formula ϕ_P is defined with the property that for any formula F, $P \models F$ if and only if $\phi_P \rightarrow F$ is a valid formula. Thus deciding whether or not a process satisfies a property is reduced to testing for the validity of a formula in the underlying modal logic.

Much of this work is carried out within the framework of *pure* process algebras, where processes are determined by their ability to perform atomic or uninterpreted actions and the modal formula describe certain aspects of their behaviour vis. a vis. these actions. Thus satisfying the modal property $[a](\langle b\rangle \mathsf{tt} \vee [c]\mathsf{ff})$ means that every time a process performs the atomic action a either it can subsequently perform the action b or it can not perform the action c. The aim of this paper is to extend this work to what we call *message passing* process algebras, i.e. process descriptions in which these actions have a certain interpretation; the reception and transmission of data along conceptual channels. Thus, for example,

$$P \Longleftarrow c?x.\mathsf{if}\ x = 0\ \mathsf{then}\ c!x.P\ \mathsf{else}\ d!(x+1).P$$

describes a process which can input a value on the channel c and output its successor on the channel d unless it is zero in which case it is output on the original channel.

Properties of such processes depend, of course, on properties of the underlying message–space and thus proof systems for inferring such properties will also have to derive theorems valid in this message–space. Here we take the approach advocated in [Hen91]; we factor out as much as possible all this auxiliary reasoning. In the extended modal logic we can express assertions of the underlying message–space but in the accompanying proof systems theorems involving such assertions are obtained for "free"; i.e. we assume the existence of an oracle which will determine the truth or otherwise of these assertions. In reality, in any implementation, we would call on an auxiliary proof system to establish such assertions. In this paper we will allow any first order formula as an assertion about the underlying message–space.

Nevertheless we do have to generalise the modal operators $\langle a\rangle$ and $[a]$ to the setting of message–passing on channels. For output actions this generalisation is straightforward. For each channel name c and data–variable x we have two modalities $\langle c!x\rangle$ and $[c!x]$. Intuitively a process P satisfies $\langle c!x\rangle F$ if it is possible for it to output on the channel c a message v and in doing so evolve to a state P' such that P' satisfies the formula $F[v/x]$; the interpretation of $[c!x]$ is similar. However input actions are somewhat more complicated and the form of the modal logic depends on the operational semantics of the process language. There are two natural generalisations of the standard operational semantics of pure processes, called *early* and *late* in [HL92, MPW92]. Here we use the *late* operational semantics which expresses the ability of processes to perform abstract actions of the form $P \xrightarrow{c?} (x)A$ where $(x)A$ is an *abstraction* which describes what will happen when a value is received on the channel c. Thus we have two further simple modalities $\langle c?\rangle$ and $[c?]$ associated with input actions but now these must be followed by formulae expressing properties of abstractions; these take the form of quantified assertions, of the form $\exists x F$ and $\forall x F$. The end result is a very powerful language for describing properties of message–passing systems. For example

$$[c?]\forall x[d!y](y = x + 1)$$

assume that *substitutions*, mappings from X to expressions, act in the standard way on expressions; we use $e[\sigma]$ to denote the expression which results from applying the substitution σ to the expression e. An *evaluation* ρ is a particular type of substitution; it maps variables to values and we assume that for every expression e, $e[\rho]$ always evaluates to a value. Moreover we assume that this evaluation depends only on the variables occurring in e; if we let $fv(e)$ to denote the variables which occur in e then $e[\rho] = e[\rho']$ whenever ρ and ρ' agree on $fv(e)$. If e contains no variables then $e[\rho]$ is independent of ρ and we denote this value by $[\![e]\!]$. In a similar manner we assume a language for boolean expressions, ranged over by b, and substitutions and evaluations act on boolean expressions in a similar manner.

