\vdash_{\text{min}} p{:}\,X}
$$

From the point of view of *automatically* constructing proofs in a goal–directed manner, it is obvious that the freedom of choice between the two $\vee$–rules and the freedom in the $\langle\rangle$–rule, due to the choice of derivative, may in a depth–first search strategy lead to non–termination. In a goal–directed construction of the proof in example 4.3, the choice of the derivation $q \xrightarrow{a} r$ in the $\langle\rangle$–application marked $*$ is crucial for the termination of the construction. Choosing the derivation $q \xrightarrow{a} p$ instead would bring us back to the initial goal (that of proving $\vdash_{\text{min}} p{:}\, X$), thus making no progress as to constructing a proof.

Non-termination of a goal-directed proof-construction for a correctness assertion $p{:}\,F$ is often the way invalidity of $p{:}\,F$ manifests itself in **Min**. To see this consider the correctness assertion $p{:}\,Y$, where Y is defined by the following recursive equation:

$$Y \Leftarrow [a]\mathrm{ff} \vee [a]Y$$

Thus, Y expresses the property of *eventual* deadlock under the minimal interpretation; a property which p clearly does not possess. Now, a goal-directed attempt of constructing a proof for $p{:}\,Y$ will necessarily lead to a construction of the following shape:

$$\cfrac{\cfrac{\cfrac{\cfrac{\vdots}{\vdash_{\min} p{:}\,Y} \qquad \cfrac{\vdots}{\vdash_{\min} q{:}\,Y}}{\vdash_{\min} p{:}\,[a]Y}\,[\,]}{\vdash_{\min} p{:}\,[a]\mathrm{ff} \vee [a]Y}\,\vee}{\vdash_{\min} p{:}\,Y}\,\mathit{Rec}$$

Hence, the initial goal ($\vdash_{\min} p{:}\,Y$) reappears as a subgoal and the construction will therefore diverge. On the other hand, it should be possible during the automatic construction to realise that the subgoal $\vdash_{\min} p{:}\,Y$ is identical to the initial goal and that any attempt of solving this subgoal therefore is pointless. Instead of directly designing such an algorithm, we shall in the next sections present two new complete proof systems — one for the minimal and one for the maximal interpretation — where we consider *sequents* of the form $\Gamma \vdash p{:}\,F$, Γ being a set of correctness assertions. Semantically Γ will act as a set of assumptions under the validity of which the validity of $p{:}\,F$ is to be evaluated. The proof rule for recursion in the two new systems will be of the general form:

$$\frac{\Gamma' \vdash p{:}\,F_X}{\Gamma \vdash p{:}\,X} \quad F_X = \mathcal{D}(X)$$

where Γ' augments Γ with information of $p{:}\,X$ being the initiating goal. This will enable any future subgoal of the form $\Gamma' \vdash p{:}\,X$ to be dealt with immediately — though differently under the two interpretations — by a simple inspection of the assumptions Γ'.

For the maximal interpretation it turns out to be sufficient to establish $p{:}\,F_X$ under the assumption of $p{:}\,X$ holding in order to conclude $p{:}\,X$. Intuitively this means that, provided $p{:}\,X$ does not introduce a contradiction, then $p{:}\,X$ holds under the maximal interpretation. Hence, in this case Γ' will simply be the set $\Gamma \cup \{p{:}\,X\}$.

For the minimal interpretation on the other hand, $p{:}\,F_X$ must — as is obvious from the proof system **Min** above — be established without any assumptions in order to conclude $p{:}\,X$. In fact, it turns out that we must be able to establish $p{:}\,F_X$ even under the assumptions that $p{:}\,X$ does *not* hold. Hence, Γ' may in this case be expressed as the set $\Gamma \cup \{p{:}\,\neg X\}$, assuming the introduction of the negation operator $\neg$ to our logic.

