# intrinsic semantics of termination-insensitive noninterference

by Jonathan Sterling (j.w.w. Robert Harper)
on April 26, 2022

Boston University POPV Seminar

*"type structure is a syntactic discipline for enforcing* ***levels of abstraction****."* *(Reynolds, 1983)*

*"type structure is a syntactic discipline for enforcing* ***levels of abstraction****."* *(Reynolds, 1983)*

**my take:** a type system is a ***protocol*** for the flow of information between different levels of abstraction.

## » the diversity of abstraction barriers

types are used to implement **many** forms of abstraction.

* implementation *vs.* interface.
* compiletime *vs.* runtime
* public *vs.* private
* trusted *vs.* untrusted

**this talk:** type for abstraction barriers with *security* as a running example

## » noninterference measures abstraction

it is easy to make a fancy type system; *but how do we know that it actually enforces a given abstraction?*

# » noninterference measures abstraction

it is easy to make a fancy type system; *but how do we know that it actually enforces a given abstraction?*

Goguen and Meseguer (1982) suggested **noninterference** as an objective measure of abstraction.

## » noninterference measures abstraction

it is easy to make a fancy type system; *but how do we know that it actually enforces a given abstraction?*

Goguen and Meseguer (1982) suggested **noninterference** as an objective measure of abstraction.

1. any module functor [**type** t] → [**val** b : bool] is constant
   Harper et al. (1990); Sterling and Harper (2021a)
2. any function $\tau$ @ private → bool @ public is constant
   Abadi et al. (1999); Sterling and Harper (2022)

[in this talk, "constant" means "$fx = fy$ for all $x, y$".]

## » noninterference measures abstraction

it is easy to make a fancy type system; *but how do we know that it actually enforces a given abstraction?*

Goguen and Meseguer (1982) suggested **noninterference** as an objective measure of abstraction.

1. any module functor [**type** t] → [**val** b : bool] is constant Harper et al. (1990); Sterling and Harper (2021a)
2. any function $\tau$ @ private → bool @ public is constant Abadi et al. (1999); Sterling and Harper (2022)

[in this talk, "constant" means "$fx = fy$ for all $x, y$".]

in other words, noninterference states that abstracted data is not leaked. (often a consequence of parametricity, but do not confuse the map for the territory!)

## » the need for controlled leakage of abstraction

full **noninterference** represents an *extreme* abstraction policy that is useful in many cases.

## » the need for controlled leakage of abstraction

full **noninterference** represents an *extreme* abstraction policy that is useful in many cases.

in practice, we often need *weaker* abstraction policies that allow certain sensitive data to be leaked in a controlled way.

# » the need for controlled leakage of abstraction

full **noninterference** represents an *extreme* abstraction policy that is useful in many cases.

in practice, we often need *weaker* abstraction policies that allow certain sensitive data to be leaked in a controlled way.

* inlining definitions across module boundaries (MLton)

# » the need for controlled leakage of abstraction

full **noninterference** represents an *extreme* abstraction policy that is useful in many cases.

in practice, we often need *weaker* abstraction policies that allow certain sensitive data to be leaked in a controlled way.

* inlining definitions across module boundaries (MLton)
* revealing secret votes after the auction is finished

## » the need for controlled leakage of abstraction

full **noninterference** represents an *extreme* abstraction policy that is useful in many cases.

in practice, we often need *weaker* abstraction policies that allow certain sensitive data to be leaked in a controlled way.

* inlining definitions across module boundaries (MLton)
* revealing secret votes after the auction is finished
* **this talk:** leaking through the termination channel, *i.e.* termination-insensitive noninterference / TINI

# » partial functions and termination-insensitive noninterference

**full** noninterference says that for any partial function $f$ : int @ private $\rightharpoonup$ bool @ public, either

1. $f = \lambda\_.\mathsf{ret}\,\mathsf{tt}$,
2. or $f = \lambda\_.\mathsf{ret}\,\mathsf{ff}$,
3. or $f = \lambda\_.\bot$.

**termination-insensitive** noninterference says that for any $x, y$ : int @ private such that $fx{\downarrow}$ and $fy{\downarrow}$, we have $fx = fy$.