The syntax of process expression is defined by the following BN-form:

$$T ::= \mathbf{0} \mid \tau.T \mid c?x.T \mid c!e.T \mid T + T' \mid \mathsf{if}\ b\ \mathsf{then}\ T\ \mathsf{else}\ T' \mid T|T' \mid T\backslash L \mid T[f] \mid C(e_1, \ldots, e_n)$$

We will first briefly explain these process constructions. Later a formal semantics will be given. The term $\mathbf{0}$ represents a process which does nothing while $\tau.P$ will do an internal action τ and then behave like the process P. The process $c?x.T$ will receive a value v on channel c and then behaves like the process $T[v/x]$, where we use the standard notation $T[v/x]$ to describe the substitution of v for all free occurrences of the variable x in T. Thus in $c?x.T$ the prefix $c?x$ is a binding operator for the value variable x and this leads to the standard definition of *free* and *bound* occurrences of variables in terms. The application of a substitution or evaluation, defined above on value expressions, is generalised to terms; thus for example $T[\sigma]$ denotes the result of substituting in T all free occurrences of x by $\sigma(x)$. We will also take for granted the definition of α-conversion, $\equiv_\alpha$, on terms. Returning to our informal explanation of the language $c!e.P$ will send the value of the expression e on channel c and subsequently behave like P. The operator $+$ represent choice: $P + P'$ will either proceed like P or P'. The process P orBPGrocessexpre57(y)]TJd(subst/R162

terms, i.e. terms which contain no occurrences of free variables; we use $P, Q, \ldots$ to range over these processes. Such an operational semantics is given in Figure 1. It presupposes that each process constant has associated with it a definition as

Act $\dfrac{}{\tau.P \xrightarrow{\tau} P}$

$$
\begin{array}{rcl}
F & ::= & B \mid F_1 \vee F_2 \mid F_1 \wedge F_2 \mid \langle \tau \rangle F \mid [\tau] F \mid \\
 & & \langle c!x \rangle F \mid [c!x] F \mid \langle c? \rangle G \mid [c?] G \\
G & ::= & \exists x F \mid \forall x F
\end{array}
$$

Figure 2: Syntax of the modal logic

$P \models B \qquad : \; [\![$

and universal quantification over transitions. That is, in order to satisfy a formula $\langle\alpha\rangle F$, a process must possess some α–transition leading to a configuration (a process or an abstraction) satisfying F. Dually, for a process to satisfy a formula $[\alpha]F$, any α–transition must lead to configurations satisfying F.

The logic characterizes the (late) bisimulation equivalence between message passing processes as the following theorem claims.

Theorem 2.2 (Modal characterization) *$P \sim Q$ if and only if for all (closed) modal formula F, $P \models F \Leftrightarrow Q \models F$.*

Proof: $\Rightarrow$: Suppose $P \sim Q$ and $P \models F$, we can show that $Q \models F$ by induction on the structure of F. It is easy when F has the forms $B, F_1 \wedge F_2, F_1 \vee F_2, \langle\tau\rangle F', [\tau]F'$. If F is $\langle c!x\rangle F'$, then there exists $P \xrightarrow{c!} (v, P')$ such that $P' \models F[v/x]$. Because $P \sim Q$, so $Q \xrightarrow{c!} (v, Q')$ for some Q' with $P' \sim Q'$, and by induction hypothesis $Q' \models F[v/x]$. Thus $Q \models \langle c!x\rangle F'$. Similarly we can prove the case when F is $[c!x]F'$. If F is $\langle c?\rangle G$, then there exists $P \xrightarrow{c?} (x)T$ such that $(x)T \models G$. Because $P \sim Q$, so $Q \xrightarrow{c?} (y)U$ for some $(y)U$ with

3 Proving Properties of Message Passing Processes

In this section, we will look at how to reason about whether a process satisfies a modal property. From the definition of the satisfaction relation $\models$ it seems quite natural to express whether $P \models F$ holds as a first order formula $R_{P,F}$, such that $[\![R_{P,F}]\!] = \mathtt{tt}$ just in case $P \models F$ holds. This idea is attractive because if we can do this for any process P and modal formula F, and assume that we have a first order theory about the domain V at our disposal, then the problem of whether $P \models F$ holds is reduced to the problem of whether we can establish $R_{P,F}$ in the first order theory. We will call such $R_{P,F}$ the characteristic formula of P satisfying F and write it as $P \,\mathtt{sat}\, F$ in the rest of the paper.

