Bad variables under control

Andrzej S. Murawski*

Oxford University Computing Laboratory,
Wolfson Building, Parks Road, Oxford OX1 3QD, UK

Abstract. We give a fully abstract game model for Idealized Algol with non-local control flow. In contrast to most previous papers on game semantics, we do not need to include the bad-variable constructor **mkvar** to obtain full abstraction. Using the model we show that, unlike in the "control-free" case, the presence of **mkvar** does affect observational equivalence. We conclude by discussing the effect of **mkvar** on nondeterministic and probabilistic variants of Idealized Algol.

1 Introduction

In the computer science classic "The essence of Algol" [1], Reynolds has laid out a series of principles that, in his opinion, should underlie the contemporary evolution of programming languages. He also defined a prototype language, called Idealized Algol, whose design was to be their embodiment. Based on a simple imperative language extended with the procedural mechanism of the call-by-name lambda calculus, Idealized Algol has come to be regarded as a canonical proposal for synthesizing functional and imperative programming. Its elegant definition has lent itself to a systematic analysis leading to significant progress in the field of programming language semantics [2].

One of Reynolds's insights concerned the semantics and type-theoretic treatment of assignable variables. He viewed them as dual in nature: producing values on the one hand (like expressions) and capable of accepting them on the other. This idea is reminiscent of the distinction between l- and r-values, but Reynolds took it much further: he advocated that the variable type be actually *identified* with the product of an "acceptor type" and the expression type. This decomposition enabled him to define the meaning of variables without any commitment to the structure of state, suggesting new abstract approaches to modelling state. Reynolds himself pursued one, based on functor categories, but alternatives have also emerged.

A particularly fruitful way of modelling computation turned out to be based on the idea that programs should be viewed as processes interacting with one another. Reddy calls this the object-based approach to semantics [3]. From this point of view, Reynolds's analysis of variables simply suggests viewing a variable as an object equipped with a reading- and a writing-method which it uses to

* Funded by an Advanced Research Fellowship from the UK EPSRC (EP/C539753/1).

communicate with the environment. Similar philosophy can be found in encodings of imperative programs into process algebras. In denotational semantics this approach was adopted in Abramsky and McCusker's game model of Idealized Algol [4] and Reddy's work on coherence spaces [3].

Game semantics has led to many full abstraction results ever since. These amount to defining a model such that equality in the model corresponds to program equivalence, or, in the more general inequational variant, such that observational approximation coincides with a distinguished preorder on the model. A general way of proving such results leads through a definability argument, which demonstrates that all compact elements of the model are definable, i.e. are denotations of some programs. However, when the variable type is a product type, definability fails unless an explicit pairing construct is added to the language. In the case of [4] this is the so-called "bad-variable constructor" **mkvar**, which makes a writing method M and a reading method N into a variable:

$$\frac{\Gamma \vdash M : \mathsf{exp} \to \mathsf{com} \quad \Gamma \vdash N : \mathsf{exp}}{\Gamma \vdash \mathbf{mkvar}(M, N) : \mathsf{var}}.$$

The reduction rules for $\mathbf{mkvar}(M, N)$ simply evaluate N for reading and evaluate Mv when the value v is to be written. Because, in absence of **mkvar**, definability fails, it seems likely that a game model that is (equationally or inequationally) fully abstract for a language with **mkvar**, will not be fully abstract in its absence. Thus it is natural to ask how one can repair existing game models to analyze observational approximation and equivalence in **mkvar**-free settings and whether this is really needed in all cases. This direction is also motivated by the fact that **mkvar** is not really a conventional programming construct, though languages incorporating bad variables as a feature do exist in the literature[1].

This paper uses a version of Idealized Algol that allows for side-effects in expressions and variables, to which we shall henceforth refer as IA. A fully abstract model for IA with **mkvar** was given in [4], where the preorder for inequational full abstraction was simply inclusion of *complete* plays (those in which all questions have been answered). Subsequently, McCusker [7] has introduced a different preorder on complete plays which captures observational approximation in IA. His was actually the only paper so far that has analyzed the game semantics of an IA-like language without **mkvar**. The aim of the present paper is to achieve a full abstraction result for IA enriched with non-local control flow (also without **mkvar**), i.e. in a setting where the transfer of control can violate the usual discipline of block entry and exit. Syntactically, the requisite extension of IA can be achieved by adding a **catch**-construct, in the spirit of Reynolds's escape [1] and Cartwright and Felleisen's control operators [8]. It is well known that violations of the stack discipline can be modelled in game semantics by relaxing the

[1] In the late 1960s POP-2 [5] was an attempt to define a broad-spectrum language for both numerical and symbolic computation combining the strengths of FORTRAN and LISP. The language was based on *doublets*, which are conceptually the same as **mkvar**-objects. Also, Reynolds's GEDANKEN [6] had support for *implicit references* in the spirit of **mkvar**.

bracketing condition [9]. However, full abstraction for $\mathsf{IA}+\mathbf{catch}$ is not merely a matter of applying McCusker's preorder in the unbracketed setting and we show what further refinements are necessary to accomplish it. Our model can then be used to demonstrate that, in contrast to McCusker's result on conservativity of $\mathsf{IA}+\mathbf{mkvar}$ (with respect to observational equivalence), $\mathsf{IA}+\mathbf{catch}+\mathbf{mkvar}$ does *not* extend $\mathsf{IA}+\mathbf{catch}$ conservatively. Next we go on to investigate the impact of **mkvar** on other important extensions of IA, namely, with nondeterminism and probability. It turns out that **mkvar** does affect observational equivalence in $\mathsf{IA}+\mathbf{or}$, but it does not for $\mathsf{IA}+\mathbf{coin}$.

In addition to the already-cited paper by McCusker, techniques based on nominal sets have recently been proposed as a general approach to handling the absence of **mkvar** [10, 11]. Because they bring an additional layer of combinatorial complexity to game semantics (names inside moves, name invariance), it seems a valuable goal to understand how the same results can still be achieved by "anonymous" game semantics. This paper explores this direction in the call-by-name setting of Algol-like languages, leaving call-by-value as a challenge for future work.

2 IA, $\mathsf{IA}_{\mathbf{catch}}$

We consider Reynolds's Idealized Algol [1] in which expressions and variables can produce side-effects. Its syntax is given in Figure 1. The types T are formed from the base types

$$B ::= \mathsf{com} \mid \mathsf{exp} \mid \mathsf{var}$$

using the $\rightarrow$ constructor: $T ::= B \mid T \rightarrow T$. The base types represent commands, natural-number-valued expressions and variables respectively. The (call-by-name) operational semantics of the language can be found in [12]. We assume that initially all variables have value 0 and write Ω_T for the divergent term $\mathbf{Y}_T(\lambda x^T.x) : T$. In IA the flow of control can be influenced locally through sequential composition and conditionals. Non-local flow control can be added, for example, via the construct:

$$\frac{\Gamma, x : \mathsf{com} \vdash M : \mathsf{com}}{\Gamma \vdash \mathbf{catch}\, x\, \mathbf{in}\, M : \mathsf{com}}.$$

