Proving Translation and Reduction Semantics Equivalent for Java Simple Closures

Marco Bellia and M. Eugenia Occhiuto

Dipartimento di Informatica, Università di Pisa, Italy {*bellia,occhiuto*}*@di.unipi.it*

Abstract

FGCJ is a minimal core calculus that extends Featherweight (generic) Java, FGJ, with lambda expressions. It has been used to study properties of Simple Closure in Java, including type safety and the abstraction property. Its formalization is based on a reduction semantics and a typing system that extend those of FGJ. $\mathcal{F}$ is a source-to-source, translation rule system from Java 1.5 extended with lambda expressions back to ordinary Java 1.5. It has been introduced to study implementation features of closures in Java, including assignment of non local variables and relations with anonymous class objects. In this paper we prove that the two semantics commute.

1 Introduction

In [BO10], we extend Featherweight Java [ABW01] with simple closures [BGR10] (S-closure, for short). In that paper we define a minimal core calculus FGCJ to study properties of S-closures in Java and we provide a reduction semantics, $\rightarrow$, and prove type safety and abstraction property for S-closures. In [BO09], we extend Java 1.5 with S-closures. In that paper we provide a translation semantics, $\mathcal{F}$, which translates S-closures into objects of anonymous classes, built from single method interfaces. Based on the translation semantics, we obtain an implementation of Java 1.5 with S-closures by mapping, possibly through a preprocessor, programs of the extended language into programs of ordinary Java 1.5.

In this paper, we prove that these two semantics commute. As a consequence, we have that: (a) S-closures modelled in FGCJ are those considered in $\mathcal{F}$; (b) S-closures implemented in [BO09] satisfy the properties proved in [BO10]; (c) FGCJ and $\mathcal{F}$ are a framework to study and implement closures in Java in the form of S-closures and possibly, variants of them [Goe07]. To prove equivalence, we extend Featherweight Java, FGJ, to cope with interfaces and anonymous classes, obtaining FGAJ as a minimal core calculus for Java 1.5. We do the same for FGCJ obtaining FGACJ as a minimal core calculus for Java 1.5 extended with S-closures, Section 2. We extend the reduction semantics $\rightarrow$ on the new constructs and prove type safety for such extended calculi, Section 3. We restrict the translation semantics $\mathcal{F}$ to translate from FGACJ onto FGAJ and prove that the diagram in Fig. 1, commutes, Section 4.

2 Featherweight GACJ

2.1 Notation and General Conventions

In this paper we adopt the notation used in [ABW01], accordingly $\overline{\texttt{f}}$ is a shorthand for a possibly empty sequence $\texttt{f}_1, \ldots, \texttt{f}_n$ (and similarly for $\overline{\texttt{T}}, \overline{\texttt{x}}$, etc.) and $\overline{\texttt{M}}$ is a shorthand

$$\begin{tikzcd}
\text{e}\in \text{FGACJ} \arrow[r, "\rightarrow"] \arrow[d, "\mathcal{F}"'] & \text{e'}\in \text{FGACJ} \arrow[d, "\mathcal{F}"] \\
\mathcal{F}[\![\text{e}]\!] \in \text{FGAJ} \arrow[r, "\rightarrow"'] & \mathcal{F}[\![\text{e'}]\!] \in \text{FGAJ}
\end{tikzcd}$$

Fig. 1. Commutation diagram

for $\mathtt{M}_1 \ldots \mathtt{M}_n$ (with no commas) where n is the size $|\overline{\mathtt{f}}|$, respectively $|\overline{\mathtt{M}}|$, i.e. the number of terms of the sequence. The empty sequence is $\circ$ and symbol "," denotes concatenation of sequences. Operations on pairs of sequences are abbreviated in the obvious way $\overline{\mathtt{C}}\ \overline{\mathtt{f}}$ is $\mathtt{C}_1\ \mathtt{f}_1, \ldots, \mathtt{C}_n\ \mathtt{f}_n$ and similarly $\overline{\mathtt{C}}\ \overline{\mathtt{f}}$; is $\mathtt{C}_1\ \mathtt{f}_1; \ldots \mathtt{C}_n\ \mathtt{f}_n$; and $\mathtt{this}.\overline{\mathtt{f}} = \overline{\mathtt{f}}$; is a shorthand for $\mathtt{this}.\mathtt{f}_1 = \mathtt{f}_1; \ldots \mathtt{this}.\mathtt{f}_n = \mathtt{f}_n$; Sequences of field declarations, parameters and method declaration cannot contain duplications. Cast, $(\_)\_$, and closure definition, $\#\_\_$, have lower precedence than other operators, and cast precedes closure definition. Hence $\#()(\mathtt{this!}())$ can be written as $\#()\mathtt{this!}()$. The, possibly indexed and/or primed, metavariables T, V, U, S, W range over type expressions; T ranges over type expressions which are not closures; X, Y, Z range over type variables; N, P, Q range over class types; C, D, E range over class names; f, g range over field names; e, v, d range over expressions; x, y range over variable names and M, K, L and m range respectively, over methods, constructors, classes, and method names. $[x/y]e$ denotes the result of replacing y by x in e. Eventually $FV(\overline{\mathtt{T}})$ denotes the set of type variables in $\overline{\mathtt{T}}$.

2.2 Syntax

The abstract syntax of FGJ is at the beginning of **Table1**, followed by the syntactic rules that extend FGJ with (generic) interfaces and anonymous class object creation, defining language FGAJ. A type interface $\mathtt{I}\langle\overline{\mathtt{T}}\rangle$ is an interface name I and a list $\overline{\mathtt{T}}$ of the type expressions that bind the type variables $\overline{\mathtt{X}}$ of the interface declaration (see, rules $\mathtt{T}_{\text{FGAJ}}$ and $\mathtt{L}_{\text{FGJ}}$). In Java, a type interface may have subtypes, moreover classes may implement interfaces: We omit such features in FGAJ , since we consider interfaces only in combination with the mechanism of anonymous class object creation. Analogously, we omit the use of classes in anonymous class object creation and restrict it to only interfaces (see, rule $\mathtt{e}_{\text{FGAJ}}$). The use of classes, instead of interfaces, in anonymous class object creation, is more heavy since it involves method overriding: whose formalization requires additional rules. On the other hand, such complication is unnecessary for this paper aim, since translation $\mathcal{F}$ does not use such feature, The syntax of S-closures is the one adopted in [Rei09], it includes *lambda expressions* and *function types*, it is reported in the third box of **Table1** and extends FGJ in FGCJ defining the calculus studied in [BO10]. Lambda expressions consist of closures whose body is an expression and of closures whose body is a block: Since sequencing and assignment are omitted in FGJ as well as in FGCJ, the body of a closure can only be an expression (see rule $\mathtt{F}_{\text{FGCJ}}$). Closure types extend types as rule $\mathtt{T}_{\text{FGAJ}}$ shows. A closure type $\#\mathtt{T}(\overline{\mathtt{T}})$ specifies the type sequence $(\overline{\mathtt{T}})$, possibly empty (standing for the type unit), of the arguments and the type $\mathtt{T}$ of the result. An example of closure is $\#(\mathtt{Integer\ x}, \mathtt{Integer\ y})\ (\mathtt{x} + \mathtt{y})$ which has two arguments $\mathtt{x}$ and $\mathtt{y}$, has body $\mathtt{x}+\mathtt{y}$, and type $\#\mathtt{Integer}(\mathtt{Integer}, \mathtt{Integer})$. No new generic variables can be introduced when defining a closure (reasons can be found in [Rei10]) but of course generic variables (introduced in class or method declarations) can occur in the type expressions of the arguments or be used inside closure body.

