Characteristic Formulae for Fixed-Point Semantics: A General Framework

Luca Aceto, Anna Ingolfsdottir and Joshua Sack
ICE-TCS, School of Computer Science, Reykjavik University

EXPRESS'09, Bologna, 5 September 2009

Thanks to the Icelandic Research Fund and Reykjavik University's Development Fund for partial financial support.

Verifying Correctness of Reactive Systems

Equivalence/Preorder Checking

$$Impl \equiv Spec$$

- $\equiv$ is a 'behavioural' equivalence/preorder,
- $Spec$ is expressed in the same language as $Impl$—typically in terms of (a language for describing) automata
- $Spec$ provides the (full) specification of the intended behaviour

Model Checking

$$Impl \models Property$$

- $\models$ is the satisfaction relation
- $Property$ is a (partial) specification of the intended behaviour, often expressed in a modal or temporal logic

Verifying Correctness of Reactive Systems

Equivalence/Preorder Checking

$$Impl \equiv Spec$$

- $\equiv$ is a 'behavioural' equivalence/preorder,
- $Spec$ is expressed in the same language as $Impl$—typically in terms of (a language for describing) automata
- $Spec$ provides the (full) specification of the intended behaviour

Model Checking

$$Impl \models Property$$

- $\models$ is the satisfaction relation
- $Property$ is a (partial) specification of the intended behaviour, often expressed in a modal or temporal logic

Characteristic Formulae: A Bridge Between the Worlds

Characteristic Formulae

A characteristic formula for $Spec$ modulo $\equiv$ is a formula $F(Spec)$ such that, for each $Impl$,

$$Impl \equiv Spec \text{ iff } Impl \models F(Spec) \ .$$

The Role of Characteristic Formulae

- Using characteristic-formula constructions one can effectively reduce implementation verification to model checking.
- Characteristic formulae give an indication of the expressiveness of a logical property language.
- Characteristic formulae are 'a perfect form of reverse engineering.'

Characteristic Formulae: A Bridge Between the Worlds

Characteristic Formulae

A characteristic formula for $Spec$ modulo $\equiv$ is a formula $F(Spec)$ such that, for each $Impl$,

$$Impl \equiv Spec \text{ iff } Impl \models F(Spec) \ .$$

The Role of Characteristic Formulae

- Using characteristic-formula constructions one can effectively reduce implementation verification to model checking.
- Characteristic formulae give an indication of the expressiveness of a logical property language.
- Characteristic formulae are 'a perfect form of reverse engineering.'

A Bit of History

- Recursion-free CCS terms $\longrightarrow$ characteristic formulae modulo observational congruence. (Graf and Sifakis, 1986)
- Finite labelled transition systems (LTSs) $\longrightarrow$ characteristic formulae in Hennessy-Milner logic with recursion modulo strong bisimilarity. (Ingolfsdottir, Godskesen and Zeeberg, 1987)
- Finite Kripke structures $\longrightarrow$ characteristic formulae in CTL modulo strong bisimilarity. (Browne, Clarke and Grümberg, 1988)
- Finite LTSs with divergence $\longrightarrow$ characteristic formulae in intuitionistic Hennessy-Milner logic with recursion modulo some partial bisimilarity. (Ingolfsdottir and Steffen, 1994)
- Lots more! Characteristic formulae are part of our genetic heritage!

Our Question and The Main Message

Our Motivating Question and Aim

Can one give a unified treatment of (some of) the aforementioned results in terms of general principles?

Aim: Recover extant constructions in a principled fashion, and possibly obtain novel characteristic-formula constructions 'for free'.

The Message in a Bottle

Yes! We give a general view of characteristic formulae that are expressed in terms of logics with a facility for the recursive definition of formulae. The proposed framework applies to behavioural relations that are defined as fixed points of suitable monotonic functions.

Our Question and The Main Message

Our Motivating Question and Aim

Can one give a unified treatment of (some of) the aforementioned results in terms of general principles?

Aim: Recover extant constructions in a principled fashion, and possibly obtain novel characteristic-formula constructions 'for free'.

The Message in a Bottle

Yes! We give a general view of characteristic formulae that are expressed in terms of logics with a facility for the recursive definition of formulae. The proposed framework applies to behavioural relations that are defined as fixed points of suitable monotonic functions.

Outline for the Rest of the Talk

1. A motivating example
2. The main theorem
3. Applications of the main theorem
4. Concluding remarks

And now for some technical content...