Operationally, $\mathbf{catch}\, x\, \mathbf{in}\, M$ amounts to encapsulating M in a block so that any occurrence of x in M will trigger a forward jump out of the block. Some typical control constructs found in programming languages, e.g. `break` and `continue` of C, can easily be expressed in IA augmented with **catch**. Using side-effects one can also detect whether an early exit has taken place. For instance, it is possible to simulate Cartwright and Felleisen's control operator $\mathbf{catch}_{\mathsf{exp}}$ [8]:

$$\frac{\Gamma, x : \mathsf{exp} \vdash M : \mathsf{exp}}{\Gamma \vdash \mathbf{catch}_{\mathsf{exp}}\, x\, \mathbf{in}\, M : \mathsf{exp}},$$

which returns 0 for early exit and $n+1$ if M evaluates to n, by

$$\mathbf{new}\, X, Y\, \mathbf{in}\, \mathbf{catch}\, z\, \mathbf{in}\, X := M[(Y := 1; z; 0)/x]; \mathbf{if}\, !Y\, \mathbf{then}\, 0\, \mathbf{else}\, \mathbf{succ}(!X).$$

$$\frac{}{\Gamma \vdash \mathbf{skip} : \mathsf{com}} \qquad \frac{i \in \mathbb{N}}{\Gamma \vdash i : \mathsf{exp}} \qquad \frac{}{\Gamma, x : T \vdash x : T}$$

$$\frac{\Gamma \vdash M : \mathsf{exp}}{\Gamma \vdash \mathbf{succ}(M) : \mathsf{exp}} \qquad \frac{\Gamma \vdash M : \mathsf{exp}}{\Gamma \vdash \mathbf{pred}(M) : \mathsf{exp}}$$

$$\frac{\Gamma \vdash M : \mathsf{exp} \quad \Gamma \vdash M_0, M_1 : B}{\Gamma \vdash \mathbf{if}\, M\, \mathbf{then}\, M_1\, \mathbf{else}\, M_0 : B} \qquad \frac{\Gamma \vdash M : \mathsf{com} \quad \Gamma \vdash N : B}{\Gamma \vdash M; N : B}$$

$$\frac{\Gamma, x : T \vdash M : T'}{\Gamma \vdash \lambda x^T.M : T \to T'} \qquad \frac{\Gamma \vdash M : T \to T' \quad \Gamma \vdash N : T}{\Gamma \vdash MN : T'}$$

$$\frac{\Gamma \vdash M : \mathsf{var}}{\Gamma \vdash !M : \mathsf{exp}} \qquad \frac{\Gamma \vdash M : \mathsf{var} \quad \Gamma \vdash N : \mathsf{exp}}{\Gamma \vdash M := N : \mathsf{com}}$$

$$\frac{\Gamma, x : \mathsf{var} \vdash M : B}{\Gamma \vdash \mathbf{new}\, x\, \mathbf{in}\, M : B} \qquad \frac{}{\Gamma \vdash \mathbf{Y}_T : (T \to T) \to T}$$

Fig. 1. IA syntax.

catch can be regarded as an atom of non-local control: a minimalistic, yet expressive, mechanism capable of modelling the effect of more complicated control constructs such as labelled jumps (`goto`) and call-with-current-continuation (`callcc`) [9].

In the paper we shall consider extensions $\mathcal{L} = \mathsf{IA}{+}\Box$ of IA, written $\mathsf{IA}_\Box$, where $\Box \subseteq \{\mathbf{catch}, \mathbf{mkvar}, \mathbf{or}\}$. The operational semantics of each of these languages induces a notion of termination $M \Downarrow_{\mathcal{L}}$ for closed terms $\vdash M : \mathsf{com}$. Then we can define observational approximation and equivalence as follows.

Definition 1. *Suppose $\Gamma \vdash M_1, M_2 : T$ are terms of $\mathcal{L}$. $\Gamma \vdash M_1 : T$ approximates $\Gamma \vdash M_2 : T$ (written $\Gamma \vdash M_1 \sqsubseteq\!\!\!\approx_{\mathcal{L}} M_2$) iff, for all $\mathcal{L}$-contexts $C[-]$ such that $\vdash C[M_1], C[M_2] : \mathsf{com}$ holds, $\mathcal{C}[M] \Downarrow_{\mathcal{L}}$ implies $\mathcal{C}[N] \Downarrow_{\mathcal{L}}$. Two $\mathcal{L}$-terms are equivalent (written $\Gamma \vdash M_1 \cong_{\mathcal{L}} M_2$) if they $\sqsubseteq\!\!\!\approx_{\mathcal{L}}$-approximate each other.*

3 Games

The focus of this section is a full abstraction result for $\mathsf{IA}_{\mathbf{catch}+\mathbf{mkvar}}$, which can readily be synthesized from [12] and [4]. The first to study control operators in game semantics was Laird [9], who discovered that their presence can be modelled by relaxing the *bracketing* condition.

Definition 2. *An* arena *is a triple $A = \langle M_A, \lambda_A, \vdash_A \rangle$, where*

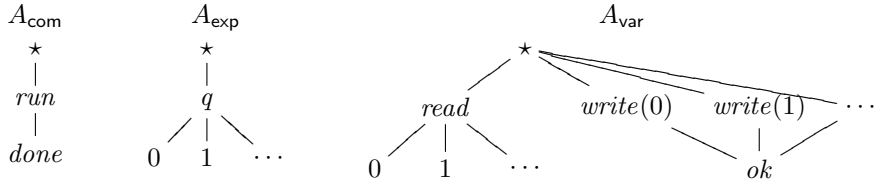
- *M_A is a set of* moves*;*
- *$\lambda_A : M_A \to \{O, P\} \times \{Q, A\}$ is a function indicating to which player (O or P) a move belongs and of what kind it is (question or answer);*
- *$\vdash_A \subseteq (M_A + \{\star\}) \times M_A$ is the so-called* enabling *relation, which must satisfy the following conditions.*

- *If $\star$ enables a move then it is an O-question without any other enabler. A move like this is called* initial *and we shall write I_A for the set containing all initial moves of A.*
- *If one move enables another then the former must be a question and the two moves must belong to different players.*

Product and arrow arenas can be constructed as follows:

$$\begin{array}{ll} M_{A\times B} = M_A + M_B & M_{A\Rightarrow B} = M_A + M_B \\ \lambda_{A\times B} = [\lambda_A, \lambda_B] & \lambda_{A\Rightarrow B} = [\overline{\lambda}_A, \lambda_B] \\ \vdash_{A\times B} = \vdash_A \, + \vdash_B & \vdash_{A\Rightarrow B} = \vdash_B \, + (I_B \times I_A) \, + \, (\vdash_A \cap (M_A \times M_A)) \end{array}$$

$\overline{\lambda}_A$ reverses the ownership of moves in A while preserving their kind. Here are the arenas used interpret the base types of IA (the moves at the bottom are answer-moves).



Given an IA-type T, we shall write $[\![T]\!]$ for the corresponding arena obtained compositionally from A_{com}, A_{exp} and A_{var} using the $\Rightarrow$ construction.

