Lógica de pruebas para certificación de computación móvil

Federico Feller

14 de octubre de 2009

Contenido

1. Introducción
   - Motivación
   - PCC
   - Modelo
2. Lógica de pruebas
   - Lógicas modales y LP
   - ILPnd
3. Cálculo para certificados de código móvil
   - Términos y certificados
   - Sistema de tipos
   - Semántica operacional
4. Propiedades
   - Seguridad de tipos
   - Normalización fuerte
5. Extensiones
   - Booleanos
   - Números naturales
   - Concatenación de certificados
6. Conclusiones
   - Trabajo a Futuro
   - Conclusiones

Contenido

1. Introducción
   - Motivación
   - PCC
   - Modelo
2. Lógica de pruebas
   - Lógicas modales y LP
   - ILPnd
3. Cálculo para certificados de código móvil
   - Términos y certificados
   - Sistema de tipos
   - Semántica operacional
4. Propiedades
   - Seguridad de tipos
   - Normalización fuerte
5. Extensiones
   - Booleanos
   - Números naturales
   - Concatenación de certificados
6. Conclusiones
   - Trabajo a Futuro
   - Conclusiones

Motivación

- Estudio de entornos distribuidos de computación, específicamente computaciones móviles (traslado de código)

Motivación

- Estudio de entornos distribuidos de computación, específicamente computaciones móviles (traslado de código)
- Productor de código : Escribe software que se compila y se transmite dentro de la red para su ejecución
- Consumidor de código: Recibe el código generado por el productor y lo ejecuta

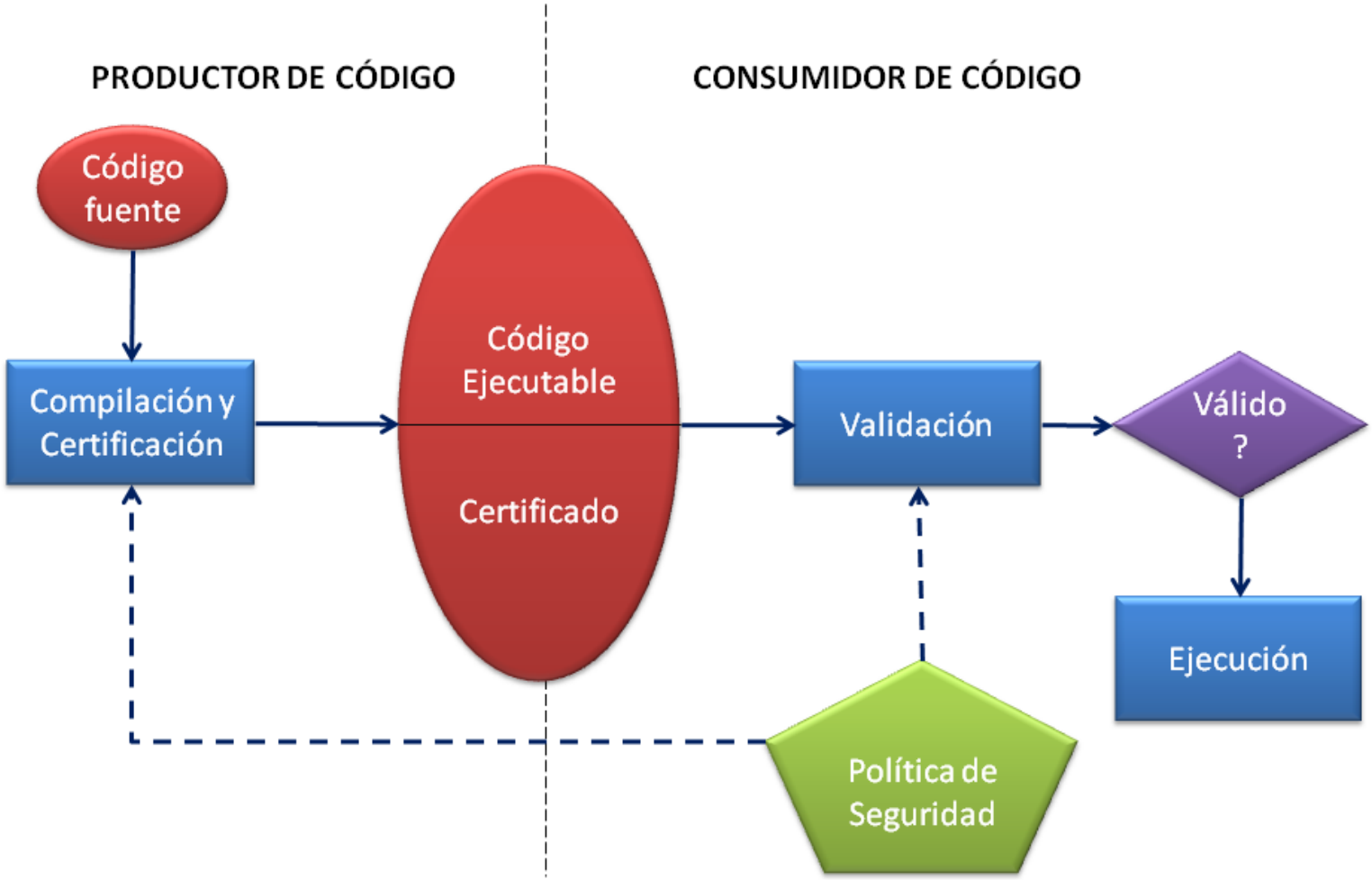
Motivación

- Estudio de entornos distribuidos de computación, específicamente computaciones móviles (traslado de código)
- Productor de código : Escribe software que se compila y se transmite dentro de la red para su ejecución
- Consumidor de código: Recibe el código generado por el productor y lo ejecuta
- Problemas de seguridad
  - El productor generar por error código que pone en riesgo al consumidor
  - El programa es capturado y modificado por un tercero mal intencionado

Motivación

- Estudio de entornos distribuidos de computación, específicamente computaciones móviles (traslado de código)
- Productor de código : Escribe software que se compila y se transmite dentro de la red para su ejecución
- Consumidor de código: Recibe el código generado por el productor y lo ejecuta
- Problemas de seguridad
  - El productor generar por error código que pone en riesgo al consumidor
  - El programa es capturado y modificado por un tercero mal intencionado
- Proof Carrying Code (PCC)

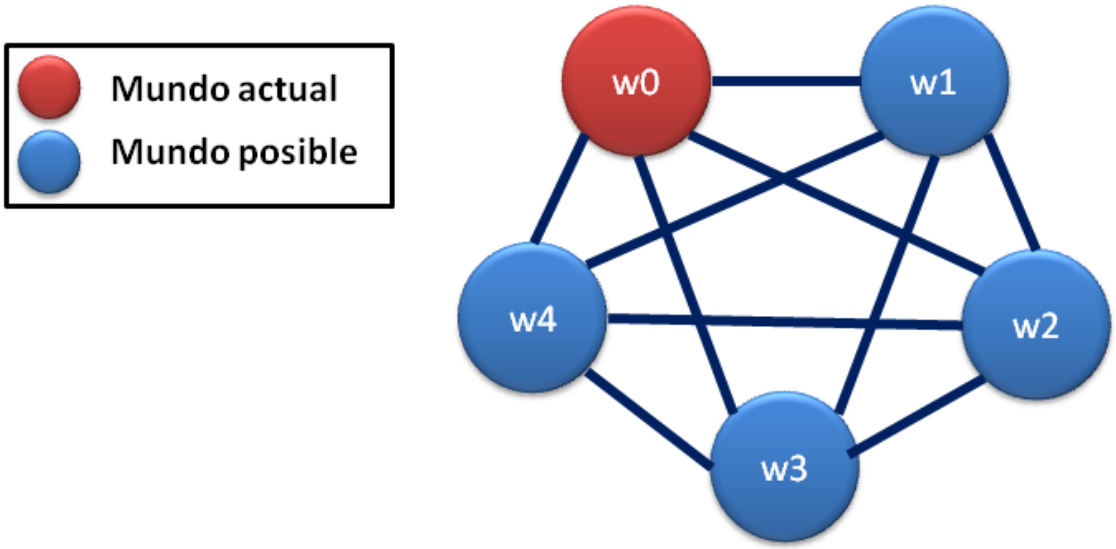
PCC


PRODUCTOR DE CÓDIGO
CONSUMIDOR DE CÓDIGO
Código fuente
Compilación y Certificación
Código Ejecutable
Certificado
Validación
Válido ?
Ejecución
Política de Seguridad

