Towards FaceTT

A generalization of intensional type systems with Glue

Tex Schönlank[1], Andreas Nuyts[2], Dominique Devriese[3]

[1,2,3] DistriNet, KU Leuven, Belgium
TYPES 2026

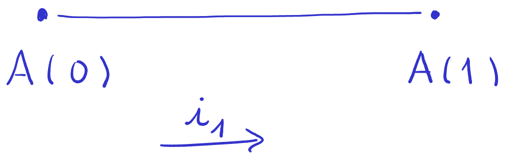
[1] tex.schonlank@kuleuven.be

2026-05-06

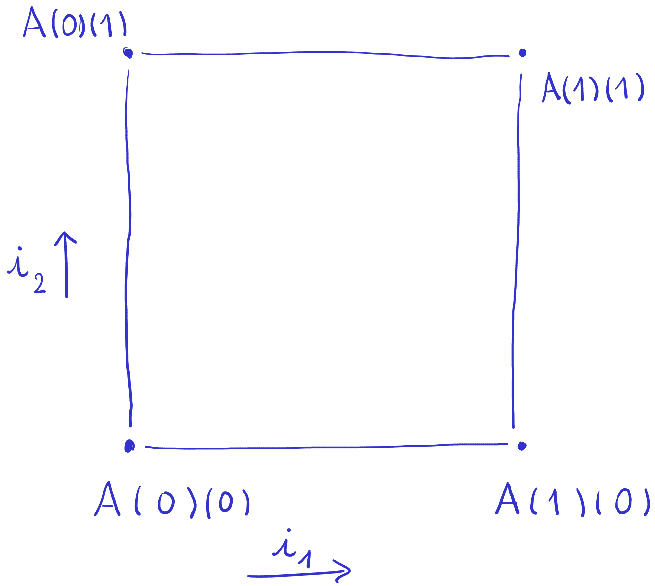
1. CCHM's Cubical type theory [Coh+15]

A

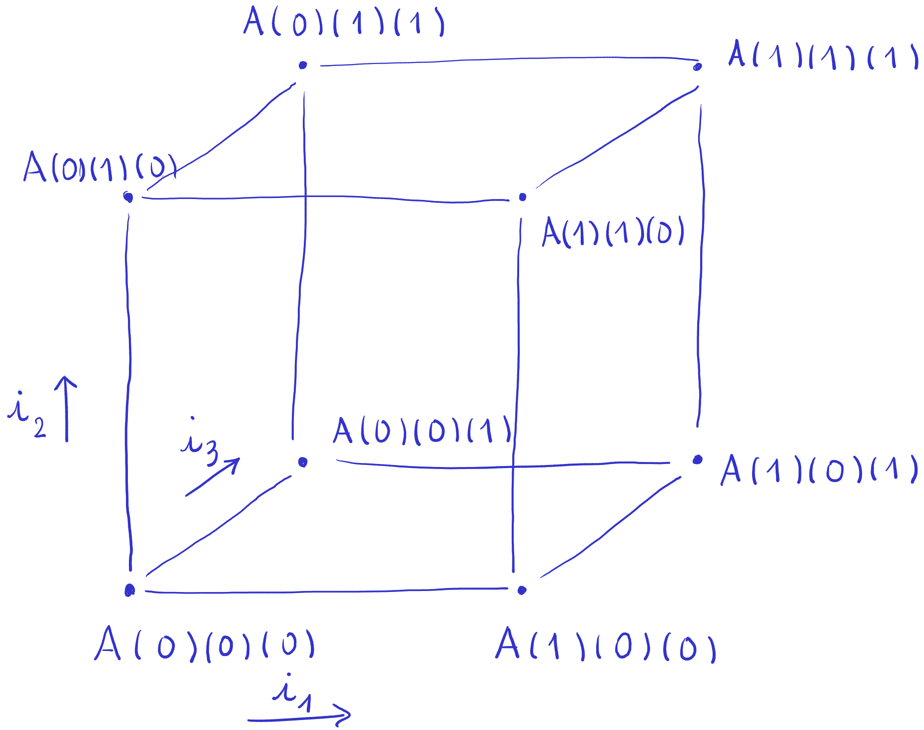
$$\Gamma \quad\quad \vdash A \quad\quad \mathsf{type}$$