**TINI:** leak data through (only) the termination channel

**contribution of this work:**
new denotational semantics for TINI

# dependency core calculus

## » a core calculus of dependency (Abadi et al., 1999)

Abadi et al. (1999) proposed a simple and elegant core calculus **DCC** for information flow; monadic metalanguage + idempotent monads for each security level:

$$A ::= A \to B \mid A \times B \mid A_\perp \mid \mathsf{T}_l A$$

judgment $\boxed{A \text{ sealed } @\, l}$ means that $\eta : A \to \mathsf{T}_l A$ is an iso.

$$\frac{\Gamma \vdash M : \mathsf{T}_l A \qquad \Gamma, x : A \vdash Nx : B \qquad B \text{ sealed } @\, l}{\Gamma \vdash x \leftarrow M; Nx : B}$$

# » closure properties of sealing in DCC

antitone family of exponential ideals:

$$\frac{l \sqsubseteq l'}{\mathsf{T}_{l'} A \text{ sealed } @\, l} \qquad \frac{A \text{ sealed } @\, l}{\mathsf{T}_{l'} A \text{ sealed } @\, l}$$

$$\frac{A \text{ sealed } @\, l \qquad B \text{ sealed } @\, l}{A \times B \text{ sealed } @\, l} \qquad \frac{B \text{ sealed } @\, l}{A \to B \text{ sealed } @\, l}$$

# » noninterference in DCC

noninterference holds in DCC because you can't "get out" of the monad $\mathsf{T}_l$. but how do we prove it?

1. construct a **denotational semantics** $[\![-]\!]$ that validates noninterference
2. prove that DCC is **computationally adequate** wrt. $[\![-]\!]$, *i.e.* for $\cdot \vdash M : \mathsf{unit}_\perp$ we have:

$$[\![M]\!]\downarrow \iff (\cdot \vdash M = \mathsf{ret}\,() : \mathsf{unit}_\perp)$$

Abadi et al. (1999) employ a *relational model* over dcpos.

# » Abadi et al.'s relational model of DCC

**dcpos** model the "Moggi fragment" $A \times B, A \to B, A_\perp$.

# » Abadi et al.'s relational model of DCC

**dcpos** model the "Moggi fragment" $A \times B, A \to B, A_{\perp}$.
more structure needed to interpret **sealing**.

# » Abadi et al.'s relational model of DCC

**dcpos** model the "Moggi fragment" $A \times B, A \to B, A_\perp$.
more structure needed to interpret **sealing**.

**Abadi et al.:** constrain dcpos by $|\mathcal{L}|$-indexed *binary relations*, where $(\mathcal{L}, \sqsubseteq)$ is the poset of security levels.

# » indexed logical relation on dcpos

Abadi et al. (1999) define an indexed logical relation $\mathcal{DC}$:

* an **object** is a pair of a dcpo $A$ and a family of admissible binary relation $R_{A,l} \subseteq A \times A$ for $l \in \mathcal{L}$;
* a **morphism** from $(A, R_A)$ to $(B, R_B)$ is a continuous function $f : A \to B$ such that for all $(x, y) \in R_{A,l}$ we have $(fx, fy) \in R_{B,l}$.

$\mathcal{DC}$ is cartesian closed; in fact, it is the admissible sub-gluing of the functor dcpo $\to$ dcpo$^{|\mathcal{L}|}$ sending $A$ to $(l \mapsto A \times A)$ where $|\mathcal{L}|$ is the underlying set of the poset $\mathcal{L}$.

# » relational interpretation of nontermination and sealing

Abadi et al. interpret a type as a pair $[\![A]\!] = (|A|, R_A) \in \mathcal{DC}$.

**nontermination:**

$$|A_\perp| = |A|_\perp$$
$$R_{A_\perp,l} = R_{A,l} \cup \{(\perp,\perp)\}$$

**sealing:**

$$|\mathsf{T}_l A| = |A|$$
$$R_{\mathsf{T}_l A,l'} = \begin{cases} R_{A,l'} & \text{if } l \sqsubseteq l' \\ \top & \text{otherwise} \end{cases}$$

# » noninterference via relational model

fix a function $f : \mathsf{T}_{\mathsf{private}}\mathsf{int} \to (\mathsf{T}_{\mathsf{public}}\mathsf{bool})_\perp$.

# » noninterference via relational model

fix a function $f : \mathsf{T}_{\mathsf{private}}\mathsf{int} \to (\mathsf{T}_{\mathsf{public}}\mathsf{bool})_\perp$.