A *justified sequence* s in an arena A is a sequence of moves in which every move $m \notin I_A$ must have a pointer to an earlier move n in s such that $n \vdash_A m$. n is then said to be the *justifier* of m. It follows that every justified sequence must begin with an O-question. The view $\ulcorner s \urcorner$ of a justified sequence s is defined by

$$\begin{array}{rl} \ulcorner \epsilon \urcorner = \epsilon & \\ \ulcorner sm \urcorner = m & \text{if } m \text{ is initial in } A \\ \ulcorner s_1 \, m \, \overbrace{s_2} \, n \urcorner = \ulcorner s \urcorner \, m \, n & \end{array}$$

A justified sequence s satisfies the *visibility condition* iff in any prefix $s'm$ of s such that m is not initial, the justifier of m lies in $\ulcorner s' \urcorner$. A justified sequence satisfies the *bracketing* condition if any answer-move is justified by the latest unanswered question that precedes it.

Definition 3. *A justified sequence is a* play *iff O- and P-moves alternate and the visibility condition is satisfied. We write P_A for the set of plays in A.*

Note that plays do not satisfy the bracketing condition. This notion of play suffices to define a game model of $\mathsf{IA}_{\mathbf{catch+mkvar}}$, in which terms are interpreted as strategies.

Definition 4. *A* strategy *in an arena A, written $\sigma : A$, is a non-empty set of even-length plays in A which is closed under taking even prefixes and satisfies the* determinacy *condition: $smn_1, smn_2 \in \sigma$ entails $n_1 = n_2$ (targets of pointers from n_1 and n_2 are also required to be the same).*

For any arena A, the strategy on $A \Rightarrow A$ that copies moves between the two instances of A is called the *identity* strategy. Arenas and strategies form a category in which a morphism between A and B is a strategy on $A \Rightarrow B$. In order to compose two strategies $\sigma : A \Rightarrow B$, $\tau : B \Rightarrow C$, one first defines *interaction sequences* on A, B, C, which are sequences of moves from arenas A, B and C together with justification pointers from all moves except those initial in C. The set of all such sequences will be denoted by $\mathsf{int}(A, B, C)$. Given an interaction sequence u, we write $u \upharpoonright A, B$ for its subsequence consisting of all A- and B-moves as well as pointers between them (pointers from/to moves of C in u are erased, though). $u \upharpoonright B, C$ is defined analogously. $u \upharpoonright A, C$ is defined similarly except that, whenever a pointer from an A-move m_A points at a B-move m_B which in turn has a pointer to a C-move m_C, we add a pointer from m_A to m_C. Then one takes $\sigma; \tau : A \Rightarrow C$ to be

$$\{u \upharpoonright A, C \mid u \in \mathsf{int}(A, B, C),\ u \upharpoonright A, B \in \sigma,\ u \upharpoonright B, C \in \tau\}.$$

Arenas and strategies form a category in which identity strategies are indeed the identity maps.

A play is called *single-threaded* if it contains just one occurrence of an initial move. In general, a play may consist of several interleaved single-threaded plays. Strategies determined completely by their single-threaded plays will be called *single-threaded*: they consist of all plays that are interleavings of the single-threaded plays belonging to the strategy.

Arenas and single-threaded strategies turn out to form a cartesian closed category, which provides a canonical interpretation of λ-abstraction and application. The inclusion relation on strategies enriches the category with the structure of a complete partial order needed to interpret recursion. Other features of $\mathsf{IA}_{\mathbf{catch+mkvar}}$ can be interpreted by composition with special designated strategies, which we list in Figure 2 along with their single-threaded complete plays. For illustration, we give the two maximal single-threaded plays of the strategy $\mathsf{catch} : [\![(\mathsf{com}_2 \to \mathsf{com}_1) \to \mathsf{com}_0]\!]$ used to interpret **catch**:

$$run_0\ run_1\ done_1\ done_0 \qquad\qquad run_0\ run_1\ run_2\ done_0\,.$$

We have used subscripts to indicate the copies of com from which the moves originate. Then $[\![\Gamma \vdash \mathbf{catch}\, x\, \mathbf{in}\, M]\!] = [\![\Gamma \vdash \lambda x^{\mathsf{com}}.M]\!]; \mathsf{catch}$.

Theorem 1. *Arenas and single-threaded strategies ordered by inclusion are an inequationally fully abstract model of* $\mathsf{IA}_{\mathbf{catch+mkvar}}$*: for any* $\mathsf{IA}_{\mathbf{catch+mkvar}}$*-terms* $\Gamma \vdash M_1, M_2 : T$*:*

$$\Gamma \vdash M_1 \sqsubseteq\!\!\!\approx_{\mathsf{IA}_{\mathbf{catch+mkvar}}} M_2 \iff [\![\Gamma \vdash M_1]\!] \subseteq [\![\Gamma \vdash M_2]\!].$$

To our knowledge, this theorem has not appeared in the literature so far, though it seems to be known within the community. It can be proved in a similar way to the characterization of program approximation via complete plays for $\mathsf{IA}_{\mathbf{mkvar}}$ [4]. The argument relies on the fact that any finite strategy is definable by a term of $\mathsf{IA}_{\mathbf{catch+mkvar}}$, which can be shown using the techniques of [12].

$\mathsf{skip} : [\![\mathsf{com}]\!]$ $\quad run\ done$

$\mathsf{i} : [\![\mathsf{exp}]\!]$ $\quad q\, i$

$\mathsf{succ} : [\![\mathsf{exp}]\!]_1 \Rightarrow [\![\mathsf{exp}]\!]_0$ $\quad q_0\, q_1\, \sum_{i\in\mathbb{N}} i_1\, (i+1)_0$

$\mathsf{pred} : [\![\mathsf{exp}]\!]_1 \Rightarrow [\![\mathsf{exp}]\!]_0$ $\quad q_0\, q_1\, \sum_{i\in\mathbb{N}^+} i_1\, (i-1)_0$

$\mathsf{if}_B : [\![\mathsf{exp}]\!]_3 \Rightarrow [\![B]\!]_2 \Rightarrow [\![B]\!]_1 \Rightarrow [\![B]\!]_0$

$$\sum_{m\vdash [\![B]\!]^n} (m_0\, q_3\, 0_3\, m_1\, n_1\, n_0 + m_0\, q_3\, (\sum_{i\in\mathbb{N}^+} i_3)\, m_2\, n_2\, n_0)$$

$\mathsf{seq}_B : [\![\mathsf{com}]\!]_2 \Rightarrow [\![B]\!]_1 \Rightarrow [\![B]\!]_0$ $\quad \sum_{m\vdash [\![B]\!]^n} m_0\, run_2\, done_2\, m_1\, n_1\, n_0$

$\mathsf{deref} : [\![\mathsf{var}]\!]_1 \Rightarrow [\![\mathsf{exp}]\!]_0$ $\quad q_0\, read_1\, \sum_{i\in\mathbb{N}} i_1\, i_0$

$\mathsf{assign} : [\![\mathsf{var}]\!]_2 \Rightarrow [\![\mathsf{exp}]\!]_1 \Rightarrow [\![\mathsf{com}]\!]_0$ $\quad run_0\, q_1\, \sum_{i\in\mathbb{N}} i_1\, write(i)_2\, ok_2\, done_0$

$\mathsf{cell}_B : ([\![\mathsf{var}]\!]_2 \Rightarrow [\![B]\!]_1) \Rightarrow [\![B]\!]_0$

$$\sum_{m\vdash [\![B]\!]^n} m_0 m_1 (read_2\, 0_2\,)^* (\sum_{i\in\mathbb{N}} write(i)_2\, ok_2 (read_2\, i_2)^*)^* n_1 n_0$$

$\mathsf{mkvar} : [\![\mathsf{exp} \to \mathsf{com}]\!]_2 \Rightarrow [\![\mathsf{exp}]\!]_1 \Rightarrow [\![\mathsf{var}]\!]_0$

$$read_0\, q_1\, (\sum_{i\in\mathbb{N}} i_1\, i_0) + \sum_{i\in\mathbb{N}} write(i)_0\, run_2\, (q_2\, i_2)^*\, done_2\, ok_0$$

Fig. 2. Special strategies.

