Categories, Institutions, Theories, Abstract Data Types, and Development Graphs

Florian Rabe

Part of the course on Computational Logic by Michael Kohlhase

Fall 2007, Jacobs University Bremen

24.09.2007

Motivation

Two views on logic

- proof-oriented $\Rightarrow$ calculi, axiomatizations, proof trees $\Rightarrow$ Michael so far and later
- model-oriented $\Rightarrow$ models, interpretations, categories $\Rightarrow$ me now and later

Related via soundness and completeness

Category Theory

- Abstraction from set-theoretical notions
- Extremely hard to understand
- Occurrence of same pattern in apparently unrelated branches of mathematics
- Extremely helpful to have understood
- Tip: Intuition and images crucial for understanding, more so than the formulas themselves

References

- Introduced by Samuel Eilenberg and Saunders Mac Lane in 1942 – 1945
- Standard reference:

```
@Book{categories,
 author = {S. Mac Lane},
 title = {Categories for the working mathematician
     },
 year = {1998},
 publisher = {Springer}
 }
```

- Recommended read: Steve Awodey's lecture notes, `http://www.andrew.cmu.edu/course/80-413-713/notes/cats.pdf`

Category

A category $\mathcal{C}$ consists of

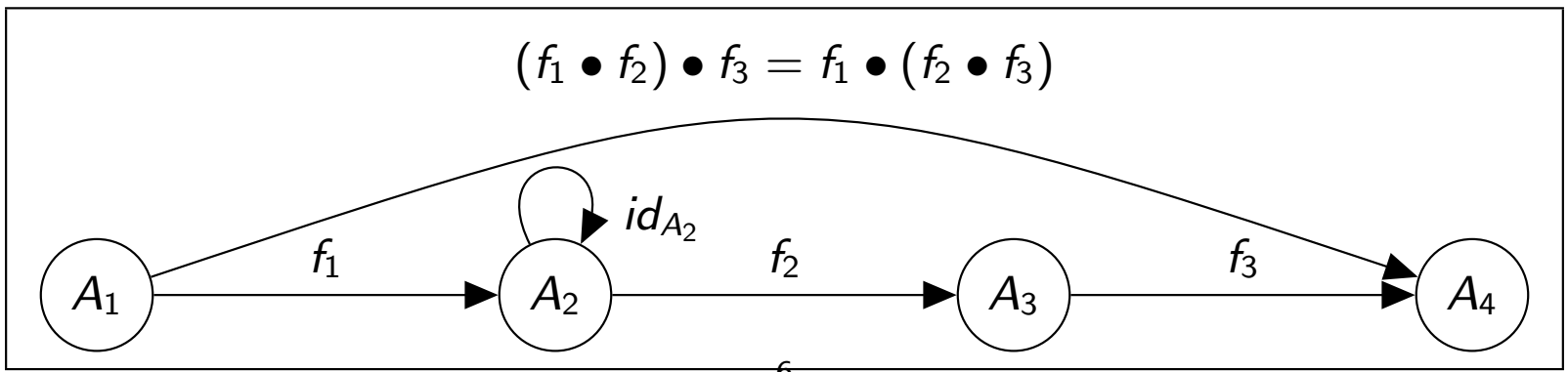
- a collection $|\mathcal{C}|$ of objects
- for two objects $A, B \in |\mathcal{C}|$, a collection $\mathcal{C}(A, B)$ of morphisms (arrows) from A to B

with the operations

- id_-, which assigns to every object $A \in |\mathcal{C}|$ its identity morphism $id_A \in \mathcal{C}(A, A)$
- $- \bullet -$, which assigns to three objects $A_1, A_2, A_3 \in |C|$ and two morphisms $f_1 \in \mathcal{C}(A_1, A_2)$ and $f_2 \in \mathcal{C}(A_2, A_3)$ their composition $f_1 \bullet f_2$

such that for all objects $A_i \in |\mathcal{C}|$ for $i = 1, 2, 3, 4$ and all morphisms $f_i \in \mathcal{C}(A_i, A_{i+1})$ for $i = 1, 2, 3$

- $f_1 \bullet id_{A_2} = f_1$ and $id_{A_2} \bullet f_2 = f_2$ (identity laws)
- $(f_1 \bullet f_2) \bullet f_3 = f_1 \bullet (f_2 \bullet f_3)$ (associativity law)



Example: Sets

The category $\mathcal{S}\mathrm{et}$ is given by

- $|\mathcal{S}\mathrm{et}|$: all sets
- $\mathcal{S}\mathrm{et}\,(A, B)$: mappings from A to B
- Identity and composition: as for mappings

Example: Graphs

A graph G with nodes N and edges $E \subseteq N \times N$, induces a category G^* by

- $|G^*| = N$
- $G^*(a, b) = \{(n_0, \ldots, n_r) \in N^r \mid r \in \mathbb{N}, n_0 = a, n_r = b, (n_{i-1}, n_i) \in E \text{ for all } 1 \leq i \leq r\}$
- Identity: $id_a = (a)$
- Composition: $(n_0, \ldots, n_r) \bullet (m_0, \ldots, m_s) = (n_0, \ldots n_r, m_1, \ldots, m_s)$

In other words: Morphisms from a to b are paths in G from a to b, the identities are the empty paths, and composition is concatenation.

Example: Abstracting from a Set Theoretic Concept

- A morphism $f \in \mathcal{C}(A, B)$ is called an isomorphism iff there is a morphism $f^{-1} \in \mathcal{C}(B, A)$ such that $f \bullet f^{-1} = id_A$ and $f^{-1} \bullet f = id_B$.
- In that case, we say that A and B are isomorphic and write $A \cong B$.
- Isomorphisms in $\mathcal{S}\mathrm{et}$ are exactly the bijections.
- For a graph G, the only isomorphisms in G^* are the identities.

The Category of Signatures

Recall: A FOL signature is a triple (Σ_f, Σ_p, ar) where $ar : \Sigma_f \cup \Sigma_p \to \mathbb{N}$ assigns to each symbol its arity.

- The FOL signatures form a category *Sig* as follows:
  - $|Sig|$: all FOL signatures
  - $\sigma \in Sig\big((\Sigma_f, \Sigma_p, ar), (\Sigma'_f, \Sigma'_p, ar')\big)$ iff $\sigma : \left\{ \begin{array}{l} \Sigma_f \to \Sigma'_f \\ \Sigma_p \to \Sigma'_p \end{array} \right\}$ such that $ar(s) = ar'(\sigma(s))$ for all $s \in \Sigma_f \cup \Sigma_p$.
  - Identity: identity mapping
  - Composition: composition of mappings

The Category of Models

Recall: A model of the FOL signature $\Sigma = (\Sigma_f, \Sigma_p, ar)$ is a pair (U, I) where U is the universe and I is the interpretation function assigning an n-ary function or an n-ary relation to every n-ary function or predicate symbol, respectively.

The models of Σ form a category Mod_Σ as follows:

- $|Mod_\Sigma|$: all Σ-models
- $\varphi \in Mod_\Sigma((U, I), (U', I'))$ iff: $\varphi : U \to U'$ such that
  - for every $f \in \Sigma_f$ with $ar(f) = n$ and all $u_i \in U$:

$$\varphi(f^I(u_1, \ldots, u_n)) = f^{I'}(\varphi(u_1), \ldots, \varphi(u_n))$$

  - for every $p \in \Sigma_p$ with $ar(p) = n$ and all $u_i \in U$:

$$(u_1, \ldots, u_n) \in p^I \Rightarrow (\varphi(u_1), \ldots, \varphi(u_n)) \in p^{I'}$$

- Identity: identity mapping
- Composition: composition of mappings

Remarks on Notation

- For $|\mathcal{C}|$, sometimes the notation $\mathcal{C}_0$ is used.
- For $\mathcal{C}(A, B)$, the notations $Mor_{\mathcal{C}}(A, B)$ and $Hom_{\mathcal{C}}(A, B)$ are also common. The set $\bigcup_{A,B \in |\mathcal{C}|} \mathcal{C}(A, B)$ is sometimes written $\mathcal{C}_1$.
- The notation $f \bullet g$ is not common in the literature. Instead $f ; g$ is used. It is also common to write $g \circ f$ (i.e., with reversed composition order as for the composition of mappings).

