Admissibility of Substitution for Multimode Type Theory

Joris Ceulemans Andreas Nuyts Dominique Devriese

KU Leuven, Belgium

EuroProofNet WG6 Meeting
Leuven, Belgium
4 April 2024

Motivating Example: Guarded Recursion

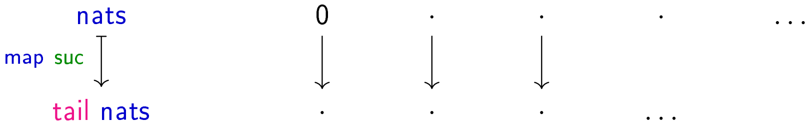
Productivity check in Agda is sometimes too restrictive:

```
nats : Stream ℕ
head nats = 0
tail nats = map suc nats
```

Motivating Example: Guarded Recursion

Productivity check in Agda is sometimes too restrictive:

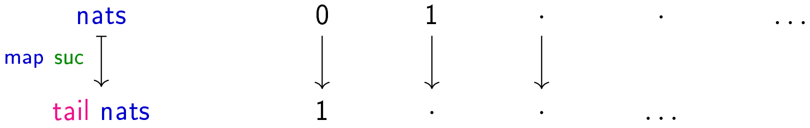
```
nats : Stream ℕ
head nats = 0
tail nats = map suc nats
```



Motivating Example: Guarded Recursion

Productivity check in Agda is sometimes too restrictive:

$$\mathsf{nats} : \mathsf{Stream}\ \mathbb{N}$$
$$\mathsf{head}\ \mathsf{nats} = 0$$
$$\mathsf{tail}\ \mathsf{nats} = \mathsf{map}\ \mathsf{suc}\ \mathsf{nats}$$



Motivating Example: Guarded Recursion

Productivity check in Agda is sometimes too restrictive:

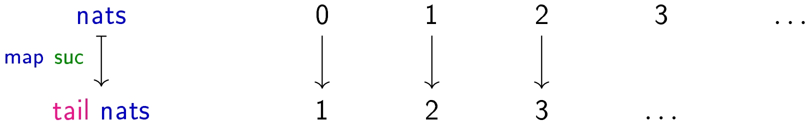
```
nats : Stream ℕ
head nats = 0
tail nats = map suc nats
```

nats	0	1	2	·	...
map suc ↓	↓	↓	↓		
tail nats	1	2	·	...	

Motivating Example: Guarded Recursion

Productivity check in Agda is sometimes too restrictive:

```
nats : Stream ℕ
head nats = 0
tail nats = map suc nats
```



Motivating Example: Guarded Recursion

Productivity check in Agda is sometimes too restrictive:

$$\mathsf{nats} : \mathsf{Stream}\ \mathbb{N}$$
$$\mathsf{head}\ \mathsf{nats} = 0$$
$$\mathsf{tail}\ \mathsf{nats} = \mathsf{map}\ \mathsf{suc}\ \mathsf{nats}$$

$$\mathsf{map}\ \mathsf{suc} : \mathsf{Stream}\ \mathbb{N} \to \mathsf{Stream}\ \mathbb{N}$$

nats	0	1	2	3	...
map suc ↓	↓	↓	↓		
tail nats	1	2	3	...	

Idea: express recursion behaviour in type via modalities.

A Brief Introduction to Multimode Type Theory (MTT)
(Gratzer et al., 2020)

Parametrised by mode theory (≈ small 2-category):

m n

A Brief Introduction to Multimode Type Theory (MTT)

(Gratzer et al., 2020)

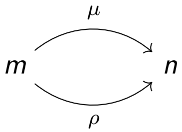
Parametrised by mode system (≈ small 2-category):

m n

A Brief Introduction to Multimode Type Theory (MTT)

(Gratzer et al., 2020)

Parametrised by mode system (≈ small 2-category):



New primitive modal operations:

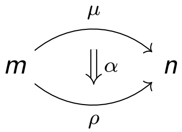
$$\vdash \Gamma \,.\, 🔒_\mu \ \mathsf{Ctx} \ @ \ m \quad \leftarrow \quad \vdash \Gamma \ \mathsf{Ctx} \ @ \ n$$

$$\Gamma \,.\, 🔒_\mu \vdash T \ \mathsf{Ty} \ @ \ m \quad \rightarrow \quad \Gamma \vdash \langle \mu \mid T \rangle \ \mathsf{Ty} \ @ \ n$$

A Brief Introduction to Multimode Type Theory (MTT)

(Gratzer et al., 2020)

Parametrised by mode system (≈ small 2-category):



New primitive modal operations:

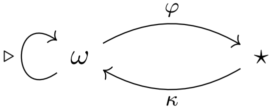
$$\vdash \Gamma\,.\,\text{🔒}_\mu \ \mathsf{Ctx} \ @ \ m \qquad \leftarrow \qquad \vdash \Gamma \ \mathsf{Ctx} \ @ \ n$$

$$\Gamma\,.\,\text{🔒}_\mu \vdash T \ \mathsf{Ty} \ @ \ m \qquad \rightarrow \qquad \Gamma \vdash \langle \mu \mid T \rangle \ \mathsf{Ty} \ @ \ n$$

Intuitively, $\mathsf{coe}^{\alpha}_{T} : \langle \mu \mid T \rangle \to \langle \rho \mid T \rangle \ @ \ n$

Example: Guarded Recursion

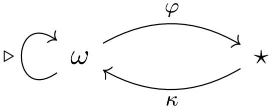
Mode system:



$\varphi \circ \triangleright = \varphi$
$\varphi \circ \kappa = \mathbb{1}$
adv $\in \mathbb{1}_\omega \Rightarrow \triangleright$ (so, next : $A \to \langle \triangleright \mid A \rangle$)

Example: Guarded Recursion

Mode system:

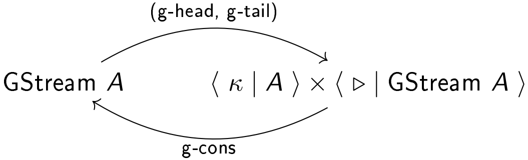


$\varphi \circ \triangleright = \varphi$

$\varphi \circ \kappa = \mathbb{1}$

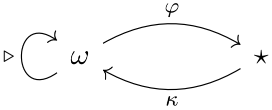
$\mathsf{adv} \in \mathbb{1}_\omega \Rightarrow \triangleright$ (so, next : $A \to \langle \triangleright \mid A \rangle$)

New non-modal type/term constructors:



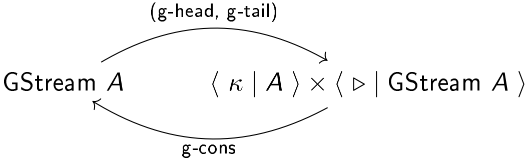
Example: Guarded Recursion

Mode system:



$$\varphi \circ \triangleright = \varphi$$
$$\varphi \circ \kappa = \mathbb{1}$$
$$\mathsf{adv} \in \mathbb{1}_\omega \Rightarrow \triangleright \qquad (\text{so, next} : A \to \langle \triangleright \mid A \rangle)$$

New non-modal type/term constructors:



$$\frac{\Gamma \,.\, (\triangleright \mid x : T) \vdash t : T \ @\ \omega}{\Gamma \vdash \text{löb}[\triangleright \mid x : T]\ t : T \ @\ \omega}$$

Implementing nats in MTT

g-nats : GStream $\mathbb{N}$
g-nats = $\{\ \}_0$

Hole	Mode	Context	Expected type
0	ω	$\cdot$	GStream $\mathbb{N}$

Implementing nats in MTT

$\text{g-nats} : \text{GStream}\ \mathbb{N}$
$\text{g-nats} = \{\text{löb}[\triangleright \mid s : \text{GStream}\ \mathbb{N}]\ ?\}0$