Soundness and Adequacy ($[\![\vdash M : \mathsf{com}]\!] = [\![\vdash \mathbf{skip}]\!]$ if and only if $M \Downarrow$) can also be proved in the standard way. The goal of this paper is to show an analogous theorem for $\mathsf{IA}_{\mathbf{catch}}$, with the inclusion ordering replaced by a different preorder.

4 The essence of mkvar

Game semantics interprets local variable allocation in $\Gamma \vdash \mathbf{new}\, x\, \mathbf{in}\, M : \mathsf{com}$ through composition of $[\![\Gamma \vdash \lambda x^{\mathsf{var}}.M]\!]$ with the strategy $\mathsf{cell}_{\mathsf{com}} : [\![(\mathsf{var}_2 \to \mathsf{com}_1) \to \mathsf{com}_0]\!]$. $\mathsf{cell}_{\mathsf{com}}$ itself denotes the term

$$\vdash \lambda f^{\mathsf{var}\to\mathsf{com}}.\mathbf{new}\, x\, \mathbf{in}\, fx : (\mathsf{var} \to \mathsf{com}) \to \mathsf{com}.$$

Single-threaded plays of $\mathsf{cell}_{\mathsf{com}}$ are prefixes of plays of the following shape:

$$run_0\, run_1\, (read_2\, 0_2)^* (\sum_{i\in\mathbb{N}} write(i)_2\, ok_2\, (read_2\, i_2)^*)^*\, done_1\, done_0$$

i.e. *read*'s trigger responses consistent with preceding *write*'s. Using **mkvar** we can easily violate the discipline, because unrelated methods can be employed for reading and writing. However, the same effect can also be achieved without it. Indeed, under call-by-name evaluation, whenever a *read* follows a *write*, we cannot really be sure that they refer to the same variable. Here is a term illustrating this behaviour

$$\vdash \lambda f^{\mathsf{var}\to\mathsf{com}}.\mathbf{new}\, X, Y\, \mathbf{in}\, f(\mathbf{if}\, !X\, \mathbf{then}\, Y\, \mathbf{else}\, X),$$

which will produce the play $run_0\, run_1\, write(1)_2\, ok_2\, read_2\, 0_2\, done_1\, done_0$. Thus the essence of **mkvar** does not boil down to breaking the logical link between *read*'s and *write*'s. We argue that it lies in uniformity instead: in absence of **mkvar** each variable operation, whether a read or a write, produces the same side-effects while it is being completed. To formalize this intuition it is useful to observe that, without **mkvar**, subterms of type var (in β-normal form) always have "tails" of the shape $fM_1 \cdots M_k : \mathsf{var}$, which may be combined using conditionals, pre-composed with commands and bound with **new** (if $k = 0$). In game semantics, when O plays a move q_O in order to explore such a subterm, the resultant plays will initially correspond to the associated side-effects (if any). These side-effects will be independent of q_O, which could well be *read*, $write(0)$ or $write(13)$. Eventually, when the "tail" is reached and f is not bound by **new**, P will play a copy q_P of q_O. Below we introduce new relations on plays and strategies to express an aspect of the uniformity that will turn out useful in subsequent technical arguments.

Definition 5. *Given an arena A corresponding to an IA-type and $q, q' \in \{read\} \cup \{write(i) \mid i \in \mathbb{N}\}$, the relation $\diamond_{\mathsf{O}}^{q,q'} \subseteq P_A \times P_A$ is defined as follows: $t \diamond_{\mathsf{O}}^{q,q'} t'$ iff $t = s_1 q s_2$, $t' = s_1 q' s_2$ and q, q' are O-moves from the same copy of A_{var} that have not been answered in t, t' respectively. $\diamond_{\mathsf{P}}^{q,q'}$ is defined in an analogous way, by replacing "O-moves" with "P-moves". We write $\diamond_{\mathsf{O}}$ for the (symmetric) relation $\bigcup_{q,q'} \diamond_{\mathsf{O}}^{q,q'}$. $\diamond_{\mathsf{P}}$ is defined similarly.*

Definition 6. *A strategy $\sigma : A$ is $\diamond$-closed iff, for any $s \in \sigma, t \in P_A$, if there exist q, q' such that $s \diamond_{\mathsf{O}}^{q,q'} t$, then $t \in \sigma$ or there exists $s' \in \sigma$ such that $t \diamond_{\mathsf{P}}^{q,q'} s'$.*

Next we turn to questions that have been answered. If the q_P from the above scenario is eventually answered, say, with a_O, P will immediately answer q_O with a copy a_P of a_O. Afterwards, the play will actually follow independently of the value of q and a. This is in contrast to the case of $k = 0$ and f being bound by **new**. Here after a series of possible side-effects (independent of q_O) P will answer q_O with a_P. Because this case corresponds to examining a genuine storage cell, what happens next could depend on the current value of the variable: if $s_1 write(0) s_1 ok s_2 \in \sigma$, it does not have to be the case that $s_1 write(2) s_2 ok s_2 \in \sigma$. Similarly, if $s_1 read s_2 0 s_3 \in \sigma$, we do not know whether $s_1 write(2) s_2 ok s_2 \in \sigma$. However, if $s_1 read s_2 0 s_3 \in \sigma$, we can be sure that $s_1 write(0) s_2 ok s_3 \in \sigma$, because overwriting a variable with its current value does not change the state. Below we define another closure property of strategies, which unifies the observations just made about answer-moves. This is essentially a more precise variant of the $\propto$-closure used in [7], adapted to general plays.

Definition 7. *Given an arena A corresponding to an IA-type we define $\lhd_{\mathsf{O}} \subseteq P_A \times P_A$ as follows: $t \lhd_{\mathsf{O}} t'$ iff $t = s_1\ read\ \widehat{s_2}\ i\ s_3$ and $t' = s_1\ write(i)\ \widehat{s_2}\ ok\ s_3$ for some $i \in \mathbb{N}$, where read and $write(i)$ are O-moves from the same copy of A_{var}. $\lhd_{\mathsf{P}}$ is defined in an analogous way.*

Definition 8. *A strategy $\sigma : A$ is $\lhd$-closed iff, for any $s \in \sigma, t \in P_A$, if $s \lhd_{\mathsf{O}} t$ then $t \in \sigma$ or there exists $s' \in \sigma$ satisfying $t \lhd_{\mathsf{P}} s'$.*

Lemma 1. *Strategies denoting* $\mathsf{IA}_{\mathbf{catch}}$*-terms are* $\diamond$*- and* $\triangleleft$*-closed.*

Proof. First one shows that $\diamond$- and $\triangleleft$-closure are preserved by composition. Then it suffices to show that the basic special strategies used to construct the model satisfy the Lemma.

Note that the strategy corresponding to **mkvar** satisfies neither $\diamond$- nor $\triangleleft$-closure.