5 The Proof System MAX

In order to obtain a sound and complete axiomatization for the maximal interpretation, it is only necessary to consider sequents, $\Gamma \vdash p{:}\,F$, where the assumptions of Γ are of the form $q{:}\,X$. I.e. whenever $q{:}\,G \in \Gamma$, then G is an identifier. We shall call such sets of correctness assertions *simple*. Simple sets behave in a rather straightforward manner with respect to satisfiability: for Γ a simple tt $\quad \Gamma \vdash_{\max} p{:}\, tt$

$\wedge \quad \dfrac{\Gamma \vdash_{\max} p{:}\, F \quad \Gamma \vdash_{\max} p{:}\, G}{\Gamma \vdash_{\max} p{:}\, F \wedge G}$

$\vee \quad \dfrac{\Gamma \vdash_{\max} p{:}\, F}{\Gamma \vdash_{\max} p{:}\, F \vee G} \qquad \dfrac{\Gamma \vdash_{\max} p{:}\, G}{\Gamma \vdash_{\max} p{:}\, F \vee G}$

$\langle\rangle \quad \dfrac{\Gamma \vdash_{\max} p'{:}\, F}{\Gamma \vdash_{\max} p{:}\, \langle a\rangle F} \quad p \xrightarrow{a} p'$

$[] \quad \dfrac{\Gamma \vdash_{\max} p_1{:}\, F \ldots\ldots \Gamma \vdash_{\max} p_n{:}\, F}{\Gamma \vdash_{\max} p{:}\, [a]F} \quad \{p_1, \ldots, p_n\} = \{p' \mid p \xrightarrow{a} p'\}$

*Rec*1 $\quad \Gamma, p{:}\, X \vdash_{\max} p{:}\, X$

*Rec*2 $\quad \dfrac{\Gamma, p{:}\, X \vdash_{\max} p{:}\, F_X}{\Gamma \vdash_{\max} p{:}\, X} \quad F_X = \mathcal{D}(X)$

Figure 2: The proof system **MAX**

set of correctness assertions the assignment $\tau_\Gamma : X \mapsto \{q \mid q{:}\, X \in \Gamma\}$ is the minimal assignment validating it. In fact $\models_\tau \Gamma$ iff $\tau_\Gamma \subseteq \tau$.

The notion of semantic validity under the maximal interpretation may now be extended to sequents in the following way:

Definition 5.1 (Validity)
Let Γ be a simple set of correctness assertions and $p{:}\, F$ a correctness assertion. Then $\Gamma \models_{\max} p{:}\, F$ iff:

$$\exists\sigma. ((\sigma \subseteq [\![\mathcal{D}]\!]\sigma \cup \tau_\Gamma) \text{ and } \models_\sigma p{:}\, F)$$

Thus, $p{:}\, F$ must hold for some assignment σ, which is *almost* a post–model for $\mathcal{D}$, except that it may use the assumptions Γ represented by τ_Γ. For $\Gamma = \emptyset$, σ is required to be a genuine post–model for $\mathcal{D}$, since $\tau_\emptyset$ maps any identifier to the empty set. Hence, $\emptyset \models_{\max} p{:}\, F \Leftrightarrow \models_{\max} p{:}\, F$, showing that the extension of $\models_{\max}$ to sequents is a conservative one.

Definition 5.2 (The MAX proof system)
Let **P** *be an image–finite process system. Let Γ be a simple set of correctness assertions, X an identifier, $p, p', p_1, p_2, \ldots$ processes and $F, G, \ldots$ formulas over* Id. *Then $\vdash_{\max}$ is the smallest set of correctness assertions satisfying the rules of figure 2.*

In the rules of figure 2 we have written $\Gamma, p{:}\, X$ for the set $\Gamma \cup \{p{:}\, X\}$. Also, we will often write $\vdash_{\max} p{:}\, F$ for $\emptyset \vdash_{\max} p{:}\, F$. In *Rec*2, the goal $p{:}\, X$ is being recorded in the assumption part. This allows any future occurrences of $p{:}\, X$ as a (sub–) goal to be dealt with immediately (and successfully) by virtue of the axiom *Rec*1.

Due to lack of space, we now state without proof the *soundness* of the proof system **MAX**. The proof of this theorem is given in a full version of this paper [L].