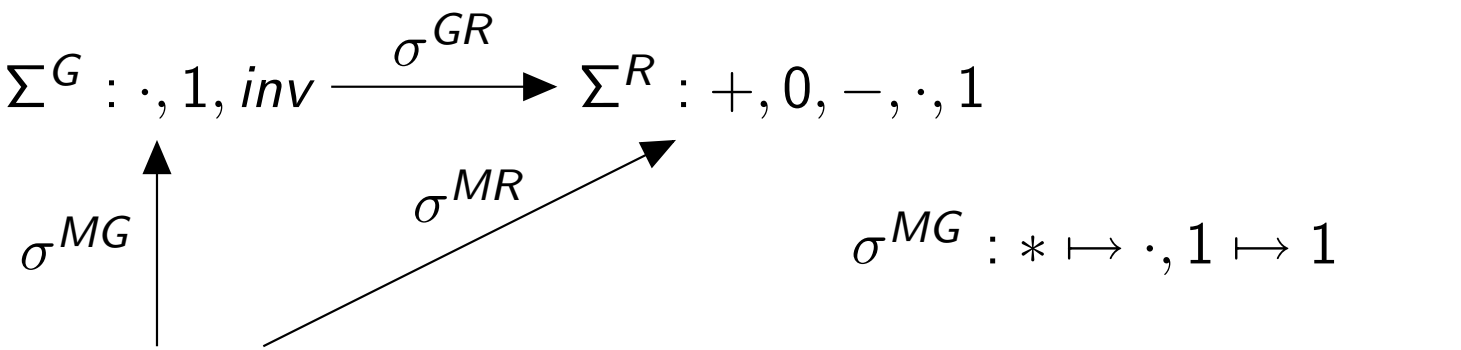
26.09.2007

Motivation: Functors

- Categories abstraction from set-theoretical notions
- Thus applications in different branches of mathematics
- Thus unification of definitions and theorems
- Full power of category theory utilized by relating categories to each other
- To do that, introduction of functors

Diagrams

- A diagram over $\mathcal{C}$ is a multigraph in which all nodes are objects of $\mathcal{C}$ and all edges from A to B are morphisms from A to B.
- Any path in a diagram induces a morphism by composing its edges. A diagram commutes if these morphisms are equal for any two paths between the same nodes.
- Example: A (non-commuting) diagram over *Sig* relating the signatures of monoids, groups, and rings:



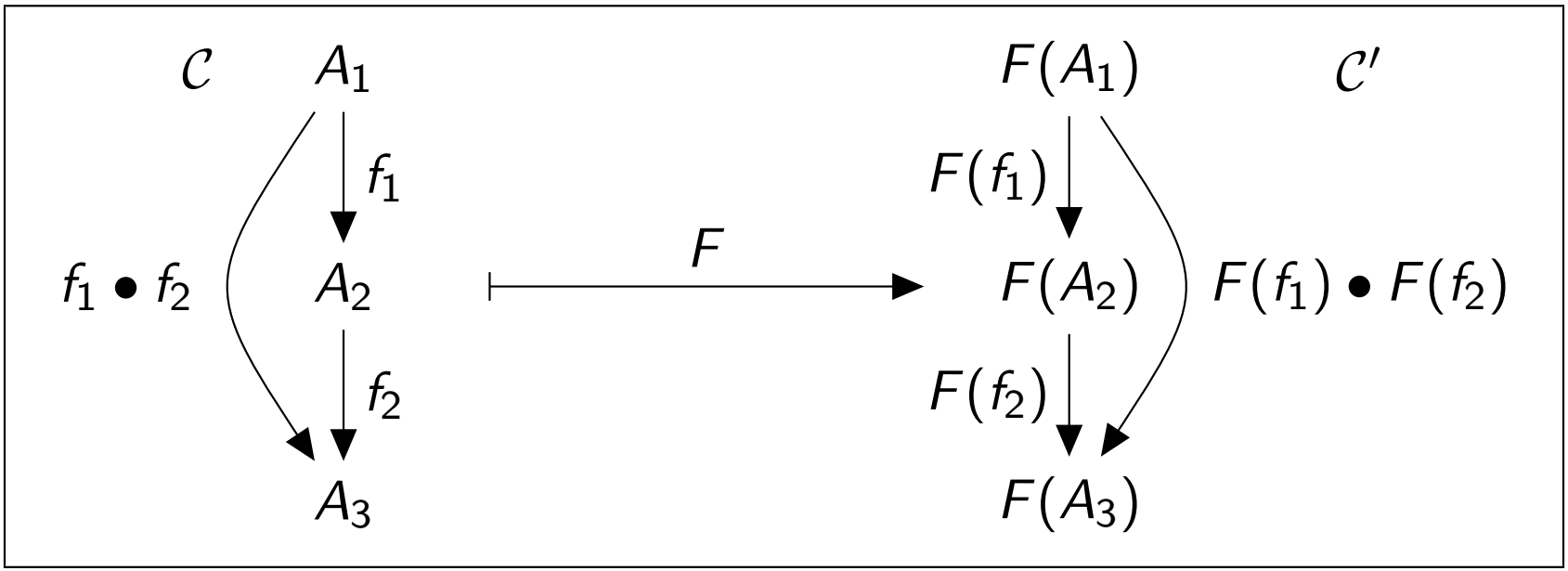
Functors

$F : \mathcal{C} \to \mathcal{C}'$ is a functor from a category $\mathcal{C}$ to a category $\mathcal{C}'$ if

- $F : \begin{cases} |\mathcal{C}| \to |\mathcal{C}'| \\ \mathcal{C}(A, B) \to \mathcal{C}'(F(A), F(B)) \quad \text{for all } A, B \in |\mathcal{C}| \end{cases}$
- such that for all $A_1, A_2, A_3 \in |\mathcal{C}|$ and all $f_i \in \mathcal{C}(A_i, A_{i+1})$ for $i = 1, 2$:

$$F(id_A) = id_{F(A)} \quad \text{and} \quad F(f_1 \bullet f_2) = F(f_1) \bullet F(f_2)$$

Functors map (commuting) diagrams to (commuting) diagrams.



The Category of Categories

$\mathcal{C}\mathrm{at}$ is a category defined by

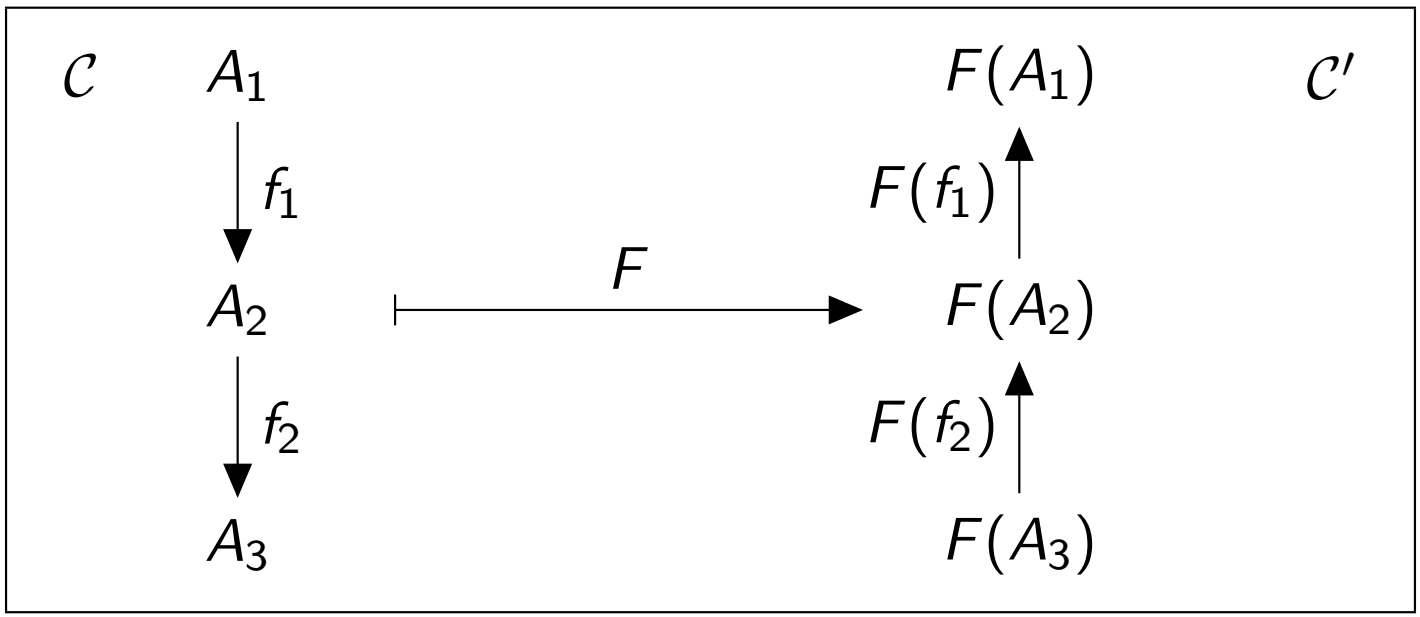
- $|\mathcal{C}\mathrm{at}|$: the collection of all categories
- $\mathcal{C}\mathrm{at}\,(\mathcal{C}, \mathcal{C}')$: the collection of all functors from $\mathcal{C}$ to $\mathcal{C}'$
- $id_{\mathcal{C}}$: maps all objects and morphisms of $\mathcal{C}$ to themselves
- $F \bullet G$: maps an object A to $G(F(A))$ and a morphism f to $G(F(f))$