5 What difference does mkvar make?

Because the addition of new syntactic features makes the discriminating power of contexts stronger, it is natural to expect that some approximations in $\mathsf{IA}_{\mathbf{catch}}$ will no longer hold in $\mathsf{IA}_{\mathbf{catch+mkvar}}$. For IA and $\mathsf{IA}_{\mathbf{mkvar}}$, it was shown in [7] that essentially all examples of IA approximations that fail in $\mathsf{IA}_{\mathbf{mkvar}}$ are based on approximating reads with matching writes, as in

$$x : \mathsf{var} \vdash \mathbf{if}\, !x\, \mathbf{then}\, \Omega\, \mathbf{else}\, \mathbf{skip} \sqsubseteq_{\mathsf{IA}} x{:=}0.$$

The same idea can also be used to demonstrate the difference between $\mathsf{IA}_{\mathbf{catch}}$ and $\mathsf{IA}_{\mathbf{catch+mkvar}}$, but we will also have another class of examples, relying on variable operations immediately followed by divergence:

$$x : \mathsf{var} \vdash \mathbf{if}\, !x\, \mathbf{then}\, \Omega\, \mathbf{else}\, \Omega \quad \cong_{\mathsf{IA}_{\mathbf{catch}}} \quad x{:=}0; \Omega \quad \cong_{\mathsf{IA}_{\mathbf{catch}}} \quad x{:=}1; \Omega.$$

Because the terms generate different plays, these equivalences do not hold in $\mathsf{IA}_{\mathbf{catch+mkvar}}$ so, in contrast to $\mathsf{IA}_{\mathbf{mkvar}}$, $\mathsf{IA}_{\mathbf{catch+mkvar}}$ turns out not to extend $\mathsf{IA}_{\mathbf{catch}}$ conservatively even for observational equivalence. In the remainder of the paper we show how to capture approximation in $\mathsf{IA}_{\mathbf{catch}}$ with a preorder based on $\diamond_{\mathsf{P}}$ and $\triangleleft_{\mathsf{P}}$, which will make it clear that equivalences above do hold.

Definition 9. *Suppose* $\sigma, \tau : A$ *are single-threaded. We define* $\sigma \sqsubseteq \tau$ *to hold iff for any* $s \in \sigma$ *there exists* $t \in \tau$ *such that* $s\,(\diamond_{\mathsf{P}} \cup \triangleleft_{\mathsf{P}})^*\, t$.

Because σ and τ are single-threaded, the quantification over $s \in \sigma$ could well range over single-threaded plays only. In the next section we aim to prove:

Theorem 2 (Full abstraction). *For any* $\mathsf{IA}_{\mathbf{catch}}$*-terms* $\Gamma \vdash M_1, M_2 : T$ *we have* $\Gamma \vdash M_1 \sqsubseteq_{\mathsf{IA}_{\mathbf{catch}}} M_2$ *if and only if* $[\![\Gamma \vdash M_1]\!] \sqsubseteq [\![\Gamma \vdash M_2]\!]$.

6 Proof of full abstraction

The left-to-right direction hinges on the fact that $[\![\Gamma \vdash M_1]\!] \sqsubseteq [\![\Gamma \vdash M_2]\!]$ implies $[\![\vdash C[M_1]]\!] \sqsubseteq [\![\vdash C[M_2]]\!]$. Indeed, more generally, one can show that composition of $\diamond$- and $\triangleleft$-closed strategies is monotone with respect to $\sqsubseteq$, which implies the above.

Lemma 2. *For any* $\triangleleft$*- and* $\diamond$*-closed strategies* $\sigma_1, \sigma_2 : A \Rightarrow B$ *and* $\tau_1, \tau_2 : B \Rightarrow C$*: if* $\sigma_1 \sqsubseteq \sigma_2$ *and* $\tau_1 \sqsubseteq \tau_2$ *then* $\sigma_1; \tau_1 \sqsubseteq \sigma_2; \tau_2$.

Proof. By repeated alternate applications of closure rules for σ_i and τ_i.

Lemma 3. *For any* $\mathsf{IA}_{\mathbf{catch}}$*-terms* $\Gamma \vdash M_1, M_2 : T$ *if* $[\![\Gamma \vdash M_1]\!] \sqsubseteq [\![\Gamma \vdash M_2]\!]$ *then* $\Gamma \vdash M_1 \sqsubseteq\!\!\!\!\sim_{\mathsf{IA}_{\mathbf{catch}}} M_2$.

Proof. Suppose $\vdash \mathcal{C}[M_1] : \mathsf{com} \Downarrow$. Then, by Soundness (of the game model of $\mathsf{IA}_{\mathbf{catch+mkvar}}$), $[\![\vdash \mathcal{C}[M_1] : \mathsf{com}]\!] = [\![\vdash \mathbf{skip}]\!]$. Because $[\![\Gamma \vdash M_1]\!] \sqsubseteq [\![\Gamma \vdash M_2]\!]$, we have $[\![\vdash \mathbf{skip}]\!] = [\![\vdash \mathcal{C}[M_1]]\!] \sqsubseteq [\![\vdash \mathcal{C}[M_2]]\!]$. Hence, $[\![\vdash C[M_2]]\!] = [\![\vdash \mathbf{skip}]\!]$ and, by Adequacy, $\mathcal{C}[M_2] \Downarrow$.

To establish the converse we need a new definability argument. Because the strategies involved are $\diamond$- and $\triangleleft$-closed, it is no longer impossible to prove definability for single positions. We shall restrict ourselves to plays of the shape $run \cdots done$, as these suffice for the reconstruction of contexts used in the definition of $\sqsubseteq\!\!\!\!\sim_{\mathsf{IA}_{\mathbf{catch}}}$.

Proposition 1. *Let* $s = run \cdots done$ *be a single-threaded play in* $[\![T]\!]$*. Then there exists an* $\mathsf{IA}_{\mathbf{catch}}$*-term* $\vdash M_s : T$ *such that the set of single-threaded complete plays of* $[\![\vdash M_s]\!]$ *equals* $\{t \mid s\ (\diamond_{\mathsf{O}} \cup \triangleleft_{\mathsf{O}})^*\ t\}$.

We will say that an answer-move occurring in a play is *well-bracketed*, if the question that justifies it is the most recent unanswered question in the view calculated right before the answer has been played. A strategy is called well-bracketed if in each of its plays any P-answer is well-bracketed. Thanks to the factorization techniques developed in [9, 12], which factor out non-well-bracketed P-answers using catch, it suffices to prove the above Proposition for plays s in which P-answers are *well-bracketed*. To that end we first identify a family of *innocent* strategies which are definable in IA without **new**. Innocence, first defined in [13], guarantees that P's responses depend only on the current view of the play rather than the whole play.

6.1 Innocent strategies without mkvar

Let $A_{\square} = \langle \{\square_q, \square_a\}, \{(\square_q, (Q, O)), (\square_a, (A, P))\}, \{(\star, \square_q), (\square_q, \square_a)\} \rangle$. Given an IA type T, let $[\![T]\!]_{\mathsf{sym}}$ be the arena obtained compositionally from T using the $\times$ and $\Rightarrow$ constructions in the same way as $[\![T]\!]$ except that occurrences of var are interpreted differently: positive ones by $A_{\square}$ and negative ones by $A_{\mathsf{var}} \times A_{\square}$. Moves of $A_{\square}$ will be called *generic*, while those from A_{var} will be referred to as *concrete*. Thus, only P will have concrete moves at his disposal in $[\![T]\!]_{\mathsf{sym}}$.

