Yale University
Department of Computer Science

A Syntactic Method for Proving Observational Equivalences

Martin Odersky

Yale University, Department of Computer Science,
Box 2158 Yale Station, New Haven, CT 06520

Research Report YALEU/DCS/RR-964
May, 1993

A Syntactic Method for Proving Observational Equivalences

Martin Odersky

Yale University, Department of Computer Science,
Box 2158 Yale Station, New Haven, CT 06520

Abstract

We present a syntactic method for proving observational equivalences in reduction systems. The method is based on establishing a weak diamond property for critical pairs. It has been used successfully in proofs on the observational equivalence theories of λ_{var} and $\lambda\nu$.

1 Introduction

Observational equivalence is the most comprehensive notion of equality of between program fragments. Usually, it is what programmers have in mind when they say that two program fragments are interchangeable. The observational equivalences of a language define thus the transformations that are admissible in it. Hence, knowing what those equivalences are is important in areas such as program verification, transformational programming, partial evaluation and code optimization.

Intuitively, two terms are observationally equivalent if they cannot be distinguished by some experiment. Experiments place a program fragment in a context and observe the output of the resulting program. If each experiment yields the same output for both fragments, or if it yields no output for both fragments (due to non-termination, for instance), then we say the two fragments are observationally equivalent. What constitutes "output" in this context depends on the underlying language.

This definition of observational equivalence does not lead naturally to a simple technique for proving that a given relation is an observational equivalence. In fact, such proofs tend to be rather hard. Therefore, one often tries to prove observational equivalences indirectly.

One popular approach works with a model of the programming language instead of the terms of the language themselves. A model is *adequate* if any equality that holds in the model is also an observational equivalence. Writing $=_D$ for equality in the model and $(\cong)$ for observational equivalence, we have $=_D \subseteq \cong$. Adequate models present a sound way to prove observational equivalences. In *fully abstract* models denotational identity and observational equivalence are the same, i.e. $=_D = \cong$. Reasoning in fully abstract models is therefore sound and complete for the observational equivalence theory of a language. Unfortunately, it is often hard to construct a fully abstract model that makes reasoning about $=_D$ simpler than reasoning about $(\cong)$. For instance, in the case of PCF [10], the the only known fully abstract model [6] is defined in terms of congruence classes of $(\cong)$, and hence cannot contribute anything new to our knowledge about $(\cong)$. Riecke and O'Hearn improve over this by showing that in the presence of a context lemma only congruence classes of closed terms need to be considered [**?**].

Sometimes, properties of the language in consideration can help in observational equivalence proofs. For instance, Milner's context lemma [6] for the λ-calculus and related functional languages establishes that the only contexts one needs to consider are function applications. Or, it might be sufficient to consider only closed instantiations of the sides of an observational equivalence (as in, [5] Theorem **ciu**).

This paper presents a purely syntactic method for proving observational equivalences in arbitrary extensions of the λ-calculus. The work was motivated by the need to prove observational equivalences in the syntactic theories λ_{var} [9] and $\lambda\nu$ [8], for which no abstract models are known yet. The method is inspired by the "critical pairs" technique of the Knuth-Bendix completion algorithm [4]. The critical pairs technique of Knuth and Bendix consists of a proof that, for each critical pair M_1, M_2 with root M the following diagram can be completed.

$$\begin{tikzcd} M \arrow[r] \arrow[d] & M_2 \arrow[d, dotted, ->>] \\ M_1 \arrow[r, dotted, ->>] & N \end{tikzcd} \tag{1}$$

As usual, given nodes are connected by straight lines in this diagram, whereas nodes attached by dotted lines have to be shown to exist.

Like Knuth-Bendix completion, our technique relies on establishing some kind of diamond property for critical pairs. It requires that the following diagram can be completed for all $(\rightarrow, \sim)$-critical pairs (N, M').

$$\begin{tikzcd} M \arrow[r] \arrow[d, dash, "\sim_1"'] & M' \arrow[d, dotted, dash, "\sim_1"] \\ N \arrow[r, dotted, dash, "\cong"'] & N' \end{tikzcd} \tag{2}$$

Unlike a conventional critical pair, a $(\rightarrow, \sim)$-critical pair involves two relations, reduction $(\rightarrow)$ and parallel similarity $(\sim_1)$. Similarity $(\sim)$ is the candidate relation that needs to be shown an observational equivalence. $(\sim_1)$ is a parallel version of $(\sim)$, resulting from applying $(\sim)$ relations to independent subterms of a term. Informally, the diagram says that, whenever $M \sim_1 N$ and M reduces in one step to M' we can find a term N' such that $M' \sim_1 N'$ and N' is observationally equivalent to N. If this holds for all critical pairs, we say that $(\sim)$ is *locally stable*. The second condition we need is that $(\sim)$ *preserves answers*, i.e. one has $M \sim A \;\Rightarrow\; M = A$ for all terms M and answers A. Our first theorem (Theorem 2.26) states that if these conditions are both met then $(\sim)$ is an observational equivalence. Our second theorem (Theorem 2.27) extends this approach to deterministic reduction.

Note the similarity between this technique and bisimulation. Bisimulations are relations that were originally studied in the context of process algebras [7]. The concept has also been adapted in a functional setting [1]. A relation ($\sim$) is a bisimulation if the following diagram can be completed for all terms N, M, M' and observable actions a:

$$\begin{tikzcd} M \arrow[r, "a"] \arrow[d, no head, "\sim"'] & M' \arrow[d, dotted, no head, "\sim"] \\ N \arrow[r, dotted, "a"] & N' \end{tikzcd} \tag{3}$$

The main difference between diagram (2) and diagram (3) concerns the top and bottom edges. In the case of bisimulation, these are both reduction steps with the same observable action a. In our critical pairs method, the given reduction on the top is a single ($\rightarrow$)-step, and we require only $N \cong N'$ on the bottom. This offers a convenient way to use previous knowledge about the observational equivalence relation ($\cong$).

The rest of this paper is organized as follows. Section 2 defines observational equivalence for reduction systems and presents criteria for a ralation to be an observational equivalence. Section 3 applies these results to the $\lambda\nu$ calculus. Section 4 concludes.

