Inference, Targeting and Compatibility in a Type System for Java with SAM Typed Closures

Marco Bellia and M. Eugenia Occhiuto

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

Abstract. In this paper we consider the new type structure that has been proposed for Java closures, in the last Java Specification Language [BS11a]. This structure uses SAM types that are in fact, nominal types instead of the structural, functional types, used in the previous, proposals. In addition, it allows type inference for closures and for closure arguments. Through a technique, already consolidated in previous studies in Java extensions, we extend the calculus FGJ, [IPW01], with interfaces, anonymous classes, closures of the new form and SAM types. We define a type system and a reduction semantics for this calculus FGATCJ. Using the type system, we formalize the notions of closure context, target type, compatibility type and closure type as they emerge in [BS11a]. Eventually, we prove the soundness of the type system.

1 Introduction

The problem of extending Java with closures is widely discussed in several papers and discussion forums (see [BO11,Ope12]). In this paper we prove type soundness for Java closures in the last proposal of JSR 000335 [BS11a,BS11b]. Such a proposal uses *SAM* types [BLB06] as types for closures, but shares with the previous proposals the idea of considering closures as a shortening for anonymous single method objects. Its main characteristics are:

Closure Definition To introduce a closure a special syntax is provided, without any explicit reference to the object type. A closure definition (*lambda expression*) requires specifying the closure argument names (if any), the generic types (if any) and the closure body; arguments types are optional.

Type Inference The closure definition does not specify the type of the defined closure and can omit argument types: The type system infers them all. In case of failure during the inference process, a type error occurs.

SAM Types Interface types with a single method, named *functional interfaces* (*SAM* types) are the types of the closures. As all Java reference types, *functional interfaces* are nominal types, i.e they are different types if they have different names even though they have the same structure. For this reason, a closure can be assigned to (i.e. is *compatible* with) many different types.

Generic types Generics, specified in the closure definition, instantiate the generic variables defined in the single method signature of a suitable functional interface. A problem arises: What is a suitable interface and what is its relation with the type of the closure?

Target Types The solution adopted is to assign to a closure the *target* type, that is the expected type in the specific *context* in which the closure is used. The target type becomes the type if it is compatible with the closure.

Closure Contexts The possible contexts in which a closure can appear are:

1. Variable declaration
2. Assignment
3. Return statement
4. Array initializer
5. Method or constructor argument
6. Lambda expression body
7. Conditional expression
8. Cast expression

Type Compatibility The conditions which must hold for a closure to be compatible with a type are: *i)* The type must be a functional interface: Let m be its single method. *ii)* Number and types of the closure arguments must be the same as those of m. *iii)* Return types of the closure and of m must be compatible. *iv)* Exceptions thrown by the closure body must be allowed in the `throws` clause of m.

Closure Invocation There is no ad hoc syntax for closure invocation. The user has to specify, hence know, the name of the single method of the functional interface.

Non-local Variables Any name used but not declared in the closure must be either declared `final` or *effectively final*. The concept of *effectively final* variables, already introduced in Java SE 7, is now broadened, to mitigate the restriction on variables updating. An effectively final variable is a variable which is not declared final but its value is not modified.

Variable Shadowing As for blocks the local variables or formal parameters of a closure cannot shadow already declared names.

Meaning of `this` The self reference `this` in a closure refers to the object whose method is enclosing the closure definition and not to the defined closure (`this` transparency), thus disallowing recursive definitions through `this`. To define a recursive closure, it's necessary to associate a name to the closure, for instance through variable declaration and initialization or assignment.

The new features of this proposal are mainly concerned with (1) SAM types, (2) target types and compatibility, (3) type inference of closure arguments. The challenge here is to understand if a calculus with such features is actually type safe and in the latter case which structures are necessary to provide a type safe semantics to closures. With this aim, we use the same approach we adopted for studying the previous proposals. We extend the calculus FGJ [IPW01] by defining a type system and a reduction semantics for interfaces and anonymous classes, FGAJ [BO12], and for this kind of closures (here called, Target Closure), obtaining the new calculus FGATCJ. Then, we prove the type soundness.

Section 2 contains a brief presentation of the aims and the main features of FGJ. Section 3 defines syntax and semantics of FGATCJ. Section 4 contains the main results to prove type soundness. Section 5 contains a small example that sketches how the type system is used to assign types to closures. Section 6 concludes the paper.

2 Featherweight Generic Java

A program in FGJ [IPW01] consists of a declaration of generic class definitions and of an expression to be evaluated using the classes. The expression corresponds to the body of the 0-arguments main method of ordinary Java.

A complete definition of the syntax of FGJ consists of the grammar rules in **Table 1** that are labelled by the defined grammatical category indexed by FGJ. Symbols $\lhd$ and $\uparrow$ are a notational shorthands for Java keyword `extends` and `return`. For syntactic regularity, (a) classes always specify the super class, possibly `Object`, and have exactly one constructor definition; (b) class constructors have one parameter for each class field with the same name as the field, invoke the super constructor on the fields of the super class and initialize the remaining fields to the corresponding parameters; (c) field access always specifies the receiver (object), possibly `this`. This results in the stylized form of the constructors. Both classes and methods may have generic type parameters.

FGJ has no side effects. Hence, *sequencing* and *assignment* are strictly confined to constructor bodies. In particular, method bodies have always the form `return`, followed by an expression. The lack of Java constructs for sequencing control and for store updating (along with that of concurrency, and reflection) is the main advantage of the calculus in studying language properties that are not affected by side effects. In this way the calculus is, as much as possible, compact and takes advantage of the referential transparency. The latter one provides a simple reduction semantics which is crucial for rigorous, easy to derive, proofs of the language properties [FF87]. About compactness, FGJ has only five forms of expressions (see definition of category `e` in **Table 1**): One for variables[1], another for *field access*, and one for *Object Creation*. The remaining two forms are *method invocation* and *cast*.

The presence of *cast* in FGJ is justified from its fundamental role in compiling generic classes. We conclude this presentation considering the twofold role of referential transparency: First, evaluation is entirely formalized within the syntax of FGJ (hence, the evaluation process results in a sequence of FGJ expressions reducing the first one to the last one, if any, which represents an error or its value) second, the order in which expressions are reduced, if more than one can be selected, does not affect the final result. The reduction semantics of FGJ consists of the first three rules that appear in **Table 2: Computation**, and deal with term evaluation, and of the first five rules in **Table 2: Congruence**, that deal with the redex selection. The remaining 20 rules of the semantics of FGJ deal with the type system and with term well-formedness. The rules of FGJ have labels that are indexed by FGJin **Table 2, 4, 5**.

3 Featherweight GATCJ

The calculus defined in this paper is obtained as an extension of the calculus FGAJ, which is in turn an extension of FGJ, with interfaces, anonymous classes and consequently objects from anonymous classes creation. FGATCJ extends FGAJ with the closures defined in [BS11a]. About the non-local variables in closures, there is no matter since all the variables of FGJ can be considered effectively final. Similarly, about variable shadowing, since FGJ programs are, in effect, abstract syntax terms.

3.1 Notation and General Conventions

In this paper we adopt the notation used in [IPW01], accordingly $\overline{\mathtt{f}}$ is a shorthand for a possibly empty sequence $\mathtt{f}_1, \ldots, \mathtt{f}_n$ (and similarly for $\overline{\mathtt{T}}, \overline{\mathtt{x}}$, etc.) and $\overline{\mathtt{M}}$ is a shorthand for

[1] Variables, indicated with category `x`, include parameters and `this`. From a syntactic point of view, `this` is a keyword in Java, and is a variable in FGJ but in both languages, it has the meaning of *object self-reference* and allows method recusion, see rule `GR-Invk`, **Table 2**.