Duality

For every category $\mathcal{C}$, we define its dual (or opposite) category $\mathcal{C}^{op}$ by:

- $|\mathcal{C}^{op}| = |\mathcal{C}|$
- $\mathcal{C}^{op}(A, B) = \mathcal{C}(B, A)$
- $id_A^{\mathcal{C}^{op}} = id_A^{\mathcal{C}}$
- $f \bullet^{\mathcal{C}^{op}} g = g \bullet^{\mathcal{C}} f$

A functor from $\mathcal{C}$ to $\mathcal{C}'^{op}$ maps (commuting) diagrams over $\mathcal{C}$ to (commuting) diagrams over $\mathcal{C}'$ if all arrows are flipped around:



Duality: Example

- Flipping arrows dualizes concepts, e.g., terminal and initial are dual concepts.
- An object $A \in |\mathcal{C}|$ is called terminal in $\mathcal{C}$ iff for all $B \in |C|$ there is a unique $f \in \mathcal{C}(B, A)$.
- An object $A \in |\mathcal{C}|$ is called initial in $\mathcal{C}$ iff for all $B \in |C|$ there is a unique $f \in \mathcal{C}(A, B)$.
- Going to the opposite category cancels dualization: A is terminal in $\mathcal{C}$ iff it is initial in $\mathcal{C}^{op}$.

Motivation: Institutions

- The concept of institutions abstracts from logical notions like formulas, models, and satisfaction
- Provides common intuition and definitions
- Structures and orders the multitude of different logics
- Institution-independent theorems for general theory of logic

Reference

- Introduced in the 1980s by Joseph Goguen and Rod Burstall
- Standard reference (relatively gentle introduction and lots of examples)

```
@Article{institutions,
  author =        "J. A. Goguen and R. M. Burstall",
  title =         "Institutions: Abstract Model
      Theory for Specification and Programming",
  journal =       "Journal of the Association for
      Computing Machinery",
  volume =        "39(1)",
  pages =         "95--146",
  year =          "1992",
}
```

- Available online (use `http://citeseer.ist.psu.edu/` to find papers)

Institutions

An institution is a tuple $(\mathit{Sig}, \mathit{Sen}, \mathit{Mod}, \models)$ where

- Sig is a category (signatures)
- $\mathit{Sen} : \mathit{Sig} \to \mathcal{S}\mathrm{et}$ is a functor (assigning the set of sentences to a signature)
- $\mathit{Mod} : \mathit{Sig} \to \mathcal{C}\mathrm{at}^{op}$ is a functor (assigning the category of models to a signature)
- $\models_\Sigma \subseteq |\mathit{Mod}(\Sigma)| \times \mathit{Sen}(\Sigma)$ is a relation between Σ-models and Σ-sentences for every $\Sigma \in |\mathit{Sig}|$ (saying whether a formula holds in a model)

such that the satisfaction condition holds (see below).

For a signature morphism σ, $\mathit{Sen}(\sigma)$ is called sentence translation along σ, and $\mathit{Mod}(\sigma)$ is called model reduction along σ.

The Institution *FOL* (1)

The institution *FOL* for first-order logic is given by

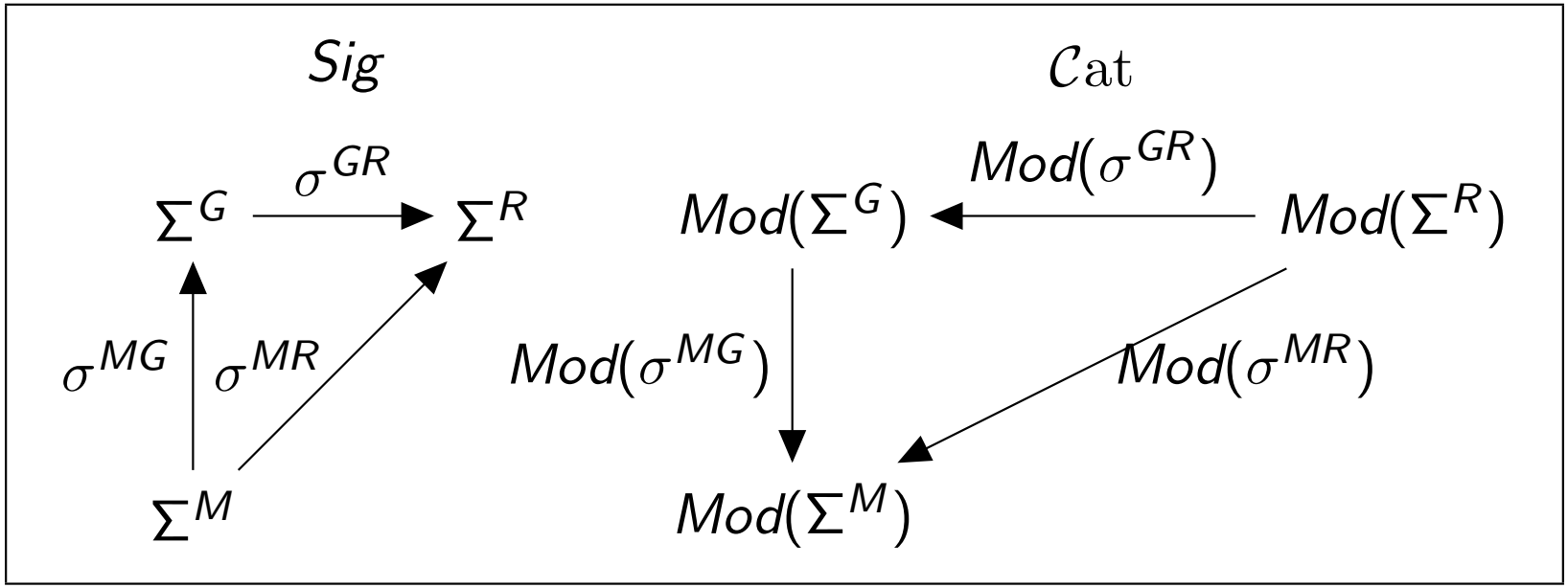
- Sig^{FOL}: signatures as defined before
- $Sen^{FOL} : Sig \to \mathcal{S}\mathrm{et}$ on objects (Σ_f, Σ_p, ar):
  $Sen^{FOL}(\Sigma_f, \Sigma_p, ar)$ is the set of first-order formulas as defined before using function symbols from Σ_f and predicate symbols from Σ_p with their respective arity given by ar
- $Sen^{FOL} : Sig \to \mathcal{S}\mathrm{et}$ on morhisms
  $\sigma : (\Sigma_f, \Sigma_p, ar) \to (\Sigma'_f, \Sigma'_p, ar')$:
  $Sen^{FOL}(\sigma)$ maps a formula $\varphi \in Sen^{FOL}(\Sigma_f, \Sigma_p, ar)$ to itself except that every function or predicate symbol s is replaced with $\sigma(s)$

The Institution *FOL* (2)

- $Mod^{FOL} : Sig \to \mathcal{C}\mathrm{at}^{op}$ on objects Σ:
  $Mod^{FOL}(\Sigma)$ is the category Mod_Σ as defined before
- Mod^{FOL} on morhisms $\sigma : \Sigma \to \Sigma'$:
  $Mod^{FOL}(\sigma) : Mod^{FOL}(\Sigma') \to Mod^{FOL}(\Sigma)$ is a functor defined as follows
  - $Mod^{FOL}(\sigma)$ on objects (U', I'): $Mod^{FOL}(\sigma)(U', I') = (U, I)$ where $U := U'$ and $s^I := \sigma(s)^{I'}$ for all $s \in \Sigma_f \cup \Sigma_p$
  - $Mod^{FOL}(\sigma)$ on morphisms $\varphi : (U_1', I_1') \to (U_2', I_2')$:
    $Mod^{FOL}(\sigma)(\varphi) = \varphi$

The Institution *FOL* (2): Example

Recall the diagram over Sig^{FOL}. Applying Mod^{FOL} and flipping the arrows yields a diagram over $\mathcal{C}\mathrm{at}$. (The superscript *FOL* is dropped below.)