2 Proving Observational Equivalences

We study observational equivalence in the context of reduction systems that extend the λ calculus. In the following, let $\mathcal{T}$ be an equational theory that extends λ with term language $Terms(\mathcal{T})$, a set of *programs* $Progs(\mathcal{T}) \subseteq Terms(\mathcal{T})$, and a set of *answers* $Ans(\mathcal{T}) \subseteq Progs(\mathcal{T})$.

2.1 Observational Equivalence

Definition 2.1 Two terms $M, N \in Terms(\mathcal{T})$ are *observationally equivalent* in $\mathcal{T}$, written

$$\mathcal{T} \models M \cong N$$

iff for all contexts C in $Terms(\mathcal{T})$ such that $C[M]$ and $C[N]$ are programs, and for all answers $A \in Ans(\mathcal{T})$,

$$\mathcal{T} \vdash C[M] = A \Leftrightarrow \mathcal{T} \vdash C[N] = A.$$

Lemma 2.2 For all $M, N \in Terms(\mathcal{T})$,

$$\mathcal{T} \models M \cong N \Leftrightarrow \forall C. \mathcal{T} \models C[M] \cong C[N].$$

Proof: "$\Rightarrow$": Assume $M \cong N$, let A be an answer, and let C be a context. Let C' be a context such that $C'[C[M]]$ and $C'[C[N]]$ are closed. Then $M \cong N$ implies

$$C'[C[M]] = A \Leftrightarrow C'[C[N]] = A.$$

Since C' was arbitrary, $C[M] \cong C[N]$.
"$\Leftarrow$": Pick $C = [\,]$. $\blacksquare$

The definition of $(\cong)$ gives us only a very cumbersome way to reason about observational equivalence, since it relies on a universal quantification over all contexts. In the following, we work out other criteria for observational equivalences that are easier to use in proofs.

2.2 Basic Definitions

Our main result requires a formal definition of rules for reduction and observational equivalence. We introduce a new alphabet of *meta-variables* $a, b, c, \ldots$. *Meta-terms* are constructed from meta-variables, the productions that form terms, and a substitution operator.

Definition 2.3 A meta-term X is one of the following:

(i) If $M ::= \mathcal{P}(M_1, \ldots, M_n)$ is a term-forming production, and $X_1, \ldots, X_n$ are meta-terms, then $\mathcal{P}(X_1, \ldots, X_n)$ is a meta-term.

(ii) A meta-variable is a meta-term.

(iii) If $X_1, \ldots, X_n$ are meta-terms, $x_1, \ldots x_n$ are variables, and a is a meta-variable then $[X_1/x_1, \ldots, X_n/x_n]\ a$ is a meta-term.

Definition 2.4 A meta-context is a meta-term with a hole $[\,]$ in place of one of its sub-meta-terms.

In the following, we will use letters K, L, M, N, ... for meta-terms as well as terms. Letters C, D will denote (meta-)contexts.

Definition 2.5 A *valuation* ρ is a mapping from meta-variables to meta-terms that maps all but a finite number of meta-variables to themselves. We will write valuations in the same way as we write substitutions, i.e. $[\lambda x.x/a]$ is the valuation that assigns $\lambda x.x$ to a. The set of all valuations for a term language $\mathcal{T}$ will be denoted $\mathcal{V}_{\mathcal{T}}$. Where it is clear from the context we will leave out the subscript.

The meaning of a valuation is extended homomorphically to a mapping from meta-terms to meta-terms. We will also sometimes extend the meaning of valuation to a mapping from meta-terms to meta-contexts by defining $\rho\,[\,] \equiv [\,]$.

Definition 2.6 Two meta-terms X and Y are syntactically equal, $X \equiv Y$, if for all valuations ρ such that ρX and ρY are terms, $\rho X \equiv \rho Y$. X and Y are observationally equivalent $X \cong Y$, if for all valuations ρ such that ρX and ρY are terms, $\rho X \cong \rho Y$.

Definition 2.7 Substitution $[X/x]Y$ on meta-terms is defined inductively in the same way as it is defined on terms, with the added rule that

$$[Y/y]\,([X_1/x_1, \ldots, X_n/x_n]\ a) \quad \equiv \quad [Y/y, ([Y/y]X_1)/x_1, \ldots, ([Y/y]X_n)/x_n]\ a.$$

Lemma 2.8 For all meta-terms M, N, P, meta-variables a,

$$M \to N \;\Rightarrow\; [P/a]M \to [P/a]N.$$

Proof: Immediate from the definition of $(\to)$ ■

Definition 2.9 Let $\mathcal{S} \;=\; \{X_i \bowtie Y_i\}_{i \in I}$ be a set of equations between meta-terms of $\mathcal{T}$. The *compatible valuation closure* of S is the relation between meta-terms given by

$$C[\rho X_i] \overset{\rho X_i}{\bowtie} C[\rho Y_i]$$

where $i \in I$, C ranges over the meta-contexts of $\mathcal{T}$, and ρ ranges over the valuations of $\mathcal{T}$. We will leave out the superscript of $\bowtie$ if it is unimportant.

In the following we assume that $\mathcal{T}$ is a reduction system given by a term language $Terms(\mathcal{T})$, and a reduction relation $(\to)$ that is the compatible valuation closure of a system $\mathcal{R}$ of reduction rules on meta-terms U_i, V_i (U_i non-variable) in $MetaTerms(\mathcal{T})$.

$$\mathcal{R} \;=\; \{U_i \to V_i\}_{i \in I}$$

As usual, we write $(\twoheadrightarrow)$ for the reflexive and transitive closure of $(\to)$, and take equality $(=)$ to be the smallest equivalence relation that contains $(\to)$.

We further assume a similarity relation $(\sim)$ that is the compatible valuation closure of a symmetric system $\mathcal{S}$ of equations between non-variable meta-terms X_j, Y_j in $MetaTerms(\mathcal{T})$.

$$\mathcal{S} \;=\; \{X_j \sim Y_j\}_{j \in J}$$

We assume that the meta-variables in $\mathcal{R}$ are distinct from those in $\mathcal{S}$.