Hole	Mode	Context	Expected type
0	ω	$\cdot$	$\text{GStream}\ \mathbb{N}$

Implementing nats in MTT

$$\text{g-nats} : \mathsf{GStream}\ \mathbb{N}$$
$$\text{g-nats} = \text{löb}[\triangleright \mid s : \mathsf{GStream}\ \mathbb{N}]\ \{\ \}_0$$

Hole	Mode	Context	Expected type
0	ω	$(\triangleright \mid s : \mathsf{GStream}\ \mathbb{N})$	$\mathsf{GStream}\ \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\text{g-nats} : \text{GStream } \mathbb{N}$$
$$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \{\text{g-cons}$$
$$?$$
$$?\}0$$

Hole	Mode	Context	Expected type
0	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N})$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$\text{g-nats} : \text{GStream } \mathbb{N}$
$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

{ }0

{ }1

Hole	Mode	Context	Expected type
0	$\star$	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\kappa}$	$\mathbb{N}$
1	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$\text{g-nats} : \text{GStream } \mathbb{N}$

$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

{0}0

{ }1

Hole	Mode	Context	Expected type
0	$\star$	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\kappa}$	$\mathbb{N}$
1	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$\text{g-nats} : \text{GStream } \mathbb{N}$
$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

0

$\{\ \}_1$

Hole	Mode	Context	Expected type
1	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\Gamma \vdash \text{g-map} : (\kappa \mid A \to B) \to \text{GStream } A \to \text{GStream } B$$

$\text{g-nats} : \text{GStream } \mathbb{N}$
$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

0

$\{\text{g-map } ?\ ?\}1$

Hole	Mode	Context	Expected type
1	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\Gamma \vdash \text{g-map} : (\kappa \mid A \to B) \to \text{GStream } A \to \text{GStream } B$$

$\text{g-nats} : \text{GStream } \mathbb{N}$
$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

$\quad 0$

$\quad (\text{g-map } \{\ \}1\ \ \{\ \}2)$

Hole	Mode	Context	Expected type
1	$\star$	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright} . 🔒_{\kappa}$	$\mathbb{N} \to \mathbb{N}$
2	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\Gamma \vdash \text{g-map} : (\kappa \mid A \to B) \to \text{GStream } A \to \text{GStream } B$$

g-nats : GStream $\mathbb{N}$
g-nats = löb[$\triangleright \mid s$: GStream $\mathbb{N}$] g-cons

0

(g-map {suc}1 { }2)

Hole	Mode	Context	Expected type
1	$\star$	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . \text{🔒}_{\triangleright} . \text{🔒}_{\kappa}$	$\mathbb{N} \to \mathbb{N}$
2	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . \text{🔒}_{\triangleright}$	GStream $\mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\Gamma \vdash \text{g-map} : (\kappa \mid A \to B) \to \text{GStream } A \to \text{GStream } B$$

$\text{g-nats} : \text{GStream } \mathbb{N}$
$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

  0

  $(\text{g-map suc } \{\ \}2)$

Hole	Mode	Context	Expected type
2	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N})\,.\,\text{🔒}_{\triangleright}$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\Gamma \vdash \text{g-map} : (\kappa \mid A \to B) \to \text{GStream } A \to \text{GStream } B$$

$\text{g-nats} : \text{GStream } \mathbb{N}$
$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

$\quad 0$

$\quad (\text{g-map suc } \{s\}2)$

Hole	Mode	Context	Expected type
2	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Implementing nats in MTT

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\Gamma \vdash \text{g-map} : (\kappa \mid A \to B) \to \text{GStream } A \to \text{GStream } B$$

$\text{g-nats} : \text{GStream } \mathbb{N}$
$\text{g-nats} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons}$

$\quad 0$

$\quad (\text{g-map suc } s)$

Hole	Mode	Context	Expected type

Variables & 2-cells

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\begin{aligned}&\text{g-toggle} : \text{GStream } \mathbb{N}\\&\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\\&\quad \text{g-cons } 0 \ (\text{g-cons } 1 \ \{\ \}0\ )\end{aligned}$$