Table 1 : **Syntax**	
FGJ	
$\mathtt{T} ::= \mathtt{X} \mid \mathtt{N}$	$(\mathtt{T}_{\mathrm{FGJ}})$
$\mathtt{N} ::= \mathtt{C}\langle\overline{\mathtt{T}}\rangle$	$(\mathtt{N}_{\mathrm{FGJ}})$
$\mathtt{L} ::= \mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\ \{\overline{\mathtt{T}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\}$	$(\mathtt{L}_{\mathrm{FGJ}})$
$\mathtt{K} ::= \mathtt{C}(\overline{\mathtt{T}}\ \overline{\mathtt{f}})\{\mathtt{super}(\overline{\mathtt{f}}); \mathtt{this}.\overline{\mathtt{f}} = \overline{\mathtt{f}};\}$	$(\mathtt{K}_{\mathrm{FGJ}})$
$\mathtt{M} ::= \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e};\}$	$(\mathtt{M}_{\mathrm{FGJ}})$
$\mathtt{e} ::= \mathtt{x} \mid \mathtt{e.f} \mid \mathtt{e.m}\langle\overline{\mathtt{T}}\rangle(\overline{\mathtt{e}}) \mid \mathtt{new}\ \mathtt{N}(\overline{\mathtt{e}}) \mid (\mathtt{N})\mathtt{e}$	$(\mathtt{e}_{\mathrm{FGJ}})$
IA: *Extensions for Interfaces and Anonymous Class Objects*	
$\mathtt{T} ::= \mathtt{I}\langle\overline{\mathtt{T}}\rangle$	$(\mathtt{T}_{\mathrm{FGAJ}})$
$\mathtt{L} ::= \mathtt{interface}\ \mathtt{I}\ \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\{\overline{\mathtt{H}}\}$	$(\mathtt{L}_{\mathrm{FGAJ}})$
$\mathtt{H} ::= \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})$	$(\mathtt{H}_{\mathrm{FGAJ}})$
$\mathtt{e} ::= \mathtt{new}\ \mathtt{I}\langle\overline{\mathtt{T}}\rangle()\ \{\overline{\mathtt{M}}\}$	$(\mathtt{e}_{\mathrm{FGAJ}})$
Cl: *Extensions for Closures*	
$\mathtt{T} ::= \#\mathtt{T}(\overline{\mathtt{T}})$	$(\mathtt{T}_{\mathrm{FGCJ}})$
$\mathtt{e} :: \mathtt{F} \mid \mathtt{e}\ !\ (\overline{\mathtt{e}})$	$(\mathtt{e}_{\mathrm{FGCJ}})$
$\mathtt{F} ::= \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\mathtt{e}$	$(\mathtt{F}_{\mathrm{FGCJ}})$
FGAJ = FGJ + IA	
FGACJ = FGJ + IA + Cl	
FGCJ = FGJ + Cl	

Eventually, at the bottom of **Table1**, the syntactic structure of the various calculi, considered in the paper, is resumed. For space convenience, the reduction rules of the semantics as well as the typing rules are not given in separate tables for each calculus. In fact, since compositionality of the semantics (we use), the rules of the various constructs are the same in all calculi containing such a construct. However, for the reader convenience, in all tables, but **Table 3**, the rules for each calculus, FGJ, FGAJ, FGCJ, FGACJ, have a label which is indexed by the name of the minimal calculus including the construct, involved in the rule. Note that $\mathtt{C}\langle\overline{\mathtt{T}}\rangle$ include $\mathtt{Object}$(since $\overline{\mathtt{T}}$ may be the empty sequence and $\mathtt{C}$ may be $\mathtt{Object}$) hence generic variables in classes and methods can be instantiated with types $\mathtt{T}$ that include interfaces or closures.

2.3 Semantics: Reduction

The reduction semantics is given through the inference rules in **Table 2**, which define the reduction relation $\mathtt{e} \longrightarrow \mathtt{e}'$ that says that “expression $\mathtt{e}$ reduces to expression $\mathtt{e}'$ in one step". The set of expressions which cannot be further reduced is the set of *normal forms* and constitute values of the calculus. In FGACJ values are objects, constructed out of an anonymous or named class, and of closures. Hence the grammatical category v defines the syntactic form of the values (domain) of the calculus FGACJ:

$$\begin{array}{rl} \mathtt{v} ::= & \mathtt{new\ N}(\overline{\mathtt{v}}) \\ & \mid \mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\} \\ & \mid \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\mathtt{e} \end{array}$$

Table 2: Computation

Computation	
$\dfrac{\mathit{fields}(\mathtt{N}) = \overline{\mathtt{T}}\ \overline{\mathtt{f}}}{(\mathtt{new\ N}(\overline{\mathtt{e}})).\mathtt{f}_i \longrightarrow \mathtt{e}_i}$	(GR-FIELD$_{\mathrm{FGJ}}$)
$\dfrac{\mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{N}) = \overline{\mathtt{x}}.\mathtt{e}}{(\mathtt{new\ N}(\overline{\mathtt{e}})).\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{d}}) \longrightarrow [\overline{\mathtt{d}}/\overline{\mathtt{x}}, \mathtt{new\ N}(\overline{\mathtt{e}})/\mathtt{this}]\mathtt{e}}$	(GR-INVK$_{\mathrm{FGJ}}$)
$\dfrac{\emptyset \vdash \mathtt{N} <: \mathtt{P}}{(\mathtt{P})(\mathtt{new\ N}(\overline{\mathtt{e}})) \longrightarrow \mathtt{new\ N}(\overline{\mathtt{e}})}$	(GR-CAST$_{\mathrm{FGJ}}$)
$\#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}!(\overline{\mathtt{d}}) \longrightarrow [\overline{\mathtt{d}}/\overline{\mathtt{x}}, \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\mathtt{e}/\mathtt{this}]\mathtt{e}$	(GR-INV-CLOS$_{\mathrm{FGCJ}}$)
$\dfrac{\mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}) = \overline{\mathtt{x}}.\mathtt{e}}{(\mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}).\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{d}}) \longrightarrow [\overline{\mathtt{d}}/\overline{\mathtt{x}}, \mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}/\mathtt{this}]\mathtt{e}}$	(GR-INVK-ANONYM$_{\mathrm{FGAJ}}$)
Congruence	
$\dfrac{\mathtt{e}_0 \longrightarrow \mathtt{e}_0'}{\mathtt{e}_0.\mathtt{f} \longrightarrow \mathtt{e}_0'.\mathtt{f}}$	(GRC-FIELD$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{e}_0 \longrightarrow \mathtt{e}_0'}{\mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\overline{\mathtt{e}}) \longrightarrow \mathtt{e}_0'.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\overline{\mathtt{e}})}$	(GRC-T-INV$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{e}_i \longrightarrow \mathtt{e}_i'}{\mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\ldots, \mathtt{e}_i, \ldots) \longrightarrow \mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\ldots, \mathtt{e}_i' \ldots)}$	(GRC-INV-ARG$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{e}_i \longrightarrow \mathtt{e}_i'}{\mathtt{new\ N}(\ldots, \mathtt{e}_i, \ldots) \longrightarrow \mathtt{new\ N}(\ldots, \mathtt{e}_i', \ldots)}$	(GRC-NEW$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{e} \longrightarrow \mathtt{e}'}{(\mathtt{N})\mathtt{e} \longrightarrow (\mathtt{N})\mathtt{e}'}$	(GRC-CAST$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{e} \longrightarrow \mathtt{e}'}{\#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e} \longrightarrow \#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}'}$	(GRC-CLOS-VAL$_{\mathrm{FGCJ}}$)
$\dfrac{\mathtt{e} \longrightarrow \mathtt{e}'}{\mathtt{e}!(\overline{\mathtt{e}}) \longrightarrow \mathtt{e}'!(\overline{\mathtt{e}})}$	(GRC-INV-CLOS$_{\mathrm{FGCJ}}$)
$\dfrac{\mathtt{e}_i \longrightarrow \mathtt{e}_i'}{\mathtt{e}!(\ldots, \mathtt{e}_i, \ldots) \longrightarrow \mathtt{e}!(\ldots, \mathtt{e}_i', \ldots)}$	(GRC-CLOS-ARG$_{\mathrm{FGCJ}}$)

This structure of values results from the reduction rules of the calculus. The rules indexed by FGJ are the same as those of calculus FGJ introduced in [ABW01], and those indexed by FGCJ are the same as those of the calculus FGCJ introduced in [BO10]. In particular, they include rule GR-Invk-Clos that reduces a closure invocation replacing it by the closure body in which the formal parameters are replaced by the corresponding actual ones, and `this` is replaced by the closure itself, thus allowing *recursive closures*.