Definition 2.10 *Parallel similarity* $(\sim_1)$ is the smallest relation closed under the following three rules.

$$(ID)\;\; M \sim_1 M \qquad\qquad (SINGLE)\;\; \frac{M \;\sim\; N}{M \sim_1 N}$$

$$(COMP)\;\; \frac{M \sim_1 N \qquad P \sim_1 Q}{[P/a]M \sim_1 [Q/a]N}$$

We write $M \overset{X}{\sim}_1 N$ if $M {\sim_1} N$ and X is the set of all subterms of M that derive from pattern instances of single similarities $(\sim)$ in $M {\sim_1} N$. Formally, $\overset{X}{\sim}_1$ is defined as follows. Augment the term language by marked terms M^*. Let $\mathcal{L}(M)$ be the set of marked subterms in M and let $|M|$ be the erasure of M, in which all marks are deleted. Let $\sim_{1'}$ be the smallest relation closed under (ID), $(COMP)$ and

$$(SINGLE')\;\; \frac{C[L] \;\sim N}{C[L^*] \sim_{1'} N}\;.$$

Then define $|M| \overset{\mathcal{L}(M)}{\sim}_1 N$ iff $M {\sim_{1'}} N$.

2.3 Critical Pairs and Local Stability

Analogously to the notion of critical pairs in rewrite systems, we define critical pairs to be the result of applying two independent modifications to overlapping parts of a common term. Unlike the situation in rewrite systems, our modifications are of two different kinds, namely reduction and similarity.

Definition 2.11 (Interference, Critical Pair) Let $L \to R \in \mathcal{R}$, $S \sim T \in \mathcal{S}$, ρ be a valuation. Two sub-meta-terms ρL and ρS of a common meta-term *interfere*, if there is a non-variable meta-term M, meta-context C such that (4) or (5) holds.

$$L \equiv C[M] \quad \wedge \quad \rho M \equiv \rho S \tag{4}$$
$$S \equiv C[M] \quad \wedge \quad \rho M \equiv \rho L \tag{5}$$

Two terms M, N form a $(\to, \sim)$-*critical pair* if there exists a root term P, a redex Δ, and pattern instances of similarities $L_1, ..., L_n$ $(n \geq 1)$, such that

$$P \xrightarrow{\Delta} M \qquad \text{and} \qquad P \overset{\{L_1,\ldots,L_n\}}{\sim_1} N$$

and Δ interferes with each L_i $(i = 1, .., n)$. The pair is *deterministically critical* if there is an evaluation context[1] E such that $P \equiv E[\Delta]$.

We will often use the notation $[N \sim_1 P \to M]$ for a $(\to, \sim)$-critical pair M, N with root P.

Definition 2.12 $(\sim)$ is *locally stable* if for all $(\to, \sim)$-critical pairs M', N with root M there is a meta-term N' such that the following diagram commutes.

$$\begin{tikzcd} M \arrow[r] \arrow[d, dash, "\sim_1"'] & M' \arrow[d, dotted, dash, "\sim_1"] \\ N \arrow[r, dotted, dash, "\cong"'] & N' \end{tikzcd} \tag{6}$$

$(\sim)$ is *deterministically locally stable* if for all deterministically $(\to, \sim)$-critical pairs M', N with root M there are meta-terms M'', N' such that the following diagram commutes.

$$\begin{tikzcd} M \arrow[r, "d"'] \arrow[d, dash, "\sim_1"'] & M' \arrow[r, dotted, two heads, "d"'] & M'' \arrow[d, dotted, dash, "\sim_1"] \\ N \arrow[rr, dotted, dash, "\cong"'] & & N' \end{tikzcd} \tag{7}$$

[1]Evaluation contexts are defined in the next sub-section.

Lemma 2.13 If $(\sim)$ is locally stable then for all terms M, M', N with $N \sim_1 M$, $M \rightarrow M'$ there exists a term N' such that the following diagram commutes.

$$\begin{tikzcd} M \arrow[r] \arrow[d, dash, "\sim_1"'] & M' \arrow[d, dotted, dash, "\sim_1"] \\ N \arrow[r, dotted, dash, "\cong"'] & N' \end{tikzcd} \tag{8}$$

Proof: Let $M \overset{X}{\sim}_1 N$ and Δ be the redex of the reduction $M \rightarrow M'$. Let $O_1, ..., O_m$ be those terms in X that interfere with Δ. Let $P_1, ..., P_n$ be those terms in $X \backslash \{O_1, ..., O_m\}$ that are contained in either Δ or some O_i $(i = 1, ..., m)$.

Let K' be the smallest subterm of M that contains Δ and $O_1, ..., O_m$. Let K be the result of replacing each subterm P_i in K by a fresh meta-variable a_i $(i = 1, ..., n)$. Then $K' \equiv [P_1/a_1...P_n/a_n]\ K$. Furthermore, there are contexts C, D, as well as terms L, $Q_1, ..., Q_n$ such that

$$\begin{array}{lcl} M & \equiv & C[[P_1/a_1...P_n/a_n]\ K] \\ N & \equiv & D[[Q_1/a_1...Q_n/a_n]\ L] \\ P_i & \overset{P_i}{\sim} & Q_i \\ K & \overset{\{O_1,...,O_m\}}{\sim_1} & L \end{array}$$

Also, since $M \sim_1 N$, we must have $C[b] \sim_1 D[b]$ for all meta-variables b.

We construct in three stages a diagram that implies (8).

Stage 1: Let R be the reduct of K under Δ. That is, $[P_1/a_1...P_n/a_n]\ K \xrightarrow{\Delta} [P_1/a_1...P_n/a_n]\ R$. Since Δ does not interfere with $P_1, ..., P_n$, the following diagram commutes:

$$\begin{tikzcd} {[P_1/a_1...P_n/a_n]\ K} \arrow[r, "\Delta"] \arrow[d, dash, "\sim_1"'] & {[P_1/a_1...P_n/a_n]\ R} \arrow[d, dash, "\sim_1"] \\ {[Q_1/a_1...Q_n/a_n]\ K} \arrow[r] & {[Q_1/a_1...Q_n/a_n]\ R} \end{tikzcd} \tag{9}$$

Stage 2: Assume first that $m \geq 1$. Since Δ interferes with $O_1, ..., O_m$, R and L form a critical pair. Since $\mathcal{T}$ is locally stable, there exists then an R' such that the following diagram commutes:

$$\begin{tikzcd} K \arrow[r, "\Delta"] \arrow[d, dash, "\sim_1"'] & R \arrow[d, dash, "\sim_1"] \\ L \arrow[r, dash, "\cong"'] & R' \end{tikzcd} \tag{10}$$