Outline for the Rest of the Talk

1. A motivating example
2. The main theorem
3. Applications of the main theorem
4. Concluding remarks

And now for some technical content. . .

Strong Bisimilarity as a Largest Fixed Point

A finite LTS is a triple $P = (\mathbf{P}, \mathbf{A}, \longrightarrow)$, where

- $\mathbf{P}$ is a finite set,
- $\mathbf{A}$ is a finite set of labels and
- $\longrightarrow \subseteq \mathbf{P} \times \mathbf{A} \times \mathbf{P}$ is a transition relation.

Strong bisimilarity is the largest fixed point of the monotonic function $(p, q) \in \mathcal{F}_{bisim}(S)$, where $S \subseteq \mathbf{P} \times \mathbf{P}$, iff for every $a \in \mathbf{A}$,

1. if $p \xrightarrow{a} p'$, then there exists some $q' \in \mathbf{P}$ such that $q \xrightarrow{a} q'$ and $(p', q') \in S$, and
2. if $q \xrightarrow{a} q'$, then there exists some $p' \in \mathbf{P}$ such that $p \xrightarrow{a} p'$ and $(p', q') \in S$.

Logic: HML with Recursion

Syntax of HML with Recursion

$$F ::= tt \mid ff \mid X \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle a \rangle F_1 \mid [a] F_1,$$

where $X \in Var$ (a set of variables) and $a \in \mathbf{A}$. A declaration D associates a formula with each variable.

Semantics

Given an LTS $P = (\mathbf{P}, \mathbf{A}, \longrightarrow)$ and an environment $\sigma : Var \to \mathcal{P}(\mathbf{P})$, define $(\sigma, p) \models F$ thus (selected rules):

$$\begin{array}{lll} (\sigma, p) \models X & \text{iff} & p \in \sigma(X) \\ (\sigma, p) \models \langle a \rangle F_1 & \text{iff} & (\sigma, p') \models F_1 \text{ for some } p' \text{ for which } p \xrightarrow{a} p' \\ (\sigma, p) \models [a] F_1 & \text{iff} & (\sigma, p') \models F_1 \text{ for all } p' \text{ for which } p \xrightarrow{a} p' \end{array}$$

Key observation: $\models$ is monotonic in σ.

Characteristic Formulae for Strong Bisimilarity

Theorem (Ingolfsdottir et al.)

$p \sim_{bisim} q$ iff $(\sigma_{\max}, p) \models X_q$, where $\sigma_{\max}$ is the largest interpretation of the declaration $D_{bisim}(X_q)$ defined thus:

$$D_{bisim}(X_q) = \bigwedge_{a \in \mathbf{A}} [a]\Big(\bigvee_{q'.q \xrightarrow{a} q'} X_{q'} \Big) \wedge \bigwedge_{a, q'.q \xrightarrow{a} q'} \langle a \rangle X_{q'}.$$

In fact, for each $S \subseteq \mathbf{P} \times \mathbf{P}$ and $p, q \in \mathbf{P}$,

$$(\sigma_S, p) \models D_{bisim}(X_q) \Leftrightarrow (p, q) \in \mathcal{F}_{bisim}(S),$$

where $\sigma_S(X_q) = \{p \in \mathbf{P} \mid (p, q) \in S\}$, for each $q \in \mathbf{P}$.

Characteristic Formulae for Strong Bisimilarity

Theorem (Ingolfsdottir et al.)

$p \sim_{bisim} q$ iff $(\sigma_{\max}, p) \models X_q$, where $\sigma_{\max}$ is the largest interpretation of the declaration $D_{bisim}(X_q)$ defined thus:

$$D_{bisim}(X_q) = \bigwedge_{a \in \mathbf{A}} [a]\Big(\bigvee_{q'.q \xrightarrow{a} q'} X_{q'} \Big) \wedge \bigwedge_{a, q'.q \xrightarrow{a} q'} \langle a \rangle X_{q'}.$$

In fact, for each $S \subseteq \mathbf{P} \times \mathbf{P}$ and $p, q \in \mathbf{P}$,

$$(\sigma_S, p) \models D_{bisim}(X_q) \Leftrightarrow (p, q) \in \mathcal{F}_{bisim}(S),$$

where $\sigma_S(X_q) = \{p \in \mathbf{P} \mid (p, q) \in S\}$, for each $q \in \mathbf{P}$.