Table 3: Classes and Interfaces

Subclassing	
$\mathtt{C} \trianglelefteq \mathtt{C}$ $\quad \dfrac{\mathtt{C} \trianglelefteq \mathtt{D} \qquad \mathtt{D} \trianglelefteq \mathtt{E}}{\mathtt{C} \trianglelefteq \mathtt{E}}$ $\quad \dfrac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle \triangleleft \mathtt{D}\ \{\overline{\mathtt{S}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\}}{\mathtt{C} \trianglelefteq \mathtt{D}}$	
Auxiliary functions	
$\mathit{fields}(\mathtt{Object}) = \circ$	(F-Object)
$\dfrac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle \triangleleft \mathtt{N}\ \{\overline{\mathtt{S}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\} \qquad \mathit{fields}([\overline{\mathtt{T}}/\overline{\mathtt{X}}]\mathtt{N}) = \overline{\mathtt{U}}\ \overline{\mathtt{g}}}{\mathit{fields}(\mathtt{C}\langle\overline{\mathtt{T}}\rangle) = \overline{\mathtt{U}}\ \overline{\mathtt{g}}, [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\overline{\mathtt{S}}\ \overline{\mathtt{f}}}$	(F-Class)
$\dfrac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle \triangleleft \mathtt{N}\ \{\overline{\mathtt{S}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\} \qquad \langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\mathtt{U\ m}\ (\overline{\mathtt{U}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e};\} \in \overline{\mathtt{M}}}{\mathit{mtype}(\mathtt{m}, \mathtt{C}\langle\overline{\mathtt{T}}\rangle) = [\overline{\mathtt{T}}/\overline{\mathtt{X}}](\langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\overline{\mathtt{U}} \to \mathtt{U})}$	(MT-Class)
$\dfrac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle \triangleleft \mathtt{N}\ \{\overline{\mathtt{S}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\} \qquad \mathtt{m} \notin \overline{\mathtt{M}}}{\mathit{mtype}(\mathtt{m}, \mathtt{C}\langle\overline{\mathtt{T}}\rangle) = \mathit{mtype}(\mathtt{m}, [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\mathtt{N})}$	(MT-Super)
$\dfrac{\mathtt{interface\ I}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle\ \{\overline{\mathtt{H}}\} \qquad \langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\mathtt{U\ m}(\overline{\mathtt{U}}\ \overline{\mathtt{x}}) \in \overline{\mathtt{H}}}{\mathit{mtype}(\mathtt{m}, \mathtt{I}\langle\overline{\mathtt{T}}\rangle) = [\overline{\mathtt{T}}/\overline{\mathtt{X}}](\langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\overline{\mathtt{U}} \to \mathtt{U})}$	(MT-Interface)
$\dfrac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle \triangleleft \mathtt{N}\ \{\overline{\mathtt{S}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\} \qquad \langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\mathtt{U\ m}\ (\overline{\mathtt{U}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e};\} \in \overline{\mathtt{M}}}{\mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{C}\langle\overline{\mathtt{T}}\rangle) = \overline{\mathtt{x}}.[\overline{\mathtt{T}}/\overline{\mathtt{X}}, \overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{e}}$	(MB-Class)
$\dfrac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle \triangleleft \mathtt{N}\ \{\overline{\mathtt{S}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\} \qquad \mathtt{m} \notin \overline{\mathtt{M}}}{\mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{C}\langle\overline{\mathtt{T}}\rangle) = \mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\mathtt{N})}$	(MB-Super)
$\dfrac{\mathtt{interface\ I}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle\ \{...\} \qquad \langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\mathtt{U\ m}\ (\overline{\mathtt{U}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e};\} \in \overline{\mathtt{M}}}{\mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}) = \overline{\mathtt{x}}.[\overline{\mathtt{T}}/\overline{\mathtt{X}}, \overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{e}}$	(MB-Interface)
Auxiliary predicates	
$\mathtt{override}(\mathtt{m}, \mathtt{Object}, \langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\overline{\mathtt{T}} \to \mathtt{T}_0)$	(Over-Object)
$\dfrac{\begin{array}{l}\mathit{mtype}(\mathtt{m}, \mathtt{N}) = \langle\overline{\mathtt{Z}} \triangleleft \overline{\mathtt{Q}}\rangle\overline{\mathtt{U}} \to \mathtt{U}_0\ \mathtt{implies} \\ \qquad ((\overline{\mathtt{P}}, \overline{\mathtt{T}}) = [\overline{\mathtt{Y}}/\overline{\mathtt{Z}}](\overline{\mathtt{Q}}, \overline{\mathtt{U}})\ \mathtt{and}\ \overline{\mathtt{Y}} <: \overline{\mathtt{P}} \vdash \mathtt{T}_0 <: [\overline{\mathtt{Y}}/\overline{\mathtt{Z}}]\mathtt{U}_0)\end{array}}{\mathtt{override}(\mathtt{m}, \mathtt{N}, \langle\overline{\mathtt{Y}} \triangleleft \overline{\mathtt{P}}\rangle\overline{\mathtt{T}} \to \mathtt{T}_0)}$	(Over)
DCast	
$\dfrac{\mathit{dcast}(\mathtt{C}, \mathtt{D}) \qquad \mathit{dcast}(\mathtt{D}, \mathtt{E})}{\mathit{dcast}(\mathtt{C}, \mathtt{E})}$ $\quad \dfrac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \triangleleft \overline{\mathtt{N}}\rangle \triangleleft \mathtt{D}\langle\overline{\mathtt{T}}\rangle\ \{\ldots\} \qquad \overline{\mathtt{X}} = FV(\overline{\mathtt{T}})}{\mathit{dcast}(\mathtt{C}, \mathtt{D})}$	(DCast)

Table 4: Typing Rules

$\Delta;\Gamma \vdash \mathtt{x} : \Gamma(\mathtt{x})$	(GT-Var$_{\text{FGJ}}$)
$\dfrac{\Delta;\Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \mathit{fields}(\mathit{bound}_\Delta(\mathtt{T}_0)) = \overline{\mathtt{T}}\ \overline{\mathtt{f}}}{\Delta;\Gamma \vdash \mathtt{e}_0.\mathtt{f}_i : \mathtt{T}_i}$	(GT-Field$_{\text{FGJ}}$)
$\dfrac{\begin{array}{c} \mathit{mtype}(\mathtt{m}, \mathit{bound}_\Delta(\mathtt{T}_0)) = \langle \overline{\mathtt{Y}} \lhd \overline{\mathtt{P}} \rangle \overline{\mathtt{U}} \rightarrow \mathtt{U} \\ \Delta;\Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \overline{\mathtt{V}}\ \mathtt{ok} \qquad \Delta \vdash \overline{\mathtt{V}} <: [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}} \\ \Delta;\Gamma \vdash \overline{\mathtt{e}} : \overline{\mathtt{S}} \qquad \Delta \vdash \overline{\mathtt{S}} <: [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{U}} \end{array}}{\Delta;\Gamma \vdash \mathtt{e}_0.\mathtt{m}\langle \overline{\mathtt{V}} \rangle (\overline{\mathtt{e}}) : [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}}$	(GT-Inv$_{\text{FGJ}}$)
$\dfrac{\begin{array}{c} \mathit{mtype}(\mathtt{m}, \mathtt{I}\langle \overline{\mathtt{T}} \rangle) = \langle \overline{\mathtt{Y}} \lhd \overline{\mathtt{P}} \rangle \overline{\mathtt{U}} \rightarrow \mathtt{U} \\ \Delta;\Gamma \vdash \mathtt{e}_0 : \mathtt{I}\langle \overline{\mathtt{T}} \rangle \qquad \Delta \vdash \overline{\mathtt{V}}\ \mathtt{ok} \qquad \Delta \vdash \overline{\mathtt{V}} <: [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}} \\ \Delta;\Gamma \vdash \overline{\mathtt{e}} : \overline{\mathtt{S}} \qquad \Delta \vdash \overline{\mathtt{S}} <: [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{U}} \end{array}}{\Delta;\Gamma \vdash \mathtt{e}_0.\mathtt{m}\langle \overline{\mathtt{V}} \rangle (\overline{\mathtt{e}}) : [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}}$	(GT-AnonymInv$_{\text{FGAJ}}$)
$\dfrac{\begin{array}{c} \Delta \vdash \mathtt{N}\ \mathtt{ok} \qquad \mathit{fields}(\mathtt{N}) = \overline{\mathtt{T}}\ \overline{\mathtt{f}} \\ \Delta;\Gamma \vdash \overline{\mathtt{e}} : \overline{\mathtt{S}} \qquad \Delta \vdash \overline{\mathtt{S}} <: \overline{\mathtt{T}} \end{array}}{\Delta;\Gamma \vdash \mathtt{new}\ \mathtt{N}(\overline{\mathtt{e}}) : \mathtt{N}}$	(GT-New$_{\text{FGJ}}$)
$\dfrac{\Delta \vdash \mathtt{I}\langle \overline{\mathtt{T}} \rangle\ \mathtt{ok} \qquad \Delta;\Gamma \vdash \overline{\mathtt{M}}\ \ \mathtt{OK\,IN}\ \mathtt{I}\langle \overline{\mathtt{T}} \rangle}{\Delta;\Gamma \vdash \mathtt{new}\ \mathtt{I}\langle \overline{\mathtt{T}} \rangle()\{\overline{\mathtt{M}}\} : \mathtt{I}\langle \overline{\mathtt{T}} \rangle}$	(GT-AnonymNew$_{\text{FGAJ}}$)
$\dfrac{\Delta;\Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \mathit{bound}_\Delta(\mathtt{T}_0) <: \mathtt{N}}{\Delta;\Gamma \vdash (\mathtt{N})\mathtt{e}_0 : \mathtt{N}}$	(GT-UCast$_{\text{FGJ}}$)
$\dfrac{\begin{array}{c} \Delta;\Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \mathtt{N}\ \mathtt{ok} \qquad \Delta \vdash \mathtt{N} <: \mathit{bound}_\Delta(\mathtt{T}_0) \\ \mathtt{N} = \mathtt{C}\langle \overline{\mathtt{T}} \rangle \qquad \mathit{bound}_\Delta(\mathtt{T}_0) = \mathtt{D}\langle \overline{\mathtt{T}} \rangle \qquad \mathit{dcast}(\mathtt{C}, \mathtt{D}) \end{array}}{\Delta;\Gamma \vdash (\mathtt{N})\mathtt{e}_0 : \mathtt{N}}$	(GT-DCast$_{\text{FGJ}}$)
$\dfrac{\begin{array}{c} \Delta;\Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \mathtt{N}\ \mathtt{ok} \\ \mathtt{N} = \mathtt{C}\langle \overline{\mathtt{T}} \rangle \qquad \mathit{bound}_\Delta(\mathtt{T}_0) = \mathtt{D}\langle \overline{\mathtt{U}} \rangle \qquad \mathtt{C} \ntrianglelefteq \mathtt{D} \qquad \mathtt{D} \ntrianglelefteq \mathtt{C} \end{array}}{\Delta;\Gamma \vdash (\mathtt{N})\mathtt{e}_0 : \mathtt{N}}$	(GT-SCast$_{\text{FGJ}}$)
$\dfrac{\Delta \vdash \overline{\mathtt{T}}\ \mathtt{ok} \qquad \Delta;\Gamma, \overline{\mathtt{x}} : \overline{\mathtt{T}}, \mathtt{this} : \#\mathtt{T}(\overline{\mathtt{T}}) \vdash \mathtt{e}\ : \mathtt{T}}{\Delta;\Gamma \vdash \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\ \mathtt{e} : \#\mathtt{T}(\overline{\mathtt{T}})}$	(GT-closure$_{\text{FGCJ}}$)
$\dfrac{\Delta;\Gamma \vdash \mathtt{e} : \#\mathtt{T}(\overline{\mathtt{T}}) \qquad \Delta;\Gamma \vdash \overline{\mathtt{e}} : \overline{\mathtt{S}} \qquad \Delta \vdash \overline{\mathtt{S}} <: \overline{\mathtt{T}}}{\Delta;\Gamma \vdash \mathtt{e!}(\overline{\mathtt{e}}) : \mathtt{T}}$	(GT-Closure-Inv$_{\text{FGCJ}}$)