Theorem 5.3 (Soundness)
Let Γ be a simple set of correctness assertions, and $p{:}\,F$ be a correctness assertion. Then the following holds:

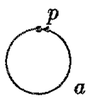
$$\Gamma \vdash_{\max} p{:}\,F \;\Rightarrow\; \Gamma \models_{\max} p{:}\,F$$

For *finite* process systems **P** and for *finite* sets of identifiers Id, the proof system **MAX** is moreover *complete*, as stated in the theorem below. Again, we refer to [L] for the proof.

Theorem 5.4 (Completeness)
Let **P** *be a finite process system and* Id *be a finite set of identifiers. Let Γ be a simple set of correctness assertions and $p{:}\,F$ be a correctness assertion. Then the following holds:*

$$\Gamma \models_{\max} p{:}\,F \;\Rightarrow\; \Gamma \vdash_{\max} p{:}\,F$$

Example 5.5 *Let* **P** *be given by the following diagram:*



and let X be given by the recursive equation:

$$X \Rightarrow \langle a\rangle\mathrm{tt} \wedge [a]X$$

i.e. X holds for a process provided the process is always or invariantly able to do an a–action. Let us now prove that p enjoys the property X using the **MAX** *system:*

$$Rec2\,\dfrac{\wedge\,\dfrac{\langle\rangle\,\dfrac{p{:}\,X \vdash_{\max} p{:}\,\mathrm{tt}}{p{:}\,X \vdash_{\max} p{:}\,\langle a\rangle\mathrm{tt}} \qquad \dfrac{p{:}\,X \vdash_{\max} p{:}\,X}{p{:}\,X \vdash_{\max} p{:}\,[a]X}\,[]}{p{:}\,X \vdash_{\max} p{:}\,\langle a\rangle\mathrm{tt} \wedge [a]X}}{\emptyset \vdash_{\max} p{:}\,X}$$

6 The Proof System MIN

An axiomatization adequate for the minimal interpretation is obtained by considering sequents, $\Gamma \vdash p{:}\,F$, where the assumptions of Γ are of the form $q{:}\neg X$. This introduces negation to our logic, but only in a very restricted sense and only in the assumption part. The notion of validity relative to an assignment σ of a negated formula of the form $q{:}\neg X$ is defined in the obvious way: $\models_\sigma q{:}\neg X \;\Leftrightarrow \not\models_\sigma q{:}\,X \;\Leftrightarrow\; q \notin \sigma(X)$.

As a dual to the notion of *simple* sets of correctness assertions we then define a set of correctness assertions Δ to be *co–simple* in case G is $\neg X$ for some identifier X whenever $q{:}\,G \in \Delta$ for some process q. Satisfiability of co–simple sets is also rather trivial: for Δ a co–simple set of correctness assertions define the assignment $\tau_\Delta : X \mapsto \{p \mid p{:}\neg X \notin \Delta\}$. It is obvious that τ_Δ validates Δ. In fact it may be shown that $\models_\tau \Delta \;\Leftrightarrow\; \tau \subseteq \tau_\Delta$.

The notion of semantic validity under the minimal interpretation may now be extended in the following way:

tt $\quad \Delta \vdash_{\min} p\colon \mathrm{tt}$

$\wedge \quad \dfrac{\Delta \vdash_{\min} p\colon F \quad \Delta \vdash_{\min} p\colon G}{\Delta \vdash_{\min} p\colon F \wedge G}$

$\vee \quad \dfrac{\Delta \vdash_{\min} p\colon F}{\Delta \vdash_{\min} p\colon F \vee G} \qquad \dfrac{\Delta \vdash_{\min} p\colon G}{\Delta \vdash_{\min} p\colon F \vee G}$

$\langle\rangle \quad \dfrac{\Delta \vdash_{\min} p'\colon F}{\Delta \vdash_{\min} p\colon \langle a\rangle F} \quad p \xrightarrow{a} p'$

