[CGB] E. Clarke, O. Grumberg and M.C. Browne. *Reasoning About Networks With Many Identical Finite–State Processes*, Carnegie Mellon University, Pittsburg, October 1986.

[CPS1] R. Cleaveland, J. G. Parrow and B. Steffen. *The Concurrency Workbench: Operating Instructions*, University of Edinburgh, Laboratory for Foundations of Computer Science, Technical Notes 10, September 1988.

[CPS2] R. Cleaveland, J. G. Parrow and B. Steffen. *The Concurrency Workbench*, In Proceedings of the Workshop on Automatic Verification Methods for Finite State Systems, LNCS 407, 1989

[CPS3] R. Cleaveland, J. G. Parrow and B. Steffen. *A Semantics based Verification Tool for Finite State Systems*, In pro. of the Ninth International Symposium on Protocol Specification, Testing, and Verification; North Holland, 1989.

[CH] R. Cleaveland and M. Hennessy. *Testing Equivalence as a Bisimulation Equivalence.* In Proceedings of the Workshop on Automatic Verification Methods for Finite State Systems, LNCS 407, 1989

[CS1] R. Cleaveland and B. Steffen. *A Preorder for Partial Process Specifications*, In Proceedings CONCUR'90, LNCS 458, 1990

[CS2] R. Cleaveland and B. Steffen. *Computing Behavioural Preorders, Logically*, submitted for ICALP'91

[GIZ] J.C.Godskesen, A.Ingólfsdóttir, M.Zeeberg. *Fra Hennessy-Milner logik til CCS-processer*, M.Sc. Thesis, Aalborg University Center 1987.

[GS1] S. Graf and J. Sifakis. *A Modal Characterization of Observational Congruence on Finite Terms of CCS*, Information and Control, pp. 125-145, Vol 68, 1986.

[GS2] S. Graf and J. Sifakis. *A Logic for the Description of Non-deterministic Programs and Their Properties*, Information and Control, vol 68, 1986.

[GS3] S. Graf and J, Sifakis. *A Logic for the Specification and Proof of Regular Controllable Processes of CCS*, Acta Informatica, 23, 1986.

[HM] M. Hennessy, R. Milner. *Algebraic Laws for Nondeterminism and Concurrency*, Journal of the Association for Computing Machinery, Vol. 32, pp. 137-161, 1985.

[Ho] S. Holmström. *Hennessy-Milner Logic with Recursion as a Specification Language, and a Refinement Calculus based on it*, Report 44 Programming Methodology Group, University of Göteborg, 1988

[Ko] D. Kozen. *Results on the Propositional Mu-Calculus*, TCS 27, pp. 333-354, North Holland, 1983.

[La] K. Larsen. *Proof Systems for Hennessy–Milner Logic with Recursion*, in Proceedings CAAP 1988.

[Mi1] R. Milner. *A Calculus for Communicating Systems*, LNCS 92.

[Mi2] R. Milner. *A Modal Characterization of Observable Machine-Behaviour*, LNCS 112, pp. 25-34, Berlin 1981.

[Mi3] R. Milner. *Communication and Concurrency*, Prentice Hall, 1989.

[PS] G. Plotkin and C. Stirling. *A Framework for Intuitionistic Modal Logics*, Theoretical Aspects of Reasoning about Knowledge, Monterey, 1986.

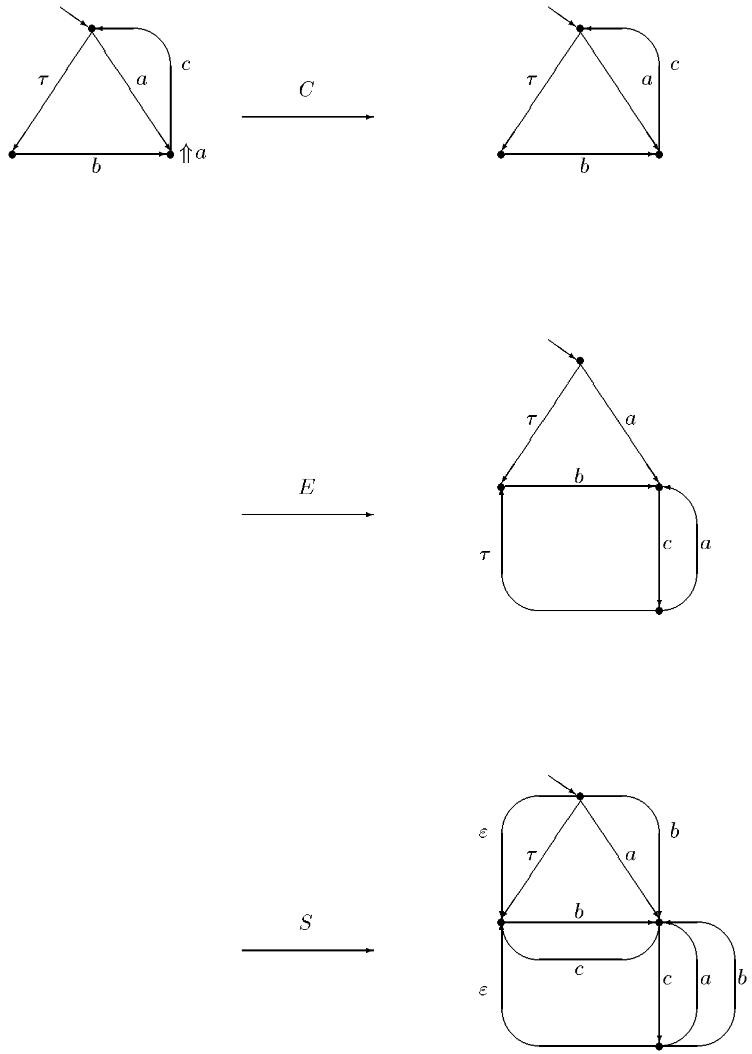
[Pn] A. Pnueli. *Linear and Branching Structures in the Semantics and Logics of Reactive Systems*, ICALP'85, LNCS 194.