$[\![f]\!]$ is a function $\mathbb{Z} \to 2_\perp$ such that for all $l \in \mathcal{L}$ and $x, y \in \mathbb{Z}$,

$$x \; R_{\mathsf{T}_{\mathsf{private}}\mathsf{int}, l} \; y \implies [\![f]\!]x = [\![f]\!]y$$

## » noninterference via relational model

fix a function $f : \mathsf{T}_{\mathsf{private}}\mathsf{int} \to (\mathsf{T}_{\mathsf{public}}\mathsf{bool})_\bot$.

$[\![f]\!]$ is a function $\mathbb{Z} \to 2_\bot$ such that for all $l \in \mathcal{L}$ and $x, y \in \mathbb{Z}$,

$$x \; R_{\mathsf{T}_{\mathsf{private}}\mathsf{int},l} \; y \implies [\![f]\!]x = [\![f]\!]y$$

setting $l :=$ public, we have:

$$\top \implies [\![f]\!]x = [\![f]\!]y$$

# » noninterference via relational model

fix a function $f : \mathsf{T}_{\mathsf{private}}\mathsf{int} \to (\mathsf{T}_{\mathsf{public}}\mathsf{bool})_\perp$.

$[\![f]\!]$ is a function $\mathbb{Z} \to 2_\perp$ such that for all $l \in \mathcal{L}$ and $x, y \in \mathbb{Z}$,

$$x \; R_{\mathsf{T}_{\mathsf{private}}\mathsf{int}, l} \; y \implies [\![f]\!]x = [\![f]\!]y$$

setting $l :=$ public, we have:

$$\top \implies [\![f]\!]x = [\![f]\!]y$$

thus the relational model satisfies noninterference. **adequacy** for the relational model then implies noninterference for DCC.

# » noninterference and discreteness

noninterference always works in the same way: you have a more indiscrete relation on the left and a more discrete relation on the right.

# » noninterference and discreteness

noninterference always works in the same way: you have a more indiscrete relation on the left and a more discrete relation on the right.

noninterference is termination-sensitive in the relational model $[\![-]\!]$ because when $R_A$ is discrete, then so is $R_{A_\perp}$.

## » noninterference and discreteness

noninterference always works in the same way: you have a more indiscrete relation on the left and a more discrete relation on the right.

noninterference is termination-sensitive in the relational model $[\![-]\!]$ because when $R_A$ is discrete, then so is $R_{A_\bot}$.

Abadi et al. (1999) go on to adapt the DCC to support termination-**in**sensitivity by changing the semantics of $A_\bot$ to be less discrete.

# » adapting DCC for termination-insensitive noninterference

Abadi et al. (1999) adapt DCC for TINI by extending the rules for sealing:

$$\frac{A \ \textsf{sealed} \ @\, l}{A_\perp \ \textsf{sealed} \ @\, l}$$

with this rule, the canonical map $(\mathsf{T}_l A)_\perp \to \mathsf{T}_l A_\perp$ is an iso.

# » adapting DCC for termination-insensitive noninterference

Abadi et al. (1999) adapt DCC for TINI by extending the rules for sealing:

$$\frac{A \text{ sealed } @\, l}{A_\bot \text{ sealed } @\, l}$$

with this rule, the canonical map $(\mathsf{T}_l A)_\bot \to \mathsf{T}_l A_\bot$ is an iso.

lastly, tweak the relational semantics:

$$R_{A_\bot,l} = R_{A,l} \cup \{(\bot,\bot)\} \cup \{(x,\bot) \mid x \in |A|\} \cup \{(\bot,x) \mid x \in |A|\}$$

# » critique of relational semantics of TINI

there are several problems with the relational semantics.
refer to $A \in \mathcal{DC}$ as $l$-sealed when $A \cong \mathsf{T}_l A$.

# » critique of relational semantics of TINI

there are several problems with the relational semantics. refer to $A \in \mathcal{DC}$ as $l$-sealed when $A \cong \mathsf{T}_l A$.

1. **failure of antitonicity:** if $A$ is $l$-sealed and $k \sqsubseteq l$, it need not be that $A$ is $k$-sealed. good behavior limited to the **image** of $[\![-]\!]$, *contra* the principles of den.sem.

# » critique of relational semantics of TINI

there are several problems with the relational semantics. refer to $A \in \mathcal{DC}$ as $l$-sealed when $A \cong \mathsf{T}_l A$.

