Type-based Dependency Analysis

RS^3 Topic Workshop on Concurrent Noninterference

UNI FREIBURG

University of Freiburg

Matthias Keil
Institute for Computer Science
Faculty of Engineering
University of Freiburg

June 2012

Outline

1 Dependencies

2 Formalization
- Dependency Type

3 Abstract Analysis

4 Soundness
- Noninterference
- Correctness
- Termination

5 Dependency-based Access Permission Contracts

$$\begin{array}{lcll} \textit{Values} & \ni & v & ::= \quad \phi, \psi, \pi \\ \textit{Expressions} & \ni & e & ::= \quad e(v) \mid \textbf{if}(v)\ e \end{array}$$

$$e(\psi) \Downarrow \phi \tag{1.1}$$

Definition (Dependency)

$$\begin{aligned} &\exists \langle \psi_i, \psi_j \rangle \mid \psi_i \neq \psi_j \ : \\ &e[\psi \mapsto \psi_i] \Downarrow \phi_i \ \wedge \ e[\psi \mapsto \psi_j] \Downarrow \phi_j \ \wedge \ \phi_i \neq \phi_j \\ &\Rightarrow \ \phi \rightsquigarrow \psi \end{aligned} \tag{1.2}$$

UNI FREIBURG

$$\begin{array}{lllll} \textit{Values} & \ni & v & ::= & \phi, \psi, \pi \\ \textit{Expressions} & \ni & e & ::= & e(v) \mid \textbf{if}(v)\ e \end{array}$$

$$e(\psi) \Downarrow \phi \tag{1.3}$$

Definition (Independency)

$$\begin{aligned} &\forall \langle \psi_i, \psi_j \rangle \mid \psi_i \neq \psi_j \ : \\ &e[\psi \mapsto \psi_i] \Downarrow \phi_i \ \wedge \ e[\psi \mapsto \psi_j] \Downarrow \phi_j \ \wedge \ \phi_i = \phi_j \\ &\Rightarrow \ \phi \not\rightsquigarrow \psi \end{aligned} \tag{1.4}$$

UNI FREIBURG

Definition (Direct Dependency [Den76, DD77])

$$e(\psi) \Downarrow \phi \;\Rightarrow\; \phi \rightsquigarrow \psi$$

Definition (Indirect Dependency [Den76, DD77])

$$\textbf{if}\ (\psi)\ e \Downarrow \phi \;\Rightarrow\; \phi \rightsquigarrow \psi$$

Definition (Transitiv Relation [Den76, DD77])

$$\phi \rightsquigarrow \pi \;\wedge\; \pi \rightsquigarrow \psi \;\Rightarrow\; \phi \rightsquigarrow \psi$$

Syntax of λ_{JS} [Int99]

UNI FREIBURG

$$
\begin{array}{llll}
\textit{Constant} & \ni c & ::= & \textit{bool} \mid \textit{num} \mid \textit{str} \mid \textbf{undefined} \mid \textbf{null} \\
\textit{Variable} & \ni x & ::= & x_0 \ldots \\
\textit{Value} & \ni v & ::= & c \mid \xi^{\ell} \\
\textit{Expression} & \ni e & ::= & v \mid x \\
 & & \mid & \textbf{let}\ (x = e)\ \textbf{in}\ e \\
 & & \mid & \textbf{op}(e \ldots) \\
 & & \mid & e.e \\
 & & \mid & e.e = e \\
 & & \mid & e(e) \\
 & & \mid & \textbf{new}^{\ell} \\
 & & \mid & \textbf{if}\ (e)\ e,\ e \\
 & & \mid & \lambda^{\ell} x.e \\
 & & \mid & e; e \\
 & & \mid & \textbf{trace}^{\iota}(e)
\end{array}
$$

UNI FREIBURG

$$
\begin{array}{llll}
\textit{Location} & \ni \xi^{\ell} & & \\
\textit{Function} & \ni \lambda^{\ell} x.e & & \\
\textit{Closure} & \ni f & ::= & \textit{Environment} \times \textit{Function} \\
\textit{Properties} & \ni \mathcal{P} & ::= & \textit{str} \rightarrow \textit{Value} \\
\textit{Object} & \ni o & ::= & \textit{Properties} \times \textit{Closure} \\
\textit{Environment} & \ni \rho & ::= & \textit{Variable} \rightarrow \textit{Value} \\
\textit{Heap} & \ni \mathcal{H} & ::= & \textit{Location} \rightarrow \textit{Object}
\end{array}
$$

Definition (Judgement λ_{JS})

$$\mathcal{H}, \rho \vdash e \Downarrow \mathcal{H}' \mid v \tag{2.1}$$

$$
\begin{array}{lrcl}
\textit{Basic Value} & \ni v & ::= & c \\
 & & \mid & \xi^{\ell} \\
\textit{Value} & \ni \omega & ::= & v : \kappa \\
\textit{Dependency Type} & \ni \kappa & ::= & \emptyset \\
 & & \mid & \iota \\
 & & \mid & \kappa \bullet \kappa
\end{array}
$$

Definition (Judgement $\lambda^{\mathcal{D}}_{JS}$)

$$\mathcal{H}, \rho, \kappa \vdash e \Downarrow \mathcal{H}' \mid \omega \tag{2.2}$$

UNI FREIBURG

(DT-Constant)

$$\frac{}{\mathcal{H}, \rho, \kappa \;\vdash\; c \;\Downarrow\; \mathcal{H} \mid c : \kappa}$$

(DT-Variable)

$$\frac{}{\mathcal{H}, \rho, \kappa \;\vdash\; x \;\Downarrow\; \mathcal{H} \mid \rho(x) \bullet \kappa}$$

(DT-Operation)

$$\frac{\begin{array}{c}\mathcal{H}, \rho, \kappa \;\vdash\; e_0 \;\Downarrow\; \mathcal{H}' \mid v_0 : \kappa_0 \\ \vdots \\ \mathcal{H}^{n-1}, \rho, \kappa \;\vdash\; e_n \;\Downarrow\; \mathcal{H}^n \mid v_n : \kappa_n \\ v_{op} \;=\; \Downarrow^{v}_{\mathbf{op}} (v_0 \ldots v_n)\end{array}}{\mathcal{H}, \rho, \kappa \;\vdash\; \mathbf{op}(e_0 \ldots e_n) \;\Downarrow\; \mathcal{H}^n \mid v_{op} : \kappa_0 \bullet \ldots \bullet \kappa_n}$$