[Ste] B. Steffen. *Characteristic Formulae*, in the Proceedings ICALP 1989.

[Sti] C. Stirling. *Modal Logics for Communicating Systems*, TCS 49, pp. 311-347, 1987.

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

[Wa] D.J. Walker. *Bisimulation and Divergence in CCS*, in Proceedings LICS 1988.



6 Conclusion

The results presented in this paper do not only demonstrate the expressive power of intuitionistically interpreted Hennessy-Milner logic with mutual recursion. They also build the theoretical background for a uniform and efficient method for the ***automatic verification*** of bisimulation–like relations between processes by means of ***model checking***: transform the underlying transition system wrt the chosen process relation as described in Section 5 [6], construct the characteristic formula for the left hand side process and check whether the right hand side process satisfies the characteristic formula constructed in the second step.

Whereas the first two steps of this method can be implemented directly along the lines described in this paper, the realization of the third step is described in [CS2]. The algorithm presented there is a variant of the model checking algorithm of Arnold and Crubille [AC], which:

- is linear in the size of the process times the size of the formula, and thus improves on the algorithm of [AC], whose worst-case complexity is proportional to the size of the process times the square of the size of the formula.
- is generalized to deal with an intuitionistic variant of the logic, which is necessary, in order to characterize preorder relations.

Thus according to Lemma 4.3, the worst case time complexity of our preorder checking algorithm is proportional to the product of the numbers of transitions of the two argument processes. In particular, for a fixed action set, it is of fourth order in the number of states of the transition systems involved. This improves on the previously published complexity result about preorder checking, which is sixth order in its worst case ([CPS2,CPS3]). The algorithm is going to be implemented in the Concurrency Workbench [CPS1,CPS2,CPS3], which is an analysis tool for finite state processes.

According to [CH], the method described in this paper can also be adapted to automatically verify testing equivalences and preorders, and only a minor modification of the definition of $\preceq$ would cover the specification–implementation relation presented in [CS1].

7 Acknowledgements

We would like to thank Luca Aceto, Rance Cleaveland, Matthew Hennessy, Michael Mendler, Robin Milner, Kim Larsen, Ernst–Rüdiger Olderog, Joachim Parrow, Andrea Peiffer, Colin Stirling and Bent Thomsen for their constructive comments. The work was supported by Science and Engineering Research Council and the ESPRIT BRA project CONCUR.

References

[AC] A. Arnold, P. Crubille. *Linear Algorithms to solve Fixed Point Equations on Transition Systems.* Information Processing Letters 29, 57-66, 1988

[6] This is the only part which depends on the process relation under consideration.

5 Application to CCS

In CCS, bisimulation–like relations are used to compare the computational power of processes. Depending on the needed sensitivity (deadlock, convergence, etc.) and the mathematical convenience (axiomatizability, computability and provability) one can choose between $\sqsubseteq$ (strong preorder), $\underset{\sim}{\sqsubset}$ (weak preorder), $\sim$ (strong equivalence), $\approx$ (observational or weak equivalence) and $\approx^c$ (observational congruence).— The definitions of the strong preorder, the weak preorder and the equivalences can be found in [Sti], [Wa] [5] and [Mi1,Mi3] respectively.— We are now going to show how to obtain characteristic formulae for each of these relations using the techniques introduced in the previous section.

The strong preorder $\sqsubseteq$ is just $\preceq$ in the context of a CCS–like transitions system, where divergence is defined by:

- $p\uparrow\varepsilon$ if p can engage in an infinite sequence of τ-moves and
- $p\uparrow a$ if either $p\uparrow\varepsilon$ or $p\xrightarrow{a}p'$ for some p', where $p'\uparrow\varepsilon$.