Given plays $s_1 \in P_{[\![T]\!]_{\mathsf{sym}}}$ and $s_2 \in P_{[\![T]\!]}$, we shall say that s_2 *matches* s_1 iff s_2 can be obtained from s_1 by replacing each occurrence of $\square_q$ (respectively $\square_a$) with a concrete question (respectively answer) coming from the same copy of var as the generic move it replaces. Thus, plays in $[\![T]\!]_{\mathsf{sym}}$ can be viewed as specifications of sets of plays in $[\![T]\!]$. Suppose $\sigma : [\![T]\!]_{\mathsf{sym}}$. Because σ is deterministic and $[\![T]\!]_{\mathsf{sym}}$ does not allow concrete O-moves, a play from $[\![T]\!]$ can match at most one play from σ. This matching can be used to define strategies $\widehat{\sigma} : [\![T]\!]$ that "match" σ but, in general, such extensions will not be unique. In what follows, we introduce a special class of strategies on $[\![T]\!]_{\mathsf{sym}}$ that can be extended to strategies on $[\![T]\!]$ in a canonical way.

Definition 10. *A well-bracketed strategy $\sigma : [\![T]\!]_{\mathsf{sym}}$ is a tail strategy iff it satisfies the following conditions.*

(i) If $s\square_a \in \sigma$ then the last move of s is a $\square_a$-move.
(ii) If $s\square_q \in \sigma$ then for any $s\,\square_q\,\widehat{s_1}\,\square_a \in P_{[\![T]\!]_{\mathsf{sym}}}$ such that $s\square_q s_1 \in \sigma$, we have

$$s\overbrace{\square_q\,\widehat{s_1}\,\square_a}\,\square_a \in \sigma.$$

Remark 1. Because tail strategies are well-bracketed, the target of the last justification in clause (ii) is uniquely determined by $s\square_q$. We shall call it the *mate* of $\square_q$. Of course, the mate of $\square_q$ must also be a $\square_q$-move. Similarly, we define the mate of a P-answer $\square_a$ in $s\square_a \in \sigma$ to be the last move of s, which by clause (i) must be an O-answer of the shape $\square_a$.

Definition 11. *Let $\sigma : [\![T]\!]_{\mathsf{sym}}$ be a tail strategy. We define the* copy-cat extension *$\widehat{\sigma} : [\![T]\!]$ of σ in the following way.*

- $\epsilon \in \widehat{\sigma}$.
- *If $s \in \widehat{\sigma}$, $sm_1 \in P_{[\![\sigma]\!]}$, $tn_1n_2 \in \sigma$ are such that tn_1 matches sm_1 then:*
  - *if n_2 is not generic then $sm_1n_1 \in \widehat{\sigma}$;*
  - *if n_2 is generic, then $s' = sm_1m_2 \in \widehat{\sigma}$, where s' is the unique play matching tn_1n_2 and such that n_2 is instantiated with the same concrete move as its mate.*

Note that when σ is innocent, so is $\widehat{\sigma}$. An innocent strategy is compactly innocent if it depends only on a finite number of views.

Lemma 4. *Suppose $\sigma : [\![T]\!]_{\mathsf{sym}}$ is a compactly innocent tail strategy. Then there exists a* **new***-free term $\vdash M : T$ of IA such that $[\![\vdash M : T]\!] = \widehat{\sigma}$.*

Proof. Follows the standard definability argument for PCF [13].

6.2 Knowingness without mkvar

Now we continue with definability for certain knowing, i.e. not necessarily innocent, strategies.

Lemma 5. *Suppose $T = \mathsf{var}_k \to \cdots \to \mathsf{var}_1 \to T'$ and let s be a single-threaded play in $[\![T]\!]_{\mathsf{sym}}$ such that any generic P-question $\square_q$ in s comes from var_i ($i = 1, \cdots, k$) and no two such questions come from the same var_i. Let σ be the least single-threaded strategy on $[\![T]\!]_{\mathsf{sym}}$ containing s. If σ is a tail strategy, then there exists a compactly-innocent tail strategy $\tau : [\![\mathsf{var} \to T]\!]_{\mathsf{sym}}$ such that $\tau; \mathsf{cell}_T^{\mathsf{sym}} = \sigma$[2]. Consequently, $\widehat{\tau}; \mathsf{cell}_T = \widehat{\sigma}$.*

[2] $\mathsf{cell}_T : [\![(\mathsf{var} \to T) \to T]\!]$ works in the same way as $\mathsf{cell}_{\mathsf{com}}$: moves are being copied from one copy of $[\![T]\!]$ to another and, when O makes a move in the var component, P's responses reflect the behaviour of a storage cell. $\mathsf{cell}_T^{\mathsf{sym}}$ works analogously except that it is a strategy in the game $(A_{\mathsf{var}} \Rightarrow [\![T]\!]_{\mathsf{sym}}) \Rightarrow [\![T]\!]_{\mathsf{sym}}$.

Proof. We modify the factorization argument from [4]. Its key idea is that τ follows σ except that it uses the additional var component for recording the single-threaded history of play. Thus, when an O-move from $[\![T]\!]_{\mathsf{sym}}$ is made following some play $t \in \tau$, τ will always play *read* to find out what the current single-threaded play looks like. The subsequent O-move is then regarded as the code of the play. τ is then able to mimic σ in an innocent way, because the code of the whole single-threaded play will be present in the relevant view. However, before τ imitates σ, it always writes the code of the resultant single-threaded play to the var component. Plays of τ thus have the shape $\cdots\, m_T^O\ \mathit{read}\ s\ \mathit{write}(\mathit{sab})\ \mathit{ok}\ m_T^P$, i.e. the procedure introduces additional moves between any O-move and the P-move that follows.

Note that, in our case, we cannot afford to adopt the above-described procedure after O-moves of the form $\square_a$, because we want τ to be a tail strategy. However, because σ is a tail strategy, we know that, after an O-move $\square_a$, P will also respond with $\square_a$. Moreover, because generic P-questions can only come from some var_i, all O-answers $\square_a$ must also come from there. Consequently, thanks to the special shape of the arena, the predecessor of any O-answer $\square_a$ in s must be the P-question $\square_q$ that enables it. This opens up the way to modifying the previous factorization: before P plays $\square_q$ in τ, τ can already write the code of $s\square_q\square_a\square_a$ to the var component, because $\square_a\square_a$ will follow anyway. When $\square_a$ is indeed played by O afterwards, τ will not read or write from var component, but will immediately reply with the same $\square_a$ as σ would. Note that this behaviour is innocent, because no two generic O-answers come from the same var_i.

Proposition 2. *Suppose $s = \mathit{run} \cdots \mathit{done}$ is a single-threaded play of $[\![T]\!]$ in which P-moves are well-bracketed. Then there exists an IA-term $\vdash M_s : T$ such that the set of single-threaded complete plays of $[\![\vdash M_s]\!]$ is $\{t \mid s\ (\triangleleft_{\mathsf{O}} \cup \diamond_{\mathsf{O}})^*\ t\}$.*

Proof. Note that the shape of the play means that $T = T_l \to \cdots \to T_1 \to \mathsf{com}$. Let k be the number of concrete P-answers in s. We will first replace s with $s' \in P_{[\![T']\!]_{\mathsf{sym}}}$ such that $T' = \mathsf{var}_k \to \cdots \to \mathsf{var}_1 \to T$ and s' satisfies the assumptions of Lemma 5. s' is obtained from s in the following way.