On the other hand, if $m = 0$ then $K \equiv L$ and (10) can be made to commute with $R' \equiv R$. Furthermore, (10) still commutes if a valuation ρ is applied to each vertex:

$$\begin{tikzcd} \rho K \arrow[r, "\Delta"] \arrow[d, dash, "\sim_1"'] & \rho R \arrow[d, dash, "\sim_1"] \\ \rho L \arrow[r, dash, "\cong"'] & \rho R' \end{tikzcd} \tag{11}$$

Stage 3: Let ρ be some arbitrary valuation. By the previous stage, $\rho L \cong \rho R'$. Hence, by Lemma 2.2, also $C[\rho L] \cong C[\rho R']$ and $D[\rho L] \cong D[\rho R']$. Therefore, the following diagram commutes:

$$\begin{tikzcd} C[\rho L] \arrow[r, "\Delta"] \arrow[d, dash, "\sim_1"'] & C[\rho R'] \arrow[d, dash, "\sim_1"] \\ D[\rho L] \arrow[r, dash, "\cong"'] & D[\rho R'] \end{tikzcd} \tag{12}$$

Setting $\rho = [Q_1/a_1 ... Q_n/a_n]$ and stacking diagrams (9), (11), and (12) on top of each other yields:

$$
\begin{tikzcd}
M \equiv C[[P_1/a_1 ... P_n/a_n]\ K] \arrow[r, "\Delta"] \arrow[d, dash, "\sim_1"'] & C[[P_1/a_1 ... P_n/a_n]\ R] \equiv M' \arrow[d, dash, "\sim_1"] \\
C[[Q_1/a_1 ... Q_n/a_n]\ K] \arrow[r] \arrow[d, dash, "\sim_1"'] & C[[Q_1/a_1 ... Q_n/a_n]\ R] \arrow[d, dash, "\sim_1"] \\
C[[Q_1/a_1 ... Q_n/a_n]\ L] \arrow[r, dash, "\cong"'] \arrow[d, dash, "\sim_1"'] & C[[Q_1/a_1 ... Q_n/a_n]\ R'] \arrow[d, dash, "\sim_1"] \\
N \equiv D[[Q_1/a_1 ... Q_n/a_n]\ L] \arrow[r, dash, "\cong"'] & D[[Q_1/a_1 ... Q_n/a_n]\ R']
\end{tikzcd}
\tag{13}
$$

Looking on the right hand column of this diagram, we have $P_i \sim Q_i$, $R \sim_1 R'$, $C[b] \sim_1 D[b]$, for all b. By repeated application of rule $(COMP)$, $M' \sim_1 D[[Q_1/a_1 ... Q_n/a_n]\ R'] \stackrel{def}{=} N'$, which implies the proposition. ■

2.4 Deterministic Local Stability

We now work towards a version of Lemma 2.13 that can be applied to (deterministic) evaluation steps instead of reduction steps. The new version is generally easier to establish than Lemma 2.13, but holds only if the theory admits an evaluation procedure that is definable as a context-machine [3].

Definition 2.14 (Evaluation Contexts, Deterministic Reduction) Let $\mathcal{E}$ be a subset of the meta-contexts of $\mathcal{T}$. We define a binary relation $\rightarrow_{\mathcal{E}}$ on terms of $\mathcal{T}$ as follows:

> $M \rightarrow_{\mathcal{E}} N$ iff there are terms M', N' and there is an meta-context $E \in \mathcal{E}$ such that $M \equiv E[M']$, $N \equiv E[N']$, and $M' \xrightarrow{M'} N'$.

Then $\mathcal{E}$ is a set of *evaluation contexts* and $\rightarrow_{\mathcal{E}}$ is a *deterministic reduction* if the following two conditions are met.

- $\rightarrow_{\mathcal{E}}$ is deterministic. $M \rightarrow_{\mathcal{E}} N_1$ and $M \rightarrow_{\mathcal{E}} N_2$ implies $N_1 \equiv N_2$.
- $\rightarrow_{\mathcal{E}}$ is sound and complete for reduction to an answer. For all terms M, answers A,

$$M \twoheadrightarrow A \Leftrightarrow M \twoheadrightarrow_{\mathcal{E}} A$$

We also use the symbol $\xrightarrow[d]{}$ for deterministic reduction if the set $\mathcal{E}$ is clear from the context.

Definition 2.15 A set $\mathcal{E}$ of evaluation contexts is *downward closed* if, for all $E \in \mathcal{E}$, meta-contexts C_1, C_2, $E \equiv C_1 \cdot C_2$ implies that $C_2 \in \mathcal{E}$.

Example 2.1 λ has a set of evaluation contexts, which is generated by the grammar

$$E ::= [\,] \mid E\ M \mid p\ E. \tag{14}$$

This is a consequence of the Curry-Feys Standardization theorem for the λ-calculus ([2], CH 11, §4).

Proposition 2.16 Evaluation contexts for λ are downward closed.

Proof: Let E be an evaluation context, and let C_1, C_2 be meta-contexts such that $E \equiv C_1 \cdot C_2$. Using an induction on the form of C_1, we show that C_2 is an evaluation context. Since $E \equiv C_1 \cdot C_2$, C_1 must be of one of the forms of (14) If $C_1 \equiv [\,]$ then $E \equiv C_2$ and hence C_2 is an evaluation context. If $C_1 \equiv E'\ M$, for some evaluation context E' and term M, then there is a meta-context C_1' such that $E' \equiv C_1' \cdot C_2$. By the induction hypothesis, C_2 is an evaluation context. Finally, if $C_1 \equiv p\ E'$, for some primitive operator p and evaluation context E', then there is again a meta-context C_1' such that $E' \equiv C_1' \cdot C_2$. By the induction hypothesis, C_2 is an evaluation context. ∎

Definition 2.17 $(\sim)$ *preserves evaluation contexts* if, for all meta-terms M, $(\sim)$-pattern instances P, meta-contexts C, and meta-variables a, if $[P/a]C$ is an evaluation context then so is C.

If evaluation contexts are defined inductively then there is a syntactic criterion for preservation of evaluation contexts that is easy to check:

Definition 2.18 A *context-pattern* is formed from the inductive definitions of meta-context, plus a new alphabet of variables that range over contexts instead of terms.