Now let us see how to construct $P \,\mathtt{sat}\, F$ for a given process P and formula F. Naturally the construction should be done inductively on the structure of F. Take the case $F_1 \wedge F_2$ as an example; assuming we have constructed $P \,\mathtt{sat}\, F_1$ and $P \,\mathtt{sat}\, F_2$, according to the definition of $\models$, it is reasonable to define $P \,\mathtt{sat}\, F_1 \wedge F_2$ as $P \,\mathtt{sat}\, F_1 \wedge P \,\mathtt{sat}\, F_2$. Take

sat ⟨∀∃∉∀∀≠⇒|∈∈{∀}⇐⟨∀∃∉∀∀≠⇒|∉∈{∈∄∉⇐⟨⇒≠⇒∃∃∃≠⇒
We avedyn cnced
a F forP
P
. This
10026(ticshumint2051 .20TdIt(e)-1 000. lsong the elati(dal)-15998.sem
Tj/49.1620 24)T0217019960]TJ[n79.80214.4Td[(w)999. 49(e)]TJ8020Td(mite)T 048.48010Td(to)Tj17.11990Td(coiorder)

Act $\dfrac{}{\tau.T \xrightarrow{\mathrm{tt},\tau} T}$ $\dfrac{}{c!e.T \xrightarrow{\mathrm{tt},c!} (e,T)}$ $\dfrac{}{c?x.T \xrightarrow{\mathrm{tt},c?} (x)T}$

Sum $\dfrac{T \xrightarrow{b,\alpha} A}{T + U \xrightarrow{b,\alpha} A}$ $\dfrac{U \xrightarrow{b,\alpha} A}{T + U \xrightarrow{b,\alpha} A}$

Cond $\dfrac{T \xrightarrow{b,\alpha} A}{\mathsf{if}\ b'\ \mathsf{then}\ T\ \mathsf{else}\ U \xrightarrow{b \wedge b',\alpha} A}$ $\dfrac{U \xrightarrow{b,\alpha} A}{\mathsf{if}\ b'\ \mathsf{then}\ T\ \mathsf{else}\ U \xrightarrow{b \wedge \neg b',\alpha} A}$

Par $T \xrightarrow{b,\tau} T$

Now look at the third Par rule, it is possible that the bound variable x in $(x)T'$ also appears free in U. Obviously these two appearances of x should not be confused. In order to avoid such confusion, we assume a function new which takes a set of variables X as argument and produce a new variable which is not in X, and use new to change the bound variable.

The first lemma shows that this operational semantics for open terms is in fact a generalization of the operational semantics for processes in Figure 1.

Lemma 3.1 *For any process P, $P \xrightarrow{\alpha} A$ if and only if $P \xrightarrow{b,\alpha} A$ for some (closed) b with $[\![b]\!] = \mathsf{tt}$.*

Proof: The statement of the lemma assumes that a closed value expression e is identified with its value $[\![e]\!]$. The proof consists of two routine inductions, one on the rules in Figure 1, and the other on those of Figure 4. □

The next result states that in some sense the symbolic operational semantics is preserved under substitutions.

Lemma 3.2 *For any term T and substitution σ, $T[\sigma] \xrightarrow{b',\alpha} A'$ if and only if $T \xrightarrow{b,\alpha} A$ for some b, A such that $b' \equiv b[\sigma], A' \equiv_\alpha A[\sigma]$.*

Proof: Each direction is proved by induction on the rules in Figure 4.