$[] \quad \dfrac{\Delta \vdash_{\min} p_1\colon F \ldots\ldots \Delta \vdash_{\min} p_n\colon F}{\Delta \vdash_{\min} p\colon [a]F} \quad \{p_1, \ldots, p_n\} = \{p' \mid p \xrightarrow{a} p'\}$

Rec $\quad \dfrac{\Delta, p\colon \neg X \vdash_{\min} p\colon F_X}{\Delta \vdash_{\min} p\colon X} \quad F_X = \mathcal{D}(X)\ ,\ p\colon \neg X \notin \Delta$

Figure 3: The proof system **MIN**

Definition 6.1 (Validity)
Let Δ be a co-simple set of correctness assertions and $p\colon F$ a correctness assertion. Then $\Delta \models_{\min} p\colon F$ iff:

$$\forall\sigma.\,((\llbracket\mathcal{D}\rrbracket\sigma \cap \tau_\Delta \subseteq \sigma) \Rightarrow \models_\sigma p\colon F)$$

Thus, $p\colon F$ must hold for any assignment σ being a pre-fixed point to $\lambda\sigma.(\llbracket\mathcal{D}\rrbracket\sigma \cap \tau_\Delta)$. Obviously, this set includes all the pre-models for $\mathcal{D}$ and will increase as τ_Δ decreases — or equivalently as Δ increases. For $\Delta = \emptyset$, τ_Δ maps any identifier to the set Pr, and the assignments under which $p\colon F$ must be valid is exactly the pre-models of $\mathcal{D}$. Hence, $\emptyset \models_{\min} p\colon F \;\Leftrightarrow\; \models_{\min} p\colon F$, showing that the above extension of $\models_{\min}$ to sequents is a conservative one.

Definition 6.2 (The MIN proof system)
Let **P** *be an image-finite process system. Let Δ be a set of co-simple correctness assertions, X an identifier, $p, p', p_1, p_2, \ldots$ processes and $F, G, \ldots$ formulas over* Id. *Then $\vdash_{\min}$ is the smallest set of correctness assertions satisfying the rules of figure 3.*

In *Rec*, the goal $p\colon X$ is being recorded in the assumption part as a negated assertion, $p\colon \neg X$. Hence, any subsequent occurrences of $p\colon X$ as a (sub-) goal will fail due to lack of applicable inferencerules resulting from the side-condition of *Rec*.

The proof system **MIN** is *sound* and appropriately *complete* as stated in the theorems below. Again — due to lack of space – we refer the reader to [L] for the proofs.

Theorem 6.3 (Soundness)
Let Δ be a co-simple set of correctness assertions and $p\colon F$ be a correctness assertion. Then the following holds:

$$\Delta \vdash_{\min} p\colon F \;\Rightarrow\; \Delta \models_{\min} p\colon F$$

Theorem 6.4 (Completeness)
Let **P** *be a finite process system and* Id *a finite set of identifiers. Let Δ be a co-simple set of correctness assertions and $p\colon F$ be a correctness assertion. Then the following holds:*

$$\Delta \models_{\min} p\colon F \;\Rightarrow\; \Delta \vdash_{\min} p\colon F$$

Example 6.5 *Let us re-examine example 5.7 in the proof system* MIN. *Thus* **P** *is given by the diagram below:*

$$\begin{tikzcd} p \arrow[r, bend left, "a"] & q \arrow[l, bend left, "a"] \arrow[r, "a"] & r \end{tikzcd}$$

and the identifier X is specified by the following recursive equation:

$$X \Leftarrow [a]\mathrm{ff} \vee \langle a\rangle X$$

We prove that p enjoys X using MIN *as follows:*

$$\begin{array}{lc}
 & - \\ \cline{2-2}
[\,] & r{:}\neg X, q{:}\neg X, p{:}\neg X \vdash_{\min} r{:}[a]\mathrm{ff} \\ \cline{2-2}
\vee & r{:}\neg X, q{:}\neg X, p{:}\neg X \vdash_{\min} r{:}[a]\mathrm{ff} \vee \langle a\rangle X \\ \cline{2-2}
Rec & q{:}\neg X, p{:}\neg X \vdash_{\min} r{:}X \\ \cline{2-2}
\langle\rangle^{*} & q{:}\neg X, p{:}\neg X \vdash_{\min} q{:}\langle a\rangle X \\ \cline{2-2}
\vee & q{:}\neg X, p{:}\neg X \vdash_{\min} q{:}[a]\mathrm{ff} \vee \langle a\rangle X \\ \cline{2-2}
Rec & p{:}\neg X \vdash_{\min} q{:}X \\ \cline{2-2}
\langle\rangle & p{:}\neg X \vdash_{\min} p{:}\langle a\rangle X \\ \cline{2-2}
\vee & p{:}\neg X \vdash_{\min} p{:}[a]\mathrm{ff} \vee \langle a\rangle X \\ \cline{2-2}
Rec & \emptyset \vdash_{\min} p{:}X
\end{array}$$

We observe that, compared with the simpler proof system Min, *the possibility of non-termination has been removed by the introduction of assumptions. Still we may choose the derivation $q \xrightarrow{a} p$ instead of $q \xrightarrow{a} r$ in the $\langle\rangle$-application marked $*$. However, in contract to the simpler proof system* Min, *this will* not *bring us back to the initial goal, but instead to that of proving $q{:}\neg X, p{:}\neg X \vdash_{\min} p{:}X$. The only rule that may possibly be applied to this goal is the rule Rec. However, the side-condition of Rec is clearly not valid and the attempted application therefore fails immediately*

7 Conclusion and Future Work

The proof systems presented in this paper are intended to extend the work of Stirling [St83,St85A,St85B] and Winskel [W84,W85]. However, in contrast to their work, we make no assumptions as to the structure of processes and our systems are therefore valid for any image–finite system of processes. Thus, we believe, that our rules for recursive formulas may be added directly to the systems of Stirling and Winskel yielding complete *compositional* proof systems for Hennessy–Milner Logic *with* recursion.

The proof systems presented in sections 5 and 6 was partly motivated from a computational point of view. Indeed, an *automatic verification system* for deciding whether a finite-state CCS-process satisfies a (possibly recursive) formula has now been implemented in PROLOG based directly on the two proof systems. However, we shall refer a presentation of this new verification system and a discussion of its relationship to that of Clarke, Emerson and Sistla [CES] to a separate paper.

Unfortunately, the proof systems presented do not allow alternations of minimal and maximal fixed points, imposing serious limitations on the expressiveness. However, our proof systems can be extended so that certain restricted types of alternations can be handled. In particular, temporal propositions involving nested $\Box$ ($\forall$G , $\exists$G) and $\Diamond$ ($\forall$F , $\exists$F) may be dealt with. A proof system which is sound and complete for the full μ–calculus [K] is however still unknown.

In the proof systems of sections 5 and 6, we may alternatively interpret a sequent of the form $\Gamma \vdash p : F$ as "the process p provides *evidence* for the satisfiability of the formula F under the assumptions of Γ. Under this interpretation, the proof systems provide a way of constructing evidence of satisfiability for a composite formula from similar evidence of satisfiability of its component formulas. Thus, the proof systems may be used as the basis for a system, which will automatically synthesize processes (implementations) satisfying a given (possibly recursive) formula (specification). Such a system would be comparable to both the method of Clarke and Emerson [CE] — which synthesizes models for specifications expressed in Branching Time Temporal Logic — and the method of Manna and Wolper [MW] — which synthesizes CSP–like programs from specifications expressed in Linear Time Temporal Logic. However, the implementation of a model–constructing system based upon the presented proof systems and a discussion of such a systems relationship to those in [CE,MW] will be presented in a future paper.

In conclusion, we believe that, being able to serve as the basis for both an automatic *model–checking* and an automatic *model–constructing* system, emphasizes the importance of the presented proof systems. Moreover, the *partial* and *total correctness* of these derived automatic systems may be based directly on the soundness and completeness theorems of **MIN** and **MAX**.