For the ring $\mathbb{Z} \in |Mod(\Sigma^R)|$ of integers, $Mod(\sigma^{GR})(\mathbb{Z})$ is the additive group of the integers, and $Mod(\sigma^{MR})(\mathbb{Z})$ is the multiplicative monoid of the integers.

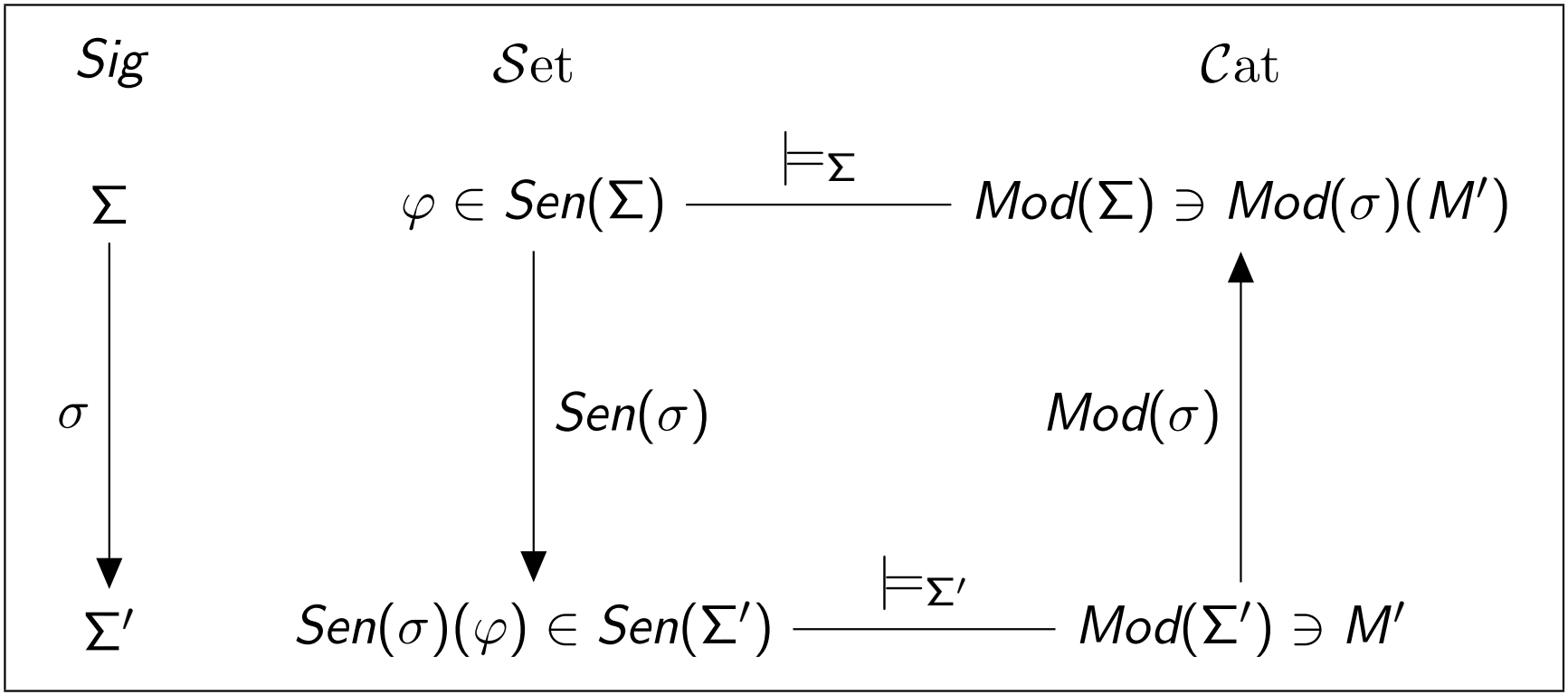
The Institution *FOL* (3)

- For a signature $\Sigma \in |Sig^{FOL}|$, a model $M \in |Mod^{FOL}(\Sigma)|$, and a formula $\varphi \in Sen^{FOL}(\Sigma)$:
  $M \models^{FOL}_{\Sigma} \varphi$ iff M satisfies φ as defined before
- The satisfaction condition will be proven as an exercise.

Satisfaction Condition

For all $\Sigma, \Sigma' \in |\mathit{Sig}|$, $\sigma \in \mathit{Sig}(\Sigma, \Sigma')$, $\varphi \in \mathit{Sen}(\Sigma)$, and $M' \in \mathit{Mod}(\Sigma')$:

$$M' \models_{\Sigma'} \mathit{Sen}(\sigma)(\varphi) \quad \text{iff} \quad \mathit{Mod}(\sigma)(M') \models_{\Sigma} F$$



A Remark on Foundation

- We leave open what a "collection of objects" is in the definition of categories. It cannot always be a set because then $|\mathcal{S}\mathrm{et}|$ would have to contain itself.
- Similarly, the category of categories cannot exist because $|\mathcal{C}\mathrm{at}|$ would have to contain $\mathcal{C}\mathrm{at}$.
- These questions are addressed by axiomatic set theory, which provides the foundation of mathematics.
- In principal, levels are introduced such that
  - Level 0 contains all sets; the elements of a set are other sets.
  - Level 1 contains classes: $|\mathcal{S}\mathrm{et}|$ is the class of all sets, and every class is a subclass of $|\mathcal{S}\mathrm{et}|$. Every set is a class, but not vice versa.
  - Level 2 contains super-classes, i.e., collections that may contain other classes. $|\mathcal{C}\mathrm{at}|$ is such a super-class.

01.10.2007

First-order Logic with Equality

- $Sig^{FOL^=} = Sig^{FOL}$
- $Sen^{FOL^=}(\Sigma)$: like $Sen^{FOL}(\Sigma)$ but with addtional case

$$A, A' \in wff(\Sigma_i) \text{ implies } A \doteq A' \in wff(\Sigma_o)$$

- $Mod^{FOL^=} = Mod^{FOL}$
- $\models^{FOL^=}_{\Sigma}$: like $\models^{FOL}_{\Sigma}$ but with additional case

$$M, \alpha \models^{FOL^=}_{\Sigma} A \doteq A' \text{ iff } [A]^{M,\alpha} = [A']^{M,\alpha}$$

for every assignment α

Subinstitutions

- Assume an institution $I = (Sig^I, Sen^I, Mod^I, \models^I)$
- Various ways to construct subinstutions I' of I:
  - make $Sig^{I'}$ a subcategory of Sig^I
  - make $Sen^{I'}(\Sigma)$ a subset of $Sen^I(\Sigma)$
  - make $Mod^{I'}(\Sigma)$ a subcategory of $Mod^I(\Sigma)$
- Several subinstitutions of $FOL^=$ are interesting
  - No equality: first-order logic FOL
  - No predicate symbols: algebraic logic
  - No predicate symbols and only $\forall$ and $\doteq$ as logical symbols: equational logic

Motivation: Theories

- $Mod(\Sigma)$ contains all possible models
- But we really want to single out certain models by imposing axioms
- Thus: Theories are pairs of a signature and a set of axioms
- Reference: Same as for institutions