1. **failure of antitonicity:** if $A$ is $l$-sealed and $k \sqsubseteq l$, it need not be that $A$ is $k$-sealed. good behavior limited to the **image** of $[\![-]\!]$, *contra* the principles of den.sem.
2. **failure of transitivity:** we think of $x\ R_{A,l}\ y$ as meaning "$x$ indistinguishable to $y$ by clients at level $l$", but $R_{A_\perp,l}$ in TINI model is not transitive!

## » critique of relational semantics of TINI

there are several problems with the relational semantics. refer to $A \in \mathcal{DC}$ as $l$-sealed when $A \cong \mathsf{T}_l A$.

1. **failure of antitonicity:** if $A$ is $l$-sealed and $k \sqsubseteq l$, it need not be that $A$ is $k$-sealed. good behavior limited to the **image** of $[\![-]\!]$, *contra* the principles of den.sem.
2. **failure of transitivity:** we think of $x\ R_{A,l}\ y$ as meaning "$x$ indistinguishable to $y$ by clients at level $l$", but $R_{A_\perp,l}$ in TINI model is not transitive!
3. **(TI)NI is bolted on:** relational model takes an **insecure** computational model and **cuts it down** to its secure part. not what I would call den.sem. for security!

(end of background)

# » intrinsic semantics of termination-insensitive noninterference

our desiderata for semantics:

1. **antitone:** if $A$ is $l$-sealed and $k \sqsubseteq l$, then $A$ is $k$-sealed.
2. **intrinsic:** rather than “cutting down” insecure dcpo model, find new kind of domain that supports security.

# » intrinsic semantics of termination-insensitive noninterference

our desiderata for semantics:

1. **antitone:** if $A$ is $l$-sealed and $k \sqsubseteq l$, then $A$ is $k$-sealed.
2. **intrinsic:** rather than “cutting down” insecure dcpo model, find new kind of domain that supports security.

these principles naturally lead to (pre)sheaves of dcpos, where TINI behavior arises *automatically* in a **startling** way.

# » indexed cbpv decomposition of DCC

we simplify our project by decomposing DCC into *value types* and *computation types*.

$$A^+ ::= \mathsf{U}X^\ominus \mid \mathsf{bool} \mid \ldots \qquad \text{(value types)}$$

$$X^\ominus ::= \mathsf{F}A^+ \mid A^+ \to X^\ominus \mid \ldots \qquad \text{(comp. types)}$$

# » indexed cbpv decomposition of DCC

we simplify our project by decomposing DCC into *value types* and *computation types*.

$$A^{+} ::= \mathsf{U}X^{\ominus} \mid \mathsf{bool} \mid \ldots \qquad \text{(value types)}$$

$$X^{\ominus} ::= \mathsf{F}A^{+} \mid A^{+} \to X^{\ominus} \mid \ldots \qquad \text{(comp. types)}$$

close only value types under sealing modalities:

$$A^{+} ::= \ldots \mid \mathsf{T}_{l}A^{+} \mid \ldots$$

# » indexed cbpv decomposition of DCC

we simplify our project by decomposing DCC into *value types* and *computation types*.

$$A^+ ::= \mathsf{U} X^\ominus \mid \mathsf{bool} \mid \ldots \qquad \text{(value types)}$$

$$X^\ominus ::= \mathsf{F} A^+ \mid A^+ \to X^\ominus \mid \ldots \qquad \text{(comp. types)}$$

close only value types under sealing modalities:

$$A^+ ::= \ldots \mid \mathsf{T}_l A^+ \mid \ldots$$

index equational theory by security levels $l \in \mathcal{L}$:

$$\boxed{\Gamma \vdash_l U : A^+} \qquad \boxed{\Gamma \vdash_l U \equiv V : A^+} \qquad \boxed{\Gamma \vdash_l M : X^\ominus}$$

$$\boxed{\Gamma \vdash_l M \equiv N : X^\ominus}$$

## » the sealing modality; declassification of termination channels

our sealing modality $\mathsf{T}_l$ is an idempotent monad like Abadi et al. (1999). but it collapses to a point under $l$:

$$\frac{k \sqsubseteq l}{\Gamma \vdash_k \star : \mathsf{T}_l A^+} \qquad \frac{k \sqsubseteq l \qquad \Gamma \vdash_k U : \mathsf{T}_l A^+}{\Gamma \vdash_k U \equiv V : \mathsf{T}_l A^+}$$