(DT-Let)

$$\frac{\mathcal{H}, \rho, \kappa \vdash e_0 \Downarrow \mathcal{H}' \mid v_0 : \kappa_0 \qquad \mathcal{H}', \rho[x \mapsto v_0 : \kappa_0], \kappa \vdash e_1 \Downarrow \mathcal{H}'' \mid v_1 : \kappa_1}{\mathcal{H}, \rho, \kappa \vdash \textbf{let } (x = e_0) \textbf{ in } e_1 \Downarrow \mathcal{H}'' \mid v_1 : \kappa_1}$$

(DT-FunctionCreation)

$$\frac{\xi^\ell \notin dom(\mathcal{H})}{\mathcal{H}, \rho, \kappa \vdash \lambda^\ell x.e \Downarrow \mathcal{H}[\xi^\ell \mapsto \langle \rho, \lambda^\ell x.e \rangle] \mid \xi^\ell : \kappa}$$

(DT-FunctionApplication)

$$\frac{\mathcal{H}, \rho, \kappa \vdash e_0 \Downarrow \mathcal{H}' \mid \xi^\ell : \kappa_0 \qquad \langle \mathcal{P}, \langle \dot{\rho}, \lambda^\ell x.e \rangle \rangle = \mathcal{H}'(\xi^\ell) \qquad \mathcal{H}', \rho, \kappa \vdash e_1 \Downarrow \mathcal{H}'' \mid v_1 : \kappa_1 \qquad \mathcal{H}'', \dot{\rho}[x \mapsto v_1 : \kappa_1], \kappa \bullet \kappa_0 \vdash e \Downarrow \mathcal{H}''' \mid v : \kappa_v}{\mathcal{H}, \rho, \kappa \vdash e_0(e_1) \Downarrow \mathcal{H}''' \mid v : \kappa_v}$$

(DT-ObjectCreation)

$$\frac{\xi^\ell \notin dom(\mathcal{H})}{\mathcal{H}, \rho, \kappa \vdash \mathbf{new}^\ell \Downarrow \mathcal{H}[\xi^\ell \mapsto \emptyset] \mid \xi^\ell : \kappa}$$

(DT-PropertyReference)

$$\frac{\mathcal{H}, \rho, \kappa \vdash e_0 \Downarrow \mathcal{H}' \mid \xi^\ell : \kappa_{\xi^\ell} \quad \mathcal{H}', \rho, \kappa \vdash e_1 \Downarrow \mathcal{H}'' \mid str : \kappa_{str}}{\mathcal{H}, \rho, \kappa \vdash e_0.e_1 \Downarrow \mathcal{H}'' \mid \mathcal{H}''(\xi^\ell)(str) \bullet \kappa_{\xi^\ell} \bullet \kappa_{str}}$$

(DT-PropertyAssignment)

$$\frac{\mathcal{H}, \rho, \kappa \vdash e_0 \Downarrow \mathcal{H}' \mid \xi^\ell : \kappa_{\xi^\ell} \quad \mathcal{H}', \rho, \kappa \vdash e_1 \Downarrow \mathcal{H}'' \mid str : \kappa_{str} \quad \mathcal{H}'', \rho, \kappa \vdash e_2 \Downarrow \mathcal{H}''' \mid v : \kappa_v}{\mathcal{H}, \rho, \kappa \vdash e_0.e_1 = e_2 \Downarrow \mathcal{H}'''[\xi^\ell, str \mapsto v : \kappa_v \bullet \kappa_{\xi^\ell} \bullet \kappa_{str}] \mid v : \kappa_v}$$

UNI FREIBURG

(DT-ConditionTrue)

$$\frac{v_0 = \mathit{true} \qquad \mathcal{H}, \rho, \kappa \vdash e_0 \Downarrow \mathcal{H}' \mid v_0 : \kappa_0 \qquad \mathcal{H}', \rho, \kappa \bullet \kappa_0 \vdash e_1 \Downarrow \mathcal{H}''_1 \mid v_1 : \kappa_1}{\mathcal{H}, \rho, \kappa \vdash \textbf{if}\ (e_0)\ e_1,\ e_2 \Downarrow \mathcal{H}''_1 \mid v_1 : \kappa_1}$$

(DT-ConditionFalse)

$$\frac{v_0 \neq \mathit{true} \qquad \mathcal{H}, \rho, \kappa \vdash e_0 \Downarrow \mathcal{H}' \mid v_0 : \kappa_0 \qquad \mathcal{H}', \rho, \kappa \bullet \kappa_0 \vdash e_2 \Downarrow \mathcal{H}''_2 \mid v_2 : \kappa_2}{\mathcal{H}, \rho, \kappa \vdash \textbf{if}\ (e_0)\ e_1,\ e_2 \Downarrow \mathcal{H}''_2 \mid v_2 : \kappa_2}$$

UNI FREIBURG

(DT-Sequence)

$$\frac{\mathcal{H}, \rho, \kappa \vdash e_0 \Downarrow \mathcal{H}' \mid v_0 : \kappa_0 \quad \mathcal{H}', \rho, \kappa \vdash e_1 \Downarrow \mathcal{H}'' \mid v_1 : \kappa_1}{\mathcal{H}, \rho, \kappa \vdash e_0; e_1 \Downarrow \mathcal{H}'' \mid v_1 : \kappa_1}$$

(DT-Trace)

$$\frac{\mathcal{H}, \rho, \kappa \bullet \iota \vdash e \Downarrow \mathcal{H}' \mid v : \kappa_v}{\mathcal{H}, \rho, \kappa \vdash \mathbf{trace}^{\iota}\ (e) \Downarrow \mathcal{H}' \mid v : \kappa_v}$$

Lattice Value

$$
\begin{array}{lcl}
\mathit{Undefined} & ::= & \{\texttt{undefined}, \bot\} \\
\mathit{Null} & ::= & \{\texttt{null}, \bot\} \\
\mathit{Bool} & ::= & \{\texttt{true}, \texttt{false}, \bot\} \\
\mathit{Infinity} & ::= & \{+\texttt{Infinity}, -\texttt{Infinity}\} \\
\mathit{UInt} & ::= & \{0 \ldots 4294967295\} \\
\mathit{NotUInt} & ::= & \{\ldots, -1, -1.1, 1.1, \ldots\} \\
\mathit{Num} & ::= & \mathit{Infinity} \cup \mathit{UInt} \cup \mathit{NotUInt} \cup \{\texttt{NaN}, \bot\} \\
\mathit{UIntString} & ::= & \{''0'' \ldots '' 4294967295''\} \\
\mathit{NotUIntString} & ::= & \{''a'', '' b'', \ldots\} \\
\mathit{String} & ::= & \mathit{UIntString} \cup \mathit{NotUIntString} \cup \{\bot\}
\end{array}
$$

$$
\mathit{Lattice\ Value} \ni \mathcal{L} ::= \mathit{Undefined} \times \mathit{Null} \times \mathit{Bool} \times \mathit{Num} \times \mathit{String}
$$