Thus the strong preorder case is covered by means of the Main Theorem 4.2. For the other four relations we define four process transformations: W (for 'weakening' the transition relation by collapsing τ–actions, which are regarded as ***internal*** or ***unobservable***), C (to obtain convergence), E (to extend the corresponding transition system in a way that excludes the start state as the destination of a transition) and S (for 'weakening', while maintaining the sensitivity for τ-moves of the start state). As before we represent processes as rooted transition systems (i.e. as pairs (T,p_0), where p_0 is the start state):

- $W((\mathbf{P},\longrightarrow,Act,\uparrow),p_0) =_{df} ((\mathbf{P},\Rightarrow,(Act\setminus\{\tau\})\cup\{\epsilon\},\Uparrow),p_0)$, where
  1. $\forall p,q\in\mathbf{P}\ \forall a\in Act\setminus\{\tau\}.\ p\overset{a}{\Longrightarrow}q$ iff $p\xrightarrow{\tau}{}^{*}\xrightarrow{a}\xrightarrow{\tau}{}^{*}q$
  2. $\forall p,q\in\mathbf{P}.\ p\overset{\epsilon}{\Longrightarrow}q$ iff $p\xrightarrow{\tau}{}^{*}q$, and
  3. $p\Uparrow\mu$ if there is a p' such that $p\overset{\epsilon}{\Longrightarrow}p'$ and $p'\uparrow\mu$ or $p'\uparrow\epsilon$
- C only changes the divergence predicate $\uparrow$ to the predicate that is constantly false.
- $E((\mathbf{P},\longrightarrow,Act,\uparrow),p_0) =_{df} ((\mathbf{P}\cup\{\bar{p_0}\},\longrightarrow',Act,\uparrow'),\bar{p_0})$, where $\longrightarrow'$ and $\uparrow'$ are given by:
  1. $\forall a\in Act.\ \overset{a}{\longrightarrow}{}' = \xrightarrow{a}\cup\{(\bar{p_0},p)\mid p_0\xrightarrow{a}p\}$ and
  2. $\forall\mu\in Act\cup\{\varepsilon\}.\ \uparrow'\mu = \begin{cases}\uparrow\mu\cup\{\bar{p_0}\} & \text{if } p_0\uparrow\mu\\ \uparrow\mu & \text{otherwise.}\end{cases}$

  In other words, $\longrightarrow'$ and $\uparrow'$ are the smallest extensions of $\longrightarrow$ and $\uparrow$, respectively, such that $\bar{p_0}$ and p_0 possess identical branching structures and divergence potentials (i.e. $\bar{p_0}$ is a copy of p_0).
- $S((\mathbf{P},\longrightarrow,Act,\uparrow),p_0) =_{df} ((\mathbf{P},\Rightarrow,Act\cup\{\epsilon\},\Uparrow),p_0)$, where

[5] Walker denotes the weak preorder by $\sqsubseteq$.

1. $\forall p,q\in\mathbf{P}\ \forall a\in Act\setminus\{\tau\}.\ p\overset{a}{\Longrightarrow}q$ iff $p\xrightarrow{\tau}{}^{*}\xrightarrow{a}\xrightarrow{\tau}{}^{*}q$
2. $\forall p,q\in\mathbf{P}.\ p\overset{\epsilon}{\Longrightarrow}q$ iff $p\xrightarrow{\tau}{}^{*}q$
3. $\forall q\in\mathbf{P}.\ p_0\overset{\tau}{\Longrightarrow}q$ iff $p_0\xrightarrow{\tau}{}^{+}q.$, and
4. $p\Uparrow\mu$ if there is a p' such that $p\overset{\epsilon}{\Longrightarrow}p'$ and $p'\uparrow\mu$ or $p'\uparrow\epsilon$

In particular, the transition relation "$\Rightarrow$" resulting from this transformation records τ–moves of the start state, i.e.: $p_0\overset{\tau}{\Longrightarrow}$ iff $p_0\xrightarrow{\tau}{}^{+}$.

Note, the distinction between ε and ϵ. This distinction originates from the fact that both, the weakening transformation and the total divergence predicate require their private actions. However, it is possible and often convenient to identify ε and ϵ, which actually both represent unobservable actions. One only must change the satisfaction relation $\models_\sigma$ for atomic sentences A as follows:

$$p\models_\sigma A \text{ iff } \{\mu\mid p\uparrow\mu\vee(\mu\neq\varepsilon\wedge\exists q\in\mathbf{P}.\,p\xrightarrow{\mu}q)\}\subseteq A.$$

According to the Main Theorem 4.2 characteristic formulae for the remaining four relations can now be obtained by means of the following properties of the transformations just defined:

Theorem 5.1 *Let $p,q\in\mathbf{P}$. Then we have:*

1. $p\underset{\sim}{\sqsubset}q$ *iff* $W(p)\sqsubseteq W(q)$
2. $p\sim q$ *iff* $C(p)\sqsubseteq C(q)$
3. $p\approx q$ *iff* $W(C(p))\sqsubseteq W(C(q))$
4. $p\approx^c q$ *iff* $S(E(C(p)))\sqsubseteq S(E(C(q)))$

W is the standard transformation for changing from the strong to the weak version of equivalence and preorder, and obviously, $\sqsubseteq$ (strong preorder) coincides with $\sim$ (strong equivalence) in the absence of divergence. This yields the first three properties of the transformations.

