KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First cycle, 15 credits

A Type System for Ensuring Safe, Structured Concurrency in Scala

FAKE A. STUDENT
FAKE B. STUDENT

Stockholm, Sweden, 2024

A Type System for Ensuring Safe, Structured Concurrency in Scala

FAKE A. STUDENT

FAKE B. STUDENT

Bachelor’s Programme in Information and Communication Technology
Date: March 8, 2024

Supervisors: A. Busy Supervisor, Another Busy Supervisor, Third Busy Supervisor
Examiner: Gerald Q. Maguire Jr.
School of Electrical Engineering and Computer Science
Host company: Företaget AB
Swedish title: Ett typsystem för säker och strukturerad

Contents

Chapter 1

Introduction

Chapter 2

Background

2.1 Type Systems

2.2 Structured Concurrency

2.3 Concurrent Determinism

Chapter 3

Related Work

3.1 LaCasa

3.2 DPJ

3.3 Rust

3.4 Deterministic Concurrency Using Lattices

Chapter 4

Proposed Extension

4.1 Overview

4.2 Formalization

Chapter 5

Properties

5.1 Progress

5.2 Preservation

5.3 Confluence

Chapter 6

Conclusion

6.1 Future Work

Chapter 7

Appendix

7.1 Proofs

7.2 Inference Rules

7.2.1 Extension

7.2.1.1 Typing

$$\text{T-ASYNC}\ \frac{\begin{array}{cc} Perm[Q] \in \Gamma & \Gamma \setminus Perm[Q]; a \vdash s : \sigma \\ \Gamma; a \vdash b : Q \rhd Box[C] & x : C; ocap \vdash t : \tau \end{array}}{\Gamma; a \vdash async(b, x \Rightarrow t)\{s\} : \bot}$$

$$\text{T-FINISH}\ \frac{\Gamma; a \vdash t : \tau}{\Gamma; a \vdash finish\{t\} : null}$$

7.2.1.2 Evaluation

$$\text{E-ASYNC}\ \frac{\begin{array}{cc} T_1 = (f, false, \langle [x \to o],\ t,\ \emptyset\rangle^{\epsilon}) & L(b) = b(o, p) \\ T_2 = (f, true, \langle L,\ s,\ P \setminus \{p\}\rangle^{\epsilon}) & p \in P \end{array}}{\begin{array}{l} H,\{(f, k, \langle L,\ async(b, x \Rightarrow t)\{s\},\ P\rangle^{l} \circ FS)\} \uplus TS \\ \rightsquigarrow\ H,\{T_1, T_2\} \uplus TS \end{array}}$$

$$\text{E-FINISH1}\ \frac{T = (f', true, \langle L,\ t,\ P\rangle^{\epsilon}) \qquad f' fresh}{\begin{array}{l} H,\{(f, k, \langle L,\ let\ x = finish\{\ t\ \}\ in\ s,\ P\rangle^{l} \circ FS)\} \uplus TS \\ \rightsquigarrow\ H,\{(f, k, \langle FINISH\, f'\rangle^{m} \circ \langle L,\ s,\ P\rangle^{l} \circ FS)\} \uplus \{T\} \uplus TS \end{array}}$$

$$\text{E-FINISH2}\ \frac{\nexists (f', b', FS) \in TS}{\begin{array}{c} H, \{(f, k, \langle \mathit{FINISH}\, f' \rangle^{l} \circ \langle L,\ t,\ P \rangle^{l} \circ FS)\} \uplus TS \\ \rightsquigarrow\ H, \{(f, k, \langle L[l \to \mathit{null}],\ t,\ P \rangle^{l} \circ FS)\} \uplus TS \end{array}}$$

$$\text{E-TASK-DONE}\ \frac{}{H, \{(f, k, \epsilon)\} \uplus TS \rightsquigarrow TS}$$