- Any concrete O-question is replaced by $\square_q$ from the same copy var as the question.
- Any concrete P-answer is replaced with $\square_q^j, \square_a^j\,\square_a$, $\square_q^j, \square_a^j$ come from var_j, $\square_a$ comes from the same copy of var as the answer and the answer in question is the jth concrete P-answer in s.

By Lemma 5 $\widehat{\sigma} = \widehat{\tau}; \mathsf{cell}_{T'}$, where τ is a compactly-innocent tail strategy on $[\![\mathsf{var} \to T']\!]_{\mathsf{sym}}$. By Lemma 4 there exists a IA-term $\vdash M : \mathsf{var} \to T'$ such that $[\![\vdash M]\!] = \widehat{\tau}$. Thus, putting $M' \equiv \lambda x_k \cdots x_1 y_l \cdots y_1.\mathbf{new}\, X\, \mathbf{in}\, M X x_k \cdots x_1 y_l \cdots y_1$, we get $[\![\vdash M' : T']\!] = \widehat{\sigma}$. To obtain $\vdash M'' : T$ satisfying the current Proposition it now suffices to take

$$\lambda y_l \cdots y_1.\mathbf{new}\, x_1, \cdots, x_k\, \mathbf{in}\, \mathit{INIT}; M' x_k \cdots x_1 y_l \cdots y_1; \mathit{FINIT},$$

where $INIT \equiv INIT_1; \cdots; INIT_k, FINIT \equiv FINIT_1; \cdots; FINIT_k$,

$$INIT_j \equiv \begin{cases} x_j{:=}i+1 & j\text{th concrete } P\text{-answer in } s \text{ is } ok \text{ justified by } write(i) \\ x_j{:=}i & j\text{th concrete } P\text{-answer in } s \text{ is } i \text{ (justified by } read) \end{cases}$$

and $FINIT_j \equiv \mathbf{if}\,(!x_j = i)\,\mathbf{then\,skip\,else}\,\Omega$, where the jth concrete P-answer is i (justified by *read*) or or *ok* justified by $write(i)$.

By previous remarks Proposition 2 implies Proposition 1.

Lemma 6. *For any* $\mathsf{IA}_{\mathbf{catch}}$*-terms* $\Gamma \vdash M_1, M_2 : T$*, if* $\Gamma \vdash M_1 \sqsubseteq\!\!\!\sim_{\mathsf{IA}_{\mathbf{catch}}} M_2$ *then* $[\![\Gamma \vdash M_1]\!] \sqsubseteq [\![\Gamma \vdash M_2]\!]$.

Proof. W.l.o.g. assume that Γ is empty (other cases can be reduced to this case by λ-abstraction). Suppose $s \in [\![\vdash M_1 : T]\!]$. Let $s' = run\, s\, done$. By Proposition 1 there exists a term $\vdash M_{s'} : T \to \mathsf{com}$ whose set of single-threaded and complete plays is $\{t \mid s'\,(\diamond_{\mathsf{O}} \cup \triangleleft_{\mathsf{O}})^*\,t\}$. Let $\mathcal{C}[-] = M_{s'}([-])$. Then $[\![\vdash \mathcal{C}[M_1]]\!] = [\![\vdash \mathbf{skip}]\!]$, so $\mathcal{C}[M_1] \Downarrow$. Because $\Gamma \vdash M_1 \sqsubseteq\!\!\!\sim_{\mathsf{IA}_{\mathbf{catch}}} M_2$, we also have $\mathcal{C}[M_2] \Downarrow$. Hence, $[\![\vdash \mathcal{C}[M_2]]\!] = [\![\vdash \mathbf{skip}]\!]$. But this implies, by definition of composition of strategies, that there must exists $t \in [\![\vdash M_2]\!]$ such that $s(\diamond_{\mathsf{P}} \cup \triangleleft_{\mathsf{P}})^* t$.

Putting together Lemmas 3 and 6 we obtain Theorem 2.

7 On conservativity

Harmer and McCusker have investigated the game semantics of nondeterminism and showed how $\mathsf{IA}_{\mathbf{mkvar+or}}$ can be modelled by nondeterministic strategies [14]. Analogously to $\mathsf{IA}_{\mathbf{catch+mkvar}}$ and $\mathsf{IA}_{\mathbf{mkvar}}$, observational approximation (based on may-convergence) in $\mathsf{IA}_{\mathbf{catch+mkvar+or}}$ and $\mathsf{IA}_{\mathbf{mkvar+or}}$ can be shown to correspond to containment of induced plays and complete plays respectively. So, in these two cases, extensions by **or** are conservative both with respective to observational approximation and equivalence. The same turns out to apply to $\mathsf{IA}_{\mathbf{catch}}$ and IA. Indeed, our argument for $\mathsf{IA}_{\mathbf{catch}}$ is immediately applicable to $\mathsf{IA}_{\mathbf{catch+or}}$: nondeterministic strategies ordered by $\sqsubseteq$ form a fully abstract model of $\mathsf{IA}_{\mathbf{catch+or}}$. Similarly, McCusker's preorder for IA [7] also gives full abstraction for $\mathsf{IA}_{\mathbf{or}}$. Consequently, $\mathsf{IA}_{\mathbf{or}}$ and $\mathsf{IA}_{\mathbf{catch+or}}$ are conservative extensions of IA and $\mathsf{IA}_{\mathbf{catch}}$ respectively.

In contrast, extensions of IA, $\mathsf{IA}_{\mathbf{or}}$, $\mathsf{IA}_{\mathbf{or+mkvar}}$ by **catch** are not conservative, even for observational equivalence. In game semantics this manifests itself in the reliance of full abstraction results for **catch**-free languages on complete plays only.

The inclusion of **mkvar** also turns out to affect observational approximation in any of IA, $\mathsf{IA}_{\mathbf{catch}}$, $\mathsf{IA}_{\mathbf{or}}$, $\mathsf{IA}_{\mathbf{catch+or}}$. For IA the effect of **mkvar** was captured by McCusker [7], $\mathsf{IA}_{\mathbf{catch}}$ was examined in this paper and, as we already mentioned, the two results apply to $\mathsf{IA}_{\mathbf{or}}$ and $\mathsf{IA}_{\mathbf{catch+or}}$. As for observational equivalence, **mkvar** is "conservative" for IA, as shown in [7], but not $\mathsf{IA}_{\mathbf{catch}}$,

as demonstrated in this paper. It is interesting to note that, for program equivalence, $\mathsf{IA}_{\mathbf{or+mkvar}}$ does *not* extend $\mathsf{IA}_{\mathbf{or}}$ conservatively either. Let M_1, M_2 be the terms from Section 5 such that $M_1 \sqsubseteq\!\!\!\approx_{\mathsf{IA}} M_2$ and $\neg(M_1 \sqsubseteq\!\!\!\approx_{\mathsf{IA}_{\mathbf{mkvar}}} M_2)$. Then $M_1 \,\mathbf{or}\, M_2 \cong_{\mathsf{IA}_{\mathbf{or}}} M_2$, but the terms are not equivalent in $\mathsf{IA}_{\mathbf{or+mkvar}}$.