$\mathtt{M}_1 \ldots \mathtt{M}_n$ (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, $\_ \to \_$, have lower precedence than other operators, and cast precedes closure definition. Hence $() \to (\mathtt{this.invoke}())$ can be written as $() \to \mathtt{this.invoke}()$. The, possibly indexed and/or primed, metavariables $\mathtt{T}$, $\mathtt{V}$, $\mathtt{U}$, $\mathtt{S}$, $\mathtt{W}$ range over type expressions; $\mathtt{T}$, $\mathtt{X}$, $\mathtt{Y}$, $\mathtt{Z}$ range over type variables; $\mathtt{N}$, $\mathtt{P}$, $\mathtt{Q}$ range over class types; $\mathtt{C}$, $\mathtt{D}$, $\mathtt{E}$ range over class names; $\mathtt{f}$, $\mathtt{g}$ range over field names; $\mathtt{e}$, $\mathtt{v}$, $\mathtt{d}$ range over expressions; $\mathtt{x}$, $\mathtt{y}$ range over variable names and $\mathtt{M}$, $\mathtt{K}$, $\mathtt{L}$ and $\mathtt{m}$ 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 free type variables in $\overline{\mathtt{T}}$.

3.2 Syntax

The syntax of FGATCJis reported in **Table 1**. Besides interface definition and anonymous classes already contained in FGAJand FGACJin [BO12], in FGATCJit is possible to define closures through *lambda expressions* as proposed in [BS11a][2]. Accordingly, syntax is extended only to construct a closure, for which two cases are considered: closures with and without argument types. A further extension to expressions is concerned with cast to interface type, which is maintained distinct from cast to class type, for technical convenience.

Table 1 : Syntax

FGJ	
$\mathtt{T}, \mathtt{V} ::= \mathtt{X} \mid \mathtt{N}$	$(\mathtt{T}_{\mathrm{FGJ}})$
$\mathtt{N} ::= \mathtt{C}\langle\overline{\mathtt{T}}\rangle$	$(\mathtt{N}_{\mathrm{FGJ}})$
$\mathtt{L} ::= \mathtt{class}\ \mathtt{C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\ \{\overline{\mathtt{T}}\ \overline{\mathtt{f}}; \mathtt{K}\ \overline{\mathtt{M}}\}$	$(\mathtt{L}_{\mathrm{FGJ}})$
$\mathtt{K} ::= \mathtt{C}(\overline{\mathtt{T}}\ \overline{\mathtt{f}})\{\mathtt{super}(\overline{\mathtt{f}}); \mathtt{this}.\overline{\mathtt{f}} = \overline{\mathtt{f}};\}$	$(\mathtt{K}_{\mathrm{FGJ}})$
$\mathtt{M} ::= \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\{\uparrow \mathtt{e};\}$	$(\mathtt{M}_{\mathrm{FGJ}})$
$\mathtt{e} ::= \mathtt{x} \mid \mathtt{e.f} \mid \mathtt{new}\ \mathtt{N}(\overline{\mathtt{e}}) \mid \mathtt{e.m}\langle\overline{\mathtt{T}}\rangle(\overline{\mathtt{e}}) \mid (\mathtt{N})\mathtt{e}$	$(\mathtt{e}_{\mathrm{FGJ}})$
IA: *Extensions for Interfaces and Anonymous Class Objects*	
$\mathtt{T} ::= \mathtt{I}\langle\overline{\mathtt{T}}\rangle$	$(\mathtt{T}_{\mathrm{FGAJ}})$
$\mathtt{L} ::= \mathtt{interface}\ \mathtt{I}\ \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\{\overline{\mathtt{H}}\}$	$(\mathtt{L}_{\mathrm{FGAJ}})$
$\mathtt{H} ::= \langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \mathtt{T}\ \mathtt{m}(\overline{\mathtt{T}}\ \overline{\mathtt{x}})$	$(\mathtt{H}_{\mathrm{FGAJ}})$
$\mathtt{e} ::= \mathtt{new}\ \mathtt{I}\langle\overline{\mathtt{T}}\rangle()\ \{\overline{\mathtt{M}}\}$	$(\mathtt{e}_{\mathrm{FGAJ}})$
TC: *Extensions for Targeted Closures*	
$\mathtt{e} ::= \mathtt{a} \mid (\mathtt{I}\langle\overline{\mathtt{T}}\rangle)\mathtt{e}$	$(\mathtt{e}_{\mathrm{FGATCJ}})$
$\mathtt{a} ::= \langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}}) \to \mathtt{e} \mid \langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{x}}) \to \mathtt{e}$	$(\mathtt{a}_{\mathrm{FGATCJ}})$
FGAJ = FGJ + IA FGATCJ = FGJ + IA + TC	

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

[2] The concrete syntax adopted in the actual proposal in [BS11a] is different from the one contained in **Table 1** which is to be considered an abstract syntax

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 and FGATCJ, have a label which is indexed by the name of the minimal calculus including the construct, involved in the rule. Note that $\mathtt{C}\langle\overline{\mathtt{T}}\rangle$ includes `Object`(since $\overline{\mathtt{T}}$ may be the empty sequence and `C` may be `Object`) hence generic variables in classes and methods can be instantiated with types `T` that include interfaces.

3.3 Semantics: Reduction

Table 2: Computation

Computation

$$\frac{\mathit{fields}(\mathtt{N}) = \overline{\mathtt{T}}\ \overline{\mathtt{f}}}{(\mathtt{new\ N}(\overline{\mathtt{e}})).\mathtt{f}_i \rightsquigarrow \mathtt{e}_i} \qquad (\text{GR-FIELD}_{\text{FGJ}})$$

$$\frac{\mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{N}) = \overline{\mathtt{x}}.\mathtt{e}}{(\mathtt{new\ N}(\overline{\mathtt{e}})).\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{d}}) \rightsquigarrow [\overline{\mathtt{d}}/\overline{\mathtt{x}}, \mathtt{new\ N}(\overline{\mathtt{e}})/\mathtt{this}]\mathtt{e}} \qquad (\text{GR-INVK}_{\text{FGJ}})$$

$$\frac{\emptyset \vdash \mathtt{N} <: \mathtt{P}}{(\mathtt{P})(\mathtt{new\ N}(\overline{\mathtt{e}})) \rightsquigarrow \mathtt{new\ N}(\overline{\mathtt{e}})} \qquad (\text{GR-CAST}_{\text{FGJ}})$$

$$\frac{\mathit{mbody}(\mathtt{m}\langle\overline{\mathtt{V}}\rangle, \mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}) = \overline{\mathtt{x}}.\mathtt{e}}{(\mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}).\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{d}}) \rightsquigarrow [\overline{\mathtt{d}}/\overline{\mathtt{x}}, \mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\}/\mathtt{this}]\mathtt{e}} \qquad (\text{GR-INVK-ANONYM}_{\text{FGAJ}})$$

$$(\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}}) \to \mathtt{e}).\mathtt{m}\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{d}}) \rightsquigarrow [\overline{\mathtt{d}}/\overline{\mathtt{x}}]\mathtt{e} \qquad (\text{GR-CLOS-INV-TYPE}_{\text{FGATCJ}})$$

$$(\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{x}}) \to \mathtt{e}).\mathtt{m}\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{d}}) \rightsquigarrow [\overline{\mathtt{d}}/\overline{\mathtt{x}}]\mathtt{e} \qquad (\text{GR-CLOS-INV}_{\text{FGATCJ}})$$

$$(\mathtt{I}\langle\overline{\mathtt{T}}\rangle)\ \mathtt{a} \rightsquigarrow \mathtt{a} \qquad (\text{GR-CCAST}_{\text{FGATCJ}})$$