We have only one new rule, GR-Invk-Anonym$_{\text{FGAJ}}$, which is indexed by FGACJ and gives the semantics of invocation with anonymous class objects. The rule is similar to the one of method invocation with object of named classes. In fact, the two kinds of invocation may be formulated similarly provided that the auxiliary functions `mtype` and `mbody`, introduced in **Table3**, are suitably extended to select the type and the body of anonymous class objects (see rules MT-Interface and MB-Interface). Moreover, since anonymous class object creation is formulated as a new expression that extends the calculus FGJ (resp. FGCJ), the rules of congruence of [ABW01] (resp. [BO10]) are unchanged.

2.4 Semantics: Typing

The typing rules are given through inference rules that use two different kinds of environment, Δ (for type variables) and Γ (for value variables), and five different typing judgements: one for each different term structure of the language. A (well formed) type environment Δ is a mapping from type variables to (well formed, in Δ) types written as a list of $\mathtt{X}{<:}\mathtt{T}$, meaning that type variable $\mathtt{X}$ must be bound to a subtype of type $\mathtt{T}$: $\Delta(\mathtt{X}) = \mathtt{T}$ if Δ contains $\mathtt{X}{<:}\mathtt{T}$, undefined otherwise (i.e. $\mathtt{X} \notin \mathit{dom}(\Delta)$). An environment Γ is a mapping from variables to types written as a list of $\mathtt{x} : \mathtt{T}$ meaning that "$\mathtt{x}$ has type $\mathtt{T}$": $\Gamma(\mathtt{x}) = \mathtt{T}$ if Γ contains $\mathtt{x} : \mathtt{T}$, undefined otherwise (i.e. $\mathtt{x} \notin \mathit{dom}(\Gamma)$). When needed and without loss of generality, variable renaming is used to avoid name collision among environment bindings. The judgement for a (generic) type $\mathtt{T}$ (see **Table 5**) has the form $\Delta \vdash \mathtt{T}\ \mathtt{ok}$ meaning that "$\mathtt{T}$ is a well-formed type in the (well formed) type environment Δ". The typing judgements for subtyping (see **Table 5**) has the form $\Delta \vdash \mathtt{S}{<:}\mathtt{T}$ meaning that "$\mathtt{S}$ is a subtype of $\mathtt{T}$ in Δ". The judgement for classes (see rule GT-Class$_{\text{FGJ}}$ in **Table 4a**) has the form $\mathtt{C}\ \mathtt{OK}$ meaning that "$\mathtt{C}$ is well typed". The typing judgements for methods (see GT-Method$_{\text{FGJ}}$ in **Table 4a**) has the form $\mathtt{M}\ \mathtt{OK\,IN}\ \mathtt{C}$ meaning that "$\mathtt{M}$ is well typed when its declaration occurs in class $\mathtt{C}$". The judgement for expressions (see the rules of **Table 4**) has the form $\Delta; \Gamma \vdash \mathtt{e} : \mathtt{T}$ meaning that expression $\mathtt{e}$ has type $\mathtt{T}$ in the typing environment Δ and in the (variable) environment Γ. The typing rules are contained in **Table 4** and extends those of FGJ. Two rules have been added for closure construction and closure invocation. Such rules simply assert the correctness of the involved types. Four rules have been added for typing (GT-AnonymInv$_{\text{FGAJ}}$), judgement OK (GT-Interface$_{\text{FGAJ}}$), judgement OK IN (GT-Interf$_{\text{FGAJ}}$, GT-Anonym$_{\text{FGAJ}}$). The rules for subtypes and wellformed types are reported in **Table 5**.

Table 4a: Typing Rules

Classes, Interfaces, Methods

$$\frac{\begin{array}{c}\Delta = \overline{\mathtt{X}}{<:}\overline{\mathtt{N}},\ \overline{\mathtt{Y}}{<:}\overline{\mathtt{P}} \qquad \Delta \vdash \overline{\mathtt{T}}, \mathtt{T}, \overline{\mathtt{P}}\ \mathtt{ok}\\ \Delta; \overline{\mathtt{x}} : \overline{\mathtt{T}}, \mathtt{this} : \mathtt{C}\langle\overline{\mathtt{X}}\rangle \vdash \mathtt{e}_0 : \mathtt{S} \qquad \Delta \vdash \mathtt{S}{<:}\mathtt{T}\\ \mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\{...\} \qquad \mathtt{override}(\mathtt{m}, \mathtt{N}, \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\overline{\mathtt{T}} \rightarrow \mathtt{T})\end{array}}{\langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{T}\ \ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e}_0;\}\ \mathtt{OK\,IN}\ \ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle} \quad (\text{GT-Method}_{\text{FGJ}})$$

$$\frac{\overline{\mathtt{Y}}{<:}\overline{\mathtt{P}}, \overline{\mathtt{X}}{<:}\overline{\mathtt{N}} \vdash \overline{\mathtt{T}}, \mathtt{T}, \overline{\mathtt{P}}\ \mathtt{ok}}{\langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{T}\ \ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\ \mathtt{OK\,IN}\ \ \mathtt{I}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle} \quad (\text{GT-Header}_{\text{FGAJ}})$$

$$\frac{\begin{array}{c}\Delta' = \Delta, \overline{\mathtt{X}}{<:}\overline{\mathtt{N}},\ \overline{\mathtt{Y}}{<:}\overline{\mathtt{P}} \qquad \Delta'; \Gamma, \overline{\mathtt{x}} : \overline{\mathtt{T}}, \mathtt{this} : \mathtt{I}\langle\overline{\mathtt{V}}\rangle \vdash \mathtt{e}_0 : \mathtt{S}\\ \Delta' \vdash \overline{\mathtt{T}}, \mathtt{T}, \overline{\mathtt{P}}\ \mathtt{ok} \qquad \Delta' \vdash \overline{\mathtt{V}}{<:}[\overline{\mathtt{V}}/\overline{\mathtt{X}}]\overline{\mathtt{N}} \qquad \Delta' \vdash \mathtt{S}{<:}\mathtt{T}\\ \mathtt{interface}\ \mathtt{I}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\{\overline{\mathtt{H}}\} \qquad \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\ \in\ \overline{\mathtt{H}}\end{array}}{\Delta; \Gamma \vdash \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e}_0;\}\ \mathtt{OK\,IN}\ \ \mathtt{I}\langle\overline{\mathtt{V}}\rangle} \quad (\text{GT-Anonym}_{\text{FGAJ}})$$