Finally, let us consider probabilistic game semantics. Probabilistic strategies were introduced by Danos and Harmer [15] as functions $\sigma : P_A^{\mathrm{ev}} \to [0,1]$ where P_A^{ev} stands for the set of even-length plays on A. For probabilistic programs, instead of presence or lack of termination, one talks about the probability of termination, denoted by $\Downarrow_p$. Observational approximation can then be defined as follows.

Definition 12. *Suppose $\Gamma \vdash M_1, M_2 : T$ are terms of $\mathcal{L} = \mathsf{IA}_{\mathbf{coin}}, \mathsf{IA}_{\mathbf{coin+mkvar}}$. $\Gamma \vdash M_1 : T$ approximates $\Gamma \vdash M_2 : T$ iff, for all $\mathcal{L}$-contexts $C[-]$ such that $\vdash C[M_1], C[M_2] : \mathsf{com}$ holds there exist p, q such that $p \leq q$, $\mathcal{C}[M] \Downarrow_p$ and $\mathcal{C}[N] \Downarrow_q$.*

For $\mathsf{IA}_{\mathbf{mkvar}}$ the above notion can be characterized via complete plays [16]: $\Gamma \vdash M_1 \sqsubseteq\!\!\!\approx_{\mathsf{IA}_{\mathbf{coin+mkvar}}} M_2$ iff for all single-threaded complete plays s we have $[\![\Gamma \vdash M_1]\!](s) \leq [\![\Gamma \vdash M_2]\!](s)$. The left-to-right implication depends on the fact that for any single-threaded complete play s one can construct an $\mathsf{IA}_{\mathbf{mkvar}}$-term $\Gamma \vdash M$ such that s is the only single-threaded complete play in $[\![\Gamma \vdash M]\!]$. Using McCusker's definability result for IA [7], which says that there exists an IA-term generating exactly the single-threaded complete plays from $\{t \mid s \lhd_{\mathsf{O}}^* t\}$[3], one can show the following result.

Lemma 7. *For any $\mathsf{IA}_{\mathbf{coin}}$-terms $\Gamma \vdash M_1, M_2 : T$, if $\Gamma \vdash M_1 \sqsubseteq\!\!\!\approx_{\mathsf{IA}_{\mathbf{coin}}} M_2$ then for any single-threaded complete play s we have*

$$\sum_{\{t \mid s \lhd_{\mathsf{P}}^* t\}} [\![\Gamma \vdash M_1]\!](t) \leq \sum_{\{t \mid s \lhd_{\mathsf{P}}^* t\}} [\![\Gamma \vdash M_2]\!](t).$$

The converse of the Lemma is not true. The conceptual reason for the failure is that a single complete play can be extended to a $\diamond$- and $\lhd$-closed strategy in a number of ways, while the definability result in [7] (and in this paper) only explores the simplest one. This approach is fruitful for languages in which termination requires only a single terminating run, but turns out insufficient in the probabilistic setting, when termination is quantitative and all evaluation paths have to be taken into account when comparing programs. We leave the generalization of the definability results for future research. In any case, although the above lemma could not be extended to a full abstraction result, it has an important application in the proof of our last result.

Proposition 3. $\mathsf{IA}_{\mathbf{coin+mkvar}}$ *is a conservative extension of* $\mathsf{IA}_{\mathbf{coin}}$ *for observational equivalence.*

Proof. Suppose $\Gamma \vdash M_1 \cong_{\mathsf{IA}_{\mathbf{coin}}} M_2$. By Lemma 7, for any single-threaded complete s, we have $\sum_{\{t \mid s \lhd_{\mathsf{P}}^* t\}} [\![\Gamma \vdash M_1]\!](t) = \sum_{\{t \mid s \lhd_{\mathsf{P}}^* t\}} [\![\Gamma \vdash M_2]\!](t)$. Note

[3] Proposition 1 specializes to it when both O- and P-answers are well-bracketed.

that the set $\{t \mid s \lhd_{\mathsf{P}}^{*} t\}$ is finite and partially-ordered by $\lhd_{\mathsf{P}}^{*}$. By induction with respect to the reverse order, one can then prove that for all complete s we have $[\![\Gamma \vdash M_1]\!](s) = [\![\Gamma \vdash M_2]\!](s)$, from which $\Gamma \vdash M_1 \cong_{\mathsf{IA}_{\mathbf{coin}+\mathbf{mkvar}}} M_2$ follows.

Thus, for program equivalence, **mkvar** is conservative for IA and $\mathsf{IA}_{\mathbf{coin}}$, but not for $\mathsf{IA}_{\mathbf{catch}}$ or $\mathsf{IA}_{\mathbf{or}}$.

References

1. Reynolds, J.C.: The essence of Algol. In de Bakker, J.W., van Vliet, J., eds.: Algorithmic Languages. North Holland (1978) 345–372
2. O'Hearn, P.W., Tennent, R.D., eds.: ALGOL-like Languages. Progress in Theoretical Computer Science. Birkhäuser, Boston (1997) Two volumes.
3. Reddy, U.S.: Global state considered unnecessary: An introduction to object-based semantics. Lisp and Symbolic Computation **9** (1996) 7–76
4. Abramsky, S., McCusker, G.: Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions. In O'Hearn, P.W., Tennent, R.D., eds.: Algol-like languages, Birkhaüser (1997) 297–329
5. Burstall, R.M., Popplestone, R.J.: Pop-2 reference manual. Machine Intelligence **2** (1968) 205–246
6. Reynolds, J.C.: GEDANKEN–A simple typeless language based on principle of completeness and reference concept. CACM **13** (1970) 308–319
7. McCusker, G.: On the semantics of Idealized Algol without the bad-variable constructor. In: Proceedings of MFPS'03, ENTCS 83.
8. Cartwright, R., Felleisen, M.: Observable sequentiality and full abstraction (preliminary version). In: Proceedings of POPL'92.
9. Laird, J.: Full abstraction for functional languages with control. In: Proceedings of LICS'97.
10. Abramsky, S., Ghica, D.R., Murawski, A.S., Ong, C.H.L., Stark, I.D.B.: Nominal games and full abstraction for the nu-calculus. In: Proceedings of LICS'04.
11. Laird, J.: A game semantics of local names and good variables. In: Proceedings of FOSSACS'04, LNCS 2987.
12. Abramsky, S., McCusker, G.: Game semantics. In Schwichtenberg, H., Berger, U., eds.: Logic and Computation. Springer-Verlag (1998) Proceedings of the 1997 Marktoberdorf Summer School.
13. Hyland, J.M.E., Ong, C.H.L.: On Full Abstraction for PCF. Information and Computation **163(2)** (2000) 285–408
14. Harmer, R., McCusker, G.: A fully abstract game semantics for finite nondeterminism. In: Proceedings of LICS'99.
15. Danos, V., Harmer, R.: Probabilistic game semantics. In: Proceedings of LICS'00.
16. Murawski, A.S., Ouaknine, J.: On probabilistic program equivalence and refinement. In: Proceedings of CONCUR'05, LNCS 3653.