Congruence

$$\frac{\mathtt{e}_0 \rightsquigarrow \mathtt{e}_0'}{\mathtt{e}_0.\mathtt{f} \rightsquigarrow \mathtt{e}_0'.\mathtt{f}} \qquad (\text{GRC-FIELD}_{\text{FGJ}})$$

$$\frac{\mathtt{e}_0 \rightsquigarrow \mathtt{e}_0'}{\mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\overline{\mathtt{e}}) \rightsquigarrow \mathtt{e}_0'.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\overline{\mathtt{e}})} \qquad (\text{GRC-T-INV}_{\text{FGJ}})$$

$$\frac{\mathtt{e}_i \rightsquigarrow \mathtt{e}_i'}{\mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\ldots,\mathtt{e}_i,\ldots) \rightsquigarrow \mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{T}}\rangle(\ldots,\mathtt{e}_i'\ldots)} \qquad (\text{GRC-INV-ARG}_{\text{FGJ}})$$

$$\frac{\mathtt{e}_i \rightsquigarrow \mathtt{e}_i'}{\mathtt{new\ N}(\ldots,\mathtt{e}_i,\ldots) \rightsquigarrow \mathtt{new\ N}(\ldots,\mathtt{e}_i',\ldots)} \qquad (\text{GRC-NEW}_{\text{FGJ}})$$

$$\frac{\mathtt{e} \rightsquigarrow \mathtt{e}'}{(\mathtt{N})\mathtt{e} \rightsquigarrow (\mathtt{N})\mathtt{e}'} \qquad (\text{GRC-CAST}_{\text{FGJ}})$$

$$\frac{\mathtt{e} \rightsquigarrow \mathtt{e}'}{(\mathtt{I}\langle\overline{\mathtt{T}}\rangle)\mathtt{e} \rightsquigarrow (\mathtt{I}\langle\overline{\mathtt{T}}\rangle)\mathtt{e}'} \qquad (\text{GRC-CCAST}_{\text{FGATCJ}})$$

The reduction semantics is given through the inference rules in **Table 2**, which define the reduction relation $\mathtt{e} \rightsquigarrow \mathtt{e}'$ that says that "expression `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 FGATCJ values are objects, constructed out of anonymous or named classes, and of closures. Hence the grammatical category `v` defines the syntactic form of the values of the calculus FGATCJ:

$$\begin{aligned}\mathtt{v} ::= {} & \mathtt{new\ N}(\overline{\mathtt{v}}) \\ | {} & \mathtt{new\ I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\} \\ | {} & \langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{x}}) \to \mathtt{e} \\ | {} & \langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}}) \to \mathtt{e}\end{aligned}$$

The structure of values results from the reduction rules of the calculus. The rules indexed by FGJ in **Table 2** are the same as those of calculus FGJ introduced in [IPW01], and the one indexed by FGAJ is the same as those of the calculus FGAJ introduced in [BO12], which include only one new rule, GR-Invk-Anonym$_{\text{FGAJ}}$. This rule defines the semantics of invocation with anonymous class objects, quite similar to the one of method

Table 3: Classes and Interfaces

Subclassing

$$\texttt{C} \trianglelefteq \texttt{C} \qquad \frac{\texttt{C} \trianglelefteq \texttt{D} \qquad \texttt{D} \trianglelefteq \texttt{E}}{\texttt{C} \trianglelefteq \texttt{E}} \qquad \frac{\texttt{class C}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle \lhd \texttt{D}\ \{\overline{\texttt{S}}\ \overline{\texttt{f}}; \texttt{K}\ \overline{\texttt{M}}\}}{\texttt{C} \trianglelefteq \texttt{D}}$$

Auxiliary functions

$$fields(\texttt{Object}) = \circ \qquad \text{(F-Object)}$$

$$\frac{\texttt{class C}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle \lhd \texttt{N}\ \{\overline{\texttt{S}}\ \overline{\texttt{f}}; \texttt{K}\ \overline{\texttt{M}}\} \qquad fields([\overline{\texttt{T}}/\overline{\texttt{X}}]\texttt{N}) = \overline{\texttt{U}}\ \overline{\texttt{g}}}{fields(\texttt{C}\langle \overline{\texttt{T}} \rangle) = \overline{\texttt{U}}\ \overline{\texttt{g}}, [\overline{\texttt{T}}/\overline{\texttt{X}}]\overline{\texttt{S}}\ \overline{\texttt{f}}} \qquad \text{(F-Class)}$$

$$\frac{\texttt{class C}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle \lhd \texttt{N}\ \{\overline{\texttt{S}}\ \overline{\texttt{f}}; \texttt{K}\ \overline{\texttt{M}}\} \qquad \langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \texttt{U m}\ (\overline{\texttt{U}}\ \overline{\texttt{x}})\{\uparrow \texttt{e};\} \in \overline{\texttt{M}}}{mtype(\texttt{m}, \texttt{C}\langle \overline{\texttt{T}} \rangle) = [\overline{\texttt{T}}/\overline{\texttt{X}}](\langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \overline{\texttt{U}} \mapsto \texttt{U})} \qquad \text{(MT-Class)}$$

$$\frac{\texttt{class C}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle \lhd \texttt{N}\ \{\overline{\texttt{S}}\ \overline{\texttt{f}}; \texttt{K}\ \overline{\texttt{M}}\} \qquad \texttt{m} \notin \overline{\texttt{M}}}{mtype(\texttt{m}, \texttt{C}\langle \overline{\texttt{T}} \rangle) = mtype(\texttt{m}, [\overline{\texttt{T}}/\overline{\texttt{X}}]\texttt{N})} \qquad \text{(MT-Super)}$$

$$\frac{\texttt{interface I}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle\ \{\overline{\texttt{H}}\} \qquad \langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \texttt{U m}(\overline{\texttt{U}}\ \overline{\texttt{x}}) \in \overline{\texttt{H}}}{mtype(\texttt{m}, \texttt{I}\langle \overline{\texttt{T}} \rangle) = [\overline{\texttt{T}}/\overline{\texttt{X}}](\langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \overline{\texttt{U}} \mapsto \texttt{U})} \qquad \text{(MT-Interface)}$$

$$\frac{\texttt{class C}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle \lhd \texttt{N}\ \{\overline{\texttt{S}}\ \overline{\texttt{f}}; \texttt{K}\ \overline{\texttt{M}}\} \qquad \langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \texttt{U m}\ (\overline{\texttt{U}}\ \overline{\texttt{x}})\{\uparrow \texttt{e};\} \in \overline{\texttt{M}}}{mbody(\texttt{m}\langle \overline{\texttt{V}} \rangle, \texttt{C}\langle \overline{\texttt{T}} \rangle) = \overline{\texttt{x}}.[\overline{\texttt{T}}/\overline{\texttt{X}}, \overline{\texttt{V}}/\overline{\texttt{Y}}]\texttt{e}} \qquad \text{(MB-Class)}$$

$$\frac{\texttt{class C}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle \lhd \texttt{N}\ \{\overline{\texttt{S}}\ \overline{\texttt{f}}; \texttt{K}\ \overline{\texttt{M}}\} \qquad \texttt{m} \notin \overline{\texttt{M}}}{mbody(\texttt{m}\langle \overline{\texttt{V}} \rangle, \texttt{C}\langle \overline{\texttt{T}} \rangle) = mbody(\texttt{m}\langle \overline{\texttt{V}} \rangle, [\overline{\texttt{T}}/\overline{\texttt{X}}]\texttt{N})} \qquad \text{(MB-Super)}$$

$$\frac{\texttt{interface I}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle\ \{\ldots\} \qquad \langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \texttt{U m}\ (\overline{\texttt{U}}\ \overline{\texttt{x}})\{\uparrow \texttt{e};\} \in \overline{\texttt{M}}}{mbody(\texttt{m}\langle \overline{\texttt{V}} \rangle, \texttt{new I}\langle \overline{\texttt{T}} \rangle()\{\overline{\texttt{M}}\}) = \overline{\texttt{x}}.[\overline{\texttt{T}}/\overline{\texttt{X}}, \overline{\texttt{V}}/\overline{\texttt{Y}}]\texttt{e}} \qquad \text{(MB-Interface)}$$

$$\frac{\texttt{interface I}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle\ \{\overline{\texttt{H}}\} \qquad |\overline{\texttt{H}}| = 1 \qquad \Delta \vdash \overline{\texttt{V}} <: [\overline{\texttt{V}}/\overline{\texttt{X}}]\overline{\texttt{N}} \qquad \langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \texttt{U m}\ (\overline{\texttt{U}}\ \overline{\texttt{x}}) = \overline{\texttt{H}}}{\Delta \vdash met(\texttt{I}\langle \overline{\texttt{V}} \rangle) = \langle \overline{\texttt{Y}} \lhd [\overline{\texttt{V}}/\overline{\texttt{X}}]\overline{\texttt{P}} \rangle [\overline{\texttt{V}}/\overline{\texttt{X}}]\texttt{U m}\ ([\overline{\texttt{V}}/\overline{\texttt{X}}]\overline{\texttt{U}}\ \overline{\texttt{x}})} \qquad \text{(Method)}$$