Notation

- For this section, assume a fixed institution $(Sig, Sen, Mod, \models)$
- Abbreviate for a set of sentences T and a signature morphism σ:

$$Sen(\sigma)(T) = \{Sen(\sigma)(F) \mid F \in T\}$$

Theories

- A theory is a pair (Σ, T) for $\Sigma \in |Sig|$ and $T \subseteq Sen(\Sigma)$
- The elements of T are called the axioms of the theory
- Example: The theory of monoids is (Σ^M, T^M) where

$$T^M = \{\forall x, y, z.(x*y)*z \doteq x*(y*z),\ \forall x.(x*1 \doteq x \wedge 1*x \doteq x)\}$$

Entailment

- Define $T \models_\Sigma F$ as:
  for all $M \in |Mod(\Sigma)|$
  if $M \models_\Sigma H$ for all $H \in T$, then $M \models_\Sigma F$
- $T \models_\Sigma F$ means
  - F is a theorem of (Σ, T)
  - F is a (semantic) consequence of T
  - T entails F

The Category of Theories

- A signature morphism $\sigma : \Sigma \to \Sigma'$ is a theory morphism from (Σ, T) to (Σ', T') if

$$T' \models_{\Sigma'} Sen(\sigma)(F) \quad \text{for all } F \in T$$

- In particular: if $Sen(\sigma)(T) \subseteq T'$, then σ is a theory morphism
- For any institution I, theories and theory morphisms form a category, denoted by Th^I
- If $\sigma : (\Sigma, T) \to (\Sigma', T')$ is a theory morphism, then:

$$T \models_{\Sigma} F \text{ implies } T' \models_{\Sigma'} Sen(\sigma)(F)$$

- Theorem reuse: Theorem F of (Σ, T) can be moved to (Σ', T') along σ

Theories: Examples

- Groups: (Σ^G, T^G) where

  $$T^G = Sen^{FOL^=}(\sigma^{MG})(T^M) \cup \{\forall x.x \cdot inv(x) \doteq 1 \wedge inv(x) \cdot x \doteq 1\}$$

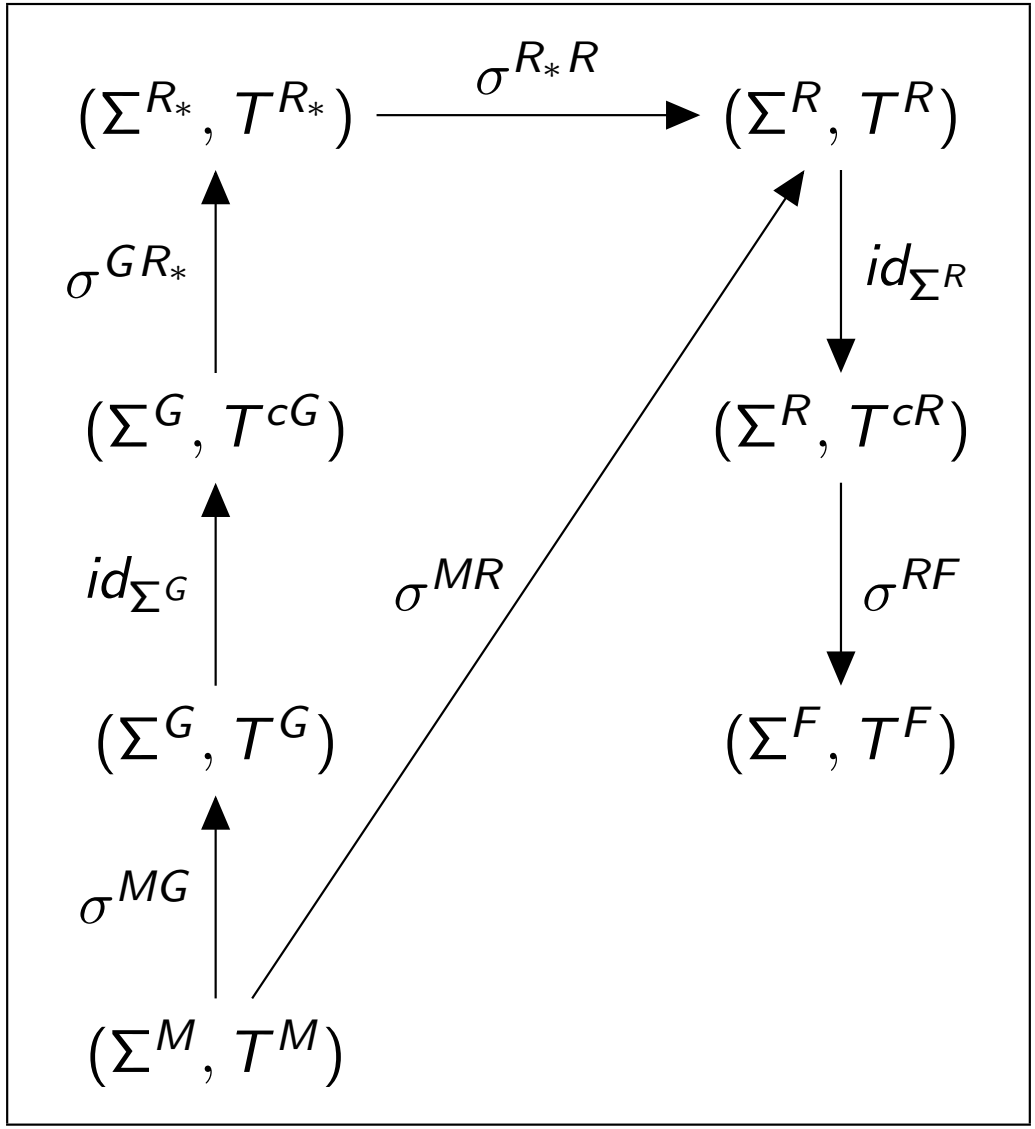
- Commutative groups: (Σ^G, T^{cG}) where
  $T^{cG} = T^G \cup \{\forall x, y.x \cdot y \doteq y \cdot x\}$
- Rings (without 1): (Σ^{R_*}, T^{R_*}) where
  - Σ^{R_*}: $+, 0, -, \cdot$
  - σ^{GR_*}: $\cdot \mapsto +$, $1 \mapsto 0$, $inv \mapsto -$
  - $T^{R_*} = Sen^{FOL^=}(\sigma^{GR_*})(T^{cG}) \cup \Phi$ where Φ contains associativity of $\cdot$ and distributivity of $\cdot$ over $+$
- Rings (with 1): (Σ^R, T^R) where
  - Σ^R: as Σ^{R_*} but with 1
  - $T^R = T^{R_*} \cup \Phi$ where Φ contains neutrality of 1 for $\cdot$
  - σ^{R_*R}: inclusion from Σ^{R_*} to Σ^R

Theories: Examples (2)

- Commutative rings: (Σ^R, T^{cR}) where $T^{cR} = T^R \cup \{\forall x, y.x \cdot y \doteq y \cdot x\}$
- Fields: (Σ^F, T^F) where
  - Σ^F: as Σ^R but with unary *inv*
  - $T^F = T^{cR} \cup \{\forall x.(\neg x \doteq 0 \Rightarrow x \cdot inv(x) \doteq 1\}$
  - σ^{RF}: inclusion from Σ^R to Σ^F

Hierarchy of Algebraic Theories

In $Th^{FOL^{=}}$:



Forgetful Functor

There is a functor $Th \longrightarrow Sig$ given by

- $(\Sigma, T) \mapsto \Sigma$
- $\sigma : (\Sigma, T) \to (\Sigma', T') \quad \mapsto \quad \sigma : \Sigma \to \Sigma'$

It is called forgetful because it forgets the axioms of a theory.