Objetivo

Estudiar un modelo de lenguaje para el productor en PCC en donde la generación de código y certificado se realiza en un marco unificado

Modelo I



- Conjunto finito de nodos denominados mundos
- Todos conectados entre sí
- En un momento, la ejecución ocurre en un mundo, pero puede trasladarse

Modelo II

La definición del modelo consiste en tres partes:

1. Lenguaje de programación recortado denominado $\lambda_{\square}^{\text{Cert}}$
2. Un sistema de tipos
3. Una semántica operacional que defina el comportamiento de los programas

Contenido

1 Introducción
- Motivación
- PCC
- Modelo

2 Lógica de pruebas
- Lógicas modales y LP
- ILPnd

3 Cálculo para certificados de código móvil
- Términos y certificados
- Sistema de tipos
- Semántica operacional

4 Propiedades
- Seguridad de tipos
- Normalización fuerte

5 Extensiones
- Booleanos
- Números naturales
- Concatenación de certificados

6 Conclusiones
- Trabajo a Futuro
- Conclusiones

Isomorfismo de Curry-DeBrujiun-Howard

- Para definir el cálculo y su sistema de tipos, se utiliza una técnica basada en el isomorfismo de Curry-DeBrujiun-Howard:

Lógica

$$A \supset B \supset A$$

$$\dfrac{\dfrac{A, B \vdash A}{A \vdash B \supset A}}{\vdash A \supset B \supset A}$$

Computación

$$A \to B \to A$$

$$\dfrac{\dfrac{x : A, y : B \vdash x : A}{x : A \vdash \lambda y.x : B \to A}}{\vdash \lambda x.\lambda y.x : A \to B \to A}$$

Lógicas modales

- El primer paso para definir el cálculo es la elección adecuada de la lógica base.

Lógicas modales

- El primer paso para definir el cálculo es la elección adecuada de la lógica base.
- Lógicas modales: Permiten razonar el valor de verdad de una proposición desde diferentes mundos posibles. Operador $\Box A$.

Lógicas modales

- El primer paso para definir el cálculo es la elección adecuada de la lógica base.
- Lógicas modales: Permiten razonar el valor de verdad de una proposición desde diferentes mundos posibles. Operador $\Box A$.
- Ya se han utilizado para modelar computaciones móviles:
  - Mundos posibles de la lógica como nodos de una red.
  - Proposición $\Box A$ como el tipo de un programa que representa código móvil y calcula un valor de tipo A.
  - La prueba de $\Box A$ se traduce a un programa que representa código móvil.
  - Ej: $\Box P \implies (\Box(P \implies Q) \implies \Box Q)$

Lógica de pruebas

- Las lógicas modales sirven para representar código móvil, falta inclusión de certificado
- Se ha optado por utilizar la lógica de pruebas o LP
- En LP, el operador $\Box A$ se reemplaza por $[s]A$ y se lee 's es una prueba de A'
  - Al término s se lo denomina polinomio de prueba y codifica la prueba de que A es verdadero.
- Se dice que LP es capaz de internalizar sus propias derivaciones:
  - Si $\vdash_{LP} A$ entonces $\vdash_{LP} [s]A$, para algún polinomio de prueba s
- Esta característica lo convierte en un candidato natural para la generación de certificados de código.

Lógica de pruebas II

Idea

Interpretar $[s]A$ como el tipo de un programa que representa código móvil y que calcula un valor de tipo A, con certificado s.

- Para definir $\lambda_{\Box}^{\mathsf{Cert}}$ se utiliza una versión intuicionista de LP denominada ILP
  - Visión análoga al uso de la lógica intuicionista para obtener el lambda cálculo simplemente tipado
- Se utiliza una representación en deducción natural de ILP denominada ILPnd.

ILPnd

- Representación en deducción natural de ILP
- Considera dos conjuntos de hipótesis: de verdad y de validez.
- Juicio hipotético con evidencia explícita:

$$\Delta; \Gamma \rhd A \mid s$$

- "A es verdadero con evidencia s bajo las hipótesis de validez Δ y las hipótesis de verdad Γ"

ILPnd

- Representación en deducción natural de ILP
- Considera dos conjuntos de hipótesis: de verdad y de validez.
- Juicio hipotético con evidencia explícita:

$$\Delta; \Gamma \rhd A \mid s$$

- "A es verdadero con evidencia s bajo las hipótesis de validez Δ y las hipótesis de verdad Γ"

$$\begin{array}{rcl} s, t & ::= & x \mid s \cdot t \mid \lambda a : A.s \mid !s \mid \text{LETC } s \text{ BE } v : A \text{ IN } t \\ A, B & ::= & P \mid A \Longrightarrow B \mid [s]A \\ \Gamma & ::= & \cdot \mid \Gamma, a : A \\ \Delta & ::= & \cdot \mid \Delta, v : A \end{array}$$

ILPnd II

Fragmento de la lógica proposicional minimal

$$\frac{}{\Delta; \Gamma, a : A, \Gamma' \rhd A \mid a}\ \text{oVar}$$

$$\frac{\Delta; \Gamma, a : A \rhd B \mid s}{\Delta; \Gamma \rhd A \implies B \mid \lambda a : A.s}\ \implies \text{I}$$

$$\frac{\Delta; \Gamma \rhd A \implies B \mid s \quad \Delta; \Gamma \rhd A \mid t}{\Delta; \Gamma \rhd B \mid s \cdot t}\ \implies \text{E}$$

ILPnd II

Fragmento de la lógica proposicional minimal

$$\frac{}{\Delta; \Gamma, a : A, \Gamma' \rhd A \mid a}\ \text{oVar}$$

$$\frac{\Delta; \Gamma, a : A \rhd B \mid s}{\Delta; \Gamma \rhd A \implies B \mid \lambda a : A.s} \implies \text{I}$$

$$\frac{\Delta; \Gamma \rhd A \implies B \mid s \quad \Delta; \Gamma \rhd A \mid t}{\Delta; \Gamma \rhd B \mid s \cdot t} \implies \text{E}$$

ILPnd II

Fragmento de la lógica proposicional minimal

$$\frac{}{\Delta; \Gamma, a : A, \Gamma' \rhd A \mid a}\ \text{oVar}$$

$$\frac{\Delta; \Gamma, a : A \rhd B \mid s}{\Delta; \Gamma \rhd A \implies B \mid \lambda a : A.s} \implies \text{I}$$

$$\frac{\Delta; \Gamma \rhd A \implies B \mid s \quad \Delta; \Gamma \rhd A \mid t}{\Delta; \Gamma \rhd B \mid s \cdot t} \implies \text{E}$$

ILPnd II

Fragmento de la lógica proposicional minimal

$$\frac{}{\Delta; \Gamma, a : A, \Gamma' \rhd A \mid a}\ \text{oVar}$$

$$\frac{\Delta; \Gamma, a : A \rhd B \mid s}{\Delta; \Gamma \rhd A \implies B \mid \lambda a : A.s} \implies \text{I}$$

$$\frac{\Delta; \Gamma \rhd A \implies B \mid s \quad \Delta; \Gamma \rhd A \mid t}{\Delta; \Gamma \rhd B \mid s \cdot t} \implies \text{E}$$

ILPnd II

Fragmento de demostrabilidad

$$\frac{}{\Delta, v : A, \Delta'; \Gamma \rhd A \mid v}\ \mathsf{mVar}$$

$$\frac{\Delta; \cdot \rhd A \mid s}{\Delta; \Gamma \rhd [s]A \mid !s}\ \Box\mathsf{I}$$

$$\frac{\Delta; \Gamma \rhd [r]A \mid s \quad \Delta, v : A; \Gamma \rhd C \mid t}{\Delta; \Gamma \rhd C\{v/r\} \mid \mathrm{LETC}\ s\ \mathrm{BE}\ v : A\ \mathrm{IN}\ t}\ \Box\mathsf{E}$$