$$\frac{\begin{array}{c}\overline{\mathtt{X}} <: \overline{\mathtt{N}} \vdash \overline{\mathtt{N}}, \mathtt{N}, \overline{\mathtt{T}}\ \mathtt{ok} \qquad \overline{\mathtt{M}}\ \ \mathtt{OK\,IN}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\\ \mathit{fields}(\mathtt{N}) = \overline{\mathtt{U}}\ \overline{\mathtt{g}} \qquad \mathtt{K} = \mathtt{C}(\overline{\mathtt{U}}\ \overline{\mathtt{g}}, \overline{\mathtt{T}}\ \overline{\mathtt{f}})\{\text{super}(\overline{\mathtt{g}});\ \text{this}.\overline{\mathtt{f}} = \overline{\mathtt{f}};\}\end{array}}{\mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\{\overline{\mathtt{T}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\}\ \mathtt{OK}} \quad (\text{GT-Class}_{\text{FGJ}})$$

$$\frac{\overline{\mathtt{X}} <: \overline{\mathtt{N}} \vdash \overline{\mathtt{N}}\ \mathtt{ok} \qquad \overline{\mathtt{H}}\ \ \mathtt{OK\,IN}\ \ \mathtt{I}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle}{\mathtt{interface}\ \mathtt{I}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\{\overline{\mathtt{H}}\}\ \mathtt{OK}} \quad (\text{GT-Interf}_{\text{FGAJ}})$$

Table 5: Subtypes

Subtypes	
$bound_{\Delta}(\mathtt{X}) = \Delta(\mathtt{X})$	(B-Var$_{\mathrm{FGJ}}$)
$bound_{\Delta}(\mathtt{N}) = \mathtt{N}$	(B-Class$_{\mathrm{FGJ}}$)
$\Delta \vdash \mathtt{T} <: \mathtt{T}$	(S-Refl$_{\mathrm{FGJ}}$)
$\dfrac{\Delta \vdash \mathtt{S} <: \mathtt{T} \qquad \Delta \vdash \mathtt{T} <: \mathtt{U}}{\Delta \vdash \mathtt{S} <: \mathtt{U}}$	(S-Trans$_{\mathrm{FGJ}}$)
$\Delta \vdash \mathtt{C}\langle\overline{\mathtt{T}}\rangle <: [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\mathtt{N}$	(S-Var$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\{\ldots\}}{\Delta \vdash \mathtt{C}\langle\overline{\mathtt{T}}\rangle <: [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\mathtt{N}}$	(S-Class$_{\mathrm{FGJ}}$)
Well-formed types	
$\Delta \vdash \text{Object}\ \mathtt{ok}$	(WF-Object$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{X} \in dom(\Delta)}{\Delta \vdash \mathtt{X}\ \mathtt{ok}}$	(WF-Var$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\{\ldots\} \qquad \Delta \vdash \overline{\mathtt{T}}\ \mathtt{ok} \qquad \Delta \vdash \overline{\mathtt{T}} <: [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\overline{\mathtt{N}}}{\Delta \vdash \mathtt{C}\langle\overline{\mathtt{T}}\rangle\ \mathtt{ok}}$	(WF-Class$_{\mathrm{FGJ}}$)
$\dfrac{\mathtt{interface}\ \mathtt{I}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\{\ldots\} \qquad \Delta \vdash \overline{\mathtt{T}}\ \mathtt{ok} \qquad \Delta \vdash \overline{\mathtt{T}} <: [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\overline{\mathtt{N}}}{\Delta \vdash \mathtt{I}\langle\overline{\mathtt{T}}\rangle\ \mathtt{ok}}$	(WF-Interf$_{\mathrm{FGAJ}}$)
$\dfrac{\Delta \vdash \overline{\mathtt{T}}\ \mathtt{ok} \qquad \Delta \vdash \mathtt{T}\ \mathtt{ok}}{\Delta \vdash \#\mathtt{T}(\overline{\mathtt{T}})\ \mathtt{ok}}$	(WF-Closure$_{\mathrm{FGCJ}}$)

3 Properties

Semantics is useful to prove language properties: We extend to FGACJ type soundness and backward compatibility already proved for FGJ [ABW01] and for FGCJ [BO10]. Then, we extend to FGACJ the closure abstraction property already proved for FGCJ [BO11b]. All Lemma and Theorem proofs can be found in the paper extended version [BO11c].

Theorem 1 (Progress). *Suppose* e *is a well-typed expression. If* e *includes as a subexpression:*

1. $\mathtt{new}\ \mathtt{N}(\overline{\mathtt{e}}).\mathtt{f}$ *then* $fields(\mathtt{N}) = \overline{\mathtt{T}}\ \overline{\mathtt{f}}$*, for some* $\overline{\mathtt{T}}$ *and* $\overline{\mathtt{f}}$*, and* $\mathtt{f} \in \overline{\mathtt{f}}$.
2. $\mathtt{new}\ \mathtt{N}(\overline{\mathtt{e}}).\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{d}})$ *then* $mbody(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{N}) = \overline{\mathtt{x}}.\mathtt{e}_0$*, for some* $\overline{\mathtt{x}}$ *and* $\mathtt{e}_0$*, and* $|\overline{\mathtt{x}}| = |\overline{\mathtt{d}}|$.
3. $\mathtt{new}\ \mathtt{I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}.\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{d}})$ *then* $mbody(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{new}\ \mathtt{I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}) = \overline{\mathtt{x}}.\mathtt{e}_0$*, for some* $\overline{\mathtt{x}}$ *and* $\mathtt{e}_0$*, and* $|\overline{\mathtt{x}}| = |\overline{\mathtt{d}}|$.
4. $\mathtt{F}!(\overline{\mathtt{d}})$ *then* $\mathtt{F} = \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\ \mathtt{e}_0$*, for some* $\overline{\mathtt{T}}$*,* $\overline{\mathtt{x}}$ *and* $\mathtt{e}_0$*, and* $|\overline{\mathtt{x}}| = |\overline{\mathtt{d}}|$. □

Theorem 2 (Type Soundness). *If* $\emptyset; \emptyset \vdash_{\mathrm{FGACJ}} \mathtt{e} : \mathtt{T}$ *and* $\mathtt{e} \rightarrow^{*}_{\mathrm{FGACJ}} \mathtt{e}'$ *with* $\mathtt{e}'$ *a normal form, then* $\mathtt{e}'$ *is a value* $\mathtt{w}$ *with* $\emptyset; \emptyset \vdash_{\mathrm{FGACJ}} \mathtt{w} : \mathtt{S}$ *and* $\emptyset \vdash_{\mathrm{FGACJ}} \mathtt{S} <: \mathtt{T}$ □

Theorem 3 (Abstraction Property). *Let* $\Delta \vdash_{\text{FGACJ}} \texttt{T ok}$, $\texttt{H}[\bullet]$ *be any context,* $\texttt{G}[\bullet]$ *be any context of type* $(\Gamma, \texttt{T})$[1] *and with no free occurrences of* `this`*. Let* $\texttt{e}_2$ *be any expression such that its free variables are not bound in* $\texttt{e}_1 \equiv \texttt{G}[\texttt{e}_2]$ *(but possibly, in* $\texttt{H}[\bullet]$*). Then* $\texttt{H}[(\#(\texttt{T x})\texttt{G}[\texttt{x}])!(\texttt{e}_2)] \approx \texttt{H}[\texttt{e}_1]$*, for any fresh variable* `x`. $\square$

Theorem 4 (Backward compatibility). *If an* FGACJ *program is well typed under the* FGCJ *rules it is also well typed under the* FGACJ *rules. Moreover, for all* FGCJ *programs e and e′ (whether well typed or not)* $\texttt{e} \rightarrow_{\text{FGACJ}} \texttt{e}' \iff \texttt{e} \rightarrow_{\text{FGACJ}} \texttt{e}'$. $\square$

4 The Translation Semantics of Java Simple Closures

The translation semantics $\mathcal{F}[\![]\!]_\tau$ of S-closures has been defined in [BO11b], it is based on the structures of *interfaces*, *anonymous classes*, and *classes* (of variable objects) and translates S-closures into a composition of such structures. In this paper, we simplifies $\mathcal{F}[\![]\!]_\tau$ in order to apply it to FGACJ, instead of Java 1.5 extended with S-closures and to translate onto FGAJ instead of ordinary Java 1.5. FGACJ (resp FGAJ) is used as a minimal core calculus for Java 1.5 extended with closures (resp. ordinary Java 1.5), to formalize the translation semantics of closures with anonymous class objects. It is shown in Fig. 2, where J is a syntactic projection of Java 1.5 onto FGAJ.