Abstract Data Types (ADT)

- An ADT is a pair $(\Sigma, \mathcal{M})$ for $\Sigma \in |Sig|$ and $\mathcal{M} \subseteq |Mod(\Sigma)|$
- A signature morphism $\sigma : \Sigma \to \Sigma'$ is an ADT morphism from $(\Sigma, \mathcal{M})$ to $(\Sigma', \mathcal{M}')$ if $Mod(\sigma)(M') \in \mathcal{M}$ for all $M' \in \mathcal{M}'$
- ADTs and ADT morphisms form a category ADT

Adjointness of Syntax and Semantics

- Define the ADT of the theory (Σ, T):

$$(\Sigma, T)^* = \left(\Sigma, \left\{M \in |Mod(\Sigma)| \;\middle|\; M \models_\Sigma F \text{ for all } F \in T\right\}\right)$$

- Define the theory the ADT $(\Sigma, \mathcal{M})$:

$$(\Sigma, \mathcal{M})^* = \left(\Sigma, \left\{F \in Sen(\Sigma) \;\middle|\; M \models_\Sigma F \text{ for all } M \in \mathcal{M}\right\}\right)$$

- $-^*$ is a pair of functors $Th \leftrightarrow ADT$
- Define the closure of (Σ, T): $T^\bullet = (\Sigma, T)^{**}$
- Define the closure of $(\Sigma, \mathcal{M})$: $\mathcal{M}^\bullet = (\Sigma, \mathcal{M})^{**}$

Adjointness of Syntax and Semantics (2)

- $-^*$ and $-^\bullet$ satisfy very nice properties, namely that of an adjunction
- If Σ is fixed and omitted, their properties are known as a Galois connection:
  - $T \subseteq T'$ implies $T^* \supseteq T'^*$
  - $\mathcal{M} \subseteq \mathcal{M}'$ implies $\mathcal{M}^* \supseteq \mathcal{M}'^*$
  - $T \subseteq T^\bullet$ and $\mathcal{M} \subseteq \mathcal{M}^\bullet$
  - $T \subseteq T'$ implies $T^\bullet \subseteq T'^\bullet$ and $\mathcal{M} \subseteq \mathcal{M}'$ implies $\mathcal{M}^\bullet \subseteq \mathcal{M}'^\bullet$
  - $T^{\bullet\bullet} = T^\bullet$ and $\mathcal{M}^{\bullet\bullet} = \mathcal{M}^\bullet$
- Entailment: $T \models_\Sigma F$ iff $F \in T^\bullet$

ADT Specification

- ADT specification is the process of finding a theory (Σ, T) such that $(\Sigma, T)^* = (\Sigma, \mathcal{M})$ for a given ADT $(\Sigma, \mathcal{M})$
- The theories of monoids, groups, etc. specify the ADTs of monoids, groups, etc.
- The ADTs $(\Sigma^{\mathbb{N}}, \{M \mid M \cong \mathbb{N}\})$ and $(\Sigma^{F}, \{M \mid M \cong \mathbb{R}\})$ cannot be specified in $FOL^{=}$
  Here $\Sigma^{\mathbb{N}}$ is the signature with the symbols $0, 1, +, \cdot$.
- The ADT $(\Sigma^{F}, \{M \mid M \cong \mathbb{Q}\})$ can be specified in $FOL^{=}$.

08.10.2007

Motivation

- Use morphisms to build big theories out of smaller ones (Little Theories Approach)
- Modularity
- Reuse
- Management of change

References

```
@Article{HSTstructured,
  author =  "R. Harper and D. Sannella and A. Tarlecki
      ",
  title =  "Structured Presentations and Logic
      Representations",
  journal =  "Annals of Pure and Applied Logic",
  year =  1994,
  volume =  67,
  pages =  "113--160",
}

@InProceedings{devgraphs,
  title = "Towards an Evolutionary Formal Software-
      Development Using {CASL}",
  author = "S. Autexier and D. Hutter and H. Mantel and
       A. Schairer",
  series = "Lecture Notes in Computer Science",
  year =  "1999",
  volume = "1827",
  pages = "73--88",
}
```

Structured Specifications

- A language to build theories over an arbitrary institution
- Syntax: The collection $STH(\Sigma)$ of structured theories with signature Σ is given by
  - Presentations: for a theory (Σ, T) with finite T, $(\Sigma, T) \in STH(\Sigma)$
  - Union: If $(\vartheta_i) \in STH$ for $i = 1, 2$, then $\vartheta_1 \cup \vartheta_2 \in STH(\Sigma)$
  - Translations: If $\vartheta \in STH(\Sigma)$ and $\sigma : \Sigma \to \Sigma'$, then $\sigma(\vartheta) \in STH(\Sigma')$
  - Derivation/Hiding: If $\vartheta \in STH(\Sigma')$ and $\sigma : \Sigma \to \Sigma'$, then $\sigma^{-1}(\vartheta) \in STH(\Sigma)$

Notation

- For sets $M' \subseteq M$, $N' \subseteq N$, and a map $f : M \longrightarrow N$, we write

$$f(M') = \{f(m) \in N \mid m \in M'\}$$

and

$$f^{-1}(N') = \{m \in M \mid f(m) \in N'\}$$

- In particular: $Sen(\sigma)(T)$ is the set of translations along σ of formulas in T. And $Sen(\sigma)^{-1}(T)$ is the set of formulas that are translated along σ to a formula in T.

Semantics of Structured Specifications

- Every structured theory is supposed to abbreviate an unstructured theory. We can flatten out these abbreviatons by defining a map $f : STH(\Sigma) \to \mathcal{P}(Sen(\Sigma))$:
-
  - $f(\Sigma, T) = T^\bullet$
  - $f(\vartheta_1 \cup \vartheta_2) = (f(\vartheta_1) \cup f(\vartheta_2))^\bullet$
  - $f(\sigma(\vartheta)) = (Sen(\sigma)(f(\vartheta)))^\bullet$
  - $f(\sigma^{-1}(\vartheta)) = Sen(\sigma)^{-1}(f(\vartheta))$
- Remark: $Sen(\sigma)^{-1}(T)$ is closed if T is closed.

Structured Specifications and Theory Morphisms

- Let $\vartheta \in STH(\Sigma)$, $\vartheta' \in STH(\Sigma')$, and let $\sigma : \Sigma \to \Sigma'$ be a signature morphism.
- Then σ is a theory morphism $(\Sigma, f(\vartheta)) \to (\Sigma', f(\sigma(\vartheta)))$.
- Similarly, σ is a theory morphism $(\Sigma, f(\sigma^{-1}(\vartheta'))) \to (\Sigma', f(\vartheta'))$.

Example

- Let $\vartheta^M = (\Sigma^M, T^M)$ be the theory of monoids, and similarly for the other example theories from Slide 39.
- Let $\vartheta^i(\Sigma^G, T^i)$ be the theory containing only the group axiom for the inverse element.
- Then ϑ^G can be written in a structured way as $\sigma^{MG}(\vartheta^M) \cup \vartheta^i$.
- (The closure of) ϑ^G can be obtained as $(\sigma^{GR_*} \bullet \sigma^{RR_*})^{-1}(\vartheta^R)$.

Motivation: Development Graphs

- Structured theories are somewhat inconvenient.
- Tool support is easier if the structure is more explicit.
- Structured specifications do not handle difficult theory morphisms, only those that exist by construction.