Auxiliary predicates

$$\texttt{override}(\texttt{m}, \texttt{Object}, \langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \overline{\texttt{T}} \mapsto \texttt{T}_0) \qquad \text{(Over-Object)}$$

$$\frac{mtype(\texttt{m}, \texttt{N}) = \langle \overline{\texttt{Z}} \lhd \overline{\texttt{Q}} \rangle \overline{\texttt{U}} \mapsto \texttt{U}_0 \text{ implies } ((\overline{\texttt{P}}, \overline{\texttt{T}}) = [\overline{\texttt{Y}}/\overline{\texttt{Z}}](\overline{\texttt{Q}}, \overline{\texttt{U}}) \text{ and } \overline{\texttt{Y}} <: \overline{\texttt{P}} \vdash \texttt{T}_0 <: [\overline{\texttt{Y}}/\overline{\texttt{Z}}]\texttt{U}_0)}{\texttt{override}(\texttt{m}, \texttt{N}, \langle \overline{\texttt{Y}} \lhd \overline{\texttt{P}} \rangle \overline{\texttt{T}} \mapsto \texttt{T}_0)} \qquad \text{(Over)}$$

$$\texttt{isClos}(\langle \overline{\texttt{V}} \rangle (\overline{\texttt{x}}) \to \texttt{e}) \qquad \text{(IsClosure)}$$

$$\texttt{isClos}(\langle \overline{\texttt{V}} \rangle (\overline{\texttt{T}}\ \overline{\texttt{x}}) \to \texttt{e}) \qquad \text{(IsClosureTyped)}$$

$$\frac{\texttt{interface I}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle\ \{\overline{\texttt{H}}\} \qquad \Delta \vdash \overline{\texttt{V}} <: [\overline{\texttt{V}}/\overline{\texttt{X}}]\overline{\texttt{N}} \qquad |\overline{\texttt{H}}| = 1}{\Delta \vdash \texttt{Fun}(\texttt{I}\langle \overline{\texttt{V}} \rangle)} \qquad \text{(Fun)}$$

DCast

$$\frac{dcast(\texttt{C}, \texttt{D}) \qquad dcast(\texttt{D}, \texttt{E})}{dcast(\texttt{C}, \texttt{E})} \qquad \frac{\texttt{class C}\langle \overline{\texttt{X}} \lhd \overline{\texttt{N}} \rangle \lhd \texttt{D}\langle \overline{\texttt{T}} \rangle\ \{\ldots\} \qquad \overline{\texttt{X}} = FV(\overline{\texttt{T}})}{dcast(\texttt{C}, \texttt{D})} \qquad \text{(DCast)}$$

invocation with object of named classes. The new rules indexed FGATCJinclude method invocation on closure values, which can be defined specifying arguments types or not, and a rule to cast closures to interface types (targeting). Also one congruence rule for casting is added. As a matter of fact, we could generalize FGJcasting rules but we prefer

here to maintain the division to point out semantics of the new constructs. Eventually, note that no congruence rule has been added for closures this means that syntactically different closures are different, in particular, $\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}}) \to \mathtt{e}$, $\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}}) \to \mathtt{e}'$ are different values even though $\mathtt{e} \rightsquigarrow \mathtt{e}'$.

3.4 Semantics: Typing

The typing rules use three different kinds of environment, Δ (for type variables), Γ (for variables), and Ψ (for closures), and six 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 $\mathtt{X}{<:}\mathtt{T}$ (with at most one binding for each type variable $\mathtt{X}$), 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. $\mathtt{X} \notin dom(\Delta)$). An environment Γ is a mapping from variables to types written as a list of $\mathtt{x} : \mathtt{T}$ (with at most one binding for each value variable $\mathtt{x}$), meaning that "$\mathtt{x}$ has type $\mathtt{T}$". An environment Ψ is a mapping from closure definitions to target (SAM) types written as a list of $a \Downarrow \mathtt{T}$ (with, at most, one binding for each closure a), meaning that closure a has target type $\mathtt{T}$. 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 judgement for sub-typing (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 4b**) has the form $\mathtt{C}\ \mathtt{OK}$ meaning that "$\mathtt{C}$ is well typed". The judgement for class methods (see GT-METHOD$_{\text{FGJ}}$ in **Table 4b**) has the form $\mathtt{M}\ \mathtt{OK}\ \mathtt{IN}\ \mathtt{C}$ meaning that "$\mathtt{M}$ is well typed when its declaration occurs in class $\mathtt{C}$". For methods in instances of anonymous classes, the judgement is the same as for class methods but the inference has to consider that these methods are defined in, runtime evaluated, interface instantiations, and they must have the same signature that has been declared in the corresponding interfaces (see GT-ANONYM$_{\text{FGAJ}}$ in **Table 4b**). The judgement for expressions (see the rules of **Table 4**) has the form $\Psi; \Delta; \Gamma \vdash \mathtt{e} : \mathtt{T}$ meaning that expression $\mathtt{e}$ has type $\mathtt{T}$ in the context of environments Ψ, Δ and Γ. The judgment for SAM types (see the rules of **Table 4a**) has the form $\Psi; \Delta; \Gamma \vdash\ a \downarrow \mathtt{T}$ meaning that closure a is compatible with SAM type $\mathtt{T}$ in the context of environments Ψ, Δ and Γ. The typing rules are contained in **Table 4** and extends those of FGJ [IPW01], and those of FGAJ [BO12], but relevant extensions have been introduced: A new environment Ψ has been added to the (semantic) context of the rules; the sixth judgment $\downarrow$ has been added, together with the corresponding inference rules (of **Table 4a**), to deal with closure type compatibility.

Environment Ψ. The requirement for a third, additional, environment, Ψ, is one of the most interesting and innovative aspect of the type system. In effect, as we noted in Section 1, a closure has, possibly several, different, compatible SAM types in each of the specific contexts in which it can occur in a program. Each compatible type of a closure, in a context, can furnish a type to the closure, but the target type of the context is actually chosen for typing the closure, provided it is one of the closure compatible types.[3] Hence, the environment Ψ is used to store, in each context of the program, the binding of each closure, used in such a context, to its target.

[3] This is an effect of the use of SAM types as the type closure domain of the calculus, and SAM types are nominal types. In case of structural types, as functional types [BO12], a closure should have only one compatible type and this type must be a supertype of the target type

Target, Compatible and Closure Type. Closure typing involves three different types. The relation among such types is fully expressed by rule GT-CLOSURE$_{\text{FGATCJ}}$, in **Table 4**. The rule says that a closure $\mathtt{a}$ has type $\mathtt{T}$ in a context $\mathcal{C} \equiv \Psi; \Delta; \Gamma$, if $\mathtt{a}$ is an instance of a closure definition $\mathtt{a}_c$, i.e. $\mathtt{a} \equiv [\overline{\mathtt{d}}/\overline{\mathtt{x}}]\mathtt{a}_c$ for a terms substitution $[\overline{\mathtt{d}}/\overline{\mathtt{x}}]$, and in context $\mathcal{C}$: $\mathtt{a}_c$ has target type $\mathtt{T}$, i.e. $\Psi \equiv \Psi_1, \mathtt{a}_c \Downarrow \mathtt{T}$ and $\mathtt{T}$ is a compatible type of $\mathtt{a}_c$.