Example 2.2 Evaluation contexts for λ are the least fixed point of the equation

$$e = \bigcup_{\rho \in \mathcal{V}} \rho([\,] \cup e\ a \cup \bigcup_{p \in Primops} \{p\ e\}) \tag{15}$$

where the expression inside the parentheses is a union of three context-patterns with context-variable e and meta-variable a.

Definition 2.19 A context-pattern C *overlaps* with a non-variable meta-term M if there is a nonvariable sub-meta-term N of C and a valuation ρ such that $\rho M \equiv \rho N$.

Definition 2.20 Let evaluation contexts be defined by an inductive definition

$$e = \bigcup_{\rho \in \mathcal{V}} \bigcup_{i \in I} \rho\ P_i$$

where each P_i is a context-pattern. Let $(\sim)$ be the compatible valuation closure of a symmetric system $\{X_j \sim Y_j\}_{j \in J}$. Then $(\sim)$ *interferes with evaluation contexts* if there is a P_i $(i \in I)$ that overlaps with an X_j $(j \in J)$.

Note that N in the previous definition is required to be a meta-term. That is, N cannot contain a hole [ ], nor can it contain a context-variable.

Proposition 2.21 In λ no similarity relation $(\sim)$ interferes with evaluation contexts.

Proof: The only subterms in the context patterns of (15) are meta-variables. Hence, no overlap is possible. ■

Proposition 2.22 If evaluation contexts are defined inductively and $(\sim)$ does not interfere with evaluation contexts then $(\sim)$ preserves evaluation contexts.

Proof: Assume that $(\sim)$ does not preserve evaluation contexts. We show that in that case $(\sim)$ must interfere with evaluation contexts.

Let P be a pattern instance of a similarity, and let C be a meta-context such that $[P/a]C$ is an evaluation context but C is not. If a does not occur in C then C is an evaluation context, which contradicts the assumption. Assume therefore that a does occur in C. Let evaluation contexts be given by the inductive definition

$$e = \bigcup_{\rho \in \mathcal{V}} \bigcup_{i \in I} \rho\ P_i$$

for some index set I, and context patterns P_i. Then

$$[P/a]C \ \equiv\ E_1[[E_2/e]([P/a]Q)]$$

for some evaluation contexts E_1, E_2, context variable e, and valuation instance Q of a context pattern P_i that contains a. But this implies that $(\sim)$ overlaps with P_i. Hence, $(\sim)$ interferes with evaluation contexts. ■

Lemma 2.23 If $\mathcal{T}$ has downward closed evaluation contexts and $(\sim)$ is deterministically locally stable and preserves evaluation contexts then for all terms M, M', N with $N \sim_1 M$, $M \underset{d}{\rightarrow} M'$ there exist terms M'', N' such that the following diagram commutes.

$$\begin{tikzcd}
M \arrow[r, "d"'] \arrow[d, dash, "\sim_1"'] & M' \arrow[r, dotted, twoheadrightarrow, "d"'] & M'' \arrow[d, dotted, dash, "\sim_1"] \\
N \arrow[rr, dotted, dash, "\cong"'] & & N'
\end{tikzcd}$$

Proof: Largely analogous to the proof of Lemma 2.13. Let $M \overset{X}{\sim}_1 N$. Let Δ be the redex of the reduction $M \rightarrow M'$. Let $O_1, ..., O_m$ be those terms in X that interfere with Δ. Let $P_1, ..., P_n$ be those terms in $X \backslash \{O_1, ..., O_m\}$ that are contained in either Δ or some O_i $(i = 1, ..., m)$.

As in the proof of Lemma 2.13, let K' be the smallest subterm of M that contains Δ and $O_1, ..., O_m$. Let K be the result of replacing each subterm P_i in K by a fresh meta-variable a_i $(i = 1, ..., n)$.

Then $K' \equiv [P_1/a_1...P_n/a_n]\ K$. Furthermore, there are contexts C, D, as well as terms $L, Q_1, ..., Q_n$ such that

$$\begin{array}{ccc} M & \equiv & C[[P_1/a_1...P_n/a_n]\ K] \\ N & \equiv & D[[Q_1/a_1...Q_n/a_n]\ L] \\ P_i & \overset{P_i}{\sim} & Q_i \\ K & \overset{\{O_1,...,O_m\}}{\sim_1} & L \end{array}$$

Since $M {\sim_1} N$, we must have $C[b] \sim_1 D[b]$ for all meta-variables b. Since $M \underset{d}{\rightarrow} M'$ there is an evaluation context E such that $M \equiv E[\Delta]$.

Let E_0 and Δ_0 be such that $K \equiv E_0[\Delta_0]$ and $[P_1/a_1...P_n/a_n]\Delta_0 \equiv \Delta$. Let $E' \equiv [P_1/a_1....P_n/a_n]E_0$. Since

$$E[\Delta] \ \equiv \ M \ \equiv \ C[[P_1/a_1...P_n/a_n]\ K] \ \equiv \ C[E'[\Delta]],$$

one has that $E \equiv C \cdot E'$. Since E is an evaluation context and $\mathcal{T}$ is downward closed it follows that E' is also an evaluation context. Since $E' \equiv [P_1/a_1]([P_2/a_2....P_n/a_n]E_0)$, and since $(\sim)$ preserves evaluation contexts, $[P_2/a_2....P_n/a_n]E_0$ is an evaluation context. Repeating this step n times, we get that E_0 is an evaluation context.

Similarly to the proof of Lemma 2.13 we now construct in three stages a diagram that implies (Lemma 2.23). Stages 1 and 3 are exactly as in the proof of Lemma 2.13.

For Stage 2, we reason as follows. Let R be as in the proof of Lemma 2.13. Assume first that $m \geq 1$.

Let Δ' be the redex of the reduction $[Q_1/a_1...Q_n/a_n]\ K \rightarrow [Q_1/a_1...Q_n/a_n]\ R$. Since Δ interferes with $O_1, ..., O_m$, R and L form a critical pair with root K and redex Δ_0. The pair is deterministically critical, since $K \equiv E_0[\Delta_0]$, and E_0 is an evaluation context. Since $\mathcal{T}$ is deterministically locally stable, there exists then meta-terms R', R'' such that the following diagram commutes:

$$\begin{tikzcd} K \arrow[r, "d"'] \arrow[d, dash, "\sim_1"'] & R \arrow[r, dotted, two heads, "d"'] & R'' \arrow[d, dotted, dash, "\sim_1"] \\ L \arrow[rr, dotted, dash, "\cong"'] & & R' \end{tikzcd}$$

The rest of Stage 2 is as in the proof of Lemma 2.13.