Development Graphs

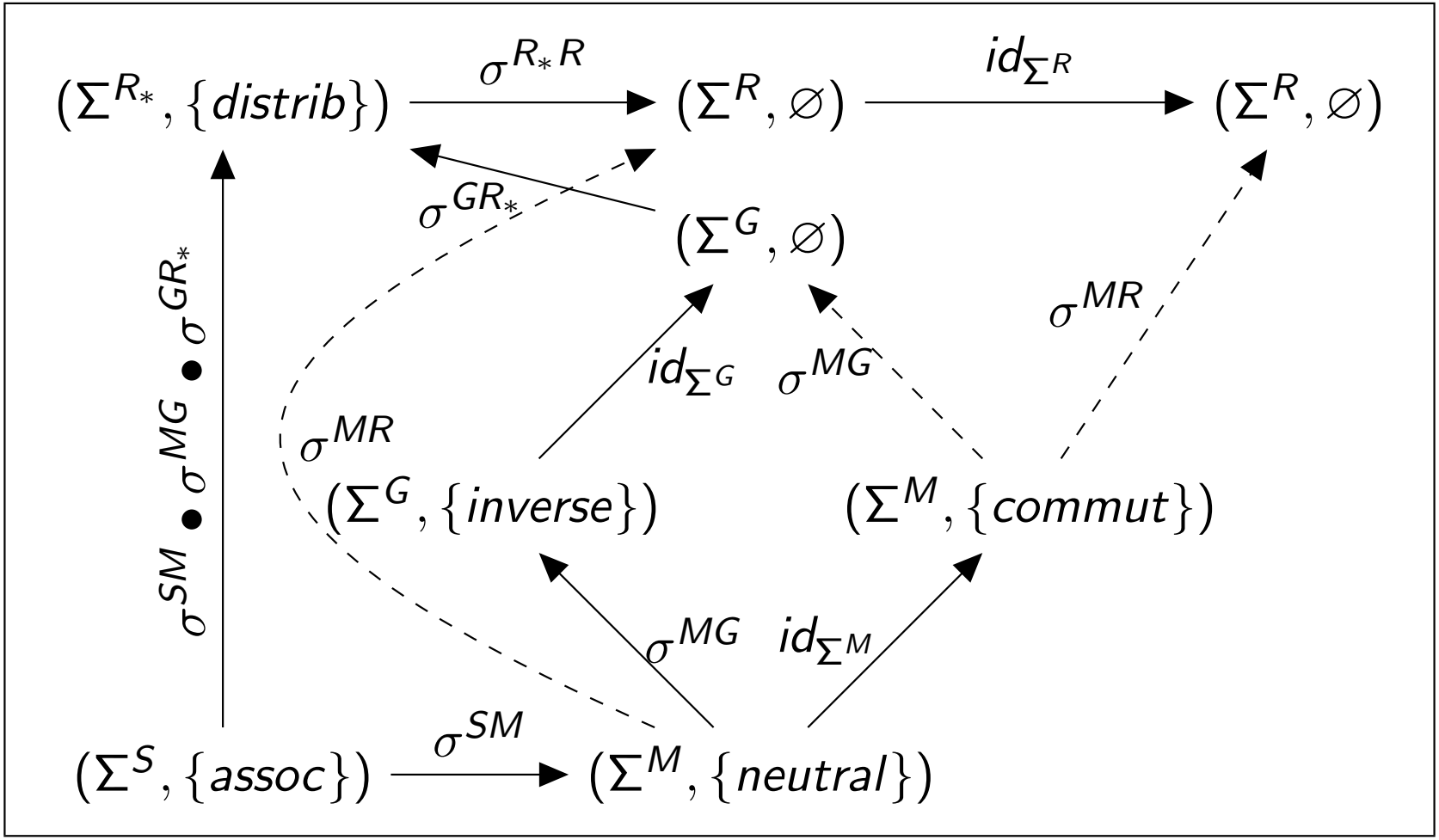
- A development graph is an acyclic graph that has
  - as nodes, theories,
  - as edges (links) from (Σ, T) to (Σ', T'): signature morphisms $\Sigma \to \Sigma'$ along with a type.
  - A type of a link is both one of definitional/theorem and one of local/global.
- Definitional links: theory morphisms by construction
- Theorem links: non-trivial theory morphisms

Flattening a Development Graph

- Intuitively, the theories in the graph are not theories, but partial theories.
- Flattening turns every node into the intended theory and every definitional link into a theory morphism.
- Flattening is defined incuctively
  - Assume a node (Σ, T) with incoming local definitional edges σ_i from (Σ_i, T_i) and incoming global definitional edges σ'_i from (Σ'_i, T'_i).
  - Then $f(\Sigma, T) = T \cup \bigcup_i Sen(\sigma_i)(T_i) \cup \bigcup_i Sen(\sigma'_i)(f(T'_i))$.
  - Well-founded due to acyclicity.

Example: Development Graph

Recall Slide 39. Dashed arrows are local. All arrows are definitional.



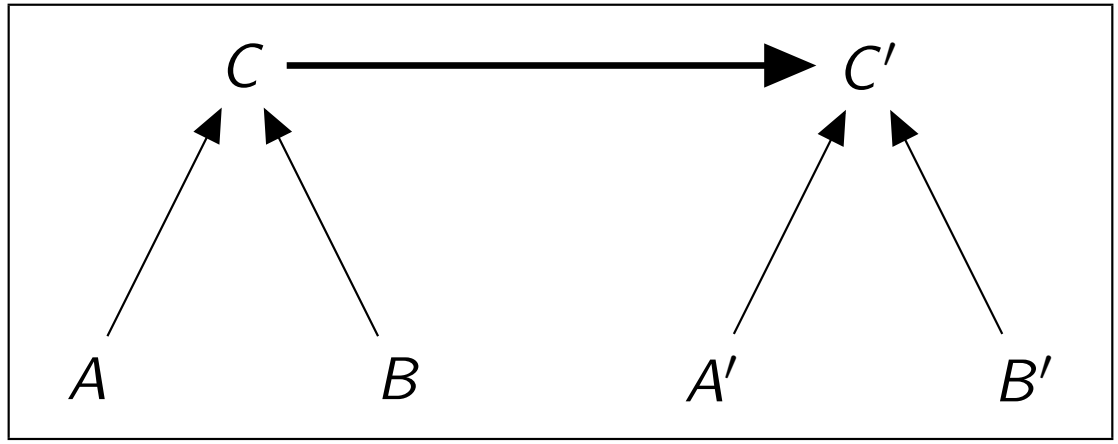
Decomposing Theorem Links

- Theorem links $\sigma : (\Sigma, T) \longrightarrow (\Sigma', T')$ must be proved to be theory morphisms.
- If σ is local, all formulas in T must be derived from $f(\Sigma', T')$.
- If σ is global, all formulas in $f(\Sigma, T)$ must be derived from $f(\Sigma', T')$.
- Global theorem links can be decomposed into local ones.
- Thus, the development structure can be used to discharge proof obligations.

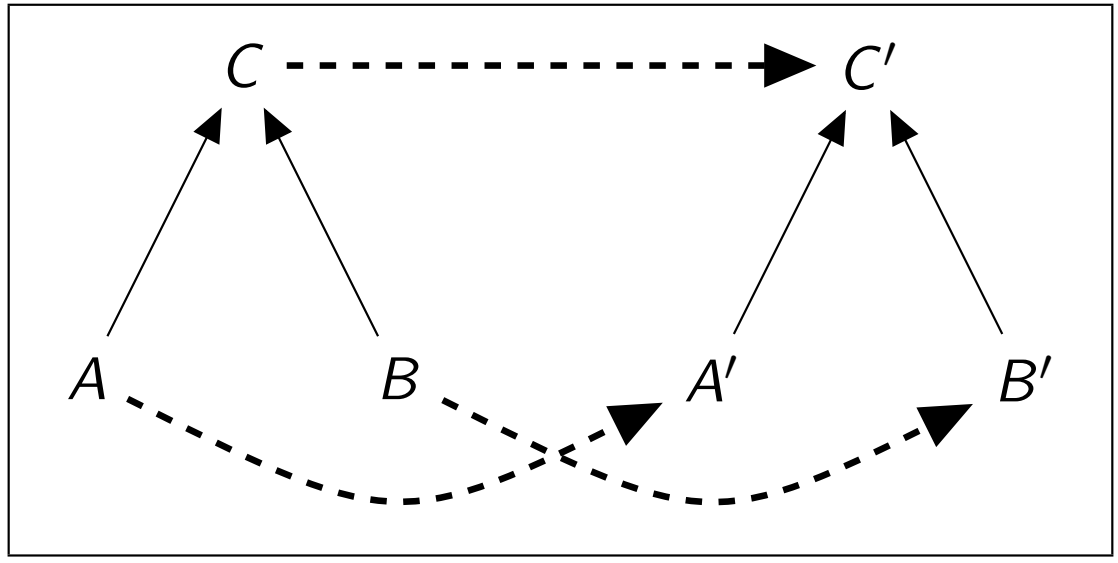
Example: Decomposing Theorem Links

Theorem links are thick.