Reverse Engineering the Proof: Ingredients

1. Strong bisimilarity is the largest fixed point of a monotonic function $\mathcal{F}_{bisim}$ over binary relations.
2. We had a monotonic logic with variables and declarations.
3. Within the logic we could write a declaration D_{bisim} that, for each $S \subseteq \mathbf{P} \times \mathbf{P}$, *expresses* $\mathcal{F}$ *up to* S—that is, for all $p, q \in \mathbf{P}$,

$$(\sigma_S, p) \models D_{bisim}(X_q) \Leftrightarrow \ (p, q) \in \mathcal{F}_{bisim}(S),$$

where $\sigma_S(X_q) = \{p \in \mathbf{P} \mid (p, q) \in S\}$, for each $q \in \mathbf{P}$.

Question: Does this depend in any way on specific properties of bisimilarity and/or HML with recursion?

The Main Theorem: It Doesn't!

$\mathcal{L}$ = monotonic logic over a set of variables Var.

Theorem

Assume that $\mathcal{F} : \mathcal{P}(\mathbf{P} \times \mathbf{P}) \to \mathcal{P}(\mathbf{P} \times \mathbf{P})$ is a monotonic function and $D : Var \to \mathcal{L}$ is a monotonic declaration such that D expresses $\mathcal{F}$ up to S for all $S \subseteq \mathbf{P} \times \mathbf{P}$. Then, for all $p, q \in \mathbf{P}$,

1. $(\sigma_{\mathsf{max}}, p) \models X_q \Leftrightarrow (p, q) \in Fix\, \mathcal{F}$ ($Fix\, \mathcal{F}$ is the largest fixed point of $\mathcal{F}$) and
2. $(\sigma_{\mathsf{min}}, p) \models X_q \Leftrightarrow (p, q) \in fix\, \mathcal{F}$ ($fix\, \mathcal{F}$ is the least fixed point of $\mathcal{F}$).

Conclusion: The ingredients we isolated justify the existence of many characteristic-formula constructions in the literature! Examples?

Ready Simulation Preorder (Bloom et al., Larsen and Skou)

Let $\mathcal{F}_{RS}(S)$ be defined such that $(p,q) \in \mathcal{F}_{RS}(S)$ iff for every $a \in \mathbf{A}$ and $q' \in \mathbf{P}$,

1. if $q \xrightarrow{a} q'$, then there exists some $p' \in \mathbf{P}$ such that $p \xrightarrow{a} p'$ and $(p',q') \in S$, and
2. if $q \not\xrightarrow{a}$ then $p \not\xrightarrow{a}$.

Fact: For each $S \subseteq \mathbf{P} \times \mathbf{P}$, we have

$$(p,q) \in \mathcal{F}_{RS}(S) \Leftrightarrow (\sigma_S, p) \models \bigwedge_{a,q'.q \xrightarrow{a} q'} \langle a \rangle X_{q'} \wedge \bigwedge_{a.q \not\xrightarrow{a}} [a]ff.$$

Corollary of the main theorem: The characteristic formula for the largest fixed point of $\mathcal{F}_{RS}$ is the largest interpretation of the declaration

$$D_{RS}(X_q) = \bigwedge_{a,q'.q \xrightarrow{a} q'} \langle a \rangle X_q \wedge \bigwedge_{a.q \not\xrightarrow{a}} [a]ff.$$

Epistemic Back-and-forth Bisimilarity (Dechesne et al.)

We augment our notion of labelled transition systems with a set $\mathcal{I}$ of identities (or agents) and a family of equivalence relations $\{\stackrel{i}{\cdot\cdot\cdot} \subseteq \mathbf{P} \times \mathbf{P} \mid i \in \mathcal{I}\}$. Let $\mathcal{F}_{bfbid}(S)$ be defined such that $(p, q) \in \mathcal{F}_{bfbid}(S)$ iff for every $a \in \mathbf{A}$ and $i \in \mathcal{I}$,

1. $\forall p' \in \mathbf{P}.\, p \xrightarrow{a} p' \Rightarrow \exists q' \in \mathbf{P}.q \xrightarrow{a} q'$ and $(p', q') \in S$,
2. $\forall q' \in \mathbf{P}.\, q \xrightarrow{a} q' \Rightarrow \exists p' \in \mathbf{P}.p \xrightarrow{a} p'$ and $(p', q') \in S$,
3. $\forall p' \in \mathbf{P}.\, p' \xrightarrow{a} p \Rightarrow \exists q' \in \mathbf{P}.q' \xrightarrow{a} q$ and $(p', q') \in S$,
4. $\forall q' \in \mathbf{P}.\, q' \xrightarrow{a} q \Rightarrow \exists p' \in \mathbf{P}.p' \xrightarrow{a} p$ and $(p', q') \in S$,
5. $\forall p' \in \mathbf{P}.\, p \stackrel{i}{\cdot\cdot\cdot} p' \Rightarrow \exists q' \in \mathbf{P}.q \stackrel{i}{\cdot\cdot\cdot} q'$ and $(p', q') \in S$ and
6. $\forall q' \in \mathbf{P}.\, q \stackrel{i}{\cdot\cdot\cdot} q' \Rightarrow \exists p' \in \mathbf{P}.p \stackrel{i}{\cdot\cdot\cdot} p'$ and $(p', q') \in S$.