7.2.2 LaCasa

7.2.2.1 Well-Formedness

$$\text{WF-VAR}\ \frac{\begin{array}{c} L(x) = \mathit{null} \vee \\ L(x) = o \wedge \mathit{typeof}(H, o) <: \Gamma(x) \vee \\ L(x) = b(o, p) \wedge \Gamma(x) = Q \rhd \mathit{Box}[C] \wedge \mathit{typeof}(H, o) <: C \end{array}}{H \vdash \Gamma; L; x}$$

$$\text{WF-PERM}\ \frac{\begin{array}{c} \gamma : permTypes(\Gamma) \longrightarrow Pinjective \\ \forall x \in dom(\Gamma). \\ (\Gamma(x) = Q \rhd \mathit{Box}[C] \wedge L(x) = b(o, p) \wedge Perm[Q] \in \Gamma) \\ \Longrightarrow \gamma(Q) = p \end{array}}{\vdash \Gamma; L; P}$$

$$\text{WF-ENV}\ \frac{\begin{array}{c} dom(\Gamma) \subseteq dom(L) \\ \forall x \in dom(\Gamma). H \vdash \Gamma; L; x \end{array}}{H \vdash \Gamma; L}$$

$$\text{WF-METHOD1}\ \frac{\Gamma_0, this : C, x : D; \epsilon \vdash t : E' \qquad E' <: E}{C \vdash defm(x : D) : E = t}$$

$$\text{WF-METHOD2}\ \frac{\Gamma = \Gamma_0, this : C, x : Q \rhd \mathit{Box}[D], Perm[Q] \qquad Qfresh \qquad \Gamma; \epsilon \vdash t : E' \qquad E' <}{C \vdash defm(x : \mathit{Box}[D]) : E = t}$$

$$\text{WF-PROGRAM}\ \frac{p \vdash \bar{cd} \qquad p \vdash \Gamma_0 \qquad \Gamma_0; \epsilon \vdash t : \sigma}{p \vdash \bar{cd}\bar{vd}t}$$

$$\text{WF-CLASS}\ \frac{\begin{array}{c} C \vdash \bar{md} \qquad D = \mathit{AnyRef} \vee p \vdash classD... \\ \forall (defm...) \in \bar{md}.\mathit{override}(m, C, D) \\ \forall varf : \sigma \in \bar{fd}.f \notin fields(D) \end{array}}{p \vdash classCextendsD\{\bar{fd}\bar{md}\}}$$

$$\text{WF-OVERRIDE}\ \frac{mtype(m, D) notdefined \vee mtype(m, D) = mtype(m, C)}{\mathit{override}(m, C, D)}$$

7.2.2.2 Typing

$$\text{T-NULL}\;\frac{}{\Gamma; a \vdash null : Null}$$

——

$$\text{T-VAR}\;\frac{x \in dom(\Gamma)}{\Gamma; a \vdash x : \Gamma(x)}$$

——

$$\text{T-LET}\;\frac{\Gamma; a \vdash e : \tau \qquad \Gamma, x : \tau; a \vdash t : \sigma}{\Gamma; a \vdash let x = e in t : \sigma}$$

——

$$\text{T-SELECT}\;\frac{\Gamma; a \vdash x : C \qquad ftype(C, f) = D}{\Gamma; a \vdash x.f : D}$$

——

$$\text{T-ASSIGN}\;\frac{\begin{array}{cc}\Gamma; a \vdash x : C & ftype(C, f) = D \\ \Gamma; a \vdash y : D' & D' <: D\end{array}}{\Gamma; a \vdash x.f = y : D}$$

——

$$\text{T-INVOKE}\;\frac{\begin{array}{cc}\Gamma; a \vdash x : C & mtype(C, m) = \sigma \to \tau \\ \Gamma; a \vdash y : \sigma' & \sigma' <: \sigma \vee \\ \multicolumn{2}{c}{(\sigma = Box[D] \wedge \sigma' = Q \rhd Box[D] \wedge Perm[Q] \in \Gamma)}\end{array}}{\Gamma; a \vdash x.m(y) : \tau}$$