Stacking the results of the three stages on top of each other yields:

$$
\begin{tikzcd}
M \equiv C[[P_1/a_1 ... P_n/a_n]\ K] \arrow[r, "\Delta"] \arrow[d, dash, "\sim_1"'] & C[[P_1/a_1 ... P_n/a_n]\ R] \equiv M' \arrow[r, two heads, "d"'] \arrow[d, dash, "\sim_1"] & C[[P_1/a_1 ... P_n/a_n]\ R''] \arrow[d, dash, "\sim_1"] \\
C[[Q_1/a_1 ... Q_n/a_n]\ K] \arrow[r, "d"'] \arrow[d, dash, "\sim_1"'] & C[[Q_1/a_1 ... Q_n/a_n]\ R] \arrow[r, two heads, "d"'] & C[[Q_1/a_1 ... Q_n/a_n]\ R''] \arrow[d, dash, "\sim_1"] \\
C[[Q_1/a_1 ... Q_n/a_n]\ L] \arrow[rr, dash, "\cong"'] \arrow[d, dash, "\sim_1"'] & & C[[Q_1/a_1 ... Q_n/a_n]\ R'] \arrow[d, dash, "\sim_1"] \\
N \equiv D[[Q_1/a_1 ... Q_n/a_n]\ L] \arrow[rr, dash, "\cong"'] & & D[[Q_1/a_1 ... Q_n/a_n]\ R']
\end{tikzcd}
\tag{16}
$$

Looking on the rightmost column of this diagram, we have $P_i \sim Q_i$, $R'' \sim_1 R'$, $C[b] \sim_1 D[b]$. By repeated application of rule $(COMP)$,

$$M'' \stackrel{def}{=} C[[P_1/a_1 ... P_n/a_n]\ R''] \sim_1 D[[Q_1/a_1 ... Q_n/a_n]\ R'] \stackrel{def}{=} N'.$$

■

2.5 Proving Observational Equivalences

We now use the previous results to develop two criteria for observational equivalences; one applying to conventional reduction, the other applying to deterministic reduction.

Definition 2.24 Let $(\sim)$ be a binary relation on terms in $\mathcal{T}$. Then $(\sim)$ *preserves answers* if, for all meta-terms M, answers A, $M \sim A \Rightarrow M \twoheadrightarrow A$.

Lemma 2.25 If $(\sim)$ preserves answers then so does $(\sim_1)$.

Proof: Assume $M \sim_1 A$, for some term M, answer A. We show $M \twoheadrightarrow A$ by an induction of the derivation of $M \sim_1 A$.

If $M \sim_1 A$ by rule (ID), then $M \equiv A$ by the premise of this rule. If $M \sim_1 A$ by rule $(SINGLE)$ then $M \twoheadrightarrow A$ by the premise of the lemma. If $M \sim_1 A$ by rule $(COMP)$, then there exist by the premise of this rule meta-terms P, Q, M', N' and a meta-variable a such that $M \equiv [P/a]M'$, $A \equiv [Q/a]N'$, $P \sim_1 Q$, $M' \sim_1 N'$. $A \equiv [Q/a]N'$ implies either $N' \equiv A$ or $N' \equiv a \wedge Q \equiv A$. We distinguish between the two cases.

If $N' \equiv A$, then $M' \twoheadrightarrow A$ by the induction hypothesis. Hence, by Lemma 2.8 $[P/a]M' \twoheadrightarrow [P/a]A \equiv A$.

On the other hand, if $N' \equiv a$ and $Q \equiv A$ then $P \twoheadrightarrow A$ by the induction hypothesis. Furthermore, $M' {\sim_1} N'$ and $N' \equiv a$ imply $M' \equiv a$ since pattern instances of similarities are non-variable terms. Hence, $[P/a]M \equiv [P/a]a \equiv P \twoheadrightarrow A$. ■

Theorem 2.26 Let $\approx$ be the transitive closure of $(\sim)$. If $(\sim)$ is locally stable and $(\sim)$ preserves answers then $\approx \subseteq \cong$.

Proof: (i) We first show a slightly simpler result: For all terms M, N, answers A,

$$M {\sim_1} N \ \wedge\ M \twoheadrightarrow A \ \Rightarrow\ N \twoheadrightarrow A \tag{17}$$

The result is shown by an induction on the length of reduction from M to A. If $M \equiv A$, then $N {\sim_1} A$, and hence $N \twoheadrightarrow A$ since $(\sim)$ preserves answers.

If $M \rightarrow M' \twoheadrightarrow A$ then by Lemma 2.13 there is a term N' such that $M' {\sim_1} N'$ and $N \cong N'$. Then by the induction hypothesis, $N' \twoheadrightarrow A$, which together with $N \cong N'$ implies $N \twoheadrightarrow A$. This shows (17).

An obvious consequence of (17) is that, for all terms M, N, contexts C, answers A,

$$C[M] {\sim_1} C[N] \ \wedge\ C[M] \twoheadrightarrow A \ \Rightarrow\ C[N] \twoheadrightarrow A.$$

Hence, $\sim_1\ \subseteq \cong$. Since $\approx$ is the transitive closure of $(\sim_1)$ and $\cong$ is transitive this implies $\approx \subseteq \cong$. ■

Theorem 2.27 Let $\approx$ be the transitive closure of $(\sim)$. If

- $\mathcal{T}$ has downward closed evaluation contexts,
- $(\sim)$ preserves evaluation contexts,
- $(\sim)$ is deterministically locally stable, and
- $(\sim)$ preserves answers

then $\approx \subseteq \cong$.