ILPnd II

Fragmento de demostrabilidad

$$\frac{}{\Delta, v : A, \Delta'; \Gamma \rhd A \mid v}\ \mathsf{mVar}$$

$$\frac{\Delta; \cdot \rhd A \mid s}{\Delta; \Gamma \rhd [s]A \mid !s}\ \Box\mathsf{I}$$

$$\frac{\Delta; \Gamma \rhd [r]A \mid s \quad \Delta, v : A; \Gamma \rhd C \mid t}{\Delta; \Gamma \rhd C\{v/r\} \mid \mathrm{LETC}\ s\ \mathrm{BE}\ v : A\ \mathrm{IN}\ t}\ \Box\mathsf{E}$$

ILPnd II

Fragmento de demostrabilidad

$$\frac{}{\Delta, v : A, \Delta'; \Gamma \rhd A \mid v}\ \mathsf{mVar}$$

$$\frac{\Delta; \cdot \rhd A \mid s}{\Delta; \Gamma \rhd [s]A \mid !s}\ \Box\mathsf{I}$$

$$\frac{\Delta; \Gamma \rhd [r]A \mid s \quad \Delta, v : A; \Gamma \rhd C \mid t}{\Delta; \Gamma \rhd C\{v/r\} \mid \mathrm{LETC}\ s\ \mathrm{BE}\ v : A\ \mathrm{IN}\ t}\ \Box\mathsf{E}$$

ILPnd II

Fragmento de demostrabilidad

$$\frac{}{\Delta, v : A, \Delta'; \Gamma \rhd A \mid v}\ \mathsf{mVar}$$

$$\frac{\Delta; \cdot \rhd A \mid s}{\Delta; \Gamma \rhd [s]A \mid !s}\ \Box\mathsf{I}$$

$$\frac{\Delta; \Gamma \rhd [r]A \mid s \quad \Delta, v : A; \Gamma \rhd C \mid t}{\Delta; \Gamma \rhd C\{v/r\} \mid \mathrm{LETC}\ s\ \mathrm{BE}\ v : A\ \mathrm{IN}\ t}\ \Box\mathsf{E}$$

Contenido

1. Introducción
   - Motivación
   - PCC
   - Modelo
2. Lógica de pruebas
   - Lógicas modales y LP
   - ILPnd
3. Cálculo para certificados de código móvil
   - Términos y certificados
   - Sistema de tipos
   - Semántica operacional
4. Propiedades
   - Seguridad de tipos
   - Normalización fuerte
5. Extensiones
   - Booleanos
   - Números naturales
   - Concatenación de certificados
6. Conclusiones
   - Trabajo a Futuro
   - Conclusiones

Definición de $\lambda_{\square}^{\mathsf{Cert}}$

- La definición de $\lambda_{\square}^{\mathsf{Cert}}$ se basa en el concepto de unidad móvil.
- Una unidad móvil representa la unión tanto código como certificado
- Tiene tipo de la forma $[s]A$, siendo A el tipo del valor que calculan y s el certificado
- El lenguaje posee primitivas para:
  - Crear unidades móviles
  - Utilizar unidades móviles, permitiendo extraer sus partes (código y certificado)
  - Trasladar las unidades móviles de un mundo a otro
  - Referenciar mundos

Sintaxis de $\lambda^{\mathsf{Cert}}_{\Box}$

tipos	A	$::=$	$P \mid A \implies B \mid [s]A$
certificados	s, t	$::=$	$a \mid v^{\circ} \mid s \cdot t \mid \lambda a : A.s \mid !s$
		$\mid$	$letc\ s\ be\ v^{\circ} : A\ in\ t \mid fetch(s)$
valores	V	$::=$	$box_s\ M \mid \lambda a.M$
términos	M, N	$::=$	$a \mid v^{\bullet} \mid V \mid M\ N$
		$\mid$	$unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N$
		$\mid$	$fetch[w]\ M$

$$(\lambda a.unpack\ a\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ v^{\bullet})(fetch[w_1]\ box_s\ M)$$

$$\lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ} \rangle\ in\ (unpack\ b\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ (box_{u^{\circ}.v^{\circ}}\ u^{\bullet} v^{\bullet}))$$

Sintaxis de $\lambda^{\mathsf{Cert}}_{\Box}$

tipos	A	$::=$	$P \mid A \Longrightarrow B \mid [s]A$
certificados	s, t	$::=$	$a \mid v^{\circ} \mid s \cdot t \mid \lambda a : A.s \mid !s$
		$\mid$	$letc\ s\ be\ v^{\circ} : A\ in\ t \mid fetch(s)$
valores	V	$::=$	$box_s\ M \mid \lambda a.M$
términos	M, N	$::=$	$a \mid v^{\bullet} \mid V \mid M\ N$
		$\mid$	$unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N$
		$\mid$	$fetch[w]\ M$

$$(\lambda a.unpack\ a\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ v^{\bullet})(fetch[w_1]\ box_s\ M)$$

$$\lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ} \rangle\ in\ (unpack\ b\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ (box_{u^{\circ}.v^{\circ}}\ u^{\bullet} v^{\bullet}))$$

Sintaxis de $\lambda^{\mathsf{Cert}}_{\Box}$

tipos	A	$::=$	$P \mid A \Longrightarrow B \mid [s]A$
certificados	s, t	$::=$	$a \mid v^{\circ} \mid s \cdot t \mid \lambda a : A.s \mid !s$
		$\mid$	$letc\ s\ be\ v^{\circ} : A\ in\ t \mid fetch(s)$
valores	V	$::=$	$box_s\ M \mid \lambda a.M$
términos	M, N	$::=$	$a \mid v^{\bullet} \mid V \mid M\ N$
		$\mid$	$unpack\ M\ to\ \langle v^{\bullet}, v^{\circ}\rangle\ in\ N$
		$\mid$	$fetch[w]\ M$

$$(\lambda a.unpack\ a\ to\ \langle v^{\bullet}, v^{\circ}\rangle\ in\ v^{\bullet})(fetch[w_1]\ box_s\ M)$$

$$\lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ}\rangle\ in\ (unpack\ b\ to\ \langle v^{\bullet}, v^{\circ}\rangle\ in\ (box_{u^{\circ}.v^{\circ}}\ u^{\bullet}v^{\bullet}))$$

Sintaxis de $\lambda^{\mathsf{Cert}}_{\Box}$

tipos	A	$::=$	$P \mid A \implies B \mid [s]A$
certificados	s, t	$::=$	$a \mid v^{\circ} \mid s \cdot t \mid \lambda a : A.s \mid !s$
		$\mid$	$letc\ s\ be\ v^{\circ} : A\ in\ t \mid fetch(s)$
valores	V	$::=$	$box_s\ M \mid \lambda a.M$
términos	M, N	$::=$	$a \mid v^{\bullet} \mid V \mid M\ N$
		$\mid$	$unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N$
		$\mid$	$fetch[w]\ M$

$$(\lambda a.unpack\ a\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ v^{\bullet})(fetch[w_1]\ box_s\ M)$$

$$\lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ} \rangle\ in\ (unpack\ b\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ (box_{u^{\circ}.v^{\circ}}\ u^{\bullet} v^{\bullet}))$$

Sintaxis de $\lambda^{\mathsf{Cert}}_{\Box}$

tipos	A	$::=$	$P \mid A \implies B \mid [s]A$
certificados	s, t	$::=$	$a \mid v^{\circ} \mid s \cdot t \mid \lambda a : A.s \mid !s$
		$\mid$	$letc\ s\ be\ v^{\circ} : A\ in\ t \mid fetch(s)$
valores	V	$::=$	$box_s\, M \mid \lambda a.M$
términos	M, N	$::=$	$a \mid v^{\bullet} \mid V \mid M\, N$
		$\mid$	$unpack\ M\ to\ \langle v^{\bullet}, v^{\circ}\rangle\ in\ N$
		$\mid$	$fetch[w]\, M$

$$(\lambda a.unpack\ a\ to\ \langle v^{\bullet}, v^{\circ}\rangle\ in\ v^{\bullet})(fetch[w_1]\, box_s\, M)$$

$$\lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ}\rangle\ in\ (unpack\ b\ to\ \langle v^{\bullet}, v^{\circ}\rangle\ in\ (box_{u^{\circ}.v^{\circ}}\ u^{\bullet} v^{\bullet}))$$