——

$$\text{T-NEW}\;\frac{a = ocap \Longrightarrow ocap(C) \qquad \forall var f : \sigma \in \bar{fd}. \exists D. \sigma = D}{\Gamma; a \vdash new C : C}$$

——

$$\text{T-OPEN}\;\frac{\Gamma; a \vdash x : Q \rhd Box[C] \qquad Perm[Q] \in \Gamma \qquad y : C; ocap \vdash t : \sigma}{\Gamma; a \vdash x.open\{y \Rightarrow t\} : Q \rhd Box[C]}$$

——

$$\text{T-BOX}\;\frac{ocap(C) \qquad Q fresh \qquad \Gamma; x : Q \rhd Box[C]; Perm[Q]; a \vdash t : \sigma}{\Gamma; a \vdash box[C]\{x \Rightarrow t\} : \bot}$$

——

$$\text{T-CAPTURE}\;\frac{\begin{array}{cc}\Gamma; a \vdash x : Q \rhd Box[C] & \Gamma; a \vdash y : Q' \rhd Box[D] \\ \{Perm[Q], Perm[Q']\} \subseteq \Gamma & D <: ftype(C, f) \\ \multicolumn{2}{c}{\Gamma \; \{Perm[Q']\}, z : Q \rhd Box[C]; a \vdash t : \sigma}\end{array}}{\Gamma; a \vdash capture(x.f, y)\{z \Rightarrow t\} : \bot}$$

——

$$\text{T-SWAP}\;\frac{\begin{array}{cc}\Gamma; a \vdash x : Q \rhd Box[C] & \Gamma; a \vdash y : Q' \rhd Box[D'] \\ \{Perm[Q], Perm[Q']\} \subseteq \Gamma & ftype(C, f) = Box[D] \\ D' <: D & R fresh \\ \multicolumn{2}{c}{\Gamma \; \{Perm[Q']\}, z : R \rhd Box[D], Perm[R]; a \vdash t : \sigma}\end{array}}{\Gamma; a \vdash swap(x.f, y)\{z \Rightarrow t\} : \bot}$$

——

$$\text{T-EMPFS}\;\frac{}{H \vdash \epsilon}$$

$$\text{T-FRAME1}\ \frac{\Gamma; a \vdash t : \sigma \qquad l \neq \epsilon \implies \sigma <: C \qquad H \vdash \Gamma; L \qquad \vdash \Gamma; L; P}{H \vdash \langle L,\ t,\ P\rangle^l : \sigma}$$

previously: $H \vdash \Gamma; L; P$ but that rule doesn't take a heap.

$$\text{T-FRAME2}\ \frac{\Gamma; x : \tau; a \vdash t : \sigma \qquad l \neq \epsilon \implies \sigma <: C \qquad H \vdash \Gamma; L \qquad H \vdash \Gamma; L; P}{H \vdash^{\tau}_{x} \langle L,\ t,\ P\rangle^l : \sigma}$$

$$\text{T-FRAME-NA}\ \frac{H \vdash F^{\epsilon} : \sigma \qquad H \vdash FS}{H \vdash F^{\epsilon} \circ FS}$$

$$\text{T-FRAME-NA2}\ \frac{H \vdash^{\tau}_{x} F^{\epsilon} : \sigma \qquad H \vdash FS}{H \vdash^{\tau}_{x} F^{\epsilon} \circ FS}$$

$$\text{T-FRAME-A}\ \frac{H \vdash F^{x} : \sigma \qquad H \vdash^{\sigma}_{x} FS}{H \vdash F^{x} \circ FS}$$

$$\text{T-FRAME-A2}\ \frac{H \vdash^{\tau}_{x} F^{y} : \sigma \qquad H \vdash^{\sigma}_{y} FS}{H \vdash^{\tau}_{x} F^{y} \circ FS}$$

$$\text{T-TS}\ \frac{\forall (f, k, FS) \in TS. H \vdash FS}{H \vdash TS}$$

7.2.2.3 Evaluation

$$\text{E-NULL}\ \frac{}{H, \langle L,\ let x = null int,\ P\rangle^l \to H, \langle L[x \to null],\ t,\ P\rangle^l}$$