Abstract Semantic Domains

$$
\begin{array}{llll}
\textit{SourceLocation} & \ni \ell, \iota & ::= & \textit{sourcefile} \times \textit{linenumber} \\
\textit{Label} & \ni \Xi & ::= & \{\textit{SourceLocation} \ldots\} \\
 & & & \\
\textit{Abstract Closure} & \ni \Lambda^{\ell} & ::= & \textit{Scope} \times \textit{Function} \\
\textit{PropertyMap} & \ni \Delta & ::= & \textit{Lattice Value} \rightarrow \textit{Abstract Value} \\
 & & & \\
\textit{Abstract Value} & \ni \vartheta & ::= & \textit{Lattice Value} \times \textit{Label} \times \textit{Dependency} \\
\textit{Abstract Object} & \ni \theta & ::= & \textit{PropertyMap} \times \textit{Abstract Closure} \\
 & & & \\
\textit{FunctionStore} & \ni \mathcal{F} & ::= & \textit{State} \times \textit{Abstract Value} \times \\
 & & & \textit{State} \times \textit{Abstract Value} \\
\textit{Scope} & \ni \sigma & ::= & \textit{Variable} \rightarrow \textit{Abstract Value} \\
\textit{State} & \ni \Gamma & ::= & (\textit{SourceLocation} \rightarrow \textit{Abstract Object}) \\
 & & & \times \textit{Dependency}
\end{array}
$$

$$
\begin{array}{lcll}
\textit{Trace} & \ni \tau & ::= & \textit{SourceLocation} \to \wp(\textit{Abstract Value}) \\
\textit{Dependency} & \ni \mathcal{D} & ::= & \emptyset \\
 & & \mid & \tau \\
 & & \mid & \mathcal{D}_0 \sqcup \mathcal{D}_1
\end{array}
$$

$$
\begin{aligned}
\circledv(\iota) &= \tau_\iota && (3.1) \\
\circledv(\iota, \vartheta) &\equiv \langle \mathcal{V}(\vartheta), \mathcal{D}_\vartheta \sqcup \tau_\iota \rangle && (3.2)
\end{aligned}
$$

UNI FREIBURG

$$\Gamma, \sigma \vdash e \Downarrow \Gamma' \mid \vartheta \tag{3.3}$$

(A-Constant)

$$\frac{}{\Gamma, \sigma \vdash c \Downarrow \Gamma \mid \langle c, \emptyset, \mathcal{D}_\Gamma \rangle}$$

(A-Variable)

$$\frac{}{\Gamma, \sigma \vdash x \Downarrow \Gamma \mid \sigma(x) \sqcup \mathcal{D}_\Gamma}$$

(A-Let)

$$\frac{\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \vartheta_0 \qquad \Gamma', \sigma[x \mapsto \vartheta_0] \vdash e_1 \Downarrow \Gamma'' \mid \vartheta_1}{\Gamma, \sigma \vdash \textbf{let } (x = e_0) \textbf{ in } e_1 \Downarrow \Gamma'' \mid \vartheta_1}$$

(A-Operation)

$$\frac{\begin{array}{c}\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \vartheta_0 \\ \vdots \\ \Gamma^{n-1}, \sigma \vdash e_n \Downarrow \Gamma^n \mid \vartheta_n \\ \langle \mathcal{L}, \Xi \rangle = \Downarrow^{\vartheta}_{\mathbf{op}} (\mathcal{V}(\vartheta_0) \ldots \mathcal{V}(\vartheta_n))\end{array}}{\Gamma, \sigma \vdash \mathbf{op}(e_0 \ldots e_n) \Downarrow \Gamma^n \mid \langle \mathcal{L}, \Xi, \mathcal{D}_{\vartheta_0} \sqcup \cdots \sqcup \mathcal{D}_{\vartheta_n} \rangle}$$

(A-ObjectCreation1)

$$\frac{\ell \notin \mathit{dom}(\Gamma)}{\Gamma, \sigma \ \vdash \ \mathbf{new}^{\ell} \ \Downarrow \ \Gamma[\ell \mapsto \emptyset] \mid \langle \mathcal{L}_{\perp}, \{\ell\}, \mathcal{D}_{\Gamma} \rangle}$$

(A-ObjectCreation2)

$$\frac{\ell \in \mathit{dom}(\Gamma)}{\Gamma, \sigma \ \vdash \ \mathbf{new}^{\ell} \ \Downarrow \ \Gamma \mid \langle \mathcal{L}_{\perp}, \{\ell\}, \mathcal{D}_{\Gamma} \rangle}$$

(A-FunctionCreation1)

$$\frac{\ell \notin \mathit{dom}(\Gamma) \qquad \mathcal{F}[\ell \mapsto \langle \Gamma_\perp, \vartheta_\perp, \Gamma_\perp, \vartheta_\perp \rangle]}{\Gamma, \sigma \;\vdash\; \lambda^\ell x.e \;\Downarrow\; \Gamma[\ell \mapsto \langle \sigma, \lambda^\ell x.e \rangle] \mid \langle \mathcal{L}_\perp, \{\ell\}, \mathcal{D}_\Gamma \rangle}$$

(A-FunctionCreation2)

$$\frac{\ell \in \mathit{dom}(\Gamma) \qquad \langle \dot{\sigma}, \lambda^\ell x.e \rangle = \Gamma(\ell)_{\Lambda^\ell}}{\Gamma, \sigma \;\vdash\; \lambda^\ell x.e \;\Downarrow\; \Gamma[\ell \mapsto \langle \sigma \sqcup \dot{\sigma}, \lambda^\ell x.e \rangle] \mid \langle \mathcal{L}_\perp, \{\ell\}, \mathcal{D}_\Gamma \rangle}$$

(A-FunctionApplication)

$$\frac{\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \langle \mathcal{L}_0, \Xi_0, \mathcal{D}_0 \rangle \quad \Gamma', \sigma \vdash e_1 \Downarrow \Gamma'' \mid \vartheta_1 \quad \Gamma''[\mathcal{D} \mapsto \mathcal{D}_{\Gamma''} \sqcup \mathcal{D}_0] \vdash^{\Pi}_{\mathbf{FA}} \Gamma'(\Xi_0), \vartheta_1 \Downarrow \Gamma''' \mid \vartheta}{\Gamma, \sigma \vdash e_0(e_1) \Downarrow \langle \mathcal{S}(\Gamma'''), \mathcal{D}_\Gamma \rangle \mid \vartheta}$$