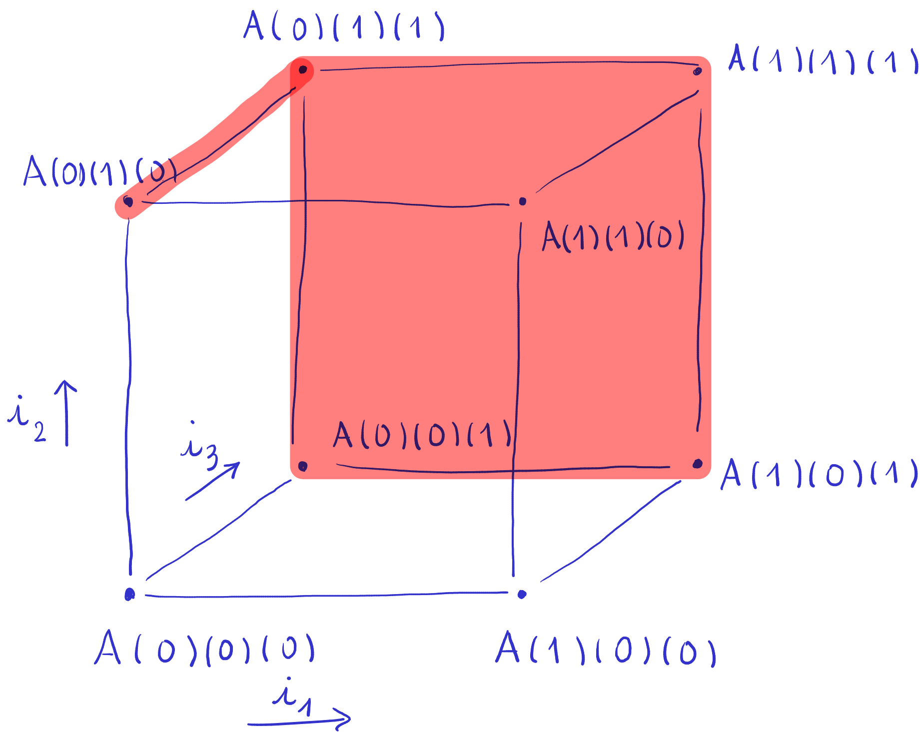
$$\Gamma, i_1 : \mathbb{I} \vdash A(i_1) \quad \text{type}$$



$$\Gamma, i_1 : \mathbb{I}, i_2 : \mathbb{I} \qquad \vdash A(i_1)(i_2) \qquad \textsf{type}$$



$$\Gamma, i_1 : \mathbb{I}, i_2 : \mathbb{I}, i_3 : \mathbb{I} \vdash A(i_1)(i_2)(i_3) \ \mathsf{type}$$



$$\Gamma, i_1 : \mathbb{I}, i_2 : \mathbb{I}, i_3 : \mathbb{I} \vdash A(i_1)(i_2)(i_3) \ \mathsf{type}$$

$$\Gamma, i_1 : \mathbb{I}, i_2 : \mathbb{I}, i_3 : \mathbb{I}, (i_3 = 1 \vee (i_1 = 0 \wedge i_2 = 1)) \vdash B \ \mathsf{type}$$

CCHM use the Glue type former:

$$\Gamma, \varphi \quad \vdash \quad A \xleftarrow[f]{\sim} T$$

$$\Gamma \quad \vdash \quad A$$

CCHM use the Glue type former:

$$\Gamma, \varphi \quad \vdash \quad A \xleftarrow[f]{\sim} T = \mathsf{Glue}(\top, T, A, f)$$

$$\Gamma \quad \vdash \quad A \xleftarrow[\mathsf{unglue}]{} \mathsf{Glue}(\varphi, T, A, f)$$

CCHM use the Glue type former:

$$\Gamma, \varphi \quad \vdash \quad A \xleftarrow[f]{\sim} T = \mathsf{Glue}(\top, T, A, f)$$

$$\Gamma \quad \vdash \quad A \xleftarrow[\mathsf{unglue}]{} \mathsf{Glue}(\varphi, T, A, f)$$