$$\begin{tikzcd}
\text{e} \in \text{Java1.5}^{\text{E}} \arrow[r, "\mathcal{F}"] \arrow[d, "J"'] & \text{e'} \in \text{Java1.5} \arrow[d, "J"] \\
J(\text{e}) \in \text{FGACJ} \arrow[r, "\mathcal{F}"'] & J(\text{e'}) \in \text{FGAJ}
\end{tikzcd}$$

Fig. 2. FGACJ is a Minimal Core Calculus for Translation $\mathcal{F}$ on Java 1.5 with S-Closures

A first and most evident simplification, is the elimination of parameter τ. In the original translation τ contained bindings for Java variables, which are not present in FGACJ [2]. In **Table 6** we report the definition of $\mathcal{F}[\![]\!]$ restricted to FGACJ. The inference rules of the definition are written following the compact notation used in [BO11a], however:

$$\mathcal{F}[\![U]\!] = L' \quad \texttt{with } U \equiv L \quad \texttt{and } C_1, ..., C_n \qquad \texttt{stands for:} \quad \frac{C_1, ..., C_n, Z_1, ..., Z_k}{L \quad \rightarrow_{\mathcal{F}} \quad L'_Z}$$

[1] A *context of type* $(\Gamma, \texttt{T})$ is any context $\texttt{H}[\bullet]$, in FGCJ, such that $\Delta; \Gamma, \texttt{x} : \texttt{T} \vdash \texttt{H}[\texttt{x}] : \texttt{S}$ for some $\Delta \vdash \texttt{T ok}$ and type `S`, and fresh variable `x`[BO11b]. The self reference `this` occurs bound, in a context (or an expression), only when it occurs inside a closure or a method defined in such a context (or expression). In all the other cases, `this` occurs free.

[2] FGACJ contains parameters and class fields whose life cycle is different from the one of non-local variables [BO11a]

where: U ranges over the syntactic domains of the language, $L \in U_{\text{FGACJ}}$ (i.e. syntactic domain U of FGACJ), $L' \in U_{\text{FGAJ}}$ (i.e. domain U of language FGAJ), premises C_i are judgments (possibly including typing judgments), $\rightarrow_{\mathcal{F}}$ is the translation judgement.

Table 6: $\mathcal{F}\llbracket\rrbracket$ Translation Semantics

Translation Rules	
1t. $\mathcal{F}\llbracket\mathtt{T}\rrbracket = \mathtt{T}$	with $\mathtt{T} \equiv \mathtt{X}$
2t. $\mathcal{F}\llbracket\mathtt{T}\rrbracket = \mathtt{I\$n}\langle\mathcal{F}\llbracket\overline{\mathtt{S}}\rrbracket\mathcal{F}\llbracket\mathtt{S}\rrbracket\rangle$	with $\mathtt{T} \equiv \#\mathtt{S}(\overline{\mathtt{S}}),\ \mathtt{n} = \lvert\overline{\mathtt{T}}\rvert + 1$
3t. $\mathcal{F}\llbracket\mathtt{T}\rrbracket = \mathtt{A}\langle\mathcal{F}\llbracket\overline{\mathtt{T}}\rrbracket\rangle$	with $\mathtt{T} \equiv \mathtt{A}\langle\overline{\mathtt{T}}\rangle,\ (\mathtt{A} \equiv \mathtt{C}$ or $\mathtt{A} \equiv \mathtt{I})$
1l. $\mathcal{F}\llbracket\mathtt{L}\rrbracket = \mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \mathcal{F}\llbracket\overline{\mathtt{N}}\rrbracket\rangle \lhd \mathcal{F}\llbracket\mathtt{N}\rrbracket\ \{$ $\mathcal{F}\llbracket\overline{\mathtt{T}}\rrbracket\ \overline{\mathtt{f}}; \mathcal{F}\llbracket\mathtt{K}\rrbracket\mathcal{F}\llbracket\overline{\mathtt{M}}\rrbracket\}$	with $\mathtt{L} \equiv \mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\ \{\overline{\mathtt{T}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\}$
2l. $\mathcal{F}\llbracket\mathtt{L}\rrbracket = \mathtt{interface}\ \mathtt{I}\langle\overline{\mathtt{X}} \lhd \mathcal{F}\llbracket\overline{\mathtt{N}}\rrbracket\rangle\{\mathcal{F}\llbracket\overline{\mathtt{H}}\rrbracket\}$	with $\mathtt{L} \equiv \mathtt{interface}\ \mathtt{I}\ \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\{\overline{\mathtt{H}}\}$
k. $\mathcal{F}\llbracket\mathtt{K}\rrbracket = \mathtt{C}(\mathcal{F}\llbracket\overline{\mathtt{T}}\rrbracket\ \overline{\mathtt{f}})\{\mathtt{super}(\overline{\mathtt{f}}); \mathtt{this}.\overline{\mathtt{f}} = \overline{\mathtt{f}};\}$	with $\mathtt{C}(\overline{\mathtt{T}}\ \overline{\mathtt{f}})\{\mathtt{super}(\overline{\mathtt{f}}); \mathtt{this}.\overline{\mathtt{f}} = \overline{\mathtt{f}};\}$
m. $\mathcal{F}\llbracket\mathtt{M}\rrbracket = \langle\overline{\mathtt{X}} \lhd \mathcal{F}\llbracket\overline{\mathtt{N}}\rrbracket\rangle\ \mathcal{F}\llbracket\,\mathtt{T}\,\rrbracket\,\mathtt{m}(\mathcal{F}\llbracket\,\overline{\mathtt{T}}\,\rrbracket\ \overline{\mathtt{x}})\{\mathcal{F}\llbracket\uparrow \mathtt{e}\rrbracket\}$	with $\mathtt{M} \equiv \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e}\}$
h. $\mathcal{F}\llbracket\mathtt{H}\rrbracket = \langle\overline{\mathtt{X}} \lhd \mathcal{F}\llbracket\overline{\mathtt{N}}\rrbracket\rangle\ \mathcal{F}\llbracket\,\mathtt{T}\,\rrbracket\,\mathtt{m}(\mathcal{F}\llbracket\,\overline{\mathtt{T}}\,\rrbracket\ \overline{\mathtt{x}})$	with $\mathtt{H} \equiv \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})$
1e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = \mathtt{x}$	with $\mathtt{e} \equiv \mathtt{x}$
2e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = \mathcal{F}\llbracket\mathtt{e}_0\rrbracket.\mathtt{f}$	with $\mathtt{e} \equiv \mathtt{e}_0.\mathtt{f}$
3e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = \mathcal{F}\llbracket\mathtt{e}_0\rrbracket.\mathtt{m}\langle\mathcal{F}\llbracket\overline{\mathtt{T}}\rrbracket\rangle(\mathcal{F}\llbracket\overline{\mathtt{e}}\rrbracket)$	with $\mathtt{e} \equiv \mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\overline{\mathtt{e}})$
4e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = \mathtt{new}\ \mathcal{F}\llbracket\mathtt{N}\rrbracket(\mathcal{F}\llbracket\overline{\mathtt{e}}\rrbracket)$	with $\mathtt{e} \equiv \mathtt{new}\ \mathtt{N}(\overline{\mathtt{e}})$
5e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = (\mathcal{F}\llbracket\mathtt{N}\rrbracket)(\mathcal{F}\llbracket\mathtt{e}_0\rrbracket)$	with $\mathtt{e} \equiv (\mathtt{N})\mathtt{e}_0$
6e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = \mathcal{F}\llbracket\mathtt{e}_0\rrbracket.\mathtt{invoke}(\mathcal{F}\llbracket\overline{\mathtt{e}}\rrbracket)$	with $\mathtt{e} \equiv \mathtt{e}_0!(\overline{\mathtt{e}})$
7e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = \mathtt{new}\ \mathtt{I\$n}\langle\mathcal{F}\llbracket\,\overline{\mathtt{T}}\,\rrbracket\mathcal{F}\llbracket\,\mathtt{T}\,\rrbracket\rangle()\{$ $\mathcal{F}\llbracket\mathtt{T}\rrbracket\ \mathtt{invoke}(\mathcal{F}\llbracket\overline{\mathtt{T}}\rrbracket\ \overline{\mathtt{x}})\{\uparrow \mathcal{F}\llbracket\mathtt{e}_0\rrbracket\}\}$	with $\mathtt{e} \equiv \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\mathtt{e}_0,\ \mathtt{n} = \lvert\overline{\mathtt{T}}\rvert + 1$ and $\Delta; \Gamma \vdash \mathtt{e} : \#\mathtt{T}(\overline{\mathtt{T}})$
8e. $\mathcal{F}\llbracket\mathtt{e}\rrbracket = \mathtt{new}\ \mathtt{I}\langle\mathcal{F}\llbracket\overline{\mathtt{T}}\rrbracket\rangle()\{\mathcal{F}\llbracket\overline{\mathtt{M}}\rrbracket\}$	with $\mathtt{e} \equiv \mathtt{new}\ \mathtt{I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}$
Translation Structures	
$\mathtt{interface}\ \mathrm{I\$n}\langle\overline{\mathtt{X}}, \mathtt{X} \lhd \overline{\mathtt{Object}}, \mathtt{Object}\ \rangle\{\mathtt{X}\ \mathtt{invoke}(\overline{\mathtt{X}}\ \overline{\mathtt{x}})\}$	*with* $\mathtt{n} = \lvert\overline{\mathtt{X}}, \mathtt{X}\rvert$