(FA-Iteration1)

$$\frac{\Gamma \vdash^{\Lambda^\ell}_{\mathbf{FA}} \Lambda^\ell, \vartheta \Downarrow \Gamma' \mid \vartheta' \quad \Gamma' \vdash^{\Pi}_{\mathbf{FA}} \Pi, \vartheta \Downarrow \Gamma'' \mid \vartheta''}{\Gamma \vdash^{\Pi}_{\mathbf{FA}} \Lambda^\ell; \Pi, \vartheta \Downarrow \Gamma'' \mid \vartheta' \sqcup \vartheta''}$$

(FA-Iteration2)

$$\frac{}{\Gamma \vdash^{\Pi}_{\mathbf{FA}} \emptyset, \vartheta \Downarrow \Gamma \mid \vartheta_\bot}$$

(FA-FunctionStore1)

$$\frac{\langle \Gamma, \vartheta \rangle \sqsubseteq \mathcal{F}(\Lambda^\ell)_{\Lambda^\ell_{In}} \qquad \langle \Gamma', \vartheta' \rangle = \mathcal{F}(\ell)_{\Lambda^\ell_{Out}}}{\Gamma \vdash^{\Lambda^\ell}_{\mathbf{FA}} \Lambda^\ell, \vartheta \Downarrow \Gamma' \mid \vartheta'}$$

(FA-FunctionStore2)

$$\frac{\begin{array}{cc} \langle \Gamma, \vartheta \rangle \not\sqsubseteq \mathcal{F}(\Lambda^\ell)_{\Lambda^\ell_{In}} & \langle \dot{\sigma}, \lambda^\ell x.e \rangle = \theta_{\Lambda^\ell} \\ \langle \bar{\Gamma}, \bar{\vartheta} \rangle = \mathcal{F}(\ell)_{\Lambda^\ell_{In}} \sqcup \langle \Gamma, \vartheta \rangle & \mathcal{F}[\ell, \Lambda^\ell_{In} \mapsto \langle \bar{\Gamma}, \bar{\vartheta} \rangle] \\ \bar{\Gamma}, \dot{\sigma}[x \mapsto \bar{\vartheta}] \vdash e \Downarrow \bar{\Gamma}' \mid \bar{\vartheta}' & \mathcal{F}[\ell, \Lambda^\ell_{Out} \mapsto \langle \bar{\Gamma}', \bar{\vartheta}' \rangle] \end{array}}{\Gamma \vdash^{\Lambda^\ell}_{\mathbf{FA}} \Lambda^\ell, \vartheta \Downarrow \bar{\Gamma}' \mid \bar{\vartheta}'}$$

(A-PropertyReference)

$$\frac{\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \langle \mathcal{L}_0, \Xi_0, \mathcal{D}_0 \rangle \quad \Gamma', \sigma \vdash e_1 \Downarrow \Gamma'' \mid \langle \mathtt{str}, \Xi_1, \mathcal{D}_1 \rangle \quad \vdash^{\Theta}_{\mathbf{PR}} \Gamma''(\Xi_0), \mathtt{str} \Downarrow \vartheta}{\Gamma, \sigma \vdash e_0.e_1 \Downarrow \Gamma'' \mid \langle \mathcal{V}(\vartheta), \mathcal{D}_0 \sqcup \mathcal{D}_1 \sqcup \mathcal{D}_\vartheta \rangle}$$

(PR-Iteration1)

$$\frac{\vdash^{\Delta}_{\mathbf{PR}} \Delta, \mathcal{L} \Downarrow \vartheta \quad \vdash^{\Theta}_{\mathbf{PR}} \Theta, \mathcal{L} \Downarrow \vartheta'}{\vdash^{\Theta}_{\mathbf{PR}} \langle \Delta, \Lambda^{\ell} \rangle : \Theta, \mathcal{L} \Downarrow \vartheta \sqcup \vartheta'}$$

(PR-Iteration2)

$$\frac{}{\vdash^{\Theta}_{\mathbf{PR}} \emptyset, \mathcal{L} \Downarrow \vartheta_{\bot}}$$

(PR-Intersection)

$$\frac{(\mathcal{L} \sqcap \mathcal{L}_i \neq \bot) \quad \vdash^{\Theta}_{\mathbf{PR}} \ \Delta, \mathcal{L} \ \Downarrow \ \vartheta'}{\vdash^{\Delta}_{\mathbf{PR}} \ (\mathcal{L}_i : \vartheta_i); \Delta, \mathcal{L} \ \Downarrow \ \vartheta_i \sqcup \vartheta'}$$

(PR-NonIntersection)

$$\frac{\mathcal{L} \sqcap \mathcal{L}_i = \bot \quad \vdash^{\Theta}_{\mathbf{PR}} \ \Delta, \mathcal{L} \ \Downarrow \ \vartheta'}{\vdash^{\Delta}_{\mathbf{PR}} \ (\mathcal{L}_i : \vartheta_i); \Delta, \mathcal{L} \ \Downarrow \ \vartheta'}$$

(PR-Empty)

$$\frac{}{\vdash^{\Delta}_{\mathbf{PR}} \ \emptyset, \mathcal{L} \ \Downarrow \ \vartheta_{undef}}$$

(A-PopertyAssignment)

$$\frac{\begin{array}{c}\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \langle \mathcal{L}_0, \Xi_0, \mathcal{D}_0 \rangle \\ \Gamma', \sigma \vdash e_1 \Downarrow \Gamma'' \mid \langle \mathtt{str}, \Xi_1, \mathcal{D}_1 \rangle \\ \Gamma'', \sigma \vdash e_2 \Downarrow \Gamma''' \mid \vartheta \\ \Gamma''' \vdash^{\overline{\Xi}}_{\mathbf{PA}} \Xi_0, \mathtt{str}, \langle \mathcal{V}(\vartheta), \mathcal{D}_0 \sqcup \mathcal{D}_1 \sqcup \mathcal{D}_\vartheta \rangle \Downarrow \Gamma''''\end{array}}{\Gamma, \sigma \vdash e_0.e_1 = e_2 \Downarrow \Gamma'''' \mid \vartheta}$$