Sintaxis de $\lambda^{\mathsf{Cert}}_{\Box}$

tipos	A	$::=$	$P \mid A \implies B \mid [s]A$
certificados	s, t	$::=$	$a \mid v^{\circ} \mid s \cdot t \mid \lambda a : A.s \mid !s$
		$\mid$	$letc\ s\ be\ v^{\circ} : A\ in\ t \mid fetch(s)$
valores	V	$::=$	$box_s\ M \mid \lambda a.M$
términos	M, N	$::=$	$a \mid v^{\bullet} \mid V \mid M\ N$
		$\mid$	$unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N$
		$\mid$	$fetch[w]\ M$

$$(\lambda a.unpack\ a\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ v^{\bullet})(fetch[w_1]\ box_s\ M)$$

$$\lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ} \rangle\ in\ (unpack\ b\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ (box_{u^{\circ}.v^{\circ}}\ u^{\bullet} v^{\bullet}))$$

Sintaxis de $\lambda^{\mathsf{Cert}}_{\Box}$

tipos	A	$::=$	$P \mid A \implies B \mid [s]A$
certificados	s, t	$::=$	$a \mid v^{\circ} \mid s \cdot t \mid \lambda a : A.s \mid !s$
		$\mid$	$letc\ s\ be\ v^{\circ} : A\ in\ t \mid fetch(s)$
valores	V	$::=$	$box_s\, M \mid \lambda a.M$
términos	M, N	$::=$	$a \mid v^{\bullet} \mid V \mid M\, N$
		$\mid$	$unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N$
		$\mid$	$fetch[w]\, M$

$$(\lambda a.unpack\ a\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ v^{\bullet})(fetch[w_1]\, box_s\, M)$$

$$\lambda a.\lambda b.unpack\ a\ to\ \langle u^{\bullet}, u^{\circ} \rangle\ in\ (unpack\ b\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ (box_{u^{\circ}.v^{\circ}}\ u^{\bullet} v^{\bullet}))$$

Sistema de tipos - Juicio

$$\Sigma; \Delta; \Gamma \rhd M : A@w \mid s$$

- "Bajo las asunciones de validez en Δ y las asunciones de verdad en Γ, M tiene tipo A en w con prueba s"
  - $\Sigma = \{w_1, \ldots w_n\}$
  - $\Delta = \cdot \mid \Delta, v : A@w$ (contexto móvil)
  - $\Gamma = \cdot \mid \Gamma, a : A@w$ (contexto local)

$$\Sigma \vdash w$$

- "El mundo w pertenece al Σ"

Reglas del sistema de tipos I

$$\frac{\Sigma \vdash w}{\Sigma; \Delta; \Gamma, a : A@w, \Gamma' \rhd a : A@w \,|\, a}\ \mathrm{VarT}$$

$$\frac{\Sigma; \Delta; \Gamma, a : A@w \rhd M : B@w \,|\, s}{\Sigma; \Delta; \Gamma \rhd \lambda a.M : A \implies B@w \,|\, \lambda a : A.s} \implies I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : A \implies B@w \,|\, s \quad \Sigma; \Delta; \Gamma \rhd N : A@w \,|\, t}{\Sigma; \Delta; \Gamma \rhd M\,N : B@w \,|\, s \cdot t} \implies E$$

Reglas del sistema de tipos I

$$\frac{\Sigma \vdash w}{\Sigma; \Delta; \Gamma, a : A@w, \Gamma' \rhd a : A@w \,|\, a}\ \mathrm{VarT}$$

$$\frac{\Sigma; \Delta; \Gamma, a : A@w \rhd M : B@w \,|\, s}{\Sigma; \Delta; \Gamma \rhd \lambda a.M : A \implies B@w \,|\, \lambda a : A.s} \implies I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : A \implies B@w \,|\, s \quad \Sigma; \Delta; \Gamma \rhd N : A@w \,|\, t}{\Sigma; \Delta; \Gamma \rhd M\,N : B@w \,|\, s \cdot t} \implies E$$

Reglas del sistema de tipos I

$$\frac{\Sigma \vdash w}{\Sigma; \Delta; \Gamma, a : A@w, \Gamma' \rhd a : A@w \,|\, a}\ \mathrm{VarT}$$

$$\frac{\Sigma; \Delta; \Gamma, a : A@w \rhd M : B@w \,|\, s}{\Sigma; \Delta; \Gamma \rhd \lambda a.M : A \implies B@w \,|\, \lambda a : A.s}\ \Longrightarrow I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : A \implies B@w \,|\, s \quad \Sigma; \Delta; \Gamma \rhd N : A@w \,|\, t}{\Sigma; \Delta; \Gamma \rhd M\,N : B@w \,|\, s \cdot t}\ \Longrightarrow E$$

Reglas del sistema de tipos I

$$\frac{\Sigma \vdash w}{\Sigma; \Delta; \Gamma, a : A@w, \Gamma' \rhd a : A@w \mid a}\ \mathrm{VarT}$$

$$\frac{\Sigma; \Delta; \Gamma, a : A@w \rhd M : B@w \mid s}{\Sigma; \Delta; \Gamma \rhd \lambda a.M : A \implies B@w \mid \lambda a : A.s} \implies I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : A \implies B@w \mid s \quad \Sigma; \Delta; \Gamma \rhd N : A@w \mid t}{\Sigma; \Delta; \Gamma \rhd M\,N : B@w \mid s \cdot t} \implies E$$

Reglas del sistema de tipos II

$$\frac{\Sigma \vdash w}{\Sigma; \Delta, v : A@w, \Delta'; \Gamma \rhd v^\bullet : A@w \mid v^\circ}\ \mathrm{VarV}$$

$$\frac{\Sigma; \Delta; \cdot \rhd M : A@w \mid s}{\Sigma; \Delta; \Gamma \rhd box_s\, M : [s]A@w \mid !s}\ \Box I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [r]A@w \mid s \quad \Sigma; \Delta, v : A@w; \Gamma \rhd N : C@w \mid t}{\Sigma; \Delta; \Gamma \rhd unpack\ M\ to\ \langle v^\bullet, v^\circ \rangle\ in\ N : C\{v^\circ/r\}@w \mid letc\ s\ be\ v : A\ in\ t}\ \Box E$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [s]A@w' \mid t \quad \Sigma \vdash w}{\Sigma; \Delta; \Gamma \rhd fetch[w']\, M : [s]A@w \mid fetch(t)}\ \mathrm{Fetch}$$

Reglas del sistema de tipos II

$$\frac{\Sigma \vdash w}{\Sigma; \Delta, v : A@w, \Delta'; \Gamma \rhd v^\bullet : A@w \mid v^\circ}\ \mathrm{VarV}$$

$$\frac{\Sigma; \Delta; \cdot \rhd M : A@w \mid s}{\Sigma; \Delta; \Gamma \rhd box_s\, M : [s]A@w \mid !s}\ \Box I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [r]A@w \mid s \quad \Sigma; \Delta, v : A@w; \Gamma \rhd N : C@w \mid t}{\Sigma; \Delta; \Gamma \rhd unpack\ M\ to\ \langle v^\bullet, v^\circ \rangle\ in\ N : C\{v^\circ/r\}@w \mid letc\ s\ be\ v : A\ in\ t}\ \Box E$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [s]A@w' \mid t \quad \Sigma \vdash w}{\Sigma; \Delta; \Gamma \rhd fetch[w']\, M : [s]A@w \mid fetch(t)}\ \mathrm{Fetch}$$

Reglas del sistema de tipos II

$$\frac{\Sigma \vdash w}{\Sigma; \Delta, v : A@w, \Delta'; \Gamma \rhd v^{\bullet} : A@w \mid v^{\circ}}\ \mathrm{VarV}$$

$$\frac{\Sigma; \Delta; \cdot \rhd M : A@w \mid s}{\Sigma; \Delta; \Gamma \rhd box_s\ M : [s]A@w \mid !s}\ \Box I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [r]A@w \mid s \quad \Sigma; \Delta, v : A@w; \Gamma \rhd N : C@w \mid t}{\Sigma; \Delta; \Gamma \rhd unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N : C\{v^{\circ}/r\}@w \mid letc\ s\ be\ v : A\ in\ t}\ \Box E$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [s]A@w' \mid t \quad \Sigma \vdash w}{\Sigma; \Delta; \Gamma \rhd fetch[w']\ M : [s]A@w \mid fetch(t)}\ \mathrm{Fetch}$$