… to prove *univalence*:

$$\Gamma \vdash \ldots : (A \simeq B) \to \left(A \underset{\mathcal{U}}{\equiv} B\right)$$

There are other systems

- Parametric Dependent Type Theory [NVD17] (without modalities)
- Cartesian Cubical Type Theory [Ang+21]
- [RS17]'s type theory for synthetic $(\infty, 1)$-categories

There are other systems

- Parametric Dependent Type Theory [NVD17] (without modalities)
- Cartesian Cubical Type Theory [Ang+21]
- [RS17]'s type theory for synthetic $(\infty, 1)$-categories

… that use (a selection of) similar features:

- shape pseudotypes ($\mathbb{I}$)
- shape operations ($0 \sqcup (i_1 \sqcap i_2)$)
- face formula operations ($\varphi \land \psi$)
- a Glue type that uses these features

2. Goal: generalize!

2.1 Goals

- Define type system FaceTT.
  Parametrized in a *face theory* ft.

2.1 Goals

- Define type system FaceTT.
  Parametrized in a *face theory* ft.

- Have FaceTT(ft) be *almost* the existing type systems.
  (For well-chosen ft.)
  (Kan-fibrancy is a non-goal.)

2.1 Goals

- Define type system FaceTT.
  Parametrized in a *face theory* ft.
- Have $\mathsf{FaceTT}(\mathsf{ft})$ be *almost* the existing type systems.
  (For well-chosen ft.)
  (Kan-fibrancy is a non-goal.)
- Prove soundness and computational properties once and for all.

2.1 Goals

- Define type system FaceTT.
  Parametrized in a *face theory* ft.

- Have FaceTT(ft) be *almost* the existing type systems.
  (For well-chosen ft.)
  (Kan-fibrancy is a non-goal.)

- Prove soundness and computational properties once and for all.

- This saves redundant work in defining type systems.

2.1 Goals

- Define type system FaceTT.
  Parametrized in a *face theory* ft.
- Have FaceTT(ft) be *almost* the existing type systems.
  (For well-chosen ft.)
  (Kan-fibrancy is a non-goal.)
- Prove soundness and computational properties once and for all.
- This saves redundant work in defining type systems.

(Continuing [Nuy20])

2.2 Structure

	Syntax & derivation	Semantics	Algorithmics & type checking
In	ft $: \mathrm{FaceTheory}$	sem $: \mathrm{SemAssump}(\mathsf{ft})$	ass $: \mathrm{AlgoAssump}(\mathsf{ft})$
Out	$\mathsf{FaceTT}(\mathsf{ft})$ $: \mathrm{TypeSystem}$	$\mathrm{model}(\mathsf{ft}, \mathrm{sem})$ $: \mathrm{Model}(\mathsf{FaceTT}(\mathsf{ft}))$	$\mathrm{algoProof}(\mathsf{ft}, \mathrm{ass})$ $: \mathrm{AlgoProp}(\mathrm{FaceTT}(\mathsf{ft}))$

2.2 Structure

	Syntax & derivation	Semantics	Algorithmics & type checking
In	ft $: \mathrm{FaceTheory}$	sem $: \mathrm{SemAssump}(\mathsf{ft})$	ass $: \mathrm{AlgoAssump}(\mathsf{ft})$
Out	$\mathsf{FaceTT}(\mathsf{ft})$ $: \mathrm{TypeSystem}$	$\mathrm{model}(\mathsf{ft}, \mathrm{sem})$ $: \mathrm{Model}(\mathsf{FaceTT}(\mathsf{ft}))$	$\mathrm{algoProof}(\mathsf{ft}, \mathrm{ass})$ $: \mathrm{AlgoProp}(\mathrm{FaceTT}(\mathsf{ft}))$

2.3 FaceTheory

Definition A *face theory* is (approximately) a tuple of

- **Shape pseudotypes**
  - CCHM: $\mathbb{I}$
  - [RS17]: I, $\mathbb{2}$

2.3 FaceTheory

Definition A *face theory* is (approximately) a tuple of

- **Shape pseudotypes**
  - CCHM: $\mathbb{I}$
  - [RS17]: I, $\mathbb{2}$
- **Shape operations**
  - CCHM: $\sqcup, \sqcap, (1 - \_), 0, 1$