Hole	Mode	Context	Expected type
0	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright} . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Variables & 2-cells

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\begin{aligned}&\text{g-toggle} : \text{GStream } \mathbb{N}\\&\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\\&\quad\quad \text{g-cons } 0 \ (\text{g-cons } 1 \ \{s\}0\ )\end{aligned}$$

Hole	Mode	Context	Expected type
0	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_{\triangleright} . 🔒_{\triangleright}$	$\text{GStream } \mathbb{N}$

Variables & 2-cells

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\text{adv} \in \mathbb{1}_\omega \Rightarrow \triangleright \qquad (\text{adv} \circ \triangleright) \in \triangleright \Rightarrow \triangleright^2$$

$$\begin{aligned}
&\text{g-toggle} : \text{GStream } \mathbb{N} \\
&\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}] \\
&\qquad \text{g-cons } 0 \; (\text{g-cons } 1 \; \{s^{\text{adv}\circ\triangleright}\}0 \;)
\end{aligned}$$

Hole	Mode	Context	Expected type
0	ω	$(\triangleright \mid s : \text{GStream } \mathbb{N}) . 🔒_\triangleright . 🔒_\triangleright$	$\text{GStream } \mathbb{N}$

Variables & 2-cells

$$\Gamma \vdash \text{g-cons} : (\kappa \mid A) \to (\triangleright \mid \text{GStream } A) \to \text{GStream } A$$

$$\text{adv} \in \mathbb{1}_\omega \Rightarrow \triangleright \qquad\qquad (\text{adv} \circ \triangleright) \in \triangleright \Rightarrow \triangleright^2$$

$\text{g-toggle} : \text{GStream } \mathbb{N}$
$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]$

$$\text{g-cons } 0 \ (\text{g-cons } 1 \ s^{\text{adv} \circ \triangleright})$$

Hole	Mode	Context	Expected type

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv} \circ \triangleright})$$

We want $\text{g-toggle} \equiv \text{g-cons } 0\ (\text{g-cons } 1\ \text{g-toggle})$

General idea:

$$\text{löb}[\triangleright \mid x : T]\ t \equiv t\ [x \mapsto \text{löb}[\triangleright \mid x : T]\ t]$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv} \circ \triangleright})$$

We want g-toggle $\equiv$ g-cons 0 (g-cons 1 g-toggle)

General idea:

$$\text{löb}[\triangleright \mid x : T]\ t \equiv t\ [x \mapsto \text{löb}[\triangleright \mid x : T]\ t]$$

However:

$$\frac{\mu : m \to n \qquad \vdash \Gamma\ \text{Ctx} \ @\ n \qquad \Gamma . 🔒_{\mu} \vdash S\ \ \text{Ty} \ @\ m}{\vdash \Gamma . (\mu \mid x : S)\ \text{Ctx} \ @\ n}$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

We want g-toggle $\equiv$ g-cons 0 (g-cons 1 g-toggle)

General idea:

$$\text{löb}[\triangleright \mid x : T]\ t \equiv t\ [x \mapsto \text{löb}[\triangleright \mid x : T]\ t]$$

However:

$$\frac{\mu : m \to n \qquad \vdash \Gamma\ \text{Ctx} @ n \qquad \Gamma . 🔒_{\mu} \vdash S\ \ \text{Ty} @ m}{\vdash \Gamma . (\mu \mid x : S)\ \text{Ctx} @ n}$$

$$\frac{\Gamma . 🔒_{\mu} \vdash s : S\ @ m}{\Gamma \vdash (x \mapsto s) : \Gamma . (\mu \mid x : S)\ @ n}$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

We want $\text{g-toggle} \equiv \text{g-cons } 0\ (\text{g-cons } 1\ \text{g-toggle})$

General idea:

$$\text{löb}[\triangleright \mid x : T]\ t \equiv t\ [x \mapsto \text{löb}[\triangleright \mid x : T]\ t]$$