# » the sealing modality; declassification of termination channels

our sealing modality $\mathsf{T}_l$ is an idempotent monad like Abadi et al. (1999). but it collapses to a point under $l$:

$$\frac{k \sqsubseteq l}{\Gamma \vdash_k \star : \mathsf{T}_l A^+} \qquad \frac{k \sqsubseteq l \qquad \Gamma \vdash_k U : \mathsf{T}_l A^+}{\Gamma \vdash_k U \equiv V : \mathsf{T}_l A^+}$$

for TINI, we add an explicit operation to declassify side effects while protecting return values:

$$\frac{\Gamma \vdash_k V : \mathsf{T}_l \mathsf{U}\mathsf{F} A^+ \qquad A^+ \ \mathsf{sealed} \ @\, l}{\Gamma \vdash_k \mathsf{tdcl}_l V : \mathsf{F} A^+}$$

$$\frac{}{\Gamma \vdash_k \mathsf{tdcl}_l(\eta_l\,(\mathsf{ret}\, V)) \equiv \mathsf{ret}\, V}$$

# » denotational semantics: total fragment

let $\mathcal{L}$ be a meet semilattice and consider the presheaf topos $\mathrm{Pr}\,\mathcal{L} = [\mathcal{L}^{\mathsf{op}}, \mathsf{Set}]$.

# » denotational semantics: total fragment

let $\mathcal{L}$ be a meet semilattice and consider the presheaf topos $\mathsf{Pr}\,\mathcal{L} = [\mathcal{L}^{\mathsf{op}}, \mathsf{Set}]$.

each security level $l \in \mathcal{L}$ gives rise to a proposition $\langle l \rangle$ in the internal language of $\mathsf{Pr}\,\mathcal{L}$:

$$\langle l \rangle k = (k \leq l)$$

# » denotational semantics: total fragment

let $\mathcal{L}$ be a meet semilattice and consider the presheaf topos $\mathsf{Pr}\,\mathcal{L} = [\mathcal{L}^{\mathsf{op}}, \mathsf{Set}]$.

each security level $l \in \mathcal{L}$ gives rise to a proposition $\langle l \rangle$ in the internal language of $\mathsf{Pr}\,\mathcal{L}$:

$$\langle l \rangle k = (k \leq l)$$

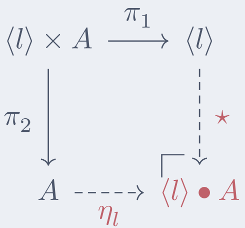
intuitive meaning of $\langle l \rangle$ is "I am unauthorized to see $l$".
we will interpret the sealing modality as redaction.

## » security levels induce phase distinctions

$\langle l \rangle$ gives rise to two reflective subcategories:

1. the presheaves $A$ such that $A \cong A^{\langle l \rangle}$ are $\langle l \rangle$-transparent
2. the presheaves $A$ such that $A \times \langle l \rangle \cong \langle l \rangle$ are $\langle l \rangle$-sealed

to *seal* a presheaf $A$, we take a pushout (quotient of sum):




$$\langle l \rangle \bullet A \cong (\langle l \rangle + A)/\sim$$

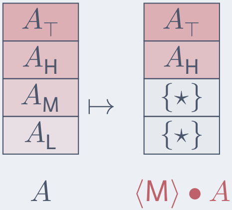
**where**

$$u \sim v \iff (\langle l \rangle = \top) \lor (u = v)$$

we will interpret $[\![\mathsf{T}_l A^+]\!] := \langle l \rangle \bullet [\![A^+]\!]$.

# » visualizing the sealing modality

let $\mathcal{L} = \{\mathsf{L} \sqsubset \mathsf{M} \sqsubset \mathsf{H} \sqsubset \top\}$, and fix $A \in \mathsf{Pr}\,\mathcal{L}$.



# » noninterference for total maps

in Pr $\mathcal{L}$, noninterference for *total* maps is **immediate**:

**Theorem**

*if $A$ is $\langle l \rangle$-sealed and $B$ is $\langle l \rangle$-transparent, then any function $A \to B$ is constant.*

# » noninterference for total maps

in Pr $\mathcal{L}$, noninterference for *total* maps is **immediate**:

**Theorem**

*if $A$ is $\langle l \rangle$-sealed and $B$ is $\langle l \rangle$-transparent, then any function $A \to B$ is constant.*

**Corollary**

*any function $\langle l \rangle \bullet \mathbb{Z} \to 2$ is constant.*

that's because $2$ is constant and thus $\langle l \rangle$-transparent.