2.3 FaceTheory

Definition A *face theory* is (approximately) a tuple of

- **Shape pseudotypes**
  - CCHM: $\mathbb{I}$
  - [RS17]: I, $\mathbb{2}$
- **Shape operations**
  - CCHM: $\sqcup, \sqcap, (1 - \_), 0, 1$
- **Face formula formers** on top of $\land, \lor, \bot, \top, (i \doteqdot j)$
  - [RS17]: $(i \leq j)$

2.3 FaceTheory

Definition A *face theory* is (approximately) a tuple of

- **Shape pseudotypes**
  - CCHM: $\mathbb{I}$
  - [RS17]: $I, \mathbb{2}$
- **Shape operations**
  - CCHM: $\sqcup, \sqcap, (1 - \_), 0, 1$
- **Face formula formers** on top of $\wedge, \vee, \bot, \top, (i \doteqdot j)$
  - [RS17]: $(i \leq j)$
- **Face formula axioms**
  - CCHM: $(\vdash 0 \sqcup i \doteqdot i), \ldots$
  - [RS17]: $(i \leq j \wedge j \leq k \vdash i \leq k), \ldots$

	Syntax & derivation	Semantics	Algorithmics & type checking
In	ft $: \mathrm{FaceTheory}$	sem $: \mathrm{SemAssump}(\mathsf{ft})$	ass $: \mathrm{AlgoAssump}(\mathsf{ft})$
Out	$\mathsf{FaceTT}(\mathsf{ft})$ $: \mathrm{TypeSystem}$	$\mathrm{model}(\mathsf{ft}, \mathrm{sem})$ $: \mathrm{Model}(\mathsf{FaceTT}(\mathsf{ft}))$	$\mathrm{algoProof}(\mathsf{ft}, \mathrm{ass})$ $: \mathrm{AlgoProp}(\mathrm{FaceTT}(\mathsf{ft}))$

2.5 FaceTT(ft)

Definition Given a face theory ft, then FaceTT(ft) is a dependent type system[1] with:

- Σ, Π, $\mathbf{0}$, $\mathbf{1}$, intensional $(=)$

[1] Essentially a (SO)GAT

2.5 FaceTT(ft)

Definition Given a face theory ft, then FaceTT(ft) is a dependent type system[1] with:

- Σ, Π, $\mathbf{0}$, $\mathbf{1}$, intensional $(=)$
- ft-independent rules: $\dfrac{\Gamma \vdash \mathsf{ctx}}{\Gamma \vdash \top\ \mathsf{fform}}, \dfrac{\Gamma \vdash \varphi, \psi\ \mathsf{fform}}{\Gamma \vdash \varphi \wedge \psi\ \mathsf{fform}}, \ldots$

[1] Essentially a (SO)GAT

2.5 FaceTT(ft)

Definition Given a face theory ft, then FaceTT(ft) is a dependent type system[1] with:

- Σ, Π, $\mathbf{0}$, $\mathbf{1}$, intensional $(=)$
- ft-independent rules: $\dfrac{\Gamma \vdash \mathsf{ctx}}{\Gamma \vdash \top\ \mathsf{fform}}$, $\dfrac{\Gamma \vdash \varphi, \psi\ \mathsf{fform}}{\Gamma \vdash \varphi \wedge \psi\ \mathsf{fform}}$, …
- ft-specific rules.