However:

$$\frac{\triangleright : \omega \to \omega \qquad \vdash \Gamma\ \text{Ctx} \ @\ \omega \qquad \Gamma . \text{🔒}_{\triangleright} \vdash T\ \text{Ty} \ @\ \omega}{\vdash \Gamma . (\triangleright \mid x : T)\ \text{Ctx} \ @\ \omega}$$

$$\frac{\Gamma . \text{🔒}_{\triangleright} \vdash ? : T \ @\ \omega}{\Gamma \vdash (x \mapsto ?) : \Gamma . (\triangleright \mid x : T) \ @\ \omega}$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

We want $\text{g-toggle} \equiv \text{g-cons } 0\ (\text{g-cons } 1\ \text{g-toggle})$

General idea:

$$\text{löb}[\triangleright \mid x : T]\ t \equiv t\left[x \mapsto (\text{löb}[\triangleright \mid x : T]\ t)\left[\text{🔑}^{\text{adv}}_{\Gamma}\right]\right]$$

However:

$$\frac{\triangleright : \omega \to \omega \qquad \vdash \Gamma\ \text{Ctx} @ \omega \qquad \Gamma . \text{🔒}_{\triangleright} \vdash T\ \text{Ty} @ \omega}{\vdash \Gamma . (\triangleright \mid x : T)\ \text{Ctx} @ \omega}$$

$$\frac{\Gamma . \text{🔒}_{\triangleright} \vdash ? : T @ \omega}{\Gamma \vdash (x \mapsto ?) : \Gamma . (\triangleright \mid x : T) @ \omega}$$

$$\Gamma . \text{🔒}_{\triangleright} \vdash \text{🔑}^{\text{adv}}_{\Gamma} : \Gamma$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

$$\Gamma \quad \vdash \sigma \quad = \left(s \mapsto \text{g-toggle}\left[\text{🔑}^{\text{adv}}_{\Gamma}\right]\right) \quad : \Gamma\,.\,(\triangleright \mid s : \text{GStream } \mathbb{N})$$

$$\text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,]$$

$$= \text{g-cons}$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\,\triangleright \mid s : \text{GStream } \mathbb{N}\,]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

$$\Gamma\,.\,🔒_\kappa \vdash \sigma\,.\,🔒_\kappa = \left(s \mapsto \text{g-toggle}\left[\,🔑^{\text{adv}}_{\Gamma}\,\right]\right).\,🔒_\kappa : \Gamma\,.\,(\triangleright \mid s : \text{GStream } \mathbb{N})\,.\,🔒_\kappa$$

$$\text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,]$$

$$= \text{g-cons } (0\ [\,\sigma\,.\,🔒_\kappa\,])\ \ldots$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream}\ \mathbb{N}]\ \text{g-cons}\ 0\ (\text{g-cons}\ 1\ s^{\text{adv}\circ\triangleright})$$

$$\Gamma . \text{🔒}_{\triangleright} \vdash \sigma . \text{🔒}_{\triangleright} = \left(s \mapsto \text{g-toggle}\left[\text{🔑}^{\text{adv}}_{\Gamma}\right]\right) . \text{🔒}_{\triangleright} : \Gamma . (\triangleright \mid s : \text{GStream}\ \mathbb{N}) . \text{🔒}_{\triangleright}$$

$$\begin{aligned} &\text{g-cons}\ 0\ (\text{g-cons}\ 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,] \\ &= \text{g-cons}\ (0\ [\,\sigma . \text{🔒}_{\kappa}\,])\ \left((\text{g-cons}\ 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma . \text{🔒}_{\triangleright}\,]\right) \end{aligned}$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

$$\Gamma \quad \vdash \sigma \quad = \left(s \mapsto \text{g-toggle}\left[\text{🔑}^{\text{adv}}_{\Gamma}\right]\right) \quad : \Gamma\,.\,(\triangleright \mid s : \text{GStream } \mathbb{N})$$