## » noninterference for total maps

in Pr $\mathcal{L}$, noninterference for *total* maps is **immediate**:

**Theorem**

*if $A$ is $\langle l \rangle$-sealed and $B$ is $\langle l \rangle$-transparent, then any function $A \to B$ is constant.*

**Corollary**

*any function $\langle l \rangle \bullet \mathbb{Z} \to 2$ is constant.*

that's because $2$ is constant and thus $\langle l \rangle$-transparent.

leads to TINI, as *partial functions* $A \rightharpoonup B$ are encoded by total functions $A \to \sum_{\phi:\mathsf{Prop}} B^\phi$ which is not constant.

$$\mathsf{Prop}(l) = \{\phi \subseteq \mathcal{L} \downarrow l \mid \phi \text{ closed under precomposition}\}$$

## » termination-insensitive noninterference for partial maps

in Pr $\mathcal{L}$, a *partial map* $A \rightharpoonup B$ is given by a total map $A \to \mathsf{L}B$ into the partial map classifier:

$$\mathsf{L}B := \sum_{\phi:\mathsf{Prop}} B^{\phi}$$

## » termination-insensitive noninterference for partial maps

in Pr $\mathcal{L}$, a *partial map* $A \rightharpoonup B$ is given by a total map $A \to \mathsf{L}B$ into the partial map classifier:

$$\mathsf{L}B := \sum_{\phi:\mathsf{Prop}} B^{\phi}$$

**Theorem (termination-insensitive noninterference)**

*for any function $f : \langle l \rangle \bullet \mathbb{Z} \to \mathsf{L}2$, if $fx{\downarrow}$ and $fy{\downarrow}$ then $fx = fy$.*

## » termination-insensitive noninterference for partial maps

in Pr $\mathcal{L}$, a *partial map* $A \rightharpoonup B$ is given by a total map $A \to \mathsf{L}B$ into the partial map classifier:

$$\mathsf{L}B := \sum_{\phi:\mathsf{Prop}} B^{\phi}$$

**Theorem (termination-insensitive noninterference)**

*for any function $f : \langle l \rangle \bullet \mathbb{Z} \to \mathsf{L}2$, if $fx{\downarrow}$ and $fy{\downarrow}$ then $fx = fy$.*

**Proof.**

the partial function $f$ restricts to a total function $\tilde{f} : U \to 2$ where $U \subseteq \langle l \rangle \bullet \mathbb{Z}$ is the set of values on which $f$ is defined; but $2$ is $\langle l \rangle$-transparent. □

# » internal domain theory for recursion

the TINI property is observed already for partial maps between presheaves, but we need to interpret recursion.
**idea:** replace ordinary dcpos with ***internal*** dcpos in $\mathrm{Pr}\,\mathcal{L}$!

# » internal domain theory for recursion

the TINI property is observed already for partial maps between presheaves, but we need to interpret recursion.
**idea:** replace ordinary dcpos with ***internal*** dcpos in Pr $\mathcal{L}$!

partial map classifier extends to a *lifting monad* on dcpos:

$$u \leq_{\mathsf{L}A} v \iff \forall x : A.u = (\top, a) \implies \exists y : A.v = (\top, b) \land x \leq_A y$$

# » Eilenberg–Moore model of cbpv DCC

1. $A^+$ is interpreted as an internal dcpo $[\![A^+]\!]$ in dcpo(Pr $\mathcal{L}$);
2. $X^\ominus$ is interpreted as an algebra for L in dcpo(Pr $\mathcal{L}$).

$$[\![\mathsf{U}X^\ominus]\!] = [\![X]\!] \qquad [\![\mathsf{F}A^+]\!] = \mathsf{L}[\![A^+]\!]$$

# » Eilenberg–Moore model of cbpv DCC

1. $A^+$ is interpreted as an internal dcpo $[\![A^+]\!]$ in $\mathsf{dcpo}(\mathsf{Pr}\,\mathcal{L})$;
2. $X^\ominus$ is interpreted as an algebra for $\mathsf{L}$ in $\mathsf{dcpo}(\mathsf{Pr}\,\mathcal{L})$.

$$[\![\mathsf{U}X^\ominus]\!] = [\![X]\!] \qquad [\![\mathsf{F}A^+]\!] = \mathsf{L}[\![A^+]\!]$$

$\mathsf{dcpo}(\mathsf{Pr}\,\mathcal{L})$ is cocomplete over $\mathsf{Pr}\,\mathcal{L}$, so we can interpret $[\![\mathsf{T}_l A^+]\!]$ as the pushout $\langle l \rangle \bullet [\![A^+]\!]$.