CCHM: $\dfrac{}{\Gamma \vdash 0 : \mathbb{I}}$, $\dfrac{\Gamma \vdash a, b : \mathbb{I}}{\Gamma \vdash a \sqcup b : \mathbb{I}}$, $\dfrac{\Gamma \vdash a, b : \mathbb{I}}{\Gamma \vdash (a \doteqdot b)\ \mathsf{fform}}$, $\dfrac{\Gamma \vdash a : \mathbb{I}}{\Gamma \vdash a \doteqdot a}$, …

[RS17]: $\dfrac{}{\Gamma \vdash 0 : 2}$, $\dfrac{\Gamma \vdash a, b : 2}{\Gamma \vdash (a \leq b)\ \mathsf{fform}}$, $\dfrac{\Gamma \vdash a : 2}{\Gamma \vdash 0 \leq a}$, …

[1] Essentially a (SO)GAT

	Syntax & derivation	Semantics	Algorithmics & type checking
In	ft $: \mathrm{FaceTheory}$	sem $: \mathrm{SemAssump}(\mathsf{ft})$	ass $: \mathrm{AlgoAssump}(\mathsf{ft})$
Out	$\mathsf{FaceTT}(\mathsf{ft})$ $: \mathrm{TypeSystem}$	$\mathrm{model}(\mathsf{ft}, \mathrm{sem})$ $: \mathrm{Model}(\mathsf{FaceTT}(\mathsf{ft}))$	$\mathrm{algoProof}(\mathsf{ft}, \mathrm{ass})$ $: \mathrm{AlgoProp}(\mathsf{FaceTT}(\mathsf{ft}))$

2.7 SemAssump(ft)

Definition Given a face theory ft, a *semantics* $[\![-]\!]$ of ft consists of

- A category $\mathcal{W}$ (the *base category*).

2.7 SemAssump(ft)

Definition Given a face theory ft, a *semantics* $[\![-]\!]$ of ft consists of

- A category $\mathcal{W}$ (the *base category*).
- Interpretations of ft concepts into the category:

$$\mathsf{ft} \xrightarrow{\quad [\![-]\!] \quad} \mathcal{W}$$

$$\mathbb{I} \qquad\qquad [\![\mathbb{I}]\!]$$

$$\mathsf{op} : \mathbb{I}_1 \times \ldots \times \mathbb{I}_k \to \mathbb{J} \qquad\qquad [\![\mathbb{I}_1]\!] \times \ldots \times [\![\mathbb{I}_k]\!] \xrightarrow[{[\![\mathsf{op}]\!]}]{} [\![\mathbb{J}]\!]$$

$$\vdots \qquad\qquad \vdots$$

	Syntax & derivation	Semantics	Algorithmics & type checking
In	ft $: \mathrm{FaceTheory}$	sem $: \mathrm{SemAssump}(\mathsf{ft})$	ass $: \mathrm{AlgoAssump}(\mathsf{ft})$
Out	$\mathsf{FaceTT}(\mathsf{ft})$ $: \mathrm{TypeSystem}$	$\mathrm{model}(\mathsf{ft}, \mathrm{sem})$ $: \mathrm{Model}(\mathsf{FaceTT}(\mathsf{ft}))$	$\mathrm{algoProof}(\mathsf{ft}, \mathrm{ass})$ $: \mathrm{AlgoProp}(\mathrm{FaceTT}(\mathsf{ft}))$

2.9 $\mathrm{model}(\mathsf{ft}, \mathrm{sem}) : \mathrm{ModelOf}(\mathsf{FaceTT}(\mathsf{ft}))$

(*Categories with families* (*CwFs*) model MLTT [Dyb95].)

A *model* of a type system ts is a GAT model,
i.e. a CwF with enough extra structure to model all the rules in ts.

2.9 $\mathrm{model}(\mathsf{ft}, \mathrm{sem}) : \mathrm{ModelOf}(\mathsf{FaceTT}(\mathsf{ft}))$

(*Categories with families* (*CwFs*) model MLTT [Dyb95].)

A *model* of a type system ts is a GAT model,
i.e. a CwF with enough extra structure to model all the rules in ts.

Given a face theory ft and a semantics sem of ft, we construct a CwF:

- $\mathsf{Ctx} = \mathsf{Psh}(\mathcal{W})$
- $\mathsf{Ty} : \mathsf{Psh}(\mathcal{W})^{\mathsf{op}} \to \mathbf{Set}$
- $\mathsf{Tm} : \mathsf{Psh}(\mathcal{W}/\mathsf{Ty})^{\mathsf{op}} \to \mathbf{Set}$

All according to standard presheaf semantics. [Hof97, HS97]

2.9 $\mathrm{model}(\mathsf{ft}, \mathrm{sem}) : \mathrm{ModelOf}(\mathsf{FaceTT}(\mathsf{ft}))$

(*Categories with families* (*CwFs*) model MLTT [Dyb95].)