□

With these symbolic actions we can now give the

The case $\langle c!x\rangle F$:

$$
\begin{array}{lll}
 & [\![P \,\mathtt{sat}\, \langle c!x\rangle F]\!] = \mathtt{tt} & \\
\Leftrightarrow & [\![\bigvee_{P \xrightarrow{b,c!} (e,P')} b \wedge P' \,\mathtt{sat}\, F[e/x]]\!] = \mathtt{tt} & \text{def. of } \mathtt{sat} \\
\Leftrightarrow & \text{for some } e, P', P \xrightarrow{b,c!} (e,P') \text{ with } [\![b]\!] = \mathtt{tt} & \\
 & \text{and } [\![P' \,\mathtt{sat}\, F[e/x]]\!] = \mathtt{tt} & \\
\Leftrightarrow & \text{for some } e, P', P \xrightarrow{c!} ([\![e]\!], P') \text{ and } [\![P' \,\mathtt{sat}\, F[[\![e]\!]/x]]\!] = \mathtt{tt} & \text{Lemma .1} \\
\Leftrightarrow & \text{for some } e, P', P \xrightarrow{c!} ([\![e]\!], P') \text{ with } P' \models F[[\![e]\!]/x] & \text{ind. hyp.} \\
\Leftrightarrow & P \models \langle c!x\rangle F & \text{def. of } \models
\end{array}
$$

The case $[c?]G$:

$$
\begin{array}{lll}
 & [\![P \,\mathtt{sat}\, [c?]G]\!] = \mathtt{tt} & \\
\Leftrightarrow & [\![\bigwedge_{P \xrightarrow{b,c!} (y)T} b \rightarrow (y)T \,\mathtt{sat}\, F]\!] = \mathtt{tt} & \text{def. of } \mathtt{sat} \\
\Leftrightarrow & \text{whenever } P \xrightarrow{b,c?} (y)T \text{ with } [\![b]\!] = \mathtt{tt} & \\
 & \text{then } [\![(y)T \,\mathtt{sat}\, G]\!] = \mathtt{tt} & \\
\Leftrightarrow & \text{whenever } P \xrightarrow{c?} (y)T \text{ then } [\![(y)T \,\mathtt{sat}\, G]\!] = \mathtt{tt} & \text{Lemma .1} \\
\Leftrightarrow & \text{whenever } P \xrightarrow{c?} (y)T \text{ then } (y)T \models G & \text{ind. hyp.} \\
\Leftrightarrow & P \models [c?]G & \text{def. of } \models
\end{array}
$$

The case $\forall x F$:

$$
\begin{array}{lll}
 & [\![(y)T \,\mathtt{sat}\, \forall x F]\!] = \mathtt{tt} & \\
\Leftrightarrow & [\![\forall z T[z/y] \,\mathtt{sat}\, F[z/x]]\!] = \mathtt{tt} & \text{def. of } \mathtt{sat} \\
\Leftrightarrow & \text{for all } v \in V, [\![T[v/y] \,\mathtt{sat}\, F[v/x]]\!] = \mathtt{tt} & \\
\Leftrightarrow & \text{for all } v \in V, T[v/y] \models F[v/x] & \text{ind. hyp.} \\
\Leftrightarrow & (y)T \models \forall x F & \text{def. of } \models
\end{array}
$$

□

Proposition 3.4 *For any substitution σ, the following logical equivalences hold:*

$$
\begin{array}{lll}
(T \,\mathtt{sat}\, F)[\sigma] & \equiv & T[\sigma] \,\mathtt{sat}\, F[\sigma] \\
((y)T \,\mathtt{sat}\, G)[\sigma] & \equiv & ((y)T)[\sigma] \,\mathtt{sat}\, G[\sigma]
\end{array}
$$

Proof: By induction on the stucture of F, using Lemma .2. We give four example cases.

The case $[\tau]F$:

$$\begin{array}{rl} & (T\ \mathtt{sat}\ [\tau]F)[\sigma] \\ \equiv & (\bigwedge_{T \xrightarrow{b,\tau} T'} b \to T \end{array}$$

in the process and formulae.