Reglas del sistema de tipos II

$$\frac{\Sigma \vdash w}{\Sigma; \Delta, v : A@w, \Delta'; \Gamma \rhd v^{\bullet} : A@w \mid v^{\circ}}\ \mathrm{VarV}$$

$$\frac{\Sigma; \Delta; \cdot \rhd M : A@w \mid s}{\Sigma; \Delta; \Gamma \rhd box_s\, M : [s]A@w \mid !s}\ \Box I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [r]A@w \mid s \quad \Sigma; \Delta, v : A@w; \Gamma \rhd N : C@w \mid t}{\Sigma; \Delta; \Gamma \rhd unpack\ M\ to\ \langle v^{\bullet}, v^{\circ} \rangle\ in\ N : C\{v^{\circ}/r\}@w \mid letc\ s\ be\ v : A\ in\ t}\ \Box E$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [s]A@w' \mid t \quad \Sigma \vdash w}{\Sigma; \Delta; \Gamma \rhd fetch[w']\, M : [s]A@w \mid fetch(t)}\ \mathrm{Fetch}$$

Reglas del sistema de tipos II

$$\frac{\Sigma \vdash w}{\Sigma; \Delta, v : A@w, \Delta'; \Gamma \rhd v^{\bullet} : A@w \mid v^{\circ}}\ \mathrm{VarV}$$

$$\frac{\Sigma; \Delta; \cdot \rhd M : A@w \mid s}{\Sigma; \Delta; \Gamma \rhd box_s\, M : [s]A@w \mid !s}\ \Box I$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [r]A@w \mid s \quad \Sigma; \Delta, v : A@w; \Gamma \rhd N : C@w \mid t}{\Sigma; \Delta; \Gamma \rhd \mathit{unpack}\ M\ \mathit{to}\ \langle v^{\bullet}, v^{\circ} \rangle\ \mathit{in}\ N : C\{v^{\circ}/r\}@w \mid \mathit{letc}\ s\ \mathit{be}\ v : A\ \mathit{in}\ t}\ \Box E$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : [s]A@w' \mid t \quad \Sigma \vdash w}{\Sigma; \Delta; \Gamma \rhd \mathit{fetch}[w']\, M : [s]A@w \mid \mathit{fetch}(t)}\ \mathrm{Fetch}$$

Semántica operacional. Definición I

- Basada en la definición de una máquina abstracta, con estados y transiciones
- Ejecución secuencial, donde cada paso sucede en un mundo determinado
  - No se modelan ejecuciones concurrentes

Semántica operacional. Definición I

- Basada en la definición de una máquina abstracta, con estados y transiciones
- Ejecución secuencial, donde cada paso sucede en un mundo determinado
  - No se modelan ejecuciones concurrentes

estado	$\mathbb{N}$	$::=$	$\mathbb{W}\,; w : [k, M]$
ambiente	$\mathbb{W}$	$::=$	$\{w_1 : C_1, \ldots w_n : C_n\}$
contexto	k	$::=$	$\mathrm{return}\ w \mid \mathrm{finish} \mid k \triangleleft l$
capa	l	$::=$	$\circ\ N \mid V\ \circ \mid \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N$
pila de contextos	C	$::=$	$\epsilon \mid C : : k$

Semántica operacional. Definición I

- Basada en la definición de una máquina abstracta, con estados y transiciones
- Ejecución secuencial, donde cada paso sucede en un mundo determinado
  - No se modelan ejecuciones concurrentes

estado	$\mathbb{N}$	$::=$	$\mathbb{W}\,; w : [k, M]$
ambiente	$\mathbb{W}$	$::=$	$\{w_1 : C_1, \ldots w_n : C_n\}$
contexto	k	$::=$	$\mathrm{return}\ w \mid \mathrm{finish} \mid k \triangleleft l$
capa	l	$::=$	$\circ\ N \mid V\ \circ \mid \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N$
pila de contextos	C	$::=$	$\epsilon \mid C : : k$

Semántica operacional. Definición I

- Basada en la definición de una máquina abstracta, con estados y transiciones
- Ejecución secuencial, donde cada paso sucede en un mundo determinado
  - No se modelan ejecuciones concurrentes

estado	$\mathbb{N}$	$::=$	$\mathbb{W}\,; w : [k, M]$
ambiente	$\mathbb{W}$	$::=$	$\{w_1 : C_1, \ldots w_n : C_n\}$
contexto	k	$::=$	$\mathrm{return}\ w \mid \mathrm{finish} \mid k \triangleleft l$
capa	l	$::=$	$\circ\ N \mid V\ \circ \mid \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^{\bullet}, v^{\circ} \rangle\ \mathit{in}\ N$
pila de contextos	C	$::=$	$\epsilon \mid C : : k$

Semántica operacional. Definición I

- Basada en la definición de una máquina abstracta, con estados y transiciones
- Ejecución secuencial, donde cada paso sucede en un mundo determinado
  - No se modelan ejecuciones concurrentes

estado	$\mathbb{N}$	$::=$	$\mathbb{W}\,; w : [k, M]$
ambiente	$\mathbb{W}$	$::=$	$\{w_1 : C_1, \ldots w_n : C_n\}$
contexto	k	$::=$	$\mathrm{return}\ w \mid \mathrm{finish} \mid k \triangleleft l$
capa	l	$::=$	$\circ\ N \mid V\ \circ \mid \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^{\bullet}, v^{\circ} \rangle\ \mathit{in}\ N$
pila de contextos	C	$::=$	$\epsilon \mid C : : k$

Semántica operacional. Definición I

- Basada en la definición de una máquina abstracta, con estados y transiciones
- Ejecución secuencial, donde cada paso sucede en un mundo determinado
  - No se modelan ejecuciones concurrentes

estado	$\mathbb{N}$	$::=$	$\mathbb{W}\,; w : [k, M]$
ambiente	$\mathbb{W}$	$::=$	$\{w_1 : C_1, \ldots w_n : C_n\}$
contexto	k	$::=$	$\mathrm{return}\ w \mid \mathrm{finish} \mid k \triangleleft l$
capa	l	$::=$	$\circ\ N \mid V\ \circ \mid \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N$
pila de contextos	C	$::=$	$\epsilon \mid C :: k$

Semántica operacional. Sintaxis

Para un conjunto de mundos $\Sigma = \{w_1, \ldots, w_n\}$

- Si $\mathbb{W} = \{w_1 : \epsilon, \cdots, w_n : \epsilon\}$ entonces
- $\mathbb{W}\,; w : [\mathit{finish}, M]$ es el estado inicial
- $\mathbb{W}\,; w : [\mathit{finish}, V]$ es un estado final
  - En un estado final el foco de la computación es un término evaluado en su totalidad, es decir, un valor

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{ll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] \longrightarrow \\
 & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] \longrightarrow \\
 & \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}]
\end{array}
$$

$$
\begin{array}{ll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] \longrightarrow \\
 & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] \longrightarrow \\
 & \{w : C; w_s\}; w : [k, V\{w'/w\}]
\end{array}
$$

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{ll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] \longrightarrow \\
 & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] \longrightarrow \\
 & \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}]
\end{array}
$$

$$
\begin{array}{ll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] \longrightarrow \\
 & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] \longrightarrow \\
 & \{w : C; w_s\}; w : [k, V\{w'/w\}]
\end{array}
$$

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{ll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] \longrightarrow \\
 & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] \longrightarrow \\
 & \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}]
\end{array}
$$

$$
\begin{array}{ll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] \longrightarrow \\
 & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] \longrightarrow \\
 & \{w : C; w_s\}; w : [k, V\{w'/w\}]
\end{array}
$$

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{lll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] & \longrightarrow \\
 & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] & \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] & \longrightarrow \\
 & \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}] &
\end{array}
$$

$$
\begin{array}{lll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] & \longrightarrow \\
 & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] & \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] & \longrightarrow \\
 & \{w : C; w_s\}; w : [k, V\{w'/w\}] &
\end{array}
$$

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{lll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] & \longrightarrow \\
 & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] & \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] & \longrightarrow \\
 & \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}] &
\end{array}
$$

$$
\begin{array}{lll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] & \longrightarrow \\
 & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] & \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] & \longrightarrow \\
 & \{w : C; w_s\}; w : [k, V\{w'/w\}] &
\end{array}
$$