Growth of Ψ. The combined use of compatible and target type leads to the definition of inference rules, to assign types to closures, which appear a little strange in the way they use the environment Ψ. All these rules have Ψ that increases in the conclusion of the rule. This is the case of rules GT-INV$_{\text{FGJ}}$, GT-NEW$_{\text{FGJ}}$, GT-CLOSURE$_{\text{FGAJ}}$ in **Table 4**, of rules COMPATIBILITYCB$_{\text{FGATCJ}}$, COMPATIBILITYTCB$_{\text{FGATCJ}}$ in **Table**

Table 4: Typing Rules

$$\Psi; \Delta; \Gamma_1, \mathtt{x} : \mathtt{T}, \Gamma_2 \vdash \mathtt{x} : \mathtt{T} \qquad \text{(GT-VAR}_{\text{FGJ}}\text{)}$$

$$\frac{\Psi; \Delta; \Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \mathit{fields}(\mathit{bound}_\Delta(\mathtt{T}_0)) = \overline{\mathtt{T}}\ \overline{\mathtt{f}}}{\Psi; \Delta; \Gamma \vdash \mathtt{e}_0.\mathtt{f}_i : \mathtt{T}_i} \qquad \text{(GT-FIELD}_{\text{FGJ}}\text{)}$$

$$\frac{\begin{array}{c}\mathit{mtype}(\mathtt{m}, \mathit{bound}_\Delta(\mathtt{T}_0)) = \langle \overline{\mathtt{Y}} \lhd \overline{\mathtt{P}} \rangle \overline{\mathtt{U}} \mapsto \mathtt{U} \\ \Psi; \Delta; \Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \overline{\mathtt{V}}\ \mathtt{ok} \qquad \Delta \vdash \overline{\mathtt{V}} <: [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}} \\ \mathtt{isClos}(\mathtt{e}_i) \qquad \Psi; \Delta; \Gamma \vdash \mathtt{e}_i \downarrow [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}_i \\ \Psi, \mathtt{e}_i \Downarrow [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}_i; \Delta; \Gamma \vdash \overline{\mathtt{e}} : \overline{\mathtt{S}} \qquad \Delta \vdash \overline{\mathtt{S}} <: [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{U}}\end{array}}{\Psi, \mathtt{e}_i \Downarrow [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}_i; \Delta; \Gamma \vdash \mathtt{e}_0.\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{e}}) : [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}} \qquad \text{(GT-INV}_{\text{FGJ}}\text{)}$$

$$\frac{\begin{array}{c}\Delta \vdash \mathtt{N}\ \mathtt{ok} \qquad \mathit{fields}(\mathtt{N}) = \overline{\mathtt{T}}\ \overline{\mathtt{f}} \qquad \mathtt{isClos}(\mathtt{e}_i) \\ \Psi; \Delta; \Gamma \vdash \mathtt{e}_i \downarrow \mathtt{T}_i \qquad \Psi, \mathtt{e}_i \Downarrow \mathtt{T}_i; \Delta; \Gamma \vdash \overline{\mathtt{e}} : \overline{\mathtt{S}} \qquad \Delta \vdash \overline{\mathtt{S}} <: \overline{\mathtt{T}}\end{array}}{\Psi, \mathtt{e}_i \Downarrow \mathtt{T}_i; \Delta; \Gamma \vdash \mathtt{new}\ \mathtt{N}(\overline{\mathtt{e}}) : \mathtt{N}} \qquad \text{(GT-NEW}_{\text{FGJ}}\text{)}$$

$$\frac{\Psi; \Delta; \Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \mathit{bound}_\Delta(\mathtt{T}_0) <: \mathtt{N}}{\Psi; \Delta; \Gamma \vdash (\mathtt{N})\mathtt{e}_0 : \mathtt{N}} \qquad \text{(GT-UCAST}_{\text{FGJ}}\text{)}$$

$$\frac{\begin{array}{c}\Psi; \Delta; \Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \mathtt{N}\ \mathtt{ok} \qquad \Delta \vdash \mathtt{N} <: \mathit{bound}_\Delta(\mathtt{T}_0) \\ \mathtt{N} = \mathtt{C}\langle\overline{\mathtt{T}}\rangle \qquad \mathit{bound}_\Delta(\mathtt{T}_0) = \mathtt{D}\langle\overline{\mathtt{U}}\rangle \qquad \mathit{dcast}(\mathtt{C}, \mathtt{D})\end{array}}{\Psi; \Delta; \Gamma \vdash (\mathtt{N})\mathtt{e}_0 : \mathtt{N}} \qquad \text{(GT-DCAST}_{\text{FGJ}}\text{)}$$

$$\frac{\begin{array}{c}\Psi; \Delta; \Gamma \vdash \mathtt{e}_0 : \mathtt{T}_0 \qquad \Delta \vdash \mathtt{N}\ \mathtt{ok} \\ \mathtt{N} = \mathtt{C}\langle\overline{\mathtt{T}}\rangle \qquad \mathit{bound}_\Delta(\mathtt{T}_0) = \mathtt{D}\langle\overline{\mathtt{U}}\rangle \qquad \mathtt{C} \not<: \mathtt{D} \qquad \mathtt{D} \not<: \mathtt{C}\end{array}}{\Psi; \Delta; \Gamma \vdash (\mathtt{N})\mathtt{e}_0 : \mathtt{N}} \qquad \text{(GT-SCAST}_{\text{FGJ}}\text{)}$$

$$\frac{\Delta \vdash \mathtt{I}\langle\overline{\mathtt{T}}\rangle\ \mathtt{ok} \qquad \Psi; \Delta; \Gamma \vdash \overline{\mathtt{M}}\ \ \mathtt{OK\ IN}\ \ \mathtt{I}\langle\overline{\mathtt{T}}\rangle}{\Psi; \Delta; \Gamma \vdash \mathtt{new}\ \mathtt{I}\langle\overline{\mathtt{T}}\rangle()\{\overline{\mathtt{M}}\} : \mathtt{I}\langle\overline{\mathtt{T}}\rangle} \qquad \text{(GT-ANONYMNEW}_{\text{FGAJ}}\text{)}$$

$$\frac{\Psi; \Delta; \Gamma \vdash \overline{\mathtt{d}} : \overline{\mathtt{S}} \qquad \Delta \vdash \overline{\mathtt{S}} <: \overline{\mathtt{T}} \qquad \Psi; \Delta; \Gamma, \overline{\mathtt{x}} : \overline{\mathtt{T}} \vdash \mathtt{a} \downarrow \mathtt{T}}{\Psi, \mathtt{a} \Downarrow \mathtt{T}; \Delta; \Gamma \vdash [\overline{\mathtt{d}}/\overline{\mathtt{x}}]\mathtt{a} : \mathtt{T}} \qquad \text{(GT-CLOSURE}_{\text{FGATCJ}}\text{)}$$

$$\frac{\Psi; \Delta; \Gamma \vdash \mathtt{a} : \mathtt{I}\langle\overline{\mathtt{T}}\rangle}{\Psi; \Delta; \Gamma \vdash (\mathtt{I}\langle\overline{\mathtt{T}}\rangle)\mathtt{a} : \mathtt{I}\langle\overline{\mathtt{T}}\rangle} \qquad \text{(GT-CCAST}_{\text{FGATCJ}}\text{)}$$

4.a and of rule GT-ANONYMCB$_{\text{FGAJ}}$ in **Table 4.b**. Ψ is extended only if the closure is compatible with the target type.