$$
\begin{array}{lll}
 & ((y)T \ \mathtt{sat}\ \exists x F)[\sigma] & \\
\equiv & (\exists z T[z/y] \ \mathtt{sat}\ F[z/x])[\sigma] & \text{def. of } \mathtt{sat} \\
\equiv & \exists z((T[z/y] \ \mathtt{sat}\ F[z/x])[\sigma]) & \\
\equiv & \exists z(T[z/y][\sigma] \ \mathtt{sat}\ F[z/x][\sigma]) & \text{ind. hyp.} \\
\equiv & \exists z(T[\sigma][z/y] \ \mathtt{sat}\ F[\sigma][z/x]) & \\
\equiv & (y)(T[\sigma]) \ \mathtt{sat}\ \exists x(F[\sigma]) & \text{def. of } \mathtt{sat} \\
\equiv & ((y)T)[\sigma] \ \mathtt{sat}\ (\exists x F)[\sigma] &
\end{array}
$$

□

Now we can prove the main result about characteristic formulae:

Theorem 3.5 (Characteristic formulae) *For any process term T and formula F, $T \ \mathtt{sat}\ F$ characterizes the satisfaction relation in the sense that for any evaluation ρ*

$$[\![(T \ \mathtt{sat}\ F)[\rho]]\!] = \mathtt{tt} \ \Leftrightarrow \ T[\rho] \models F[\rho]$$

Proof: Follows directly from the above two results.

□

Note that $P[\rho$

$$\textsf{Id}\quad \frac{}{B \vdash T{:}\, B} \qquad \textsf{Cut}\quad \frac{B_1 \vdash T{:}\quad \ldots\ldots B_n \vdash T{:}}{\bigvee_{1\le i\le n} B_i \vdash T{:}}\quad n \ge 0$$

$$\textsf{Cons}\ \frac{B_1 \vdash T{:}}{B_2 \vdash T{:}}\quad B_2 \Rightarrow B_1 \qquad \textsf{Ex}\quad \frac{B \vdash T{:}}{\exists x B \vdash T{:}}\quad x \text{ does not occur free in } T,$$

$$\vee\quad \frac{B \vdash T{:}\ \ _1}{B \vdash T{:}\ \ _1 \vee\ \ _2} \qquad \frac{B \vdash T{:}\ \ _2}{B \vdash T{:}\ \ _1 \vee\ \ _2}$$

$$\wedge\quad \frac{B \vdash T{:}\ \ _1 \qquad B \vdash T{:}\ \ _2}{B \vdash T{:}\ \ _1 \wedge\ \ _2}$$

$$\langle\tau\rangle\quad \frac{B \vdash T'{:}}{B \wedge b \vdash T{:}\langle\tau\rangle}\quad T \xrightarrow{b,\tau} T'$$

$$[\tau]\quad \frac{B \wedge b_1 \vdash T_1{:}\quad ,\ldots\ldots, B \wedge b_n \vdash T_n{:}}{B \vdash T{:}[\tau]}\quad \{(b_1,T_1),\ldots,(b_n,T_n)\} = \{(b,T') \mid T \xrightarrow{b,\tau} T'\}$$

$$\langle c!\rangle\quad \frac{B \vdash T'{:}\quad [e/x]}{B \wedge b \vdash T{:}\langle c!x\rangle}\quad T \xrightarrow{b,c!} (e,T')$$

$$[c!]\quad \frac{B \wedge b_1 \vdash T_1{:}\quad [e_1/x], \ldots\ldots, B \wedge b_n \vdash T_n{:}\quad [e_n/x]}{B \vdash T{:}[c!x]}$$

where $\{(b_1,T_1,e_1),\ldots,(b_n,T_n,e_n)\} = \{(b,T',e) \mid T \xrightarrow{b,c!} (e,T')\}$

$$\langle c?\rangle\quad \frac{B \vdash (y)T'{:}G}{B \wedge b \vdash T{:}\langle c?\rangle G}\quad T \xrightarrow{b,c?} (y)T'$$

$$[c?]\quad \frac{B \wedge b_1 \vdash (y_1)T_1{:}G, \ldots\ldots, B \wedge b_n \vdash (y_n)T_n{:}G}{B \vdash T{:}[c?]G}$$

where $\{(b_1,(y_1)T_1),\ldots,(b_n,(y_n)T_n)\} = \{(b,T') \mid T \xrightarrow{b,c?} (y)T'\}$

$$\forall\quad \frac{B \vdash T{:}}{B \vdash (x)T{:}\forall x}\quad x \text{ does not occur free in } B$$

$$\exists\quad \frac{B \vdash T[e/x]{:}\quad [e/x]}{B \vdash (x)T{:}\exists x} \qquad \alpha\quad \frac{B \vdash T'{:}\quad '}{B \vdash T{:}}\quad T' \equiv_\alpha T,\quad ' \equiv_\alpha$$