$$\begin{aligned}
&\text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,] \\
&= \text{g-cons } (0\ [\,\sigma\,.\,\text{🔒}_{\kappa}\,])\ \left((\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,.\,\text{🔒}_{\triangleright}\,]\right) \\
&= \text{g-cons } 0\ \ldots
\end{aligned}$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

$$\Gamma \qquad \vdash \sigma \qquad = \left(s \mapsto \text{g-toggle}\left[\text{🔑}^{\text{adv}}_{\Gamma}\right]\right) \qquad : \Gamma\,.\,(\triangleright \mid s : \text{GStream } \mathbb{N})$$

$$\begin{aligned}
&\text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,] \\
&= \text{g-cons } (0\ [\,\sigma\,.\,🔒_{\kappa}\,])\ \big((\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,.\,🔒_{\triangleright}\,]\big) \\
&= \text{g-cons } 0\ \Big(\text{g-cons }\ (1\ [\,\sigma\,.\,🔒_{\triangleright}\,.\,🔒_{\kappa}\,])\ \big(s^{\text{adv}\circ\triangleright}\ [\,\sigma\,.\,🔒_{\triangleright}\,.\,🔒_{\triangleright}\,]\big)\Big)
\end{aligned}$$

How does g-toggle unfold?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

$$\Gamma \qquad \vdash \sigma \qquad = \left(s \mapsto \text{g-toggle}\left[\text{🔑}^{\text{adv}}_{\Gamma}\right]\right) \qquad : \Gamma\,.\,(\triangleright \mid s : \text{GStream } \mathbb{N})$$

$$\begin{aligned}
&\text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,] \\
&= \text{g-cons } (0\ [\,\sigma\,.\,\text{🔒}_{\kappa}\,])\ \left((\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})\ [\,\sigma\,.\,\text{🔒}_{\triangleright}\,]\right) \\
&= \text{g-cons } 0\ \left(\text{g-cons }\ (1\ [\,\sigma\,.\,\text{🔒}_{\triangleright}\,.\,\text{🔒}_{\kappa}\,])\ \left(s^{\text{adv}\circ\triangleright}\ [\,\sigma\,.\,\text{🔒}_{\triangleright}\,.\,\text{🔒}_{\triangleright}\,]\right)\right) \\
&= \text{g-cons } 0\ \left(\text{g-cons } 1\ \left(s^{\text{adv}\circ\triangleright}\ \left[\left(s \mapsto \text{g-toggle}\left[\text{🔑}^{\text{adv}}_{\Gamma}\right]\right)\,.\,\text{🔒}_{\triangleright}\,.\,\text{🔒}_{\triangleright}\,\right]\right)\right)
\end{aligned}$$

Difficulties with Substitution in MTT

- MTT substitution $\neq$ list of terms.
  - keys, locks, ...
- MTT has explicit substitution constructor for terms.
  - I.e. substituted terms are part of syntax.
  - System of laws governing interaction with other constructors.
- Can MTT substitution be "computed away"?
  - Preferably in a structurally recursive way.

2 Modal Systems

WSMTT	SFMTT
extrinsically typed, intrinsically scoped	extrinsically typed, intrinsically scoped
explicit substitutions	no substitution constructor for terms
same substitution constructors as MTT	definition of substitution tailored to algorithm

Intrinsic scoping:

$$\frac{\hat{\Gamma} \,.\, (\mu \mid \_) \vdash_{\mathsf{sf}} t \text{ expr} @ m}{\hat{\Gamma} \vdash_{\mathsf{sf}} \lambda^{\mu}(t) \text{ expr} @ m} \qquad \frac{\hat{\Gamma} \,.\, 🔒_{\mu} \vdash_{\mathsf{sf}} t \text{ expr} @ m}{\hat{\Gamma} \vdash_{\mathsf{sf}} \mathsf{mod}_{\mu}(t) \text{ expr} @ n}$$

Substitution in SFMTT

Implementation in 3 stages:

1. Atomic renamings: $x \mapsto y$, 🔑, 🔒, ... (but no composition)
2. Atomic substitutions : $x \mapsto t$, 🔑, 🔒, ... (but no composition)
3. General substitutions: also composition