Closure Contexts The typing rules of closures take into account the context in which the closure appears in the program. In Section1 we recall that eight are the contexts that JSR 000335 [BS11a,BS11b] admits for closures. The minimal core calculus FGATCJreduces

Table 4a: Typing Rule - target compatibility

Closure compatibility

$$\frac{\begin{array}{c}\Delta \vdash \mathtt{T}{<:}\mathtt{I}\langle\overline{\mathtt{V}}\rangle \qquad \Delta \vdash \mathtt{Fun}(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) \qquad \mathtt{isClos(e)} \\ \Delta \vdash met(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) = \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{U}\ \mathtt{m}(\overline{\mathtt{S}}\ \overline{\mathtt{w}}) \qquad \Delta \vdash \overline{\mathtt{W}}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}} \\ \Psi; \Delta; \Gamma, \overline{\mathtt{x}} : [\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{S}} \vdash \mathtt{e} \downarrow [\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\mathtt{U}\end{array}}{\Psi, \mathtt{e} \Downarrow [\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\mathtt{U}; \Delta; \Gamma \vdash \langle\overline{\mathtt{W}}\rangle(\overline{\mathtt{x}}) \to \mathtt{e} \downarrow \mathtt{T}} \quad (\textsc{CompatibilityCB}_{\text{FGATCJ}})$$

$$\frac{\begin{array}{c}\Delta \vdash \mathtt{T}{<:}\mathtt{I}\langle\overline{\mathtt{V}}\rangle \qquad \Delta \vdash \mathtt{Fun}(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) \qquad \sim \mathtt{isClos(e)} \\ \Delta \vdash met(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) = \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{U}\ \mathtt{m}(\overline{\mathtt{S}}\ \overline{\mathtt{w}}) \qquad \Delta \vdash \overline{\mathtt{W}}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}} \\ \Psi; \Delta; \Gamma, \overline{\mathtt{x}} : [\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{S}} \vdash \mathtt{e}\ : \mathtt{Z} \qquad \Delta \vdash \mathtt{Z}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\mathtt{U}\end{array}}{\Psi; \Delta; \Gamma \vdash \langle\overline{\mathtt{W}}\rangle(\overline{\mathtt{x}}) \to \mathtt{e} \downarrow \mathtt{T}} \quad (\textsc{CompatibilityF}_{\text{FGATCJ}})$$

$$\frac{\begin{array}{c}\Delta \vdash \mathtt{T}{<:}\mathtt{I}\langle\overline{\mathtt{V}}\rangle \qquad \Delta \vdash \mathtt{Fun}(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) \qquad \mathtt{isClos(e)} \\ \Delta \vdash met(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) = \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{U}\ \mathtt{m}(\overline{\mathtt{S}}\ \overline{\mathtt{w}}) \qquad \Delta \vdash \overline{\mathtt{W}}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}} \\ \Delta \vdash \overline{\mathtt{T}}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{S}} \qquad \Psi; \Delta; \Gamma, \overline{\mathtt{x}} : \overline{\mathtt{T}} \vdash \mathtt{e}\ \downarrow [\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\mathtt{U}\end{array}}{\Psi, \mathtt{e}\ \Downarrow [\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\mathtt{U}; \Delta; \Gamma \vdash \langle\overline{\mathtt{W}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}}) \to \mathtt{e} \downarrow \mathtt{T}} \quad (\textsc{CompatibilityTCB}_{\text{FGATCJ}})$$

$$\frac{\begin{array}{c}\Delta \vdash \mathtt{T}{<:}\mathtt{I}\langle\overline{\mathtt{V}}\rangle \qquad \Delta \vdash \mathtt{Fun}(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) \qquad \sim \mathtt{isClos(e)} \\ \Delta \vdash met(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) = \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\mathtt{U}\ \mathtt{m}(\overline{\mathtt{S}}\ \overline{\mathtt{w}}) \qquad \Delta \vdash \overline{\mathtt{W}}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}} \\ \Delta \vdash \overline{\mathtt{T}}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\overline{\mathtt{S}} \qquad \Psi; \Delta; \Gamma, \overline{\mathtt{x}} : \overline{\mathtt{T}} \vdash \mathtt{e}\ : \mathtt{Z} \qquad \Delta \vdash \mathtt{Z}{<:}[\overline{\mathtt{W}}/\overline{\mathtt{Y}}]\mathtt{U}\end{array}}{\Psi; \Delta; \Gamma \vdash \langle\overline{\mathtt{W}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}}) \to \mathtt{e} \downarrow \mathtt{T}} \quad (\textsc{CompatibilityT}_{\text{FGATCJ}})$$

Table 4b: Typing Rules

Classes, Interfaces, Methods

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

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

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

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

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

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

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

Table 5: Subtypes

Subtypes

$$bound_{\Delta}(\mathtt{X}) = \Delta(\mathtt{X}) \qquad \text{(B-Var}_{\text{FGJ}}\text{)}$$

$$bound_{\Delta}(\mathtt{N}) = \mathtt{N} \qquad \text{(B-Class}_{\text{FGJ}}\text{)}$$

$$bound_{\Delta}(\mathtt{I}\langle\overline{\mathtt{V}}\rangle) = \mathtt{I}\langle\overline{\mathtt{V}}\rangle \qquad \text{(B-Interface}_{\text{FGJ}}\text{)}$$

$$\Delta \vdash \mathtt{T} <: \mathtt{T} \qquad \text{(S-Refl}_{\text{FGJ}}\text{)}$$

$$\frac{\Delta \vdash \mathtt{S} <: \mathtt{T} \qquad \Delta \vdash \mathtt{T} <: \mathtt{U}}{\Delta \vdash \mathtt{S} <: \mathtt{U}} \qquad \text{(S-Trans}_{\text{FGJ}}\text{)}$$

$$\Delta \vdash \mathtt{X} <: \Delta(\mathtt{X}) \qquad \text{(S-Var}_{\text{FGJ}}\text{)}$$

$$\frac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\{\ldots\}}{\Delta \vdash \mathtt{C}\langle\overline{\mathtt{T}}\rangle <: [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\mathtt{N}} \qquad \text{(S-Class}_{\text{FGJ}}\text{)}$$

Well-formed types

$$\Delta \vdash \text{Object ok} \qquad \text{(WF-Object}_{\text{FGJ}}\text{)}$$

$$\frac{\mathtt{X} \in dom(\Delta)}{\Delta \vdash \mathtt{X\ ok}} \qquad \text{(WF-Var}_{\text{FGJ}}\text{)}$$

$$\frac{\mathtt{class\ C}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle \lhd \mathtt{N}\{\ldots\} \qquad \Delta \vdash \overline{\mathtt{T}}\ \mathtt{ok} \qquad \Delta \vdash \overline{\mathtt{T}} <: [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\overline{\mathtt{N}}}{\Delta \vdash \mathtt{C}\langle\overline{\mathtt{T}}\rangle\ \mathtt{ok}} \qquad \text{(WF-Class}_{\text{FGJ}}\text{)}$$

$$\frac{\mathtt{interface\ I}\langle\overline{\mathtt{X}} \lhd \overline{\mathtt{N}}\rangle\{\ldots\} \qquad \Delta \vdash \overline{\mathtt{T}}\ \mathtt{ok} \qquad \Delta \vdash \overline{\mathtt{T}} <: [\overline{\mathtt{T}}/\overline{\mathtt{X}}]\overline{\mathtt{N}}}{\Delta \vdash \mathtt{I}\langle\overline{\mathtt{T}}\rangle\ \mathtt{ok}} \qquad \text{(WF-Interf}_{\text{FGAJ}}\text{)}$$

them to the following four: 3. Return statement; 5. Method or constructor argument; 6. Lambda expression body; 8. Cast expression. Each of these contexts expresses a target type for the closure that occurs in it. Hence the typing rules include rules that check that such a type is one of the compatible types of the closure. In particular, in **Table 4**, rule GT-Inv$_{\text{FGJ}}$ deals also with closures that are argument of method (i.e. context 5). Rule GT-New$_{\text{FGJ}}$ deals also with closures that are argument of constructor (i.e. context 5). In **Table 4.a**, rules CompatibilityCB$_{\text{FGATCJ}}$ and CompatibilityTCB$_{\text{FGATCJ}}$ deal with closures that are the body of another closure (i.e. context 7). In **Table 4.b**, rule GT-MethodCB$_{\text{FGJ}}$ deals with closures that are the return expression of a method (i.e. context 3). Rule GT-AnonymCB$_{\text{FGATCJ}}$ deals with closures that are the return expression of an interface instance method (i.e. context 3).

4 Properties

Semantics is useful to prove language properties. In this paper we consider the soundness of the type system for closures with SAM types. Analogously to [IPW01], we prove the subject reduction theorem and the progress theorem first, the type soundness immediately follows. For space problems, we give the statements of all the three but only a sketched proof of the first two theorems. Several interesting lemmas are used in the complete theorem proofs, we omit them at all, but we give the statements of two lemmas that are used in the sketched proofs.