Eventually, let $\mathcal{F}\llbracket l_1\rrbracket$,..,$\mathcal{F}\llbracket l_k\rrbracket$ be all the translated forms occurring in L' (with k=0 when none occurs). Then, L'_Z is L' where $\mathcal{F}\llbracket l_i\rrbracket$ is replaced by l'_i, for each $i \in [1..k]$, and $Z_i \equiv l_i \ \rightarrow_{\mathcal{F}}\ l'_i$.

Translation $\mathcal{F}\llbracket\rrbracket$ assigns meaning to closures by mapping closures of FGACJ into method objects of FGAJ ($\mathcal{F}$-rule 7e), type closures of FGACJ into FGAJ interfaces for method objects ($\mathcal{F}$-rule 2t), and closure invocation of FGACJ into FGAJ invocations of the method wrapped in the method object associated to the closure ($\mathcal{F}$-rule 6e). The remaining $\mathcal{F}$-rules express a sort of congruence of the rules above and allow to apply such rules in each subterm of the translated FGACJ program.

Our final goal here is to prove that the two semantics (reduction and translation semantics) commute. This is expressed primarily by Theorem 6 and also by Theorem 5. For technical convenience, we extend $\mathcal{F}$ to the value variable environment Γ: for each Γ, $\mathcal{F}\llbracket\Gamma\rrbracket$ is the environment that for each variable $\mathtt{x}$, $\mathcal{F}\llbracket\Gamma\rrbracket(\mathtt{x}) = \mathcal{F}\llbracket\Gamma(\mathtt{x})\rrbracket$. Moreover, we write $\vdash_{\text{FGAJ}}$ if derivation $\vdash$ uses only rules of the calculus FGAJ. We write $\vdash_{\text{FGACJ}}$ if derivation uses, in addition, rules of the calculus FGCJ.

Theorem 5 (Typing Preservation). *Let* $\Delta; \Gamma \vdash_{\mathit{FGACJ}} \mathtt{e} : \mathtt{T}$. *Then,* $\Delta; \mathcal{F}\llbracket\Gamma\rrbracket \vdash_{\mathit{FGAJ}} \mathcal{F}\llbracket\mathtt{e}\rrbracket : \mathcal{F}\llbracket\mathtt{T}\rrbracket$. *Moreover, let P be any well typed* FGACJ *program, i.e.* $\mathtt{A}\ \mathtt{OK}$ *for each class and interface* $\mathtt{A}$ *of P. Then, $\mathcal{F}\llbracket P\rrbracket$ is a well typed program in* FGAJ.

Proof By induction on typing derivation and case analysis on the last rule used. We show only one most intriguing case GT-CLOSURE$_{\text{FGCJ}}$

$\mathtt{e} \equiv \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\ \mathtt{e}_0 \quad \mathtt{T} \equiv \#\mathtt{T}_0(\overline{\mathtt{T}})$

$\Delta_0 \vdash \overline{\mathtt{T}}\ \mathtt{ok}$ and $\Delta_0; \Gamma_0, \overline{\mathtt{x}}{:}\overline{\mathtt{T}}, \mathtt{this}{:}\#\mathtt{T}_0(\overline{\mathtt{T}}) \vdash \mathtt{e}_0 : \mathtt{T}_0$

By Lemma **1**, $\Delta_0 \vdash \mathcal{F}[\![\overline{\mathtt{T}}]\!]\ \mathtt{ok}$. By induction, $\Delta_0; \mathcal{F}[\![\Gamma_0]\!], \overline{\mathtt{x}} : \mathcal{F}[\![\overline{\mathtt{T}}]\!], \mathtt{this} : \mathcal{F}[\![\#\mathtt{T}_0(\overline{\mathtt{T}})]\!] \vdash \mathcal{F}[\![\mathtt{e}_0]\!] : \mathcal{F}[\![\mathtt{T}_0]\!]$, where, by $\mathcal{F}$-rule 2t, $\mathcal{F}[\![\#\mathtt{T}_0(\overline{\mathtt{T}})]\!]{=}\mathtt{I\$n}\langle\mathcal{F}[\![\overline{\mathtt{T}}]\!]\mathcal{F}[\![\mathtt{T}_0]\!]\rangle$. Using GT-ANONYM$_{\text{FGAJ}}$ and letting, $\Delta \equiv \Delta_0$, $\Gamma \equiv \Gamma_0$, $\mathtt{N} \equiv \underline{\mathtt{Object}}$ ($\overline{\mathtt{Y}} \equiv \circ \equiv \overline{\mathtt{P}}$ be the empty sequence), $\overline{\mathtt{V}} \equiv \mathcal{F}[\![\overline{\mathtt{T}}]\!]\mathcal{F}[\![\mathtt{T}_0]\!]$, $\mathtt{m} \equiv \mathtt{invoke}$, $\mathtt{I} \equiv \mathtt{I\$n}$ (as defined in **Table 6** - translation Structures), we obtain: $\Delta_0; \Gamma_0 \vdash \mathtt{M}\ \mathtt{OK\,IN}\ \mathtt{I}\langle\mathcal{F}[\![\overline{\mathtt{T}}]\!]\mathcal{F}[\![\mathtt{T}_0]\!]\rangle$,
for $\mathtt{M} \equiv \mathcal{F}[\![\mathtt{T}_0]\!]\ \mathtt{invoke}(\mathcal{F}[\![\overline{\mathtt{T}}]\!]\ \overline{\mathtt{x}})\{\uparrow \mathcal{F}[\![\mathtt{e}_0]\!];\}$. Using GT-ANONYMNEW$_{\text{FGAJ}}$ and letting, $\overline{\mathtt{M}} \equiv \mathtt{M}$, we obtain: $\Delta_0; \Gamma_0 \vdash \mathtt{new}\ \mathtt{I}\langle\mathcal{F}[\![\overline{\mathtt{T}}]\!]\mathcal{F}[\![\mathtt{T}_0]\!]\rangle()\{\overline{\mathtt{M}}\}{:}\ \mathtt{I}\langle\mathcal{F}[\![\overline{\mathtt{T}}]\!]\mathcal{F}[\![\mathtt{T}_0]\!]\rangle$
By $\mathcal{F}$-rule 7e, $\mathtt{new}\ \mathtt{I}\langle\mathcal{F}[\![\overline{\mathtt{T}}]\!]\mathcal{F}[\![\mathtt{T}_0]\!]\rangle()\{\overline{\mathtt{M}}\} = \mathcal{F}[\![\#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\ \mathtt{e}_0]\!]$, and by $\mathcal{F}$-rule 2t, $\mathtt{I}\langle\mathcal{F}[\![\overline{\mathtt{T}}]\!]\mathcal{F}[\![\mathtt{T}_0]\!]\rangle = \mathcal{F}[\![\#\mathtt{T}_0(\overline{\mathtt{T}})]\!]$: This concludes the case □

Theorem 6 (Execution Preservation). *Let* $\Delta; \Gamma \vdash_{FGACJ} \mathtt{e} : \mathtt{T}$ *and* $\mathtt{e} \rightarrow_{FGACJ} \mathtt{e}'$*. Then,* $\mathcal{F}[\![\mathtt{e}]\!] \rightarrow_{FGAJ} \mathcal{F}[\![\mathtt{e}']\!]$ □

Proof The proof is given by case analysis on the last rule used in the computation. Due to space restrictions, we show here only the GR-INV-CLOS$_{\text{FGCJ}}$ case which is most meaningful one:

$\mathtt{e} \equiv \#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}_0!(\overline{\mathrm{d}}) \quad \mathtt{e}' \equiv [\overline{\mathrm{d}}/\overline{\mathtt{x}}, \#(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\mathtt{e}_0/\mathtt{this}]\mathtt{e}_0$