(PA-Iteration1)

$$\frac{\begin{array}{c}\Gamma \vdash^{\ell}_{\mathbf{PA}} \ell, \mathcal{L}, \vartheta \Downarrow \Gamma' \\ \Gamma' \vdash^{\overline{\Xi}}_{\mathbf{PA}} \Xi, \mathcal{L}, \vartheta \Downarrow \Gamma''\end{array}}{\Gamma \vdash^{\overline{\Xi}}_{\mathbf{PA}} \ell; \Xi, \mathcal{L}, \vartheta \Downarrow \Gamma''}$$

(PA-Iteration2)

$$\frac{}{\Gamma \vdash^{\overline{\Xi}}_{\mathbf{PA}} \emptyset, \mathcal{L}, \vartheta \Downarrow \Gamma}$$

(PA-Assignment1)

$$\frac{\mathcal{L} \in dom(\Gamma(l))}{\Gamma \vdash^{\ell}_{\mathbf{PA}} \ell, \mathcal{L}, \vartheta \Downarrow \Gamma[\ell, \mathcal{L} \mapsto \Gamma(\ell)(\mathcal{L}) \sqcup \vartheta]}$$

(PA-Assignment2)

$$\frac{\mathcal{L} \notin dom(\Gamma(\ell))}{\Gamma \vdash^{\ell}_{\mathbf{PA}} \ell, \mathcal{L}, \vartheta \Downarrow \Gamma[\ell, \mathcal{L} \mapsto \vartheta]}$$

(A-ConditionTrue)

$$\frac{\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \langle \mathcal{L}_0, \Xi_0, \mathcal{D}_0 \rangle \qquad \mathcal{L}_0 = \texttt{true} \qquad \Gamma'[\mathcal{D} \mapsto \mathcal{D}_{\Gamma'} \sqcup \mathcal{D}_0], \sigma \vdash e_1 \Downarrow \Gamma'' \mid \vartheta_1}{\Gamma, \sigma \vdash \textbf{if}\ (e_0)\ e_1,\ e_2 \Downarrow \langle \mathcal{S}(\Gamma''), \mathcal{D}_\Gamma \rangle \mid \vartheta_1}$$

(A-ConditionFalse)

$$\frac{\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \langle \mathcal{L}_0, \Xi_0, \mathcal{D}_0 \rangle \qquad \mathcal{L}_0 = \texttt{false} \qquad \Gamma'[\mathcal{D} \mapsto \mathcal{D}_{\Gamma'} \sqcup \mathcal{D}_0], \sigma \vdash e_2 \Downarrow \Gamma'' \mid \vartheta_2}{\Gamma, \sigma \vdash \textbf{if}\ (e_0)\ e_1,\ e_2 \Downarrow \langle \mathcal{S}(\Gamma''), \mathcal{D}_\Gamma \rangle \mid \vartheta_2}$$

(A-Condition)

$$\frac{\begin{array}{c}\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \langle \mathcal{L}_0, \Xi_0, \mathcal{D}_0 \rangle \\ \mathcal{L}_0 \neq \texttt{true} \wedge \mathcal{L}_0 \neq \texttt{false} \\ \Gamma'[\mathcal{D} \mapsto \mathcal{D}_{\Gamma'} \sqcup \mathcal{D}_0], \sigma \vdash e_1 \Downarrow \Gamma_1'' \mid \vartheta_1 \\ \Gamma'[\mathcal{D} \mapsto \mathcal{D}_{\Gamma'} \sqcup \mathcal{D}_0], \sigma \vdash e_2 \Downarrow \Gamma_2'' \mid \vartheta_2\end{array}}{\Gamma, \sigma \vdash \textbf{if}\ (e_0)\ e_1,\ e_2 \Downarrow \langle \mathcal{S}(\Gamma_1'' \sqcup \Gamma_2''), \mathcal{D}_\Gamma \rangle \mid \vartheta_1 \sqcup \vartheta_2}$$

(A-Sequence)

$$\frac{\Gamma, \sigma \vdash e_0 \Downarrow \Gamma' \mid \vartheta_0 \quad \Gamma', \sigma \vdash e_1 \Downarrow \Gamma'' \mid \vartheta_1}{\Gamma, \sigma \vdash e_0; e_1 \Downarrow \Gamma'' \mid \vartheta_1}$$

(A-Trace)

$$\frac{\tau_\iota = \circledcirc(\iota) \quad \Gamma[\mathcal{D} \mapsto \mathcal{D}_\Gamma \sqcup \tau_\iota], \sigma \vdash e \Downarrow \Gamma' \mid \vartheta}{\Gamma, \sigma \vdash \mathbf{trace}^\iota\ (e) \Downarrow \langle \mathcal{S}(\Gamma'), \mathcal{D}_\Gamma \rangle \mid \vartheta}$$

UNI FREIBURG

1. Noninterference on $\lambda^{\mathcal{D}}_{JS}$
2. Correctness ($\mathcal{C}$-Consistency)
3. Termination

UNI FREIBURG

$$\mathcal{H}, \rho, \kappa \vdash e \Downarrow \mathcal{H}' \mid v : \kappa_v \tag{4.1}$$

$$\iota \notin \kappa_v : \mathcal{H}, \rho, \kappa \vdash \bar{e} \Downarrow \tilde{\mathcal{H}}' \mid v : \kappa_v \tag{4.2}$$

$$\bar{e} = e[\iota \mapsto \tilde{e}] \tag{4.3}$$

UNI FREIBURG

Definition (Substitution of ι)

The substitution $e[\iota \mapsto \tilde{e}]$ of ι in e is defined as:

$$\forall e' \in \mathit{SubExp}(e) : \quad e'[\iota \mapsto \tilde{e}] \tag{4.4}$$

$$\mathbf{trace}^{\iota}(e_{\iota})[\iota \mapsto \tilde{e}] \equiv \mathbf{trace}^{\iota}(\tilde{e}) \tag{4.5}$$

Termination-Insensitive Noninterference

Definition (Bijection of ξ^{ℓ})

The bijection $\flat : \mathit{Location} \to \mathit{Location}$ from location ξ^{ℓ} to location $\xi^{\ell\prime}$ maps permutations on heap entries.

$$\flat \ ::= \ \emptyset \mid \flat[\xi^{\ell} \mapsto \xi^{\ell\prime}] \tag{4.6}$$

Definition (Bijection of v)

The bijection $\flat$ for values is defined as:

$$\flat(v) \ ::= \ \begin{cases} \flat(\xi^{\ell}) & v = \xi^{\ell} \\ v & v \neq \xi^{\ell} \end{cases} \tag{4.7}$$

Definition (κ-equivalence)

Two heaps $\mathcal{H}_0$,$\mathcal{H}_1$ are κ-equivalent $\mathcal{H}_0 \equiv_{\flat,\kappa} \mathcal{H}_1$ iff

$$\forall \xi^{\ell} \in \mathit{dom}(\flat) : \ \mathcal{H}_0(\xi^{\ell}) \equiv_{\flat,\kappa} \mathcal{H}_1(\flat(\xi^{\ell})) \tag{4.8}$$

The heaps $\mathcal{H}_0$,$\mathcal{H}_1$ only differ in values $v : \kappa_v$ with any intersection with κ or in one-sided locations.