Theorem 1 (Subject reduction). *If* $\Psi; \Delta; \Gamma \vdash \mathtt{e} : \mathtt{T}$ *and* $\mathtt{e} \rightsquigarrow \mathtt{e}'$ *then* $\Psi; \Delta; \Gamma \vdash \mathtt{e}' : \mathtt{T}'$, *for some* $\mathtt{T}'$ *such that* $\Delta \vdash \mathtt{T}' <: \mathtt{T}$

Proof. By induction on the reduction $\mathtt{e} \rightsquigarrow \mathtt{e}'$, with a case analysis on the reduction rule used. The proof of the corresponding theorem for FGJ (pp. 426-428, [IPW01]), must be extended with the following cases GR-Clos-Inv, GR-Clos-Inv-Type and GR-CCast. The first two are quite similar, the third one is trivial. We show GR-Clos-Inv: $\mathtt{e} = \langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{x}})\to\mathtt{e}_b.\mathtt{m}\langle\overline{\mathtt{V}}\rangle(\overline{\mathtt{e}})$ $\quad \mathtt{e}' = [\overline{\mathtt{e}}/\overline{\mathtt{x}}]\mathtt{e}_b$ By rule GT-Inv

$\Psi \equiv \Psi_1, \mathtt{e}_i \Downarrow [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}_i$ $\quad {}^{(1)}\ \Psi_1; \Delta; \Gamma \vdash \langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{x}})\to\mathtt{e}_b : \mathtt{T}_a$ $\quad \Delta \vdash \overline{\mathtt{V}}{<:}[\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{P}}$

$mtype(\mathtt{m}, bound_\Delta(\mathtt{T}_a)) = \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\ \overline{\mathtt{U}}\to\mathtt{U}$ $\quad \Delta \vdash \overline{\mathtt{V}}\ \mathtt{ok}$ $\quad \mathtt{isClos}(\mathtt{e}_i)$

$\Psi_1; \Delta, \Gamma \vdash \mathtt{e}_i \downarrow [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}_i$ $\quad \Psi_1; \Delta; \Gamma \vdash \overline{\mathtt{e}} : \overline{\mathtt{T}}$ $\quad \Delta \vdash \overline{\mathtt{T}}{<:}[\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{U}}$ $\quad \mathtt{T} \equiv [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}$

From (1), by GT-Closure, ${}^{(2)}\Psi_2; \Delta; \Gamma_e \vdash \mathtt{e}_c \downarrow \mathtt{T}_a$ for $\Psi_1 = \Psi_2, \mathtt{e}_c \Downarrow \mathtt{T}_a$ and $\Gamma_e \equiv \Gamma, \overline{\mathtt{y}} : \overline{\mathtt{T}}_y$ and ${}^{(3)}[\overline{\mathtt{d}}/\overline{\mathtt{y}}]\mathtt{e}_c \equiv \langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{x}})\to\mathtt{e}_b$. From (2), by Lemma 3 we have $\Psi_2; \Delta; \Gamma \vdash [\overline{\mathtt{d}}/\overline{\mathtt{x}}]\mathtt{e}_c \downarrow \mathtt{T}_a$ and since (3), $\Psi_2; \Delta; \Gamma \vdash \langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{x}})\to\mathtt{e}_b \downarrow \mathtt{T}_a$. We should consider the other three rules in **Table 4a**, but we consider only the case $\sim \mathtt{isClos}(\mathtt{e}_b)$. By Compatibility

$\Delta \vdash \mathtt{T}_a{<:}\mathtt{I}_a\langle\overline{\mathtt{V}}_a\rangle$ $\quad \Delta \vdash met(\mathtt{I}_a\langle\overline{\mathtt{V}}_a\rangle) = \langle\overline{\mathtt{Y}} \lhd \overline{\mathtt{P}}\rangle\ \mathtt{U}\ \mathtt{m}(\overline{\mathtt{U}}\ \overline{\mathtt{w}})$

$\Delta \vdash \mathtt{Fun}(\mathtt{I}_a(\overline{\mathtt{V}}_a))$ $\quad {}^{(4)}\ \Psi_2; \Delta; \Gamma, \overline{\mathtt{x}} : [\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\overline{\mathtt{U}} \vdash \mathtt{e}_b : \mathtt{Z}$ $\quad \Delta \vdash \mathtt{Z}{<:}[\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}$

From (4), by Lemma 2, $\Psi_2; \Delta; \Gamma \vdash [\overline{\mathtt{e}}/\overline{\mathtt{x}}]\mathtt{e}_b : \mathtt{Z}'$ for $\Delta \vdash \mathtt{Z}'{<:}\mathtt{Z}$ and by S-Trans, $\Delta \vdash \mathtt{Z}'{<:}[\overline{\mathtt{V}}/\overline{\mathtt{Y}}]\mathtt{U}$. Eventually by Lemma 1, $\Psi; \Delta; \Gamma \vdash [\overline{\mathtt{e}}/\overline{\mathtt{x}}]\mathtt{e}_b : \mathtt{Z}'$. Letting $\mathtt{T}' = \mathtt{Z}'$ finishes the case. □

Theorem 2 (Progress). *Suppose* $\mathtt{e}$ *is well-type. If* $\mathtt{e}$ *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. $(\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\to\mathtt{e}_0).\mathtt{m}\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{d}})$ *then* $|\overline{\mathtt{x}}| = |\overline{\mathtt{d}}| = |\overline{\mathtt{T}}|$ *for some* $\overline{\mathtt{T}}$*,* $\overline{\mathtt{x}}$ *and* $\mathtt{e}_0$*.*
4. $(\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{x}})\to\mathtt{e}_0).\mathtt{m}\langle\overline{\mathtt{S}}\rangle(\overline{\mathtt{d}})$ *then* $|\overline{\mathtt{x}}| = |\overline{\mathtt{d}}|$ *for some* $\overline{\mathtt{x}}$ *and* $\mathtt{e}_0$*.*

Proof. The proof is based on the analysis of all well typed expressions, which can be reduced to the above 4 cases, to conclude that either it is in normal form or it can be further reduced to obtain a normal form (see section 3.3) □

Theorem 3 (Type Soundness). *If* $\Psi; \emptyset; \emptyset \vdash \mathtt{e} : \mathtt{T}$ *and* $\mathtt{e} \rightsquigarrow^* \mathtt{e}'$ *with* $\mathtt{e}'$ *a normal form, then* $\mathtt{e}'$ *is a value* $\mathtt{v}$ *with* $\Psi; \emptyset; \emptyset \vdash \mathtt{v} : \mathtt{S}$ *and* $\emptyset \vdash \mathtt{S}{<:}\mathtt{T}$

Lemma 1. *Suppose* $\Delta, \overline{\mathtt{X}}{<:}\overline{\mathtt{N}} \vdash \overline{\mathtt{N}}\ \mathtt{ok}$ *and* $\Delta \vdash \mathtt{T}, \overline{\mathtt{T}}\ \mathtt{ok}$

1. *If* $\Delta \vdash \mathtt{S}{<:}\mathtt{T}$*, then* $\Delta, \overline{\mathtt{X}}{<:}\overline{\mathtt{N}} \vdash \mathtt{S}{<:}\mathtt{T}$
2. *If* $\Delta \vdash \mathtt{S}\ \mathtt{ok}$*, then* $\Delta, \overline{\mathtt{X}}{<:}\overline{\mathtt{N}} \vdash \mathtt{S}\ \mathtt{ok}$
3. *If* $\Psi; \Delta; \Gamma \vdash \mathtt{e} : \mathtt{T}$*, then* $\Psi; \Delta; \Gamma, \overline{\mathtt{x}} : \overline{\mathtt{T}} \vdash \mathtt{e} : \mathtt{T}$ *and* $\Psi; \Delta, \overline{\mathtt{X}}{<:}\overline{\mathtt{N}}; \Gamma \vdash \mathtt{e} : \mathtt{T}$ *and* $\Psi, \overline{\mathtt{e}} \Downarrow \overline{\mathtt{T}}; \Delta; \Gamma \vdash \mathtt{e} : \mathtt{T}$ □