Assuming: $\Delta; \Gamma \vdash_{\text{FGACJ}} \#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}_0!(\overline{\mathrm{d}}) : \mathtt{T}$, for some Δ, Γ and $\mathtt{T}$. Then:
By $\mathcal{F}$-rule 6e, $\mathcal{F}[\![\#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}_0!(\overline{\mathrm{d}})]\!] = \mathcal{F}[\![\#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}_0]\!].\mathtt{invoke}(\mathcal{F}[\![\overline{\mathrm{d}}]\!])$, and by $\mathcal{F}$-rule 7e,

$\mathcal{F}[\![\#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}_0]\!] = \mathtt{new}\ \mathtt{I\$n}\langle\mathcal{F}[\![\,\overline{\mathtt{T}}\,]\!]\mathcal{F}[\![\,\mathtt{T}'\,]\!]\rangle()\{\overline{\mathtt{M}}\}$

for $\mathtt{n} = |\overline{\mathtt{T}}| + 1$ and $\Delta; \Gamma \vdash \#(\overline{\mathtt{T}}\,\overline{\mathtt{x}})\mathtt{e}_0 : \#\mathtt{T}'(\overline{\mathtt{T}})$ and $\overline{\mathtt{M}} \equiv \mathcal{F}[\![\mathtt{T}']\!]\ \mathtt{invoke}(\mathcal{F}[\![\overline{\mathtt{T}}]\!]\ \overline{\mathtt{x}})\{\uparrow \mathcal{F}[\![\mathtt{e}_0]\!]\}$ (with $\mathtt{T}' \equiv \mathtt{T}$, since the assumption on the type of $\mathtt{e}$);
By MB-INTERFACE, letting $\mathtt{A}{\equiv}\ \mathtt{new}\ \mathtt{I\$n}\langle\mathcal{F}[\![\,\overline{\mathtt{T}}\,]\!]\mathcal{F}[\![\,\mathtt{T}\,]\!]\rangle()\{\overline{\mathtt{M}}\}$, we have

$mbody(\mathtt{invoke}, \mathtt{A}) = \overline{\mathtt{x}}.\mathcal{F}[\![\mathtt{e}_0]\!]$

By GR-INVK-ANONYM$_{\text{FGAJ}}$, on the transformed terms:

$\mathtt{A.invoke}(\mathcal{F}[\![\overline{\mathrm{d}}]\!]) \rightarrow [\mathcal{F}[\![\overline{\mathrm{d}}]\!]/\overline{\mathtt{x}}, \mathtt{A}/\mathtt{this}]\mathcal{F}[\![\mathtt{e}_0]\!]$

that we can apply since, by Theorem **7**, $\mathtt{A}$, $\mathcal{F}[\![\overline{\mathtt{T}}]\!]$, $\mathcal{F}[\![\mathtt{T}]\!]$, $\mathcal{F}[\![\overline{\mathrm{d}}]\!]$, $\mathcal{F}[\![\mathtt{e}_0]\!]$ are terms of FGAJ, and by Theorem **5**, are all well typed terms. It concludes the case. □

Theorem 7 ($\mathcal{F}[\![]\!]$ is Idempotent). *Let* $u \in$ FGACJ*. Then* $\mathcal{F}[\![\mathcal{F}[\![u]\!]]\!]{=}\mathcal{F}[\![u]\!]$ □

Lemma 1 ($\mathcal{F}[\![]\!]$ Preserves Types Structure). *(a) If* $\Delta \vdash \overline{\mathtt{T}} <: \overline{\mathtt{U}}$*, then* $\Delta \vdash \mathcal{F}[\![\overline{\mathtt{T}}]\!] <: \mathcal{F}[\![\overline{\mathtt{U}}]\!]$*. (b) If* $\Delta \vdash \overline{\mathtt{T}}\ \mathtt{ok}$*, then* $\Delta \vdash \mathcal{F}[\![\overline{\mathtt{T}}]\!]\ \mathtt{ok}$*.*

Example 1. Consider a program with two classes (of FGAJ, for simplicity):

```
class A {Object x; A (Object x){super(x); this.x=x;}}
class B {Object x; B (Object x){super(x); this.x=x;}}
```

Let $e \equiv$ `(#(B x)(new A(x)))!(new B(new Object()))` be an expression (of FGACJ) for the program.: Reducing e and then translating, elsewhere translating e and then reducing, yield the same term.

reduction $e \rightarrow_{\text{FGACJ}}$

$e \rightarrow_{GR-Inv-Clos_{\text{FGCJ}}}$ `[new B(new Object())/x] new A(x)`

$\equiv$ `new A(new B(new Object()))`

translation $\mathcal{F}[\![e]\!]$

$\mathcal{F}[\![$`#A(B x)(new A x)!(new B(new Object))`$]\!]$

$=\mathcal{F}[\![$`#A(B x)(new A x)`$]\!]$`.invoke(`$\mathcal{F}[\![$`new B(new Object())`$]\!]$
`=new` $I\$1$`<B,A>(){`$\mathcal{F}[\![$`A invoke(B x){new A(x)}`$]\!]$`.invoke(`$\mathcal{F}[\![$`new B(new Object())`$]\!]$`)`
`=new` $I\$1$`<B,A>(){`$\mathcal{F}[\![$`A invoke(B x){new A(x)}`$]\!]$`}.invoke(new B(new Object()))`
`=new` $I\$1$`<B,A>(){A invoke(B x){`$\uparrow$` new A(x)}}.invoke(new B(new Object ()))`
reduction $\mathcal{F}[\![e]\!] \rightarrow_{\texttt{FGAJ}}$
`new` $I\$1$`<B,A>(){A invoke(B x){`$\uparrow$` new A(x)}}`
`.invoke(new B(new Object ()))` $\rightarrow_{GR-Invk-Anonym_{\texttt{FGAJ}}}$
`[new B(new Object())/x]new A(x)`
$\equiv$ `new A(new B(new Object()))`

where, I$1 is the following interface: `interface I$1`$\langle$`T1,T`$\rangle$`{T invoke(T1 x);}`.

5 Conclusions

We proved that the reduction semantics defined for closures in the Straw-man proposal version and the translation semantics $\mathcal{F}[\![P]\!]$ commute preserving typing and computation. The translation semantics required interfaces and anonymous objects hence we have extended the minimal core calculus, FGCJ with such features, in a minimal version and have proved that the semantics properties, type safety and abstraction are also preserved. Open problems and interesting questions that deserve further investigation are closure conversion and contro-covariance. In [BO11b], we proved type soundness for contra-covariant S-closures, but since contraco-variance conflicts with interface subtyping properties of Java, it has been eliminated in this version.

References

[ABW01] Igarashi A., Pierce B., and P. Wadler. Featherweight Java: A Minimal Core Calculus for Java and GJ. *ACM TOPLAS*, 23:396–450, 2001.
[BGR10] A. Buckley, J. Gibbons, and M. Reinhold. *State of the Lambda.* Sun Microsystem, Inc., October 2010. http://cr.openjdk.java.net/~briangoetz/lambda/lambda-state-3.html.
[BO09] M. Bellia and M.E. Occhiuto. Java*Ω*: A Translation Semantics for Closures in Java. In *CS&P'2009*, pages 72–83. Warsaw University, 2009.
[BO10] M. Bellia and M.E. Occhiuto. Java*Ω*: Proving Type Safety for Java Simple Closures. In *CS&P'2010*, pages 61–72. Humboldt-Universitat zu Berlin, 2010.
[BO11a] M. Bellia and M.E. Occhiuto. *Java in Academia and Research*, chapter Java*Ω*: Higher Order Programming in Java, pages 166–185. iConcept Press Ltd., 2011.
[BO11b] M. Bellia and M.E. Occhiuto. Properties of Java Simple Closures. *Fundamenta Informaticae*, 110, 2011. To Appear (see: www.di.unipi.it/~occhiuto/JH).
[BO11c] M. Bellia and M.E. Occhiuto. Proving Translation and Reduction Semantics Equivalent for Java Simple Closures, Extended Version. Technical Report TR-11, University of Pisa, Dipartimento Informatica, 2011. (see: www.di.unipi.it/~occhiuto/JH).
[Goe07] B. Goetz. The Closures Debate: Should Closures be Added to the Java Language, and if so, How?, 2007. Java Theory and Practice, IBM Technical Library, www.ibm.com/developerworks/java/library/j-jtp04247.html.
[Rei09] M. Reinhold. Project Lambda: Straw-Man Proposal, 2009. //cr.openjdk.java.net/_ mr/lambda/straw-man/.
[Rei10] M. Reinhold. Project Lambda: Java Language Specificatiopn draft - Version 0.1, 2010. Sun Microsystems, http://mail.openjdk.java.net/.