# » Eilenberg–Moore model of cbpv DCC

1. $A^+$ is interpreted as an internal dcpo $[\![A^+]\!]$ in dcpo(Pr $\mathcal{L}$);
2. $X^\ominus$ is interpreted as an algebra for L in dcpo(Pr $\mathcal{L}$).

$$[\![\mathsf{U}X^\ominus]\!] = [\![X]\!] \qquad [\![\mathsf{F}A^+]\!] = \mathsf{L}[\![A^+]\!]$$

dcpo(Pr $\mathcal{L}$) is cocomplete over Pr $\mathcal{L}$, so we can interpret $[\![\mathsf{T}_l A^+]\!]$ as the pushout $\langle l \rangle \bullet [\![A^+]\!]$.

$[\![\Gamma \vdash_l V : A^+]\!]$ is a continuous map $[\![V]\!] : [\![\Gamma]\!] \times \langle l \rangle \to [\![A^+]\!]$.

# » interpretation of termination declassification

for any $l \in \mathcal{L}$ and $\langle l \rangle$-sealed $A \in \mathsf{dcpo}(\mathsf{Pr}\,\mathcal{L})$, we must construct a continuous map $\mathsf{tdcl}_l : \langle l \rangle \bullet \mathsf{L}A \to \mathsf{L}A$. use universal property of the pushout!

$$\mathsf{tdcl}_l(\star) = (\top, \star)$$
$$\mathsf{tdcl}_l(\eta_l(\phi, a)) = (\langle l \rangle \vee \phi, [\langle l \rangle \hookrightarrow \star, \phi \hookrightarrow a])$$

# » interpretation of termination declassification

for any $l \in \mathcal{L}$ and $\langle l \rangle$-sealed $A \in \mathsf{dcpo}(\mathsf{Pr}\,\mathcal{L})$, we must construct a continuous map $\mathsf{tdcl}_l : \langle l \rangle \bullet \mathsf{L}A \to \mathsf{L}A$. use universal property of the pushout!

$$\mathsf{tdcl}_l(\star) = (\top, \star)$$
$$\mathsf{tdcl}_l(\eta_l(\phi, a)) = (\langle l \rangle \vee \phi, [\langle l \rangle \hookrightarrow \star, \phi \hookrightarrow a])$$

computationally, each $\langle l \rangle \in \mathsf{Prop} \cong \mathsf{L}1$ can be thought of as an assertion that $l$ is redacted from the observer.

# » interpretation of termination declassification

for any $l \in \mathcal{L}$ and $\langle l \rangle$-sealed $A \in \mathsf{dcpo}(\mathsf{Pr}\,\mathcal{L})$, we must construct a continuous map $\mathsf{tdcl}_l : \langle l \rangle \bullet \mathsf{L}A \to \mathsf{L}A$. use universal property of the pushout!

$$\mathsf{tdcl}_l(\star) = (\top, \star)$$
$$\mathsf{tdcl}_l(\eta_l(\phi, a)) = (\langle l \rangle \vee \phi, [\langle l \rangle \hookrightarrow \star, \phi \hookrightarrow a])$$

computationally, each $\langle l \rangle \in \mathsf{Prop} \cong \mathsf{L}1$ can be thought of as an assertion that $l$ is redacted from the observer.

termination declassification runs this assertion in parallel with the sealed computation.

# » interpretation of termination declassification

for any $l \in \mathcal{L}$ and $\langle l \rangle$-sealed $A \in \mathsf{dcpo}(\mathsf{Pr}\,\mathcal{L})$, we must construct a continuous map $\mathsf{tdcl}_l : \langle l \rangle \bullet \mathsf{L}A \to \mathsf{L}A$. use universal property of the pushout!

$$\mathsf{tdcl}_l(\star) = (\top, \star)$$
$$\mathsf{tdcl}_l(\eta_l(\phi, a)) = (\langle l \rangle \vee \phi, [\langle l \rangle \hookrightarrow \star, \phi \hookrightarrow a])$$

computationally, each $\langle l \rangle \in \mathsf{Prop} \cong \mathsf{L}1$ can be thought of as an assertion that $l$ is redacted from the observer.

termination declassification runs this assertion in parallel with the sealed computation.

thus for an observer away from (above) $\langle l \rangle$, the divergence of $\mathsf{tdcl}_l(\eta_l \bot)$ is visible.