A *model* of a type system ts is a GAT model,
i.e. a CwF with enough extra structure to model all the rules in ts.

Given a face theory ft and a semantics sem of ft, we construct a CwF:

- $\mathsf{Ctx} = \mathsf{Psh}(\mathcal{W})$
- $\mathsf{Ty} : \mathsf{Psh}(\mathcal{W})^{\mathsf{op}} \to \mathbf{Set}$
- $\mathsf{Tm} : \mathsf{Psh}(\mathcal{W}/\mathsf{Ty})^{\mathsf{op}} \to \mathbf{Set}$

All according to standard presheaf semantics. [Hof97, HS97]

Proposition This CwF is a model of $\mathsf{FaceTT}(\mathsf{ft})$.
(Based on work in [Nuy20].)

	Syntax & derivation	Semantics	Algorithmics & type checking
In	ft $: \mathrm{FaceTheory}$	sem $: \mathrm{SemAssump}(\mathsf{ft})$	ass $: \mathrm{AlgoAssump}(\mathsf{ft})$
Out	$\mathsf{FaceTT}(\mathsf{ft})$ $: \mathrm{TypeSystem}$	$\mathrm{model}(\mathsf{ft}, \mathrm{sem})$ $: \mathrm{Model}(\mathsf{FaceTT}(\mathsf{ft}))$	$\mathrm{algoProof}(\mathsf{ft}, \mathrm{ass})$ $: \mathrm{AlgoProp}(\mathsf{FaceTT}(\mathsf{ft}))$

2.11 Algorithmics & type checking

Goal: decidability of type checking of $\mathsf{FaceTT}(\mathsf{ft})$.

2.11 Algorithmics & type checking

Goal: decidability of type checking of $\mathsf{FaceTT}(\mathsf{ft})$.

Requires normalization.

2.11 Algorithmics & type checking

Goal: decidability of type checking of $\mathsf{FaceTT}(\mathsf{ft})$.

Requires normalization.

Intermediate goal: prove normalization.

2.11 Algorithmics & type checking

Goal: decidability of type checking of $\mathsf{FaceTT}(\mathsf{ft})$.

Requires normalization.

Intermediate goal: prove normalization.

Requires decidability of $\varphi = \psi$ and $\varphi \vdash \psi$.

2.11 Algorithmics & type checking

Goal: decidability of type checking of $\mathsf{FaceTT}(\mathsf{ft})$.

Requires normalization.

Intermediate goal: prove normalization.

Requires decidability of $\varphi = \psi$ and $\varphi \vdash \psi$.

First goal: find user-provided assumptions necessary for proof.

At least / starting point: decidability of atomic face formula formers.

Bibliography

Angiuli, C., Brunerie, G., Coquand, T., Harper, R., (Favonia), K.-B. H., & Licata, D. R. (2021). Syntax and models of Cartesian cubical type theory. *Math. Struct. Comput. Sci.*, *31*(4), 424–468. https://doi.org/10.1017/S0960129521000347

Cohen, C., Coquand, T., Huber, S., & Mörtberg, A. (2015,). Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. *Types 2015*. https://doi.org/10.4230/LIPIcs.TYPES.2015.5

Dybjer, P. (1995). Internal Type Theory. In S. Berardi & M. Coppo (Eds.), *Types for Proofs and Programs, International Workshop TYPES'95, Torino, Italy, June 5-8, 1995, Selected Papers: Vol. 1158. Types for Proofs* *and Programs, International Workshop TYPES'95, Torino, Italy, June 5-8, 1995, Selected Papers*. https://doi.org/10.1007/3-540-61780-9_66

Hofmann, M. (1997). Syntax and Semantics of Dependent Types. In *Semantics and Logics of Computation: Semantics and Logics of Computation* (pp. 79–130). Cambridge University Press.