Lemma 2. *If* $\Psi; \Delta; \Gamma, \overline{\mathtt{x}} : \overline{\mathtt{T}} \vdash \mathtt{e} : \mathtt{T}$ *and* $\Psi; \Delta; \Gamma \vdash \overline{\mathtt{d}} : \overline{\mathtt{S}}$ *where* $\Delta \vdash \overline{\mathtt{S}}{<:}\overline{\mathtt{T}}$*, then* $\Psi; \Delta; \Gamma \vdash [\overline{\mathtt{d}}/\overline{\mathtt{x}}]\mathtt{e} : \mathtt{S}$ *for some* $\mathtt{S}$ *such that* $\Delta \vdash \mathtt{S}{<:}\mathtt{T}$ □

Lemma 3. *Let* $\mathtt{e} \equiv \langle\overline{\mathtt{W}}\rangle(\overline{\mathtt{x}})\to\mathtt{e}_b$ *or* $\mathtt{e} \equiv \langle\overline{\mathtt{W}}\rangle(\overline{\mathtt{T}}\ \overline{\mathtt{x}})\to\mathtt{e}_b$*. If* $\Psi; \Delta; \Gamma, \overline{\mathtt{y}} : \overline{\mathtt{T}}_y \vdash \mathtt{e} \downarrow \mathtt{T}$ *and* $\Psi; \Delta; \Gamma \vdash \overline{\mathtt{d}} : \overline{\mathtt{S}}_y$ *for* $\Delta \vdash \overline{\mathtt{S}}_y{<:}\overline{\mathtt{T}}_y$ *then* $\Psi; \Delta; \Gamma \vdash [\overline{\mathtt{d}}/\overline{\mathtt{y}}]\mathtt{e} \downarrow \mathtt{T}$ □

5 Example

Consider a program with a class Integer (whose definition is omitted, with two object: 2 and 3) and three interfaces that define three SAM types. In particular, $\mathtt{I}_0$ and $\mathtt{I}_2$ introduce two different SAM types and are such that for each closure a (and any context $\Psi; \Delta; \Gamma$) we have $(\Psi; \Delta; \Gamma \vdash)a \downarrow \mathtt{I}_0$ if and only if $(\Psi; \Delta; \Gamma \vdash)a \downarrow \mathtt{I}_2$. This is due to the nominal nature of the SAM types.

```
interface I0{Integer invoke()}; interface I1{I0 invoke(Integer x)};
interface I2{Integer invoke()}
```

Consider the following expression, $((\mathtt{I}_1)((\mathtt{x}) \to () \to 3)).\mathtt{invoke}(2)$, and if and how a (unique) type can be assigned to it. We use the type system to show that a (principal) type $\mathtt{T}$ and an environment Ψ exist: $\Psi;\emptyset;\emptyset \vdash ((\mathtt{I}_1)((\mathtt{x}) \to () \to 3)).\mathtt{invoke}(2) : \mathtt{T}$.
The computation is sketched below.

$\Psi;\emptyset;\emptyset \vdash ((\mathtt{I}_1)((\mathtt{x}) \to () \to 3)).\mathtt{invoke}(2) : \mathtt{T}$ by GT-Inv
- $\mathtt{e}_0 = (\mathtt{I}_1)((\mathtt{x}) \to () \to 3)$ $\quad \mathtt{T} = \mathtt{I}_1$
- $\Psi;\emptyset;\emptyset \vdash (\mathtt{I}_1)((\mathtt{x}) \to () \to 3) : \mathtt{I}_1$ by GT-CCast
  - $\Psi;\emptyset;\emptyset \vdash (\mathtt{x}) \to () \to 3 : \mathtt{I}_1$ by GT-Closure
    - $\overline{\mathtt{d}} \equiv \overline{\mathtt{x}}$ $\quad \Gamma \equiv \emptyset$
    - $\Psi_1;\emptyset;\emptyset \vdash (\mathtt{x}) \to () \to 3 \downarrow \mathtt{I}_1$ by CompatibilityCB
      - $\emptyset \vdash \mathtt{I}_1 <: \mathtt{I}_1$ $\quad \mathtt{isClos}(() \to 3)$ $\quad \emptyset \vdash \mathtt{Fun}(\mathtt{I})$
      - $\emptyset \vdash \mathit{met}(\mathtt{I}_1) = \mathtt{I}_0\ \mathtt{invoke(Integer\ x)}$
      - $\Psi_2,\emptyset;\emptyset;\vdash () \to 3 \downarrow \mathtt{I}_0$ by Compatibility
        - $\emptyset \vdash \mathtt{I}_0 <: \mathtt{I}_0$ $\quad \sim \mathtt{isClos}(3)$ $\quad \emptyset \vdash \mathtt{Fun}(\mathtt{I}_0)$
        - $\emptyset \vdash \mathit{met}(\mathtt{I}_0) = \mathtt{Integer\ invoke()}$
        - $\Psi_2,\emptyset;\emptyset;\vdash 3 : \mathtt{Integer}$
    - $\Psi_1 = \Psi_2, () \to 3 \Downarrow \mathtt{I}_0$ $\quad \Psi_2 = \emptyset$
  - $\Psi = \Psi_1, (x) \to () \to 3 \Downarrow \mathtt{I}_1$
- $\mathit{mtype}(\mathtt{invoke}, \mathtt{I}_1) = \mathtt{Integer} \to \mathtt{I}_0$
- $\Psi;\emptyset;\emptyset \vdash 2 : \mathtt{Integer}$
- $\Psi;\emptyset;\emptyset \vdash (\mathtt{I}_1)((x) \to () \to 3).\mathtt{invoke}(2) : \mathtt{I}_0$

Hence $\Psi = (x) \to () \to 3 \Downarrow \mathtt{I}_1, () \to 3 \Downarrow \mathtt{I}_0$, $\mathtt{e}$ reduces to $() \to 3$ and
$(x) \to () \to 3 \Downarrow \mathtt{I}_1, () \to 3 \Downarrow \mathtt{I}_0; \emptyset;\emptyset \vdash () \to 3 : \mathtt{I}_0$

6 Conclusions

We have provided a type system and a reduction semantics for closures with SAM types and type inference on closures and closure arguments [BS11a]. We proved the soundness of the type system. This shows that we can use nominal types instead of structural, functional types. The resulting system however, is not so easy if compared to the one defined for simple closure [BO12].

References

[BLB06] D. Lea B. Lee and J. Bloch. Concise Instance Creation Expressions: Closure without Complexity, 2006. crazybob.org/2006/10/java-closure-spectrum.html.

[BO11] M. Bellia and M.E. Occhiuto. *Java in Academia and Research*, chapter JavaΩ: Higher Order Programming in Java, pages 166–185. iConcept Press Ltd., 2011.

[BO12] M. Bellia and M.E. Occhiuto. The equivalence of Reduction and Translation Semantics of Java Simple Closures. *Fundamenta Informaticae*, 119:1–16, 2012.

[BS11a] A. Buckley and D. Smith. *JSR-000335 Lambda Expressions for the Java Programming Language - Early Draft Review: Lambda Specification, Version 0.4.2.* Oracle Corporation, December 2011. http://download.oracle.com/otndocs/jcp/lambda-0_4_2-edr-spec/index.html.

[BS11b] A. Buckley and D. Smith. *State of the Lambda.* Oracle Corporation, December 2011. http://cr.openjdk.java.net/~briangoetz/lambda/lambda-state-4.html.

[FF87] Matthias Felleisen and Daniel P. Friedman. A Reduction Semantics for Imperative Higher-Order Languages. In *PARLE (2)*, pages 206–223, 1987.

[IPW01] A. Igarashi, B. Pierce, and P. Wadler. Featherweight Java: A Minimal Core Calculus for Java and GJ. *ACM TOPLAS*, 23:396–450, 2001.

[Ope12] OpenJDK. Project lambda, 2012. http://openjdk.java.net/projects/lambda/.