Definition (κ-equivalence)

Two objects o_0, o_1 are κ-equivalent
$\langle \mathcal{P}_0, \langle \rho_0, \lambda^\ell x.e_0 \rangle \rangle \equiv_{\flat,\kappa} \langle \mathcal{P}_1, \langle \rho_1, \lambda^\ell x.e_1 \rangle \rangle$ iff

$$\begin{aligned}
&\forall str \in dom(\mathcal{P}_0) : \\
&str \in dom(\mathcal{P}_1) \ \wedge \ \mathcal{P}_0(str) = \mathcal{P}_1(str) \ \vee \qquad (4.9)\\
&\mathcal{P}_0(str) = v : \kappa_v \ \wedge \ \kappa \cap \kappa_v \neq \emptyset \\
&\forall str \in dom(\mathcal{P}_1) : \\
&str \in dom(\mathcal{P}) \ \wedge \ \mathcal{P}_0(str) = \mathcal{P}_1(str) \ \vee \qquad (4.10)\\
&\mathcal{P}_1(str) = v : \kappa_v \ \wedge \ \kappa \cap \kappa_v \neq \emptyset \\
&\rho_0 \equiv_{\flat,\kappa} \rho_1 \ \wedge \ \lambda^\ell x.e_0 \equiv_{\flat,\kappa} \lambda^\ell x.e_1 \qquad (4.11)
\end{aligned}$$

The objects o_0, o_1 only differ in values $v : \kappa_v$ with any intersection with κ.

Definition (κ-equivalence)

Two environments ρ_0,ρ_1 are κ-equivalent $\rho_0 \equiv_{\flat,\kappa} \rho_1$ iff

$$\begin{aligned} &\forall x \in dom(\rho_0) : \\ &x \in dom(\rho_1) \ \wedge \ \rho_0(x) = \rho_1(x) \ \vee \qquad (4.12) \\ &\rho_0(x) = v : \kappa_v \ \wedge \ \kappa \cap \kappa_v \neq \emptyset \\ &\forall x \in dom(\rho_1) : \\ &x \in dom(\rho_0) \ \wedge \ \rho_0(x) = \rho_1(x) \ \vee \qquad (4.13) \\ &\rho_1(x) = v : \kappa_v \ \wedge \ \kappa \cap \kappa_v \neq \emptyset \end{aligned}$$

The environments ρ_0,ρ_1 only differ in values $v : \kappa_v$ with any intersection with κ.

Definition (κ-equivalence)

Two value ω_0,ω_1 are κ-equivalent $v_0 : \kappa_0 \equiv_{\flat,\kappa} v_1 : \kappa_1$ iff

$$\kappa \cap \kappa_0 = \emptyset \ \wedge \ \kappa \cap \kappa_1 = \emptyset \ \rightarrow \ \flat(v_0) = v_1 \qquad (4.14)$$

The values ω_0,ω_1 only differ in the case of any intersection with κ.

Definition (κ-equivalence)

Two expressions e_0,e_1 are κ-equivalent $e_0 \equiv_{\flat,\kappa} e_1$ iff

$$\kappa = \{\iota_0, ..., \iota_n\} \rightarrow \exists e'_0 \ldots \exists e'_n : \; e_0 = e_1[\iota_0 \mapsto e'_0] \ldots [\iota_n \mapsto e'_n] \tag{4.15}$$

The expressions e_0,e_1 only differ below $\mathbf{trace}^{\iota}(e_\iota)$ subexpressions with $\iota \in \kappa$.

Theorem (Context Dependency)

We assume that $\forall \mathcal{H}, \rho, \kappa, e: \ \mathcal{H}, \rho, \kappa \vdash e \Downarrow \mathcal{H}' \mid v : \kappa_v$ *implies that* $\kappa \subseteq \kappa_v$.

Theorem (Noninterference)

We assume $\forall \bar{\kappa}, \forall \flat$ *that*
$\forall \mathcal{H}, \tilde{\mathcal{H}}, \rho, \tilde{\rho}, \kappa, e: \ \mathcal{H}, \rho, \kappa \vdash e \Downarrow \mathcal{H}' \mid v : \kappa_v$.
If $\iota \notin \bar{\kappa}$ *and* $\mathcal{H} \equiv_{\flat,\{\iota \mid \iota \notin \bar{\kappa}\}} \tilde{\mathcal{H}}$ *and* $\rho \equiv_{\flat,\{\iota \mid \iota \notin \bar{\kappa}\}} \tilde{\rho}$ *then*
$\tilde{\mathcal{H}}, \tilde{\rho}, \kappa \vdash \bar{e} \Downarrow \tilde{\mathcal{H}}' \mid \tilde{v} : \tilde{\kappa}_v$ *with* $\bar{e} = e[\iota \mapsto \tilde{e}]$ *and* $e \equiv_{\flat,\{\iota \mid \iota \notin \bar{\kappa}\}} \bar{e}$
and $\mathcal{H}' \equiv_{\flat,\{\iota \mid \iota \notin \bar{\kappa}\}} \tilde{\mathcal{H}}'$ *and* $v : \kappa_v \equiv_{\flat,\{\iota \mid \iota \notin \bar{\kappa}\}} \tilde{v} : \tilde{\kappa}_v$.