For observational congruence, $\approx^c$, the 'weakening' transformation must be refined to additionally record initial τ-moves. This refined transformation is realized in three steps. First, the divergence information is eliminated by setting the divergence predicate to false. Second, a new start node is created, which exactly 'behaves' like the original start node, but which is not reachable from inside the transition system (see E). This guarantees that a special treatment of the (new) start node only affects initial moves. Third, the process is transformed into its weak version, in a way, which records the possible τ-moves of the (new) start state (see S). The following figure illustrates this transformation. For the sake of clarity the last diagram does not display ε-edges that result from the reflexive closure of $\xrightarrow{\tau}$.

In the situation, where the processes p and q are modified to diverge on ε and a respectively, we would have the following characteristic declaration:

$$\begin{array}{lcl} X_p & \Longrightarrow & \langle a\rangle X_q \ \wedge \ \{a,b,c,\varepsilon\} \\ X_q & \Longrightarrow & \langle b\rangle X_p \ \wedge \ [b]X_p \ \wedge \ \langle c\rangle X_q \ \wedge \ [c]X_q \ \wedge \ \{a,b,c\}. \end{array}$$

Generalizing this idea, we arrive at the following definition of the characteristic formula for processes:

Definition 4.1 *Let $P = (\mathbf{P}, Act, \longrightarrow, \uparrow)$ be an extended transition system. Then the* characteristic declaration D_P *for P is defined by:*

$$X_p \ \Longrightarrow \ \Big(\bigwedge_{a,p'.p \xrightarrow{a} p'} \langle a\rangle X_{p'} \Big) \ \wedge \ \Big(\bigwedge_{a\in A_p . p\downarrow a} [a] \bigvee_{p'.p\xrightarrow{a}p'} X_{p'} \Big) \ \wedge \ A_p$$

with

$$A_p = \{\mu \mid p \xrightarrow{\mu} \ \ or \ \ p\uparrow\mu\}$$

for every $p \in \mathbf{P}$. The characteristic formula *of a process $p \in \mathbf{P}$ is now given by the variable X_p in this declaration. It will also be denoted by X_p.*

That this construction is indeed characteristic for $\preceq$ is subject of our main theorem:

Theorem 4.2 (Main Theorem)
Let $P = (\mathbf{P}, Act, \longrightarrow, \uparrow)$ be an extended transition system, D_P its characteristic declaration and $p, q \in \mathbf{P}$. Then we have: $p \preceq q \ \Leftrightarrow \ q \models_{\sigma_{D_P}} X_p$.

Proof:

$\Leftarrow$: Let $R = \{(p,q) \mid q \models_{\sigma_{D_P}} X_p\}$. We will show that R satisfies the defining conditions for $\preceq$, see Definition 2.1. Let $(p,q) \in R$.

In order to prove condition (1) of Definition 2.1, we can assume that $p \xrightarrow{a} p_1$. Then, as q satisfies the first part of (4.1), $\langle a\rangle X_{p_1}$, there exists a q_1 with $q \xrightarrow{a} q_1$ and $q_1 \models_{\sigma_{D_P}} X_{p_1}$. Now the definition of R yields $(p_1, q_1) \in R$ as desired.

If p is totally convergent, i.e. $p\downarrow\varepsilon$, we obtain $\varepsilon \notin A_p$. Now $q \models_{\sigma_{D_P}} A_p$ implies $q\downarrow\varepsilon$, which proves condition 2(a) of Definition 2.1.

For the proof of condition 2(b), assume that $p\downarrow a$ for $a \in Act$. Then it is enough to show that $q\downarrow a$ and, for every q_1, $q \xrightarrow{a} q_1$ implies $p \xrightarrow{a} p_1$ for some p_1 such that $(p_1, q_1) \in R$. We proceed by distinguishing two cases depending on whether $a \in A_p$ or not.

In case of $a \in A_p$, we have $p \xrightarrow{a} p_1$ for some p_1 because of $p\downarrow a$. This implies that there exists a conjunct $[a]F$ in the characteristic formula of p, forcing q to converge on a, i.e. $q\downarrow a$. Thus it remains to show that p matches the a-transitions of q. Suppose $q \xrightarrow{a} q_1$. Then q_1 satisfies $\bigvee_{p'.p\xrightarrow{a}p'} X_{p'}$, which is a non-empty disjunction because of $a \in A_p$ and $p\downarrow a$. This implies $q_1 \models_{\sigma_{D_P}} X_{p_1}$ for some p_1 such that $p \xrightarrow{a} p_1$. Thus $p \xrightarrow{a} p_1$ where $(p_1, q_1) \in R$ as desired.

Otherwise, $a \notin A_p$ implies that q converges on a, and that it does not possess any a-derivative, which proves the validity of condition 2(b) of Definition 2.1 also in this case.