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{ll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] \longrightarrow \\
 & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] \longrightarrow \\
 & \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}]
\end{array}
$$

$$
\begin{array}{ll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] \longrightarrow \\
 & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] \longrightarrow \\
 & \{w : C; w_s\}; w : [k, V\{w'/w\}]
\end{array}
$$

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{ll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] \longrightarrow \\
& \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] \longrightarrow \\
& \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}]
\end{array}
$$

$$
\begin{array}{ll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] \longrightarrow \\
& \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] \longrightarrow \\
& \{w : C; w_s\}; w : [k, V\{w'/w\}]
\end{array}
$$

Semántica operacional. Reglas de reducción

$$
\begin{array}{lll}
(1) & \mathbb{W}; w : [k, MN] & \longrightarrow \quad \mathbb{W}; w : [k \lhd \circ\ N, M] \\
(2) & \mathbb{W}; w : [k \lhd \circ\ N, V] & \longrightarrow \quad \mathbb{W}; w : [k \lhd V\ \circ, N] \\
(3) & \mathbb{W}; w : [k \lhd (\lambda a.M)\ \circ, V] & \longrightarrow \quad \mathbb{W}; w : [k, M\{a/V\}]
\end{array}
$$

$$
\begin{array}{ll}
(4) & \mathbb{W}; w : [k, \mathit{unpack}\ M\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N] \longrightarrow \\
 & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, M] \\
(5) & \mathbb{W}; w : [k \lhd \mathit{unpack}\ \circ\ \mathit{to}\ \langle v^\bullet, v^\circ \rangle\ \mathit{in}\ N, \mathit{box}_s\ M] \longrightarrow \\
 & \mathbb{W}; w : [k, N\{v^\circ/s\}\{v^\bullet/M\}]
\end{array}
$$

$$
\begin{array}{ll}
(6) & \{w : C; w_s\}; w : [k, \mathit{fetch}[w']\ M] \longrightarrow \\
 & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, M] \\
(7) & \{w : C : : k; w_s\}; w' : [\mathrm{return}\ w, V] \longrightarrow \\
 & \{w : C; w_s\}; w : [k, V\{w'/w\}]
\end{array}
$$

Semántica operacional. Sistema de tipos I

- Se define un sistema de tipos que establece las reglas para la construcción correcta de estados de la máquina
- Dos nuevos juicios de valor:
  - Juicio para estados: $\Sigma \vdash \mathbb{W} ; w_j : [k, M]$
  - Juicio para ambientes: $\Sigma \vdash \mathbb{W} ; k : A@w_j$
- Intuición para juicio de ambientes: El contexto k (junto al ambiente $\mathbb{W}$) espera un valor de tipo A en el mundo w_j

Semántica operacional. Sistema de tipos I

- Se define un sistema de tipos que establece las reglas para la construcción correcta de estados de la máquina
- Dos nuevos juicios de valor:
  - Juicio para estados: $\Sigma \vdash \mathbb{W} ; w_j : [k, M]$
  - Juicio para ambientes: $\Sigma \vdash \mathbb{W} ; k : A@w_j$
- Intuición para juicio de ambientes: El contexto k (junto al ambiente $\mathbb{W}$) espera un valor de tipo A en el mundo w_j

$$\frac{\Sigma = \{w_1, \ldots, w_n\} \quad \mathbb{W} = \{w_1 : C_1, \ldots w_n : C_n\} \quad \Sigma; \cdot ; \cdot \rhd M : A@w_j \mid s \quad \Sigma \vdash \mathbb{W} ; k : A@w_j}{\Sigma \vdash \mathbb{W} ; w_j : [k, M]}\ \mathit{MState}$$

Contenido

1. Introducción
   - Motivación
   - PCC
   - Modelo
2. Lógica de pruebas
   - Lógicas modales y LP
   - ILPnd
3. Cálculo para certificados de código móvil
   - Términos y certificados
   - Sistema de tipos
   - Semántica operacional
4. Propiedades
   - Seguridad de tipos
   - Normalización fuerte
5. Extensiones
   - Booleanos
   - Números naturales
   - Concatenación de certificados
6. Conclusiones
   - Trabajo a Futuro
   - Conclusiones

Seguridad de tipos

- Consiste en garantizar que un programa que ha sido bien tipado no puede fallar al momento de ser ejecutado

Seguridad de tipos

- Consiste en garantizar que un programa que ha sido bien tipado no puede fallar al momento de ser ejecutado
- El caso estándar es que una función reciba un parámetro de un tipo que no esperaba

Seguridad de tipos

- Consiste en garantizar que un programa que ha sido bien tipado no puede fallar al momento de ser ejecutado
- El caso estándar es que una función reciba un parámetro de un tipo que no esperaba
- Para $\lambda_{\square}^{\mathsf{Cert}}$, interesa el caso en donde código móvil se genera con un certificado que no le corresponde

Seguridad de tipos

- Consiste en garantizar que un programa que ha sido bien tipado no puede fallar al momento de ser ejecutado
- El caso estándar es que una función reciba un parámetro de un tipo que no esperaba
- Para $\lambda_{\square}^{\mathsf{Cert}}$, interesa el caso en donde código móvil se genera con un certificado que no le corresponde
- Se demuestran dos propiedades:

Progreso: Si $\Sigma \vdash \mathbb{N}$ es derivable y $\mathbb{N}$ no es terminal, entonces existe $\mathbb{N}'$ tal que $\mathbb{N} \longrightarrow \mathbb{N}'$

Preservación de tipos: Si $\Sigma \vdash \mathbb{N}$ es derivable y $\mathbb{N} \longrightarrow \mathbb{N}'$, entonces $\Sigma \vdash \mathbb{N}'$ es derivable

Seguridad de tipos II

- La demostración de progreso se realiza por análisis de casos sobre el estado $\mathbb{N}$

Seguridad de tipos II

- La demostración de progreso se realiza por análisis de casos sobre el estado $\mathbb{N}$
- La demostración de preservación de tipos se realiza por análisis de casos sobre el paso de reducción aplicado

Seguridad de tipos II

- La demostración de progreso se realiza por análisis de casos sobre el estado $\mathbb{N}$
- La demostración de preservación de tipos se realiza por análisis de casos sobre el paso de reducción aplicado

Corolario

Si $\Sigma \vdash \mathbb{N}$ es derivable y $\mathbb{N} \longrightarrow^* \mathrm{W}\,; w : [k, box_s\, M]$, entonces $\Sigma; \cdot; \cdot \rhd M : A@w \,|\, s$

Normalización fuerte

- Cualquier secuencia de pasos de reducción de la máquina abstracta siempre finaliza en un estado terminal

Normalización fuerte

- Cualquier secuencia de pasos de reducción de la máquina abstracta siempre finaliza en un estado terminal
- Para demostrar esta propiedad, se toma como base otro sistema que cumple con la propiedad de normalización fuerte: $\lambda^{\mathbf{1},\rightarrow}$

Normalización fuerte

- Cualquier secuencia de pasos de reducción de la máquina abstracta siempre finaliza en un estado terminal
- Para demostrar esta propiedad, se toma como base otro sistema que cumple con la propiedad de normalización fuerte: $\lambda^{\mathbf{1},\rightarrow}$
- Los estados de la máquina y pasos de reducción de $\lambda_{\square}^{\mathsf{Cert}}$ se traducen en términos y reducciones en $\lambda^{\mathbf{1},\rightarrow}$

Normalización fuerte

- Cualquier secuencia de pasos de reducción de la máquina abstracta siempre finaliza en un estado terminal
- Para demostrar esta propiedad, se toma como base otro sistema que cumple con la propiedad de normalización fuerte: $\lambda^{\mathbf{1},\rightarrow}$
- Los estados de la máquina y pasos de reducción de $\lambda_{\square}^{\mathsf{Cert}}$ se traducen en términos y reducciones en $\lambda^{\mathbf{1},\rightarrow}$
- Una traducción precisa alcanza, ya que si existiese una reducción infinita en $\lambda_{\square}^{\mathsf{Cert}}$, la traducción daría como resultado una reducción infinita en $\lambda^{\mathbf{1},\rightarrow}$ lo cual se sabe que no ocurre.