Acknowledgement

The author would like to thank Colin Stirling for his many helpfull suggestions and general support during the making of this paper. Also thanks to Robin Milner, David Park, Joachim Parrow and Mads Tofte for many interesting discussions on the subject of this paper.

References

[A83], Aczel, P.: *An Introduction to Inductive Definitions*, North-Holland, in the Handbook of Mathematical Logic, 1983.

[BMP], Ben-Ari, M., Pnueli, A. and Manna, Z.: *The Temporal Logic of Branching Time*, Acta Informatica, 20, pp 207-226, 1983.

[BT], Bloom and Troeger: *A Logical Characterization of Observation Equivalence*, TCS, vol 35, no 1, 1985

[BR], Brookes, S. and Rounds, R.: *Behavioural Equivalence Relations Induced by Programming Logics*, LNCS 154, 1983.

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

[CES], Clarke, E.M, Emerson, E.A, Sistla, A.P.: *Automatic Verification of Finite State Concurrent Systems Using Temporal Logic Specifications: A Practical Approach*, Proc 10th ACM POPL, 117-126, 1983.

[GS84], Graf, S. and Sifakis, J.: *A modal Characterization of observational congruence on finite terms of CCS*, LNCS 172, 1984.

[GS86], Graf, S. and Sifakis, J.: *A Logic for the Specification and Proof of Regular Controllable Processes of CCS*, Acta Informatica, 23, pp 507-527, 1986.

[GS85], Graf, S. Sifakis, J.: *A Logic for the Description of Non-deterministic Programs and Their Properties*, Information and Control, vol. 68, nos 1–3, 1986.

[HM], Hennessy, M. and Milner, R.: *Algebraic Laws for Nondeterminism and Concurrency*, Journal of the Association for Computing Machinery, 1985.

[K], Kozen, D.: *Results on the Propositional Mu-Calculus*, 9th ICALP, LNCS 140, 1982.

[L], Larsen, K.G.: *Proof Systems for Hennessy–Milner Logic with Recursion*, Aalborg University, Department of Mathematics and Computer Science, R 87–20, 1987.

[MW], Manna, Z. and Wolper, P. *Synthesis of Communicating Processes from Temporal Logic Specifications*, LNCS 131 , 1981.

[M80], Milner, R.: *A Calculus of Communicating Systems*, LNCS 92, 1980.

[M81], Milner, R. : *A Modal Characterization of Observable Machine-behaviour*, LNCS 112, 1981.

[M83], Milner, R.: *Calculi for Synchrony and Asynchrony*, TCS 25, pp 267-310, 1983.

[P] Park, D.: *Concurrency and automata on infinite sequences*, LNCS 84, 1980.

[Pl], Plotkin, G.: *A Structural Approach to Operational Semantics*, DAIMI-FN-19, Aarhus University, Computer Science Department, Denmark, 1981.

[Pn85], Pnueli, A.: *Linear and Branching Structures in the Semantics and Logics of Reactive Systems*, 12th ICALP, LNCS 194, 1985.

[St83], Stirling, C.: *A Proof Theoretic Characterization of Observationl Equivalence*, FCT-TCS Bangalore, To appear in TCS, 1983.

[St85A], Stirling, C.: *A Complete Proof System for a Subset of SCCS*, LNCS 185, to appear in CAAP'85, 1985.

[St85B], Stirling, C.: *A Compositional Modal Proof System for a Subset of CCS*, 12th ICALP, LNCS 194, full version to appear in TCS., 1985.

[St86], Stirling, C.: *Modal Logics for Communicating Systems*, to appear in TCS., 1986.

[T], Tarski, A.: *A Lattice-Theoretical Fixpoint Theorem and its Applications*, Pacific Journal of Math. 5, 1955.

[W84], Winskel, G.: *On the Composition and Decomposition of Assertions*, LNCS 197, 1984.

[W85], Winskel, G.: *A Complete Proof System for SCCS with Modal Assertions*, Technical Report, Computer Laboratory, University of Cambridge, 1985.