Figure 6: The Proof Rules

Theorem 4.1 (Soundness) *If* $B \vdash T{:}F$ *then, for any evaluation* ρ, $[\![B[\rho]]\!] = \mathtt{tt}$ *implies* $T[\rho] \models F[\rho]$.

Proof: First note that $T[\rho] \models F[\rho]$, for any such substitution ρ, if and only if $[\![(T\ \mathtt{sat}\ F)[\rho]]\!] = \mathtt{tt}$ by Theorem .5. So it is sufficient to prove $B \vdash T{:}F$ implies $B \Rightarrow T\ \mathtt{sat}\ F$. Therefore we only need to show that the rules of the proof system preserve soundness in this sense. Each case follows from simple identities involving $\Rightarrow$ and the logical operators. For example the soundness of the $[c?]$ rule relies on the following facts:

1. $B \wedge b \Rightarrow F$ if and only if $B \Rightarrow (b \rightarrow F)$,
2. $B \Rightarrow \bigwedge_{i \in S} F_i$ if and only if $B \Rightarrow F_i$ for all $i \in S$.

To prove the soundness of $\forall$ rule, we have to show that if $B \Rightarrow T\ \mathtt{sat}\ F$ and x does not occur free in B then $B \Rightarrow (x)T\ \mathtt{sat}\ \forall xF$. Notice that $(x)T\ \mathtt{sat}\ \forall xF \equiv \forall x T\ \mathtt{sat}\ F$. The result now follows since if x does not occur free in B then $B \Rightarrow T\ \mathtt{sat}\ F$ implies $B \Rightarrow \forall x T\ \mathtt{sat}\ F$.

To prove the soundness of $\exists$ rule, we have to show that if $B \Rightarrow T[e/x]\ \mathtt{sat}\ F[e/x]$ then $B \Rightarrow (x)T\ \mathtt{sat}\ \exists xF$. Here $T[e/x]\ \mathtt{sat}\ F[e/x]$ is logically equivalent to $(T\ \mathtt{sat}\ F)[e/x]$ which in turn implies $\exists x T\ \mathtt{sat}\ F$.

□

The completeness of the system depends on the following result:

Lemma 4.2 *For any term* T *and formula* F, $T\ \mathtt{sat}\ F \vdash T{:}F$.

Proof: By induction on the size of F. We give four cases.

In the case $\langle c!x\rangle F$, because of the construction of $T\ \mathtt{sat}\ \langle c!x\rangle F$, we have to show that $\bigvee_{T \xrightarrow{b,c!} (e,T')} b \wedge T'\ \mathtt{sat}\ F[e/v] \vdash T{:}\langle c!x\rangle F$. By the induction hypothesis, for all $T \xrightarrow{b,c!} (e,T')$, $T'\ \mathtt{sat}\ F[e/x] \vdash T'{:}F[e/x]$, so $b \wedge T'\ \mathtt{sat}\ F[e/x] \vdash T{:}\langle c!x\rangle F$ by rule $\langle c!\rangle$. So $\bigvee_{T \xrightarrow{b,c!} (e,T')} b \wedge T'\ \mathtt{sat}\ F[e/x] \vdash T{:}\langle c!x\rangle F$ by rule Cut.