$\Rightarrow$: Let the environment $\sigma_{\preceq}$ be defined by:

$$\sigma_{\preceq}(X_p) \ = \ \{q \mid p \preceq q\}$$

Then, according to our semantic definition for declarations, it is enough to show that $\sigma_{\preceq}$ is a post-fixpoint of the functional $[\![D_P]\!]$ that is induced by the characteristic declaration D_P, i.e.

$$\sigma_{\preceq} \ \leq \ [\![D_P]\!]\sigma_{\preceq}.$$

Thus let $p, q \in \mathbf{P}$ with $q \in \sigma_{\preceq}(X_p)$. Then we must prove that

$$q \ \in \ [\![\, ((\bigwedge_{a,p'.p\xrightarrow{a}p'} \langle a\rangle X_{p'}) \ \wedge \ (\bigwedge_{a\in A_p.p\downarrow a} [a] \bigvee_{p'.p\xrightarrow{a}p'} X_{p'}) \ \wedge \ A_p \,) \,]\!]\sigma_{\preceq}$$

or equivalently that

$$q \ \in \ \bigcap_{a,p'.p\xrightarrow{a}p'} \langle\cdot a\cdot\rangle \sigma_{\preceq}(X_{p'}) \ \cap \ \bigcap_{a\in A_p.p\downarrow a} [\cdot a\cdot] \bigcup_{p'.p\xrightarrow{a}p'} \sigma_{\preceq}(X_{p'}) \ \cap \ [\![A_p]\!]\sigma_{\preceq}.$$

This can be done set by set from left to right for the three parts of this intersection as follows:

- In the first case, we can assume that $p \xrightarrow{a} p_1$. As $p \preceq q$, the first condition of Definition 2.1 yields the existence of a q_1 satisfying $q \xrightarrow{a} q_1$ and $p_1 \preceq q_1$ and therefore $q_1 \in \sigma_{\preceq}(X_{p_1})$. This shows that q belongs to the first set of the three part intersection.
- In the second case, we can assume that $a \in A_p$ and $p\downarrow a$. Therefore according to Definition 2.1.2(b) $q\downarrow a$ and if $q \xrightarrow{a} q_1$ then $p \xrightarrow{a} p_1$ for some p_1 with $p_1 \preceq q_1$, i.e. $q_1 \in \sigma_{\preceq}(X_{p_1})$. Thus q belongs to the second set.
- In the third case, we can assume that either $q\uparrow\mu$ or both $q\downarrow\mu$ and $q \xrightarrow{\mu}$ hold, and we must show that $\mu \in A_p$. If $q\uparrow\mu$, Definition 2.1.2 immediately yields $p\uparrow\mu$, and we are done. Otherwise, $\mu \in A_p$ is obvious from Definition 2.1.2(b). $\square$

As mentioned already, allowing mutual recursion in our logic, rather than just nested recursion, yields concise characteristic formulae:

Lemma 4.3 *Given a process $p \in \mathbf{P}$, the size of its characteristic formula X_p grows linearly in the number of transitions of p.*

Here we identify the size of a characteristic formula with the size of its corresponding characteristic declaration, which is defined as the sum of the number of equations and the number of logical connectives occurring in it.

plays the role of ***true*** in the logic. On the other hand, there is no formula in M_V that plays the role of ***false***, as such a formula will not be needed in the definition of the characteristic formulae.

Notation: For each finite index set $I = \{i_1, \dots, i_k\}$, $k \geq 0$, and set of formulae $\{F_i \mid i \in I\}$, we will abbreviate $F_{i_1} \wedge \cdots \wedge F_{i_k}$ by $\bigwedge_{i \in I} F_i$. By convention, $\bigwedge_{i \in \emptyset} F_i = Act \cup \{\varepsilon\}$. Similarly, if $k \geq 1$, $\bigvee_{i \in I} F_i$ will be used to denote $F_{i_1} \vee \cdots \vee F_{i_k}$. Furthermore we will often write $p \models_\sigma F$ instead of $p \in [\![F]\!]\sigma$.

The set Env together with the partial order $\leq$, the usual pointwise order derived from the inclusion order in $\mathcal{P}(\mathbf{P})$, forms a complete lattice with greatest lower bound $\bigcap$ and least upper $\bigcup$ bound, where $\bigcap$ and $\bigcup$ are the pointwise extensions of intersection and union to environments. Moreover, as the logic does not include negation, the function

$$[\![F]\!] : Env \longrightarrow \mathcal{P}(\mathbf{P})$$

is monotonic for any $F \in M_V$. Thus the transformation

$$[\![D]\!] : Env \longrightarrow Env$$

given by

$$[\![D]\!] \;=\; \lambda\sigma.\, \lambda X.\, [\![D(X)]\!]\sigma$$

which is induced by a declaration D, is also monotonic. Applying the Knaster-Tarski's fixpoint theorem [Ta], this yields that $[\![D]\!]$ has a unique maximal fixpoint:

$$\sigma_D =_{df} \bigcup\{\, \sigma \mid \sigma \leq [\![D]\!]\sigma \,\}.$$

Thus the meaning of a formula F whose variables are bound by a declaration D is defined by:

$$[\![F]\!]\,\sigma_D.$$

Fundamental is the convergence condition in the semantic clause for $[a]$, which introduces an asymmetry between $[a]$ and $\langle a \rangle$ such that they are not duals. This is the basis for an intuitionistic understanding of modal logics as in [PS,Sti]. Indeed, the asymmetry in the intuitionistic interpretation of the logical language is necessary in order to cover the asymmetry of the preorder relation $\preceq$. A classical interpretation, for example, would already fail to deal with the existence of a smallest element, which exists in the form of the totally divergent process $nil \uparrow \varepsilon$ in our process system. An intuitionistic interpretation of the logic, however, delivers the following characterization theorem, which is a straightforward extension of the results of [Mi2,Sti]:

Theorem 3.1 *Defining M as the subset of all formulae in M_V that do not contain any variable, we have:*

$$\forall\, p, q \in \mathbf{P}.\;\; p \preceq q \;\;\textit{iff}\;\; (\,\forall\, F \in M_V\; \forall\, \sigma \in Env.\;\; p \models_\sigma F \;\textit{implies}\; q \models_\sigma F\,)$$

So, $p \preceq q$ means that q has more properties (expressible in M) than p. Classical negation would exclude the proper inclusion of the set of properties of one process in the set of properties of another process. Thus it is suitable for characterizing equivalences but not preorders.

4 The Characteristic Formula

Unlike [Mi2,Sti], where logics do not contain (mutual) recursion, there are ***characteristic formulae*** for $\preceq$ in M_V, i.e. for each process p there exists a formula, which is satisfied by a process q if and only if $p \preceq q$. Characteristic formulae have been introduced by Graf and Sifakis in [GS1], where it was shown how to derive a logical specification for finite processes up to observational congruence (i.e. characteristic formulae for $\approx^c$) in the modal mu-calculus [Ko]. We are now going to show, how to derive characteristic formulae for $\preceq$ in our setting. Originally, such a definition has been given in [Ste], where characteristic formulae are defined in terms of an intuitionistically interpreted modal mu-calculus. Here we modify this definition to use ***mutual*** recursion instead of ***nested*** recursion, along the lines of [GIZ], where such characteristic formulae for bisimulation–like equivalences are constructed. This modification reduces the complexity of the size of the characteristic formula obtained, which is now going to be linear in the number of transitions of the argument process (Lemma 4.3).

Before defining the characteristic formulae in detail, let us look at an example taken from CCS, in order to motivate our "design decisions":

Example:

Let $Act = \{a, b, c\}$ and the process p be defined by

$$p \overset{def}{=} a.q$$
$$q \overset{def}{=} b.p + c.q.$$

Then the characteristic formula should express the following:

1. p can make an a-move to become a process with the characteristic property of q.
2. p converges on a and every a-move will lead to a process satisfying the characteristic property of q.
3. a is the only possible action p can perform or diverge on.

These three properties of the characteristic formula of p can be expressed by means of the three conjuncts in the formula below:

$$\langle a \rangle X_q \;\wedge\; [a] X_q \;\wedge\; \{a\}$$

where X_q stands for the characteristic property of q. Analogously, we obtain for q:

$$\langle b \rangle X_p \;\wedge\; [b] X_p \;\wedge\; \langle c \rangle X_q \;\wedge\; [c] X_q \;\wedge\; \{b, c\}.$$

Together this leads to the following mutually recursive declaration, in which X_p and X_q characterize p and q up to $\preceq$ respectively:

$$X_p \;\Longrightarrow\; \langle a \rangle X_q \;\wedge\; [a] X_q \;\wedge\; \{a\}$$
$$X_q \;\Longrightarrow\; \langle b \rangle X_p \;\wedge\; [b] X_p \;\wedge\; \langle c \rangle X_q \;\wedge\; [c] X_q \;\wedge\; \{b, c\}.$$

weak transition relation $\Longrightarrow$ defined by[3]:

$$\stackrel{\epsilon}{\Longrightarrow} \; = \; \stackrel{\tau}{\longrightarrow}^{*} \quad \text{and} \quad \stackrel{a}{\Longrightarrow} \; = \; \stackrel{\epsilon}{\Longrightarrow} \cdot \stackrel{a}{\longrightarrow} \cdot \stackrel{\epsilon}{\Longrightarrow}$$

and the divergence predicate, which results from interpreting divergence as the potential for an infinite τ-sequence, as it is usually done in CCS, the processes p_0 and q_0 can be expressed by

$$((\{p_0, p_1\}, \{p_0 \stackrel{\epsilon}{\Longrightarrow} p_0, p_0 \stackrel{a}{\Longrightarrow} p_1, p_1 \stackrel{\epsilon}{\Longrightarrow} p_1\}, \{\epsilon, a\}, \{p_0 \Uparrow \varepsilon\}), p_0) \quad \text{and}$$
$$((\{q_0, q_1\}, \{q_0 \stackrel{\epsilon}{\Longrightarrow} q_0, q_0 \stackrel{a}{\Longrightarrow} q_1, q_1 \stackrel{\epsilon}{\Longrightarrow} q_1\}, \{\epsilon, a\}, \emptyset), q_0)$$

respectively[4].