$$\text{E-VAR}\ \frac{}{H, \langle L,\ let x = y int,\ P\rangle^l \to H, \langle L[x \to L(y)],\ t,\ P\rangle^l}$$

$$\text{E-SELECT}\ \frac{H(L(y)) = \langle C, FM\rangle \qquad f \in dom(FM)}{H, \langle L,\ let x = y.f int,\ P\rangle^l \to H, \langle L[x \to FM(f)],\ t,\ P\rangle^l}$$

$$\text{E-ASSIGN}\ \frac{L(y) = o \qquad H(o) = \langle C, FM\rangle \qquad H' = H[o \to \langle C, FM[f \to L(z)]]}{H, \langle L,\ let x = y.f = z int,\ P\rangle^l \to H', \langle L,\ let x = z int,\ P\rangle^l}$$

$$\text{E-NEW}\ \frac{o \notin dom(H) \qquad fields(C) = \bar{f} \qquad H' = H[o \to \langle C, f \to null\rangle]}{H,\langle L,\ let x = new C in t,\ P\rangle^{l} \to H',\langle L[x \to o],\ t,\ P\rangle^{l}}$$

$$\text{E-INVOKE}\ \frac{\begin{array}{c} H(L(y)) = \langle C, FM\rangle \qquad mbody(C,m) = x \to t' \\ \neg ocap(C) \Rightarrow L' = L_0[this \to L(y), x \to L(z)] \\ ocap(C) \Rightarrow L' = [this \to L(y), x \to L(z)] \\ P' = \emptyset \vee (L(z) = b(o,p) \wedge p \in P \wedge P' = \{p\}) \end{array}}{H,\langle L,\ let x = y.m(z) in t,\ P\rangle^{l} \circ FS \twoheadrightarrow H,\langle L',\ t',\ P'\rangle^{x} \circ \langle L,\ t,\ P\rangle^{l} \circ FS}$$

$$\text{E-RETURN1}\ \frac{}{H,\langle L,\ x,\ P\rangle^{y} \circ \langle L',\ t',\ P'\rangle^{l} \to H,\langle L'[y \to L(x)],\ t',\ P'\rangle^{l}}$$

$$\text{E-RETURN2}\ \frac{}{H,\langle L,\ x,\ P\rangle^{\epsilon} \circ \langle L',\ t',\ P'\rangle^{l} \to H,\langle L',\ t',\ P'\rangle^{l}}$$

$$\text{E-OPEN}\ \frac{L(y) = b(o,p) \qquad p \in P \qquad L' = [z \to o]}{H,\langle L,\ let x = y.open\{z \Rightarrow t'\} in t,\ P\rangle^{l} \circ FS \twoheadrightarrow H,\langle L',\ t',\ \emptyset\rangle^{\epsilon} \circ \langle L[x \to L(y)],\ t,\ P\rangle^{l} \circ FS}$$

$$\text{E-BOX}\ \frac{\begin{array}{c} o \notin dom(H) \qquad fields(C) = \bar{f} \\ H' = H[o \to \langle C, f \to null\rangle] \qquad p\ fresh \\ TS' = \{T \in TS.k \Rightarrow \neg ancestor(TS, T, f)\} \end{array}}{H,\{f, k, \langle L,\ box[C]\{x \Rightarrow t\},\ P\rangle^{l}\} \uplus TS \rightsquigarrow H',\{f, k, \langle L[x \to b(o,p)],\ t,\ P \cup \{p\}\rangle^{\epsilon} \circ \epsilon\} \uplus TS'}$$