Contenido

1 Introducción
- Motivación
- PCC
- Modelo

2 Lógica de pruebas
- Lógicas modales y LP
- ILPnd

3 Cálculo para certificados de código móvil
- Términos y certificados
- Sistema de tipos
- Semántica operacional

4 Propiedades
- Seguridad de tipos
- Normalización fuerte

5 Extensiones
- Booleanos
- Números naturales
- Concatenación de certificados

6 Conclusiones
- Trabajo a Futuro
- Conclusiones

Extensiones

- Se agregan una serie de extensiones a $\lambda_{\square}^{\mathsf{Cert}}$ para darle mayor expresividad al lenguaje

Extensiones

- Se agregan una serie de extensiones a $\lambda_{\square}^{\mathsf{Cert}}$ para darle mayor expresividad al lenguaje
- Las extensiones incluyen: Números naturales, Booleanos y concatenación de certificados

Extensiones

- Se agregan una serie de extensiones a $\lambda_{\square}^{\mathsf{Cert}}$ para darle mayor expresividad al lenguaje
- Las extensiones incluyen: Números naturales, Booleanos y concatenación de certificados
- Es necesario realizar modificaciones en:
  - Sintaxis del lenguaje
  - Sistema de tipos
  - Semántica de la máquina abstracta

Extensiones

- Se agregan una serie de extensiones a $\lambda_{\square}^{\mathsf{Cert}}$ para darle mayor expresividad al lenguaje
- Las extensiones incluyen: Números naturales, Booleanos y concatenación de certificados
- Es necesario realizar modificaciones en:
  - Sintaxis del lenguaje
  - Sistema de tipos
  - Semántica de la máquina abstracta
- Requieren también modificar las demostraciones de las propiedades (aunque no aportan casos interesantes)

Booleanos I

Modificaciones en la sintaxis

$$
\begin{array}{rcl}
A & ::= & \ldots \mid \mathit{Bool} \\
s, t & ::= & \ldots \mid \mathit{true} \mid \mathit{false} \mid \mathit{and}(s,t) \mid \mathit{or}(s,t) \mid \mathit{not}(s) \mid \mathit{if}(s,t,r) \\
V & ::= & \ldots \mid \mathbf{true} \mid \mathbf{false} \\
M, N, P & ::= & \ldots \mid M \&\& N \mid M \| N \mid \sim M \mid \text{if } P \text{ then } M \text{ else } N
\end{array}
$$

Booleanos I

Modificaciones en la sintaxis

$$\begin{array}{rcl} A & ::= & \ldots \mid \mathit{Bool} \\ s, t & ::= & \ldots \mid \mathit{true} \mid \mathit{false} \mid \mathit{and}(s,t) \mid \mathit{or}(s,t) \mid \mathit{not}(s) \mid \mathit{if}(s,t,r) \\ V & ::= & \ldots \mid \mathbf{true} \mid \mathbf{false} \\ M, N, P & ::= & \ldots \mid M \&\& N \mid M \| N \mid \sim M \mid \text{if } P \text{ then } M \text{ else } N \end{array}$$

Nuevas reglas del sistema de tipos (extracto)

$$\frac{}{\Sigma; \Delta; \Gamma \rhd \mathbf{true} : \mathit{Bool}@w \mid \mathit{true}}\ \mathit{True} \qquad \frac{}{\Sigma; \Delta; \Gamma \rhd \mathbf{false} : \mathit{Bool}@w \mid \mathit{false}}\ \mathit{False}$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : \mathit{Bool}@w \mid s \quad \Sigma; \Delta; \Gamma \rhd N : \mathit{Bool}@w \mid t}{\Sigma; \Delta; \Gamma \rhd M \&\& N : \mathit{Bool}@w \mid \mathit{and}(s,t)}\ \mathit{And}$$

$$\frac{\Sigma; \Delta; \Gamma \rhd P : \mathit{Bool}@w \mid r \quad \Sigma; \Delta; \Gamma \rhd M : A@w \mid s \quad \Sigma; \Delta; \Gamma \rhd N : A@w \mid t}{\Sigma; \Delta; \Gamma \rhd \text{if } P \text{ then } M \text{ else } N : A@w \mid \mathit{if}(r,s,t)}\ \mathit{If}$$

Booleanos II

Cambios en la sintaxis de la máquina abstracta

$$I \quad ::= \quad \ldots \mid \text{if } \circ \text{ then } M \text{ else } N \mid \circ \&\& N \mid \circ \| N \mid \sim \circ$$

Booleanos II

Cambios en la sintaxis de la máquina abstracta

$$I \quad ::= \quad \ldots \mid \text{if } \circ \text{ then } M \text{ else } N \mid \circ \,\&\&\, N \mid \circ \,\|\, N \mid \sim \circ$$

Nuevas reglas de reducción (extracto)

$$\begin{array}{lrcl}
(8) & \mathbb{W}; w : [k, \text{if } P \text{ then } M \text{ else } N] & \longrightarrow & \\
 & \mathbb{W}; w : [k \triangleleft \text{if } \circ \text{ then } M \text{ else } N, P] & & \\
(9) & \mathbb{W}; w : [k \triangleleft \text{if } \circ \text{ then } M \text{ else } N, \textbf{true}] & \longrightarrow & \mathbb{W}; w : [k, M] \\
(10) & \mathbb{W}; w : [k \triangleleft \text{if } \circ \text{ then } M \text{ else } N, \textbf{false}] & \longrightarrow & \mathbb{W}; w : [k, N] \\
(11) & \mathbb{W}; w : [k, M \,\&\&\, N] & \longrightarrow & \mathbb{W}; w : [k \triangleleft \circ \,\&\&\, N, M] \\
(12) & \mathbb{W}; w : [k \triangleleft \circ \,\&\&\, N, \textbf{false}] & \longrightarrow & \mathbb{W}; w : [k, \textbf{false}] \\
(13) & \mathbb{W}; w : [k \triangleleft \circ \,\&\&\, N, \textbf{true}] & \longrightarrow & \mathbb{W}; w : [k, N]
\end{array}$$

Números naturales I

Modificaciones en la sintaxis

$$
\begin{array}{rcl}
A & ::= & \ldots \mid \mathit{Nat} \\
s, t & ::= & \ldots \mid \mathit{zero} \mid \mathit{succ}(s) \mid \mathit{add}(s, t) \mid \mathit{mul}(s, t) \\
V & ::= & \ldots \mid \mathbf{s}^{(n)}\mathbf{z}\ (n \geq 0) \\
M, N, P, Q & ::= & \ldots \mid \mathrm{case}\ M\ \mathrm{of}\ \mathbf{z} \rightarrow P \boxplus \mathbf{s}a \rightarrow Q \mid \mathbf{s}M \\
 & \mid & M + N \mid M * N
\end{array}
$$

Números naturales I

Modificaciones en la sintaxis

$$\begin{array}{rcl} A & ::= & \ldots \mid Nat \\ s, t & ::= & \ldots \mid zero \mid succ(s) \mid add(s,t) \mid mul(s,t) \\ V & ::= & \ldots \mid \mathbf{s}^{(n)}\mathbf{z}\ (n \geq 0) \\ M, N, P, Q & ::= & \ldots \mid \mathrm{case}\ M\ \mathrm{of}\ \mathbf{z} \rightarrow P \boxplus \mathbf{s}a \rightarrow Q \mid \mathbf{s}M \\ & \mid & M + N \mid M * N \end{array}$$

Nuevas reglas del sistema de tipos (extracto)

$$\frac{}{\Sigma; \Delta; \Gamma \rhd \mathbf{z} : Nat@w \mid zero}\ Zero \qquad \frac{\Sigma; \Delta; \Gamma \rhd M : Nat@w \mid s}{\Sigma; \Delta; \Gamma \rhd \mathbf{s}M : Nat@w \mid succ(s)}\ Succ$$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : Nat@w \mid s \quad \Sigma; \Delta; \Gamma \rhd N : Nat@w \mid t}{\Sigma; \Delta; \Gamma \rhd M + N : Nat@w \mid add(s,t)}\ Add$$

Números naturales II

Cambios en la sintaxis de la máquina abstracta

$$I \quad ::= \quad \ldots \mid \mathbf{s}\circ \mid M \ + \ \circ \mid M \ * \ \circ \mid \mathrm{case} \ \circ \ \mathrm{of} \ \mathbf{z} \rightarrow P \boxplus \mathbf{s}a \rightarrow Q$$