These two processes are ***observationally equivalent*** in the sense of [Mi1,Mi3]

$$fix(X, \tau.X + a.nil) \;\approx\; a.nil$$

even though the first process can engage in an infinite internal chatter and therefore refuse to respond to an a-action at all. This motivates the definition of a *preorder*, $\preceq$, between processes:

Definition 2.1 *$\preceq$ is the union of all relations R, satisfying that pRq implies*

1. *$\forall a \forall p'$. if $p \stackrel{a}{\longrightarrow} p'$ then $\exists q'$. $q \stackrel{a}{\longrightarrow} q'$ and $p'Rq'$*
2. *a) If $p \downarrow \varepsilon$ then $q \downarrow \varepsilon$*

   b) $\forall a$. if $p \downarrow a$ then ($q \downarrow a$ and $\forall q'$. if $q \stackrel{a}{\longrightarrow} q'$ then $\exists p'. p \stackrel{a}{\longrightarrow} p'$ and $p'Rq'$).

The relation $\preceq$ is itself a preorder that satisfies the conditions in Definition 2.1. Thus it is the greatest such relation. Now, $\preceq$ (wrt the weak transition relation $\Longrightarrow$) distinguishes between the two example processes. Indeed, we still have:

$$fix(X, \tau.X + a.nil) \;\preceq\; a.nil,$$

but the converse fails:

$$a.nil \;\not\preceq\; fix(X, \tau.X + a.nil)$$

because $a.nil \downarrow \varepsilon$ whilst $fix(X, \tau.X + a.nil) \uparrow \varepsilon$.

3 The Logic

The modal logic we use is essentially Hennessy-Milner Logic [HM] with mutual recursion [La]. Only the operator $[a]$ is modified to reflect divergence properties, and sets of actions serve as atomic formulae that impose restrictions on the actions a process may perform or diverge upon. This modification leads to an intuitionistic understanding of the logic (cf. [PS,Sti]) and allows to characterize processes up to $\preceq$.

[3] Note the difference between ϵ and ε.

[4] For details see Section 5.

Let V be a countable set of variables. Then the logic M_V is given by:

$$F ::= X \mid A \mid F \wedge F \mid F \vee F \mid [a]F \mid \langle a \rangle F$$

where $X \in V$, $A \subseteq Act \cup \{\varepsilon\}$ and $a \in Act$. Variables are bound by declarations

$$D : V \longrightarrow M_V$$

which map variables to formulae in M_V and are written:

$$X_i \Longrightarrow D(X_i),\; i = 1, 2, \cdots$$

where we assume that the set of variables is indexed appropriately.

Declarations provide the logic with the power of specifying infinite behavioural patterns by means of mutual recursion. Usually, only a finite subset of V is non trivially defined, meaning $X \neq D(X)$. The other variables, where D is the identity, do not impose any constraints on processes. In fact, we will only need such "finite" declarations in future, and we will specify them finitely by their non trivial part: $X_i \Longrightarrow D(X_i)$, $i = 1, 2, \cdots, n$.

Formulae of M_V may be open expressions and as usual, we use an environment, which binds the free variables to sets of processes, in order to give semantics to such formulae. The set of all environments $\{\sigma \mid \sigma : V \longrightarrow \mathcal{P}(\mathbf{P})\}$ will be denoted by Env and ranged over by σ.

The semantics of a formula F in M_V can now be given as the set of all processes satisfying it. This leads to a semantic functional

$$[\![\_]\!]\_ : M_V \longrightarrow Env \longrightarrow \mathcal{P}(\mathbf{P})$$

which is defined inductively as follows:

$$[\![A]\!]\sigma \;=\; \{p \in \mathbf{P} \mid (p \uparrow \mu \text{ or } p \stackrel{\mu}{\longrightarrow}) \text{ implies } \mu \in A\}$$

$$[\![X]\!]\sigma \;=\; \sigma(X)$$

$$[\![F_1 \wedge F_2]\!]\sigma \;=\; [\![F_1]\!]\sigma \cap [\![F_2]\!]\sigma$$

$$[\![F_1 \vee F_2]\!]\sigma \;=\; [\![F_1]\!]\sigma \cup [\![F_2]\!]\sigma$$

$$[\![\langle a \rangle F]\!]\sigma \;=\; \langle \cdot a \cdot \rangle [\![F]\!]\sigma$$
where
$$\langle \cdot a \cdot \rangle U \;=\; \{p \in \mathbf{P} \mid \exists p' \in U. p \stackrel{a}{\longrightarrow} p'\}$$

$$[\![[a]F]\!]\sigma \;=\; [\cdot a \cdot][\![F]\!]\sigma$$
where
$$[\cdot a \cdot]U \;=\; \{p \in \mathbf{P} \mid p \downarrow a \text{ and } (\forall p'. p \stackrel{a}{\longrightarrow} p' \text{ implies } p' \in U)\}.$$