$$\text{E-CAPTURE}\ \frac{\begin{array}{c} L(x) = b(o,p) \qquad L(y) = b(o',p') \qquad \{p,p'\} \subseteq P \\ H(o) = \langle C, FM\rangle \qquad H' = H[o \to \langle C, FM[f \to o']\rangle] \\ TS' = \{T \in TS.k \Rightarrow \neg ancestor(TS, T, f)\} \end{array}}{H,\{f, k, \langle L,\ capture(x.f, y)\{z \Rightarrow t\},\ P\rangle^{l}\} \uplus TS \rightsquigarrow H',\{f, k, \langle L[z \to L(x)],\ t,\ P \setminus \{p'\}\rangle^{\epsilon} \circ \epsilon\} \uplus TS'}$$

$$\text{E-SWAP}\ \frac{\begin{array}{c} L(x) = b(o,p) \quad L(y) = b(o',p') \quad \{p,p'\} \subseteq P \\ H(o) = \langle C, FM \rangle \quad FM(f) = o'' \quad p''\mathit{fresh} \\ H' = H[o \to \langle C, FM[f \to o'] \rangle] \\ TS' = \{T \in TS.k \Rightarrow \neg \mathit{ancestor}(TS, T, f)\} \end{array}}{\begin{array}{c} H, \{\langle L,\ \mathit{swap}(x.f, y)\{z \Rightarrow t\},\ P\rangle^l\} \uplus TS \\ \rightsquigarrow\ H', \{\langle L[z \to b(o'', p'')],\ t,\ (P \setminus \{p'\}) \cup \{p''\}\rangle^\epsilon \circ \epsilon\} \uplus TS' \end{array}}$$

7.2.2.4 Definitions

Definition 1 (Object Type)**.** For an object identifier $o \in \mathit{dom}(H)$ where $H(o) = \langle C, FM \rangle$, $\mathit{typeof}(H, o) := C$

Definition 2 (Well-typed Heap)**.** A heap H is well-typed, written $\vdash H : \star$, iff

$$\begin{aligned} \forall o \in \mathit{dom}(H). H(o) = \langle C, FM \rangle \implies & \\ & (\mathit{dom}(FM) = \mathit{fields}(C) \wedge \\ & \forall f \in \mathit{dom}(FM). FM(f) = \mathit{null} \vee \mathit{typeof}(H, FM(f)) <: \mathit{ftype}(C, f)) \end{aligned} \tag{7.1}$$

Definition 3 (Separation)**.** Two object identifiers o and o' are separate in heap H, written $\mathit{sep}(H, o, o')$, iff $\forall q, q' \in \mathit{dom}(H). \mathit{reach}(H, o, q) \wedge \mathit{reach}(H, o', q') \implies q \neq q'$.

7.2.2.5 Other

$$\text{ANC-DIRECT}\ \frac{T = (f', k, FS) \quad FS = \langle \mathit{FINISH}\, f \rangle^l \circ FS'}{\mathit{ancestor}(TS, T, f)}$$

$$\text{ANC-INDIRECT}\ \frac{T' = (f', \mathit{true}, FS) \quad FS = \langle \mathit{FINISH}\, f \rangle^l \circ FS' \quad \mathit{ancestor}(TS, T, f')}{\mathit{ancestor}(TS, T, f)}$$

$$\text{ACC-F}\ \frac{x \to o \in L \vee (x \to b(o,p) \in L \wedge p \in P)}{\mathit{accRoot}(o, \langle L, t, P \rangle^l)}$$

$$\text{ACC-FS}\ \frac{\mathit{accRoot}(o, F) \vee \mathit{accRoot}(o, FS)}{\mathit{accRoot}(o, F \circ FS)}$$

$$\text{ISO-FS}\ \frac{\forall o, o' \in \mathit{dom}(H). (\mathit{accRoot}(o, FS) \wedge \mathit{accRoot}(o', FS')) \Rightarrow \mathit{sep}(H, o, o')}{\mathit{isolated}(H, FS, FS')}$$

$$\text{ISO-TS}\ \frac{\begin{array}{c}\forall T_1, T_2 \in TS.T_1 = (f, k, FS) \wedge T_2 = (g, k', GS) \wedge T_1 \neq T_2 \Rightarrow \\ isolated(H, FS, GS) \vee \\ FS = \langle FINISH\, f' \rangle^l \circ FS' \wedge awaits(TS, f', g) \vee \\ GS = \langle FINISH\, g' \rangle^m \circ GS' \wedge awaits(TS, g', f)\end{array}}{isolated(H, TS)}$$