Proof: We show (17) as follows. Let $M \twoheadrightarrow A$. Since $\mathcal{T}$ has downward-closed evaluation contexts, there exists a deterministic reduction from $M \underset{d}{\twoheadrightarrow} A$. We perform an induction on the length of this sequence. The base case is as in the proof of Theorem 2.26. For the induction step, assume $M \underset{d}{\rightarrow} M' \underset{d}{\twoheadrightarrow} A$. Then by Lemma 2.23 there are terms N', M'' such that $M' \underset{d}{\twoheadrightarrow} M''$, $M'' {\sim_1} N'$, and $N \cong N'$. Then by the induction hypothesis, $N' \twoheadrightarrow A$, which together with $N \cong N'$ implies $N \twoheadrightarrow A$. This shows (17), from which the proposition follows as in the proof of Theorem 2.26. ■

3 Application to $\lambda\nu$

In this section we apply Theorem 2.26 and Theorem 2.27 to show some observational equivalences for $\lambda\nu$ [8].

x	$\in$	$Idents$	λ-bound identifiers
n	$\in$	$Names = Names^c \cup Names^\nu$	names
n^c	$\in$	$Names^c$	constants
n^ν	$\in$	$Names^\nu$	ν-bound local names
p	$\in$	$Primops$	primitive operators
M	$\in$	$\Lambda\nu$	terms

$$
\begin{array}{rcl}
M & ::= & x \mid \lambda x.M \mid M_1\ M_2 \\
 & \mid & n \mid \nu n.M \mid M_1 == M_2 \\
 & \mid & (M_1, M_2) \mid p\ M
\end{array}
$$

Figure 1: Syntax of $\lambda\nu$

$$
\begin{array}{llcll}
\beta & (\lambda x.M)\ N & \rightarrow & [N/x]\ M & \\
\delta & p\ V & \rightarrow & \delta(p, V) & \\
 & & & & \\
eq & n == n & \rightarrow & true & \\
 & n == m & \rightarrow & false & (n \neq m) \\
 & & & & \\
\nu_\lambda & \nu n.\lambda x.M & \rightarrow & \lambda x.\nu n.M & \\
\nu_p & \nu n.(M_1, M_2) & \rightarrow & (\nu n.M_1,\ \nu n.M_2) & \\
\nu_n & \nu n.m & \rightarrow & m & (n \neq m)
\end{array}
$$

Figure 2: Reduction rules for $\lambda\nu$

3.1 The $\lambda\nu$ calculus

$\lambda\nu$ extends λ with *local names*. Its term language and reduction rules are given in Figures 1 and 2. The construct $\nu n.M$ binds a name n in a term M. $FN(M)$ denotes the set of names that occur free in M.

Viewed formally, the reduction relation of $\lambda\nu$ is the compatible valuation closure of the following system of equations.

$$
\begin{array}{llllll}
\{ & (\lambda x.a)\,b & \to & [b/x]\,a & \mid\ x \in Idents\} & \cup \\[1ex]
\{ & p\,V & \to & \delta(p,V) & \mid\ p \in Primops, V \in Values\} & \cup \\[1ex]
\{ & n == n & \to & true & \mid\ n \in Names\} & \cup \\
\{ & n == m & \to & false & \mid\ m, n \in Names, m \neq n\} & \cup \\[1ex]
\{ & \nu n.(a,b) & \to & (\nu n.a,\ \nu n.b) & \mid\ n \in Names\} & \cup \\
\{ & \nu n.\lambda x.a & \to & \lambda x.\nu n.a & \mid\ n \in Names, x \in Idents\} & \cup \\
\{ & \nu n.m & \to & m & \mid\ m, n \in Names, m \neq n\} &
\end{array}
$$

In [8], Theorem 4.6 it was shown that $\lambda\nu$ has a set of evaluation contexts that is generated by the grammar

$$E ::= [\,]\ \mid\ E\ M\ \mid\ p\ E\ \mid\ \nu n.E. \tag{18}$$

Proposition 3.1 $\lambda\nu$ has downward closed evaluation contexts.

Proof: Essentially identical to the proof of Proposition 2.16. The additional induction step $E \equiv \nu n.E'$ is completely analogous to the other two induction steps in Proposition 2.16. ■

Proposition 3.2 In $\lambda\nu$ no similarity relation ($\sim$) interferes with evaluation contexts.

Proof: The set of evaluation contexts of $\lambda\nu$ is the least fixed point of the equation

$$e = \bigcup_{\rho \in \mathcal{V}} \rho([\,]\ \cup\ e\ a\ \cup \bigcup_{p \in Primops} \{p\ e\}\ \cup \bigcup_{n \in Names^{\nu}} \nu n.a) \tag{19}$$

The only subterms in the context patterns of (19) are meta-variables. Hence, a side of a similarity cannot overlap with a context-pattern. ■

3.2 Observational Equivalences in $\lambda\nu$

Proposition 3.3 The following are observational equivalences in $\lambda\nu$:

$$\nu n.M \cong M \qquad \textbf{if } n \notin FV(M) \tag{20}$$

$$\nu n.\nu m.M \cong \nu m.\nu n.M \tag{21}$$

Proof: (20) corresponds to the compatible valuation closure of the symmetric system

$$
\begin{array}{l}
\{\ \nu n.M \sim M \quad \mid \quad n \in Names, M \in Terms, n \notin FN(M)\ \}\ \cup \\
\{\ M \sim \nu n.M \quad \mid \quad n \in Names, M \in Terms, n \notin FN(M)\ \}
\end{array} \tag{22}
$$

We first show that $\sim$ preserves answers. Assume that $M \sim A$. Because of the form of (22), this relation must have been derived from a similarity $nun.A \sim A$, where $n \notin FN(A)$. Since answers are names in $\lambda\nu$, A is a name, and we have $M \to A$ by a single ν_n reduction.

We now show that $\sim$ is locally stable. Matching (22) against $\lambda\nu$'s reduction rules establishes that a redex Δ interferes with a pattern instance $\nu n.M$ iff $\Delta \subset \nu n.M$. We distinguish according to the relative position of Δ and $\nu n.M$. Assume first $\Delta \subset M$ and let M' such that $M \xrightarrow{\Delta} M'$. Then there is the following instance of diagram (8):

$$\begin{tikzcd} \nu n.M \arrow[r, "\Delta"] \arrow[d, dash, "\sim"'] & \nu n.M' \arrow[d, dash, "\sim"] \\ M \arrow[r, "\Delta"] & M' \end{tikzcd}$$

The similarity $\nu n.M' \sim M'$ in this diagram follows from the premise $\nu n.M \sim M$ and the fact that reduction in $\lambda\nu$ does not create new free names, i.e. $M \to M'$ implies $FN(M') \subseteq FN(M)$.