$\forall e$:

$$\mathcal{H}, \rho, \kappa \vdash e \Downarrow \mathcal{H}' \mid \omega \tag{4.16}$$

$$\Gamma, \sigma \vdash e \Downarrow \Gamma' \mid \vartheta \tag{4.17}$$

$$\begin{aligned} &\mathcal{H} \prec_{\mathcal{C}} \Gamma \ \wedge \ \rho \prec_{\mathcal{C}} \sigma \ \wedge \ \kappa \prec_{\mathcal{C}} \mathcal{D}_{\Gamma} \ \rightarrow \\ &\mathcal{H}' \prec_{\mathcal{C}} \Gamma' \ \wedge \ \omega \prec_{\mathcal{C}} \vartheta \end{aligned} \tag{4.18}$$

Definition ($\mathcal{C}$-Consistency on dependencies $\kappa \prec_{\mathcal{C}} \mathcal{D}$)

$$\forall \iota \in \kappa : \ \tau_\iota \in \mathcal{D} \tag{4.19}$$

Definition ($\mathcal{C}$-Consistency on constants $c \prec_{\mathcal{C}} \mathcal{L}$)

$$c \in \mathcal{L} \tag{4.20}$$

Definition ($\mathcal{C}$-Consistency on location $\xi^\ell \prec_\mathcal{C} \Xi$)

$$\ell \in \Xi \tag{4.21}$$

Definition ($\mathcal{C}$-Consistency on values $\omega \prec_\mathcal{C} \vartheta$)

$$\kappa \prec_\mathcal{C} D \tag{4.22}$$

$$v \in \mathcal{V}(\vartheta) \ ::= \ \begin{cases} \ell \in \Xi, & v = \xi^\ell \\ c \in \mathcal{L}, & v = c \end{cases} \tag{4.23}$$

Definition ($\mathcal{C}$-Consistency on properties $\mathcal{P} \prec_{\mathcal{C}} \Delta$)

$$\forall str \in dom(\mathcal{P}) : \ \exists \mathcal{L} \in dom(\Delta) : \ str \in \mathcal{L} \ \wedge \ \mathcal{P}(str) \prec_{\mathcal{C}} \Delta(\mathcal{L}) \tag{4.24}$$

$$\forall str \notin dom(\mathcal{P}) : \ \textbf{undefined} \prec_{\mathcal{C}} \Delta(str) \tag{4.25}$$

Definition ($\mathcal{C}$-Consistency on objects $o \prec_{\mathcal{C}} \theta$)

$$\mathcal{P} \prec_{\mathcal{C}} \Delta \tag{4.26}$$

$$\rho \prec_{\mathcal{C}} \sigma \tag{4.27}$$

Definition ($\mathcal{C}$-Consistency on scopes $\rho \prec_{\mathcal{C}} \sigma$)

$$\forall x \in dom(\rho) : \; x \in dom(\sigma) \; \wedge \; \rho(x) \prec_{\mathcal{C}} \sigma(x) \tag{4.28}$$

Definition ($\mathcal{C}$-Consistency on heaps $\mathcal{H} \prec_{\mathcal{C}} \Gamma$)

$$\forall \xi^{\ell} \in dom(\mathcal{H}) : \; \ell \in \Sigma \; \wedge \; \mathcal{H}(\xi^{\ell}) \prec_{\mathcal{C}} \Sigma(\ell) \tag{4.29}$$

UNI FREIBURG

Lemma (Subset $\mathcal{C}$-Consistency)

$$\mathcal{H} \prec_{\mathcal{C}} \Gamma_0 \ \wedge \ \Gamma_0 \sqsubseteq \Gamma_1 \ \rightarrow \ \mathcal{H} \prec_{\mathcal{C}} \Gamma_1 \tag{4.30}$$

$$v : \kappa \prec_{\mathcal{C}} \vartheta_0 \ \wedge \ \vartheta_0 \sqsubseteq \vartheta_1 \ \rightarrow \ v : \kappa \prec_{\mathcal{C}} \vartheta_1 \tag{4.31}$$

Lemma ($\mathcal{C}$-Consistency on Property Update)

$$\forall o, \theta, \mathcal{L}, \vartheta \mid o \prec_{\mathcal{C}} \theta \ : \ o \prec_{\mathcal{C}} \theta[\mathcal{L} \mapsto \theta(\mathcal{L}) \sqcup \vartheta] \tag{4.32}$$

Theorem (Correctness Relation)

For all expressions e within the syntax of $\lambda^{\mathcal{D}}_{JS}$ *the following condition holds:* $\forall \mathcal{H}, \mathcal{H}', \rho, v, \kappa$: *If* $\mathcal{H}, \rho, \kappa \vdash e \Downarrow \mathcal{H}' \mid v$ *than* $\forall \Gamma, \sigma$ *with* $\mathcal{H} \prec_{\mathcal{C}} \Gamma$, $\rho \prec_{\mathcal{C}} \sigma$ *and* $\kappa \prec_{\mathcal{C}} \mathcal{D}_{\Gamma}$*:* $\Gamma, \sigma \vdash e \Downarrow \Gamma' \mid \vartheta$ *with* $\mathcal{H}' \prec_{\mathcal{C}} \Gamma'$ *and* $v \prec_{\mathcal{C}} \vartheta$.

UNI FREIBURG

Theorem (Termination)

$\Gamma, \sigma \vdash e \Downarrow \Gamma' \mid \vartheta$ *with arbitrary* e.

1. Monotony
2. Ascending chain condition

UNI FREIBURG

```
1 function fun() {
2     "Contract: a.b, a.b.c, a.?, a.b*.c"
3     var x = a.b;
4     a = {b:5};
5 }
```

- Dependency-based
  - *TAJS*, static dependency analysis
- *Contracts* instead of $\mathbf{trace}^{\iota}(e)$
- $\mathcal{C}$: `Contract: a.b [r,w];`

Evaluation

1. *trace* values ϑ / state Γ
2. create proof constraints $\mathcal{L}$
3. validate constraints $\mathcal{L}$

Constraint Based Proof constraints $\mathcal{L}$ at the end

Lazy Enforcement No direct enforcement of contracts $\mathcal{C}$

Dynamic Extent Nested contracts $\mathcal{C}$

Pre-State Snapshot $\Gamma, \sigma, \vartheta$ at $\mathcal{C}$

Read-Write Protection $\Gamma, \sigma, \vartheta$ at $\mathcal{C}$

$$
\begin{array}{lllll}
\textit{Contract} & \ni & \mathcal{C} & ::= & \emptyset \mid \mathcal{Q};\mathcal{C} \\
\textit{Permissions} & \ni & \mathcal{Q} & ::= & \langle \mathcal{A}, \Pi \rangle \\
 & & & & \\
\textit{AccessPath} & \ni & \mathcal{A} & ::= & \vec{v}.\mathcal{P} \\
\textit{Properties} & \ni & \mathcal{P} & ::= & \epsilon \mid \vec{p}.\mathcal{P} \mid \vec{p} * .\mathcal{P} \\
\textit{Variable} & \ni & \vec{v} & ::= & \{x \ldots\} \\
\textit{Property} & \ni & \vec{p} & ::= & \{x \ldots\} \\
 & & & & \\
\textit{PathPermission} & \ni & \Pi & ::= & \langle \pi_r, \pi_w \rangle \\
\textit{Readable} & \ni & \pi_r & ::= & \epsilon \mid r \\
\textit{Writeable} & \ni & \pi_w & ::= & \epsilon \mid w
\end{array}
$$

$$\begin{aligned} \mathit{ReadConstraint} &\ni \mathcal{R} \\ \mathit{WriteConstraint} &\ni \mathcal{W} \end{aligned}$$