Números naturales II

Cambios en la sintaxis de la máquina abstracta

$$I \quad ::= \quad \ldots \mid \mathbf{s}\circ \mid M \ + \ \circ \mid M \ * \ \circ \mid \mathrm{case} \ \circ \ \mathrm{of} \ \mathbf{z}{\rightarrow}P \boxplus \mathbf{s}a \rightarrow Q$$

Nuevas reglas de reducción (extracto)

$$\begin{array}{lrcl}
(28) & \mathbb{W}; w : [k, \mathrm{case}\ M\ \mathrm{of}\ \mathbf{z}{\rightarrow}P \boxplus \mathbf{s}a \rightarrow Q] & \longrightarrow & \\
& \mathbb{W}; w : [k \triangleleft \mathrm{case}\ \circ\ \mathrm{of}\ \mathbf{z}{\rightarrow}P \boxplus \mathbf{s}a \rightarrow Q, M] & & \\
(29) & \mathbb{W}; w : [k \triangleleft \mathrm{case}\ \circ\ \mathrm{of}\ \mathbf{z}{\rightarrow}P \boxplus \mathbf{s}a \rightarrow Q, \mathbf{z}] & \longrightarrow & \mathbb{W}; w : [k, P] \\
(30) & \mathbb{W}; w : [k \triangleleft \mathrm{case}\ \circ\ \mathrm{of}\ \mathbf{z}{\rightarrow}P \boxplus \mathbf{s}a \rightarrow Q, \mathbf{s}N] & \longrightarrow & \\
& \mathbb{W}; w : [k\triangleleft, Q\{a/N\}] & &
\end{array}$$

Concatenación de certificados

- Este operador existe en la definición original de LP

Concatenación de certificados

- Este operador existe en la definición original de LP
- Su inclusión está motivada en los nuevos operadores *if* y *case*. Por ejemplo:
  - $\Sigma; \cdot; \cdot \rhd P : Bool@w \mid r$
  - $\Sigma; \cdot; \cdot \rhd M : A@w \mid s$
  - $\Sigma; \cdot; \cdot \rhd N : A@w \mid t$

Concatenación de certificados

- Este operador existe en la definición original de LP
- Su inclusión está motivada en los nuevos operadores *if* y *case*. Por ejemplo:
  - $\Sigma; \cdot; \cdot \rhd P : Bool@w \mid r$
  - $\Sigma; \cdot; \cdot \rhd M : A@w \mid s$
  - $\Sigma; \cdot; \cdot \rhd N : A@w \mid t$
- Entonces es posible tipar $(s \neq t)$
  - $\Sigma; \cdot; \cdot \rhd box_s\, M : [s]A@w \mid !s$
  - $\Sigma; \cdot; \cdot \rhd box_t\, N : [t]A@w \mid !t$

Concatenación de certificados

- Este operador existe en la definición original de LP
- Su inclusión está motivada en los nuevos operadores *if* y *case*. Por ejemplo:
  - $\Sigma; \cdot; \cdot \rhd P : Bool@w \mid r$
  - $\Sigma; \cdot; \cdot \rhd M : A@w \mid s$
  - $\Sigma; \cdot; \cdot \rhd N : A@w \mid t$
- Entonces es posible tipar $(s \neq t)$
  - $\Sigma; \cdot; \cdot \rhd box_s\, M : [s]A@w \mid !s$
  - $\Sigma; \cdot; \cdot \rhd box_t\, N : [t]A@w \mid !t$
- No es posible tipar $\text{if } P \text{ then } box_t\, M \text{ else } box_t\, N$

Concatenación de certificados

- Este operador existe en la definición original de LP
- Su inclusión está motivada en los nuevos operadores *if* y *case*. Por ejemplo:
  - $\Sigma; \cdot; \cdot \rhd P : Bool@w \mid r$
  - $\Sigma; \cdot; \cdot \rhd M : A@w \mid s$
  - $\Sigma; \cdot; \cdot \rhd N : A@w \mid t$
- Entonces es posible tipar $(s \neq t)$
  - $\Sigma; \cdot; \cdot \rhd box_s\, M : [s]A@w \mid !s$
  - $\Sigma; \cdot; \cdot \rhd box_t\, N : [t]A@w \mid !t$
- No es posible tipar $\text{if } P \text{ then } box_t\, M \text{ else } box_t\, N$

$$\frac{\Sigma; \Delta; \Gamma \rhd M : A@w \mid s}{\Sigma; \Delta; \Gamma \rhd M : A@w \mid s + t}\, CertSum1 \qquad \frac{\Sigma; \Delta; \Gamma \rhd M : A@w \mid t}{\Sigma; \Delta; \Gamma \rhd M : A@w \mid s + t}\, CertSum2$$

Contenido

1. Introducción
   - Motivación
   - PCC
   - Modelo
2. Lógica de pruebas
   - Lógicas modales y LP
   - ILPnd
3. Cálculo para certificados de código móvil
   - Términos y certificados
   - Sistema de tipos
   - Semántica operacional
4. Propiedades
   - Seguridad de tipos
   - Normalización fuerte
5. Extensiones
   - Booleanos
   - Números naturales
   - Concatenación de certificados
6. Conclusiones
   - Trabajo a Futuro
   - Conclusiones

Trabajo a futuro

- Inclusión del operador de posibilidad $\diamond$.
  - No se incluye en la lógica de pruebas. Se puede investigar interpretación en LP o directamente agregar esquemas al cálculo:
    - $\diamond A$ denota valor de nodo remoto
    - Rompe conexión con la lógica
    - Puede tener sentido desde el punto de vista de la programación

Trabajo a futuro

- Inclusión del operador de posibilidad $\diamond$.
  - No se incluye en la lógica de pruebas. Se puede investigar interpretación en LP o directamente agregar esquemas al cálculo:
    - $\diamond A$ denota valor de nodo remoto
    - Rompe conexión con la lógica
    - Puede tener sentido desde el punto de vista de la programación
- Extensiones a la definición de $\lambda_{\square}^{\text{Cert}}$
  - Referencias y recursión (operador *fix*)
  - Polimorfismo sobre variables certificado
    - $\forall\alpha\forall\beta.[\alpha](A \implies B) \implies [\beta]A \implies [\alpha \cdot \beta]B$
  - Polimorfismo sobre variables de función

Conclusiones

- Se presentó un cálculo para modelar computaciones móviles que incluye la generación de certificados

Conclusiones

- Se presentó un cálculo para modelar computaciones móviles que incluye la generación de certificados
- Se realizó una definición precisa, partiendo del estudio de la lógica de pruebas.

Conclusiones

- Se presentó un cálculo para modelar computaciones móviles que incluye la generación de certificados
- Se realizó una definición precisa, partiendo del estudio de la lógica de pruebas.
- El sistema de tipos obtenido constituye una teoría unificada para la construcción correcta de código y certificados

Conclusiones

- Se presentó un cálculo para modelar computaciones móviles que incluye la generación de certificados
- Se realizó una definición precisa, partiendo del estudio de la lógica de pruebas.
- El sistema de tipos obtenido constituye una teoría unificada para la construcción correcta de código y certificados
- Se definió formalmente la semántica del lenguaje, basada en una máquina abstracta

Conclusiones

- Se presentó un cálculo para modelar computaciones móviles que incluye la generación de certificados
- Se realizó una definición precisa, partiendo del estudio de la lógica de pruebas.
- El sistema de tipos obtenido constituye una teoría unificada para la construcción correcta de código y certificados
- Se definió formalmente la semántica del lenguaje, basada en una máquina abstracta
- Se demostraron formalmente propiedades, incluyendo la seguridad en tipos

Conclusiones

- Se presentó un cálculo para modelar computaciones móviles que incluye la generación de certificados
- Se realizó una definición precisa, partiendo del estudio de la lógica de pruebas.
- El sistema de tipos obtenido constituye una teoría unificada para la construcción correcta de código y certificados
- Se definió formalmente la semántica del lenguaje, basada en una máquina abstracta
- Se demostraron formalmente propiedades, incluyendo la seguridad en tipos
- Se realizaron extensiones que aportaron mayor riqueza a la posterior implementación del prototipo

¿Preguntas?