Assume now that $\Delta \equiv \nu n.M$. We further distinguish according to the notion of reduction with Δ as redex. There are three possibilities:

$$\begin{array}{lcl} \nu n.\lambda x.M & \to & \lambda x.\nu n.M \\ \nu n.(M_1, M_2) & \to & (\nu n.M_1, \nu n.M_2) \\ \nu n.m & \to & m \end{array}$$

where $n \notin FN(M) \cup FN(M_1) \cup FN(M_2)$ and $n \neq m$. Diagram (8) can be made to commute for each of these, as can be seen from the following three diagrams:

$$\begin{tikzcd} \nu n.\lambda x.M \arrow[r] \arrow[d, dash, "\sim"'] & \lambda x.\nu n.M \arrow[dl, dash, "\sim"] \\ \lambda x.M & \end{tikzcd} \qquad \begin{tikzcd} \nu n.(M_1, M_2) \arrow[r] \arrow[d, dash, "\sim"'] & (\nu n.M_1, \nu n.M_2) \arrow[dl, dash, "\sim_1"] \\ (M_1, M_2) & \end{tikzcd} \qquad \begin{tikzcd} \nu n.m \arrow[r] \arrow[d, dash, "\sim"'] & m \arrow[dl, dash, "\equiv"] \\ m & \end{tikzcd}$$

With Theorem 2.26, (20) follows.

(21) corresponds to the symmetric system

$$\{\ \nu n.\nu m.a \ \sim\ \nu m.\nu n.a \quad | \quad n, m \in Names, n \neq m\}. \tag{23}$$

Let $\sim$ be the compatible valuation closure of this system. We use Theorem 2.27 to show that $(\sim)$ is an observational equivalence. From Proposition 3.1 we know that $\lambda\nu$ has downward closed evaluation contexts. From Proposition 3.2 and Proposition 2.22 we know that $(\sim)$ preserves evaluation contexts. Furthermore, since no side of (23) matches an answer, $(\sim)$ vacuously preserves answers. Hence, it only remains to show that $(\sim)$ is deterministically locally stable.

Matching (23) against $\lambda\nu$'s reduction rules establishes that there are three classes of critical pairs:

$$\begin{array}{lrclcll} [& \nu m.\nu n.\lambda x.M & \sim & \nu n.\nu m.\lambda x.M & \to & \nu n.\lambda x.\nu m.M &], \\ [& \nu m.\nu n.(M_1, M_2) & \sim & \nu n.\nu m.(M_1, M_2) & \to & \nu n.(\nu m.M_1, \nu m.M_2) &], \\ [& \nu m.\nu n.n' & \sim & \nu n.\nu m.n' & \to & \nu n.n' &] \end{array}$$

where M, M_1, M_2 are meta-terms, n, m, n' are names, and $m \neq n'$. Diagram (7) can be made to commute for each of these, as can be seen from the following three commuting diagrams:

$$\begin{tikzcd}
\nu n.\nu m.\lambda x.M \arrow[r, "d"'] \arrow[d, dash, "\sim"'] & \nu n.\lambda x.\nu m.M \arrow[r, "d"'] & \lambda x.\nu n.\nu m.M \arrow[d, dash, "\sim"] \\
\nu m.\nu n.\lambda x.M \arrow[rr, twoheadrightarrow] & & \lambda x.\nu m.\nu n.M
\end{tikzcd}$$

$$\begin{tikzcd}
\nu n.\nu m.(M_1, M_2) \arrow[r, "d"'] \arrow[d, dash, "\sim"'] & \nu n.(\nu m.M_1, \nu m.M_2) \arrow[r, "d"'] & (\nu n.\nu m.M_1, \nu n.\nu m.M_2) \arrow[d, dash, "\sim_1"] \\
\nu m.\nu n.(M_1, M_2) \arrow[rr, twoheadrightarrow] & & (\nu m.\nu n.M_1, \nu m.\nu n.M_2)
\end{tikzcd}$$

$$\begin{tikzcd}
\nu n.\nu m.n' \arrow[r] \arrow[d, dash, "\sim"'] & \nu n.n' \\
\nu m.\nu n.n' \arrow[ur, dash, "\cong"'] &
\end{tikzcd}$$

The $\cong$-diagonal of the last diagram is justified by (20), since $m \neq n'$. Hence, $(\sim)$ is deterministically locally stable. With Theorem 2.27, the transitive closure of $(\sim)$ is an observational equivalence, which implies (21). ∎

4 Conclusions

We have presented a syntactic method for proving that a given relation between terms is an observational equivalence. The method has been used succesfully in many proofs about the observational equivalence theories of λ_{var} and $\lambda\nu$. Hopefully it will be useful to others as well.

Acknowledgements This work was supported in part by grant N00014-91-J-4043 from DARPA. Many thanks to Vincent Dornic and Dan Rabin for their detailed comments on previous drafts of this paper.

References

[1] S. Abramsky. The lazy lambda calculus. In *Research Topics in Functional Programming*, The UT Year of Programming Series, chapter 4. Addison-Wesley Publishing Company, Inc., 1990.

[2] H. P. Barendregt. *The Lambda Calculus: its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, revised edition, 1984.

[3] E. Crank and M. Felleisen. Parameter-passing and the lambda-calculus. In *Proc. 18th ACM Symposium on Principles of Programming Languages, Orlando, Florida*, pages 233–244, January 1991.

[4] D. E. Knuth and P. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Compuatational Problems in Abstract Algebra*, pages 263–297. Pergamon, Oxford, 1970.

[5] I. Mason and C. Talcott. Equivalence in functional languages with side effects. *Journal of Functional Programming*, 1(3):287–327, July 1991.

[6] R. Milner. Fully abstract models of typed λ-calculi. *Theoretical Computer Science*, 4:1–22, 1977.

[7] R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1989.

[8] M. Odersky. A syntactic theory of local names. Research Report YALEU/DCS/RR-965, Department of Computer Science, Yale University, May 1993.

[9] M. Odersky and D. Rabin. The unexpurgated call-by-name, assignment, and the lambda-calculus. Research Report YALEU/DCS/RR-930, Department of Computer Science, Yale University, May 1993.

[10] G. D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.