Hofmann, M., & Streicher, T. (1997,). *Lifting Grothendieck Universes*. https://www2.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf

Nuyts, A. (2020). *Contributions to Multimode and Presheaf Type Theory* [Doctoral dissertation]. https://lirias.kuleuven.be/3065223

Nuyts, A., Vezzosi, A., & Devriese, D. (2017). Parametric quantifiers for dependent type theory. *PACMPL*, *1*(ICFP), 32:1–32:29. https://doi.org/10.1145/3110276

Riehl, E., & Shulman, M. (2017). A type theory for synthetic ∞-categories. *Higher Structures*, *1*(1), 147–224. https://journals.mq.edu.au/index.php/higher_structures/article/view/36

Schönlank, T., Nuyts, A., & Devriese, D. (2026). Towards FaceTT: a generalization of intensional type systems with Glue. *TYPES 2026*. https://types2026.cse.chalmers.se/abstracts/42.pdf

3. Backup slides

3.1 Weld

$$\Gamma, \varphi \vdash A \xrightarrow[f]{\sim} T$$

$$\Gamma \vdash A$$

3.1 Weld

$$\Gamma, \varphi \vdash A \xrightarrow[f]{\sim} T = \mathsf{Weld}(\top, T, A, f)$$

$$\Gamma \vdash A \overset{\mathsf{weld}}{\dashrightarrow} \mathsf{Weld}(\varphi, T, A, f)$$

3.2 De Morgan algebras

!!Add slide on CCHM's $(- = 1) : \mathbb{I} \to \mathbb{F}$ being a De Morgan algebra morphism.

3.3 Face formula judgement

We have *face formulas* with conjunction, disjunction, truth, falsity and proposition variables: $(\bot \wedge \alpha) \vee (\top \vee \beta)$ (No implication.)

Face formulas can

- be well-defined: $\Gamma \vdash \varphi\ \mathsf{prop}$
- hold: $\Gamma \vdash \varphi$
- go into the context: $\dfrac{\Gamma \vdash \varphi\ \mathsf{prop}}{\Gamma, \varphi \vdash \mathsf{ctx}}$

3.4 Universe of face formulas

New pseudotype judgement for a universe of face formulas:

$$\Gamma \vdash t :: \mathsf{Prop}$$

We can

- convert between the two face formula judgements:

$$\frac{\Gamma \vdash \varphi \ \mathsf{prop}}{\Gamma \vdash \ulcorner \varphi \urcorner :: \mathsf{Prop}} \quad \frac{\Gamma \vdash t :: \mathsf{Prop}}{\Gamma \vdash \mathsf{El}(t) \ \mathsf{prop}}$$

- weaken contexts with variables in the universe:

$$\frac{\Gamma \vdash \mathsf{ctx}}{\Gamma, \chi :: \mathsf{Prop} \vdash \mathsf{ctx}}$$

- have universe variables in Π-type domains:

3.4 Universe of face formulas

$$\frac{\Gamma, \chi :: \mathsf{Prop} \vdash T \ \mathsf{op}(\mathsf{type}_\ell)}{\Gamma \vdash (\chi :: \mathsf{Prop}) \to T \ \mathsf{op}(\mathsf{type}_\ell)}$$

- **not** put the pseudotype in the codomain.

3.5 Equality of face formulas

conjunction

$$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} \qquad \frac{\dots}{\Gamma \vdash \top \wedge \psi = \psi \ \mathsf{prop}}$$

3.5 Equality of face formulas

disjunction

$$\frac{\dots}{\Gamma \vdash \varphi \vee \top = \top \ \mathsf{prop}}$$

$$\frac{\begin{array}{l}\Gamma \vdash T \ \mathsf{type} \\ \Gamma, \varphi \vdash t_l : T \\ \Gamma, \psi \vdash t_r : T \\ \Gamma \vdash \varphi \vee \psi \\ \Gamma, \varphi \wedge \psi \vdash t_l = t_r : T\end{array}}{\Gamma \vdash \{\varphi ?\, t_l \mid \psi ?\, t_r\} : T}$$

$$\Gamma \vdash \{\top ?\, t_l \mid \psi ?\, t_r\} = t_l : T$$

$$\Gamma \vdash \{\varphi ?\, t_l \mid \top ?\, t_r\} = t_r : T$$