# » adequacy and TINI for cbpv dcc

we prove **adequacy** of the equational theory for the presheaf model using a Plotkin-style logical relations argument via synthetic Tait computability (Sterling and Harper, 2021*a*; Sterling, 2021).

thus TINI lifts from the model to the equational theory.

# » an example of a program with a leaky termination channel

**Example**

there exists an $l$-sealed type $A^+$ and a function $M : A^+ \to \mathsf{UF}\,\mathsf{unit}$ such that for some closed terms $U, V : A^+$ we have $MU\Downarrow$ but $MV\Uparrow$.

**Proof.**

Choose the following:

$$\begin{aligned} A^+ &:= \mathsf{T}_l\,\mathsf{bool} \\ U &:= \eta_l\,\mathsf{true} \\ V &:= \eta_l\,\mathsf{false} \\ M &:= \lambda x.\mathsf{tdcl}_l(z \leftarrow x; \eta_l(\mathsf{if}\ z\ \mathsf{then}\ \mathsf{ret}\,()\ \mathsf{else}\ \bot)) \end{aligned}$$

□

# » overview of contributions

we have contributed an **intrinsic** and **non-relational** denotational semantics for TINI.

- real den.sem.: good behavior outside im $[\![-]\!]$.
- avoids transitivity issue: no relations, no problem!
- ordinary / **naïve** Scott semantics, but in presheaves.

## » overview of contributions

we have contributed an **intrinsic** and **non-relational** denotational semantics for TINI.

- real den.sem.: good behavior outside im $[\![-]\!]$.
- avoids transitivity issue: no relations, no problem!
- ordinary / **naïve** Scott semantics, but in presheaves.

fully **synthetic** methods (not detailed in this talk!):

- logical frameworks for the equational theory,
- synthetic domain theory for the denotational semantics,
- synthetic Tait computability for the adequacy proof.

# » references

M. Abadi, A. Banerjee, N. Heintze, and J. G. Riecke. A core calculus of dependency. In *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '99, pages 147–160, San Antonio, Texas, USA, 1999. Association for Computing Machinery. ISBN 1-58113-095-3. doi: 10.1145/292540.292555.

J. A. Goguen and J. Meseguer. Security policies and security models. In *1982 IEEE Symposium on Security and Privacy*, 1982. doi: 10.1109/SP.1982.10014.

R. Harper, J. C. Mitchell, and E. Moggi. Higher-order modules and the phase distinction. In *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 341–354, San Francisco, California, USA, 1990. Association for Computing Machinery. ISBN 0-89791-343-4. doi: 10.1145/96709.96744.

# » references (cont.)

Y. Niu, J. Sterling, H. Grodin, and R. Harper. A cost-aware logical framework. *Proceedings of the ACM on Programming Languages*, 6(POPL), Jan. 2022. doi: 10.1145/3498670.

J. C. Reynolds. Types, abstraction, and parametric polymorphism. In *Information Processing*, 1983.

J. Sterling. *First Steps in Synthetic Tait Computability: The Objective Metatheory of Cubical Type Theory*. PhD thesis, Carnegie Mellon University, 2021. CMU technical report CMU-CS-21-142.

J. Sterling and R. Harper. Logical relations as types: Proof-relevant parametricity for program modules. *Journal of the ACM*, 68(6), Oct. 2021*a*. ISSN 0004-5411. doi: 10.1145/3474834.

# » references (cont.)

J. Sterling and R. Harper. A metalanguage for multi-phase modularity. ML 2021 abstract and talk, Aug. 2021b. URL https://icfp21.sigplan.org/details/mlfamilyworkshop-2021-papers/5/A-metalanguage-for-multi-phase-modularity.

J. Sterling and R. Harper. Sheaf semantics of termination-insensitive noninterference. In A. Felty, editor, *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, volume 228 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Dagstuhl, Germany, Aug. 2022. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/LIPIcs.FSCD.2022.15.