We denote the largest fixed point of $\mathcal{F}_{bfbid}$ by $\sim_{bfbid}$.

Logic

Logic: We use HML with recursion and add to it the operators:

- $\langle \overline{a} \rangle$ and $[\overline{a}]$, for each $a \in \mathbf{A}$, and
- $\langle i \rangle$ and $[i]$, for each $i \in \mathcal{I}$—cf. epistemic logic.

Semantics:

$$
\begin{array}{lll}
(\sigma, p) \models \langle \overline{a} \rangle F_1 & \text{iff} & (\sigma, p') \models F_1 \text{ for some } p' \text{ for which } p' \xrightarrow{a} p \\
(\sigma, p) \models [\overline{a}] F_1 & \text{iff} & (\sigma, p') \models F_1 \text{ for all } p' \text{ for which } p' \xrightarrow{a} p \\
(\sigma, p) \models \langle i \rangle F_1 & \text{iff} & (\sigma, p') \models F_1 \text{ for some } p' \text{ for which } p \stackrel{i}{\cdots} p' \text{ and} \\
(\sigma, p) \models [i] F_1 & \text{iff} & (\sigma, p') \models F_1 \text{ for all } p' \text{ for which } p \stackrel{i}{\cdots} p'.
\end{array}
$$

The logic is monotonic.

Characteristic-Formula Construction for $\sim_{bfbid}$

For each binary relation S over states, $(p, q) \in \mathcal{F}_{bfbid}(S)$ iff

$$(\sigma_S, p) \models \bigwedge_{a \in \mathbf{A}} [a]\Big(\bigvee_{q'.q \xrightarrow{a} q'} X_{q'}\Big) \wedge \bigwedge_{a, q'.q \xrightarrow{a} q'} \langle a \rangle X_{q'} \wedge$$
$$\bigwedge_{a \in \mathbf{A}} [\overline{a}]\Big(\bigvee_{q'.q' \xrightarrow{a} q} X_{q'}\Big) \wedge \bigwedge_{a, q'.q' \xrightarrow{a} q} \langle \overline{a} \rangle X_{q'} \wedge$$
$$\bigwedge_{i \in \mathcal{I}} [i]\Big(\bigvee_{q'.q \overset{i}{\cdots} q'} X_{q'}\Big) \wedge \bigwedge_{i, q'.q \overset{i}{\cdots} q'} \langle i \rangle X_{q'}.$$

Corollary of the main theorem: The characteristic formula for $\sim_{bfbid}$ is the largest interpretation of the declaration assigning to each X_q the above formula. (This solves an open problem of Dechesne et al.)

Other Relations We Can Handle

1. Simulation preorder and equivalence.
2. Prebisimilarity and partial bisimilarity (using intuitionistic HML with recursion).
3. Extended simulation preorder and equivalence (Thomsen 1987).
4. Weak bisimilarity and observation congruence.
5. Probabilistic bisimilarity over minimized probabilistic LTSs (Larsen and Skou 1992).

Summing up and Further Work

Executive Summary

1. We provided a general view of characteristic formulae for behavioural relations that can be defined by largest/least fixed points of monotonic functions.
2. We have explored a number of applications of this theorem, some in recovering characteristic formulae already discovered, and some being novel constructions.

Future and ongoing work: Search for further generalizations (done in part) and more applications (e.g., resource bisimulation equivalence (Corradini et al.), g-bisimulation equivalence (de Rijke) and various timed bisimilarities.)

Thanks and Shameless Self-Promotion

Thank You!
Any Questions?

Buy your copy of Reactive Systems: Modelling, Specification and Verification (Cambridge University Press) by Luca Aceto, Anna Ingolfsdottir, Kim G. Larsen and Jiri Srba!

Visit us at ICE-TCS, the Icelandic Centre of Excellence in Theoretical Computer Science! (Warning: Honey, we have no money, alas.)