(DT-Permit)

$$\frac{\begin{array}{c} \mathcal{H}, \rho, \kappa \vdash^{\mathcal{C}}_{\mathsf{Apply}} \mathcal{C}, \iota_{\mathcal{R}}, \iota_{\mathcal{W}} \Downarrow \mathcal{H}' \mid \rho' \mid \kappa' \mid \mathcal{L} \\ \mathcal{H}', \rho', \kappa' \vdash e \Downarrow \mathcal{H}'' \mid v : \kappa_v \\ \mathcal{H}'', \rho', \kappa_v \vdash^{\mathcal{C}}_{\mathsf{Check}} \mathcal{L} \end{array}}{\mathcal{H}, \rho, \kappa \vdash \mathbf{permit}^{\iota_{\mathcal{R}}, \iota_{\mathcal{W}}} \mathcal{C} \ \mathbf{in}\ e \Downarrow \mathcal{H}'' \mid v : \kappa_v}$$

UNI FREIBURG

Thank you for your attention.


T. Amtoft and A. Banerjee.
Information flow analysis in logical form.
Static Analysis, pages 33–36, 2004.


Torben Amtoft and Anindya Banerjee.
A logic for information flow analysis with an application to forward slicing of simple imperative programs.
Sci. Comput. Program., 64:3–28, January 2007.


Martín Abadi.
Access control in a core calculus of dependency.
Electron. Notes Theor. Comput. Sci., 172:5–31, April 2007.


Torben Amtoft, Sruthi Bandhakavi, and Anindya Banerjee.
A logic for information flow in object-oriented programs.
In *Conference record of the 33rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '06, pages 91–102, New York, NY, USA, 2006. ACM.


Martín Abadi, Anindya Banerjee, Nevin Heintze, and Jon 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, New York, NY, USA, 1999. ACM.


G. Balakrishnan and T. Reps.
Recency-abstraction for heap-allocated storage.
Static Analysis, pages 221–239, 2006.


P. Cousot and R. Cousot.
Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints.
In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252. ACM, 1977.


Dorothy E. Denning and Peter J. Denning.
Certification of programs for secure information flow.
Commun. ACM, 20:504–513, July 1977.


Dorothy E. Denning.
A lattice model of secure information flow.
Commun. ACM, 19:236–243, May 1976.


Jeanne Ferrante, Karl J. Ottenstein, and Joe D. Warren.
The program dependence graph and its use in optimization.
ACM Trans. Program. Lang. Syst., 9:319–349, July 1987.


Arjun Guha, Claudiu Saftoiu, and Shriram Krishnamurthi.
The essence of javascript.
In *Proceedings of the 24th European conference on Object-oriented programming*, ECOOP'10, pages 126–150, Berlin, Heidelberg, 2010. Springer-Verlag.


Phillip Heidegger, Annette Bieniusa, and Peter Thiemann.
Access permission contracts for scripting languages.
In *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '12, pages 111–122, New York, NY, USA, 2012. ACM.


Nevin Heintze and Jon G. Riecke.
The slam calculus: programming with secrecy and integrity.
In *Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '98, pages 365–377, New York, NY, USA, 1998. ACM.


ECMA International.

Standard ECMA-262, volume 3.
1999.


ECMA International.
Standard ECMA-262, volume 5.
2009.


Neil D. Jones and Steven S. Muchnick.
A flexible approach to interprocedural data flow analysis and programs with recursive data structures.
In *Proceedings of the 9th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '82, pages 66–74, New York, NY, USA, 1982. ACM.


Simon Holm Jensen, Magnus Madsen, and Anders Moller.
Modeling the HTML DOM and browser API in static analysis of JavaScript web applications.
In *Proc. 8th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE)*, September 2011.


Simon Holm Jensen, Anders Moller, and Peter Thiemann.

Type analysis for JavaScript.
In *Proc. 16th International Static Analysis Symposium, SAS '09*, volume 5673 of *LNCS*. Springer-Verlag, August 2009.


Simon Holm Jensen, Anders Moller, and Peter Thiemann.
Interprocedural analysis with lazy propagation.
In *Proc. 17th International Static Analysis Symposium (SAS)*, volume 6337 of *LNCS*. Springer-Verlag, September 2010.


J.B. Kam and J.D. Ullman.
Monotone data flow analysis frameworks.
Acta Informatica, 7(3):305–317, 1977.


Leo Meyerovich and Benjamin Livshits.
ConScript: Specifying and enforcing fine-grained security policies for Javascript in the browser.
In *IEEE Symposium on Security and Privacy*, May 2010.


Sergio Maffeis, John C. Mitchell, and Ankur Taly.
Isolating javascript with filters, rewriting, and wrappers.

In *Proceedings of the 14th European conference on Research in computer security*, ESORICS'09, pages 505–522, Berlin, Heidelberg, 2009. Springer-Verlag.


Andrew C. Myers.
Jflow: practical mostly-static information flow control.
In *Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '99, pages 228–241, New York, NY, USA, 1999. ACM.


Isabella Mastroeni and Damiano Zanardini.
Data dependencies and program slicing: from syntax to abstract semantics.
In *Proceedings of the 2008 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation*, PEPM '08, pages 125–134, New York, NY, USA, 2008. ACM.


F. Nielson, H.R. Nielson, and C. Hankin.
Principles of program analysis.
Springer-Verlag New York Inc, 1999.


Franccois Pottier and Vincent Simonet.
Information flow inference for ml.

In *Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '02, pages 319–330, New York, NY, USA, 2002. ACM.


Andrei Sabelfeld and Andrew C. Myers.
Language-based information-flow security.
IEEE Journal on Selected Areas in Communications, 21:2003, 2003.


M. Sharir and A. Pnueli.
Two approaches to interprocedural data flow analysis.
Program Flow Analysis: Theory and Applications, pages 189–234, 1981.


Peter Thiemann.
A prototype dependency calculus.
In *Proceedings of the 11th European Symposium on Programming Languages and Systems*, ESOP '02, pages 228–242, London, UK, UK, 2002. Springer-Verlag.


Dennis Volpano, Cynthia Irvine, and Geoffrey Smith.
A sound type system for secure flow analysis.
J. Comput. Secur., 4:167–187, January 1996.