Note that $[\![Act \cup \{\varepsilon\}]\!]\sigma = \mathbf{P}$, for any environment σ. Hence the atomic formula $Act \cup \{\varepsilon\}$

Characteristic Formulae for Processes with Divergence

Bernhard Steffen
Lehrstuhl für Informatik II
RWTH Aachen
Ahornstraße 55
W-5100 Aachen

Anna Ingólfsdóttir
Computer Science
School of Cognitive and Computing Sciences
University of Sussex
GB Brighton BN1 9QH

February 10, 1992

Abstract

Characteristic formulae have been introduced by Graf and Sifakis to relate equational reasoning about processes to reasoning in a modal logic, and therefore to allow proofs about processes to be carried out in a logical framework. This work, which concerned *finite* processes and bisimulation-like *equivalences*, has later on been extended to *finite state* processes and further equivalences. Based upon an intuitionistic understanding of Hennessy-Milner logic (HML) with mutual recursion, we extend these results to cover bisimulation-like *preorders*, which are sensitive to liveness properties. This demonstrates the expressive power of intuitionistically interpreted HML with mutual recursion, and it builds the theoretical basis for a uniform and efficient method to *automatically verify* bisimulation–like relations between processes by means of *model checking*[1].

[1]This paper is an extended version of [Ste], modified to use HML with mutual recursion [La] instead of the modal mu-calculus [Ko].

1 Motivation

Transition systems are often used as models for concurrent programs. By themselves they are too concrete to provide an appropriate notion of computational equivalence or simulation. Addition of an explicit equivalence or preorder relation on transition systems overcomes this problem, by identifying processes (nodes) of 'similar' computational power.

In this paper we investigate transition systems that explicitly provide divergence information [Sti] together with a notion of simulation given by a divergence sensitive preorder. (Think of divergence as the potential for infinite internal chatter.) As the main result we show how to obtain characteristic formulae for this notion of simulation within an intuitionistic interpretation [PS] of Hennessy-Milner Logic with mutual recursion [La]. Application to CCS [Mi1,Mi3] delivers characteristic formulae for bisimulation–like equivalences, congruences and preorders. This extends the existing work (cf. [CGB,GIZ,GS1,GS2,GS3,Ho]) in that it handles not only bisimulation–like equivalences but also *preorders* [Wa], which are sensitive to liveness properties.— The results of this paper are the basis for an extension of the Concurrency Workbench to uniformly and efficiently check various bisimulation–like relations by means of model checking[2].

2 Extended Transition Systems

An *extended finite state transition system* T is a quadruple $(\mathbf{P}, \longrightarrow, Act, \uparrow)$, where

1. $\mathbf{P}$ is a finite set (of *processes* or *states*);
2. Act is a finite set (of *observable actions*);
3. $\longrightarrow$ is a mapping, which associates with each action $a \in Act$, a relation $\stackrel{a}{\longrightarrow} \subseteq \mathbf{P} \times \mathbf{P}$;
4. $\uparrow$ is a mapping, which associates with each $\mu \in Act \cup \{\varepsilon\}$, where $\varepsilon \notin Act$ represents unobservable actions, a unary predicate $\uparrow\mu$ on $\mathbf{P}$.— Instead of $\uparrow\mu(p)$, we write $p\uparrow\mu$, and we require that $p\uparrow\varepsilon$ implies $p\uparrow a$ for every observable action $a \in Act$.

Typically, $\mathbf{P}$ is a set of program states, and for each $a \in Act$ the relationship $p\stackrel{a}{\longrightarrow}q$ indicates that p can evolve to q under the observation of a. The fourth component $\uparrow$ is introduced to express *divergence* potential: $p\uparrow\mu$ is intended to mean that p can be triggered by means of an offer of a μ-action to evolve autonomously for ever without responding to the environment. The predicate $\uparrow\varepsilon$ does not depend on any observable action. Therefore $\uparrow\varepsilon$ implies $\uparrow a$, for every $a \in Act$. That is the reason for calling $\uparrow\varepsilon$ the *total divergence* predicate.— The negation of $p\uparrow\mu$, μ-convergence, is written $p\downarrow\mu$, and we will represent states or processes (of a transition system) as *rooted transition systems*, i.e. as pairs (T, p_0), where p_0 is the *start state*.

Typical examples in CCS would be $p_0 = fix(X, \tau.X + a.nil)$ and $q_0 = a.nil$. Using the

[2]The Concurrency Workbench is an automated tool for the analysis of concurrent systems [CPS1,CPS2,CPS3].