When it is $[c?]G$, by the definition of $T\ \mathtt{sat}\ [c?]G$, we have to show that $\bigwedge_{T \xrightarrow{b,c?} (y)T'} b \rightarrow (y)T'\ \mathtt{sat}\ G \vdash T{:}[c?]G$. By the induction hypothesis, for all R such that $T \xrightarrow{b_R,c?} (y)R$, $(y)R\ \mathtt{sat}\ G \vdash (y)R{:}G$, then $(\bigwedge_{T \xrightarrow{b,c?} (y)T'} b \rightarrow (y)T'\ \mathtt{sat}\ G) \wedge b_R \vdash (y)R{:}G$ by rule Cons because $(\bigwedge_{T \xrightarrow{b,c?} (y)T'} b \rightarrow (y)T'\ \mathtt{sat}\ G) \wedge b_R \Rightarrow (y)R\ \mathtt{sat}\ G$. Then by rule $[c?]$, $\bigwedge_{T \xrightarrow{b,c?} (y)T'} b \rightarrow (y)T'\ \mathtt{sat}\ G \vdash T{:}[c?]G$.

When it is $\forall xF$, by the definition of $(y)T\ \mathtt{sat}\ \forall xF$, we have to show that $\forall z.T[z/y]\ \mathtt{sat}\ F[z/x] \vdash (y)T{:}\forall xF$. By the induction hypothesis,

$$T[z/y]\ \mathtt{sat}\ F[z/x] \vdash T[z/y]{:}F[z/x]$$

Because $\forall z.T[z/y]\ \mathtt{sat}\ F[z/x] \Rightarrow T[z/y]\ \mathtt{sat}\ F[z/x]$, by rule Cons we have

$$\forall z.T[z/y]\ \mathtt{sat}\ F[z/x] \vdash T[z/y]{:}F[z/x]$$

Then apply rule $\forall$ we have $\forall z.T[z/y]\,\mathtt{sat}\,F[z/x] \vdash (z)T[z/y]{:}\forall z F[z/x]$. Now $(z)T[z/y]{:}\forall z F[z/x]$ can be α

Theorem 4.6 *Let T, U be any two process and B be a boolean expression. Then $T \sim^B U$ if and only if for every modal formula F, $B' \vdash T{:}F \Leftrightarrow B' \vdash U{:}F$ for any B' such that $B' \Rightarrow B$.*

Proof:
One direction follows immediately from the above Lemma since if $B' \Rightarrow B$ and $T \sim^B U$ then $T \sim^{B'} U$. To prove the converse suppose $B' \vdash T{:}F \Leftrightarrow B' \vdash U{:}F$ for any modal formula F for any B' such that $B' \Rightarrow B$. We will show that $T \sim^B U$. For that we only need to show that if $[\![B[\rho]]\!] = \mathtt{tt}$ for any evaluation ρ then $T[\rho] \sim U[\rho]$, and by Theorem 2.2 we only need to show that for any closed F, $T[\rho] \models F \Leftrightarrow U[\rho] \models F$.

So suppose $[\![B[\rho]]\!] = \mathtt{tt}$ $T[\rho] \models F$; we will show that $U[\rho] \models F$. Because $T \,\mathtt{sat}\, F \vdash T{:}F$ by Lemma 4.2 and $B \wedge T \,\mathtt{sat}\, F \Rightarrow B$, so $B \wedge T \,\mathtt{sat}\, F \vdash T{:}F$. Thus $B \wedge T \,\mathtt{sat}\, F \vdash U{:}F$. Now because $T[\rho] \models F[\rho]$ so $[\![(T \,\mathtt{sat}\, F)[\rho]]\!] = [\![T[\rho] \,\mathtt{sat}\, F[\rho]]\!] = \mathtt{tt}$, and moreover $[\![B]\!] = \mathtt{tt}$, so $[\![(B \wedge T \,\mathtt{sat}\, F)[\rho]]\!] = \mathtt{tt}$, so $U[\rho] \models F$ by the soundness of the proof system. □

5 Conclusion

In this paper we have suggested a first–order modal logic for defining properties of message passing processes and shown how at least some of the techniques which apply to the use of propositional modal logic and pure process algebras, [Lar88, Sti87] can be extended to this setting. The first result is the definition of a characteristic formula for each process and property a formula in the first–order modal logic, which is logically equivalent to $\mathtt{tt}$ if and only if the process enjoys the property. The second is a sound and complete proof system for proving that a specific process enjoys a specific property. Of course this does not make the task of proving such a statement trivial. Moreover the proof system is modulo