Substitution in SFMTT

Implementation in 3 stages:

1. Atomic renamings: $x \mapsto y$, 🔑, 🔒, … (but no composition)
2. Atomic substitutions : $x \mapsto t$, 🔑, 🔒, … (but no composition)
3. General substitutions: also composition

Example: why atomic renamings?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}]\ \text{g-cons } 0\ (\text{g-cons } 1\ s^{\text{adv}\circ\triangleright})$$

$$\text{g-cons } 0\ \left(\text{g-cons } 1\ \left(s^{\text{adv}\circ\triangleright}\left[\left(s \mapsto \text{g-toggle}\left[\text{🔑}_{\Gamma}^{\text{adv}}\right]\right) . \text{🔒}_{\triangleright} . \text{🔒}_{\triangleright}\right]\right)\right)$$

Substitution in SFMTT

Implementation in 3 stages:

1. Atomic renamings: $x \mapsto y$, 🔑, 🔒, … (but no composition)
2. Atomic substitutions : $x \mapsto t$, 🔑, 🔒, … (but no composition)
3. General substitutions: also composition

Example: why atomic renamings?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream}\ \mathbb{N}]\ \text{g-cons}\ 0\ (\text{g-cons}\ 1\ s^{\text{adv}\circ\triangleright})$$

$$\text{g-cons}\ 0\ \left(\text{g-cons}\ 1\ \left(s^{\text{adv}\circ\triangleright}\left[\left(s \mapsto \text{g-toggle}\left[🔑_{\Gamma}^{\text{adv}}\right]\right).🔒_{\triangleright}.🔒_{\triangleright}\right]\right)\right)$$

$$= \text{g-cons}\ 0\ \left(\text{g-cons}\ 1\ \left(\left(\text{g-toggle}\left[🔑_{\Gamma}^{\text{adv}}\right]\right)\left[🔑_{\Gamma}^{\text{adv}\circ\triangleright}\right]\right)\right)$$

Substitution in SFMTT

Implementation in 3 stages:

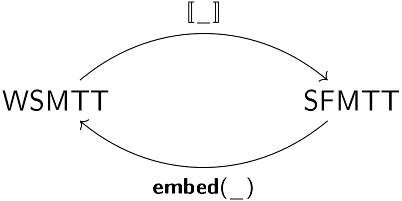
1. Atomic renamings: $x \mapsto y$, 🔑, 🔒, ... (but no composition)
2. Atomic substitutions : $x \mapsto t$, 🔑, 🔒, ... (but no composition)
3. General substitutions: also composition

Example: why atomic renamings?

$$\text{g-toggle} = \text{löb}[\triangleright \mid s : \text{GStream } \mathbb{N}] \ \text{g-cons } 0 \ (\text{g-cons } 1 \ s^{\text{adv}\circ\triangleright})$$

$$\begin{aligned} & \text{g-cons } 0 \ \left(\text{g-cons } 1 \ \left(s^{\text{adv}\circ\triangleright} \left[\left(s \mapsto \text{g-toggle} \left[\text{🔑}^{\text{adv}}_{\Gamma} \right]\right) . \text{🔒}_{\triangleright} . \text{🔒}_{\triangleright} \right]\right)\right) \\ = \ & \text{g-cons } 0 \ \left(\text{g-cons } 1 \ \left(\left(\text{g-toggle} \left[\text{🔑}^{\text{adv}}_{\Gamma} \right]\right) \left[\text{🔑}^{\text{adv}\circ\triangleright}_{\Gamma} \right]\right)\right) \\ = \ & \text{g-cons } 0 \ (\text{g-cons } 1 \ \text{g-toggle}) \end{aligned}$$

Soundness & Completeness



Theorem (Soundness)

$\textbf{\textit{embed}}([\![t]\!]) =^{\sigma} t.$

Theorem (Completeness)

If $t =^{\sigma} s$, *then* $[\![t]\!] = [\![s]\!]$.

Thank you for listening!
Questions?

preprint