$$\text{F-OK}\ \frac{\begin{array}{c}boxSep(H, F) \qquad boxObjSep(H, F) \qquad boxOcap(H, F) \\ a = ocap \implies globalOcapSep(H, F) \qquad fieldUniqueness(H, F)\end{array}}{H; a \vdash F ok}$$

$$\text{SINGFS-OK}\ \frac{H; a \vdash F ok}{H; a \vdash F \circ \epsilon ok}$$

$$\text{FS-OK}\ \frac{\begin{array}{c}H; b \vdash F^l ok \qquad H; a \vdash FS ok \\ (a = ocap \vee l = \epsilon) \Rightarrow b = ocap \\ \neg(a = ocap \vee l = \epsilon) \Rightarrow b = \epsilon \\ boxSeparation(H, F, FS) \\ uniqueOpenBox(H, F, FS) \\ openBoxPropagation(H, F^l, FS)\end{array}}{H; b \vdash F^l \circ FS ok}$$

$$\text{TS-OK}\ \frac{\begin{array}{c}\forall T \in TS.T = (f, k, \langle FINISH\, f' \rangle^l \circ FS) \Rightarrow \\ (f < f' \wedge \nexists U \in TS \setminus \{T\}.U = (f', k', \langle FINISH\, f' \rangle^l \circ FS')) \\ \exists T \in TS.(\{T' \in TS.ancestor(TS, T', T)\} \wedge \\ \forall U \in TS.U = (f, k, FS) \Rightarrow H; ocap \vdash FS ok \vee U \in TS' \wedge H; a \vdash FS ok)\end{array}}{H \vdash TS ok}$$

7.2.2.6 Predicates

$$\frac{\exists (f, k, FS) \in TS.FS = \langle FINISH\, f' \rangle^l \circ FS' \qquad awaits(TS, f', g)}{awaits(TS, f, g)}$$

$$\frac{}{awaits(TS, f, f)}$$

$$\frac{o \in dom(H)}{reach(H, o, o)}$$

$$\frac{\begin{array}{c}o \in dom(H) \qquad H(o) = \langle C, FM \rangle \\ \exists f \to o'' \in FM.reach(H, o'', o') \\ o'' \in codom(FM) \qquad reach(H, o'', o')\end{array}}{reach(H, o, o')}$$

I can't for the life of me figure out how to write a cases environment inside of the premises. Or dcases. Or any other solution. I mean, it should be obvious right? Just use some kind of box/frame/whatever command that hides the inner typesetting from the inference command but

$$\frac{x \rightarrow b(o,p) \in L \qquad p \in P}{boxRoot(o, \langle L,\ t,\ P\rangle^{l})}$$

$$\frac{boxRoot(o, F)}{boxRoot(o, F \circ \epsilon)}$$

$$\frac{boxRoot(o, F) \vee boxRoot(o, FS)}{boxRoot(o, F \circ FS)}$$

$$\frac{x \rightarrow b(o,p) \in L \qquad p \in P}{boxRoot(o, \langle L,\ t,\ P\rangle^{l}, p)}$$

$$\frac{boxRoot(o, F, p)}{boxRoot(o, F \circ \epsilon, p)}$$

$$\frac{boxRoot(o, F, p) \vee boxRoot(o, FS, p)}{boxRoot(o, F \circ FS, p)}$$

$$\frac{boxRoot(o, FS) \qquad x \rightarrow o' \in env(F) \qquad reach(H, o, o')}{openbox(H, o, F, FS)}$$

7.2.2.7 Subtyping

$$\text{<:-BOT}\ \frac{}{\bot <: \tau}$$

$$\text{<:-BOX}\ \frac{C <: D}{Box[C] <: Box[D]}$$

$$\text{<:-NULL}\ \frac{}{Null <: \tau}$$