A theorem link:



Its decomposition (The diagram must commute.):



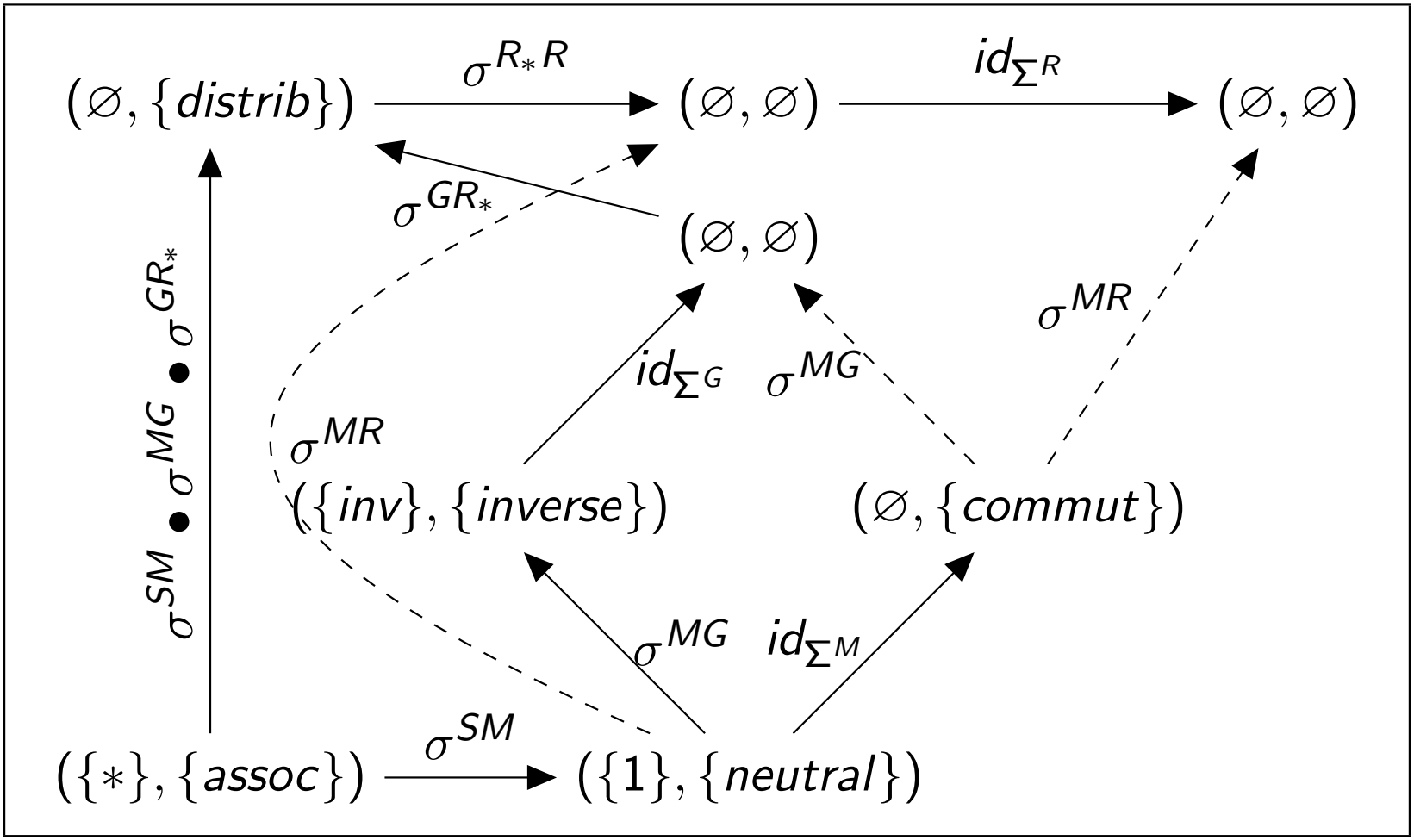
Heterogenous Development Graphs

- So far: nodes and links live in the category Th^I for a fixed institution I
- Generalization: Permit different institutions in the same graph
- Make the collection of institutions a category by defining institution translations.
- Then: Nodes are triples (I, Σ, T) of institution and theory; links from (I, Σ, T) to (I', Σ', T') are pairs (μ, σ) for an institution translation $\mu : I \longrightarrow I'$ and a theory morphism $\sigma : \mu(\Sigma) \longrightarrow \Sigma'$ in I'

Theory Graphs

- So far: modular development of the set of axioms
- Obvious extension: develop signatures by using modules, too
- Thus: Nodes (Σ, T) where Σ is a partial signature
- The complete signature of a node is obtained by flattening

Example: Theory Graphs



Theory Graphs: Problems

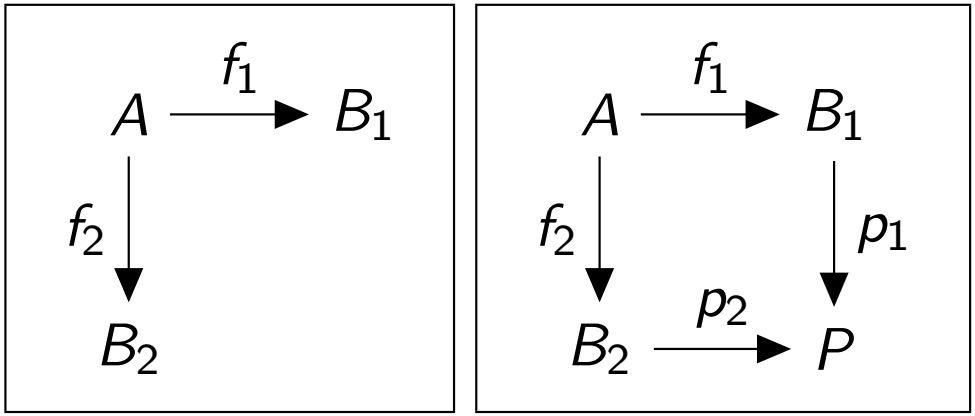
- How to define edges if we need flattening to find out what the signatures are?
- What happens if symbols are imported several times?
- Current work

Theory Graphs: Systems and Standards

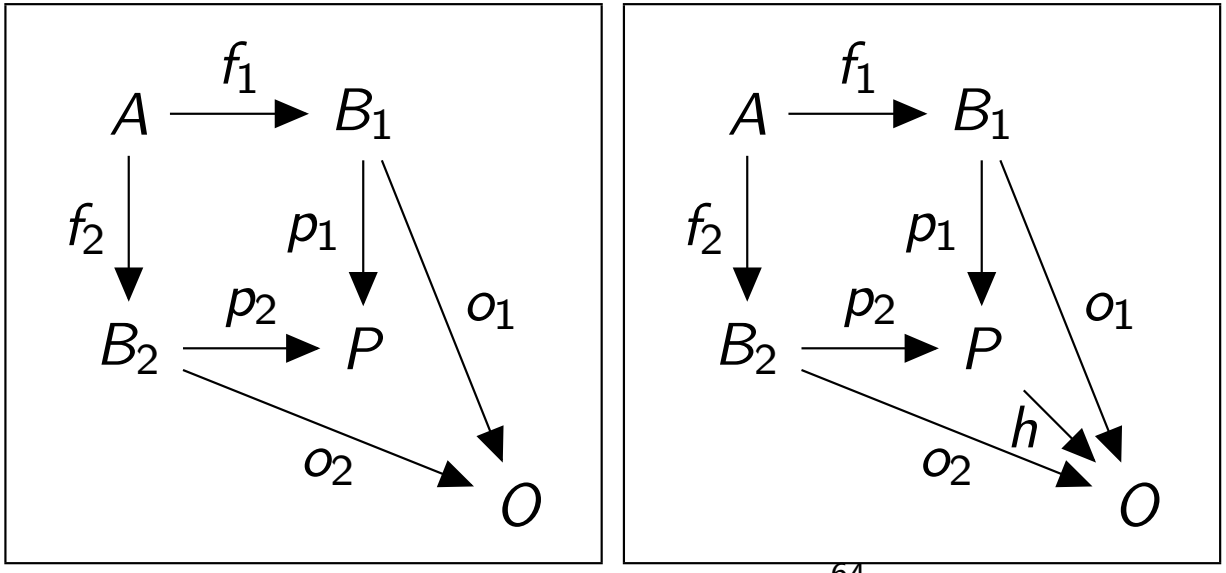
- OBJ: J. Gogues and others; `http://www.cs.ucsd.edu/users/goguen/sys/obj.html`; 1970s; influential specification language
- IMPS: W. Farmer, J. Guttman, F. Thayer; 1990; `http://imps.mcmaster.ca/`; system using theory morphisms for theorem reuse
- Maya: D. Autexier, D. Hutter, T. Mossakowski, A. Schairer; 2000 (?); `http://www-ags.dfki.uni-sb.de/~inka/maya.html`; first implementation of develoment graphs, uses $FOL^{=}$, offers management of change
- Hets: T. Mossakowski and others; 2004 (?); `http://www.informatik.uni-bremen.de/agbkb/forschung/formal_methods/CoFI/hets/index_e.htm`; extends development graphs to the heterogeneous case
- OMDoc: M. Kohlhase; 2002 (?); `http://www.omdoc.org/`; XML-based semi-formal specification language

Pushouts

Given the left diagram in a category $\mathcal{C}$. (P, p_1, p_2) is called a pushout of (f_1, f_2) if the right diagram commutes

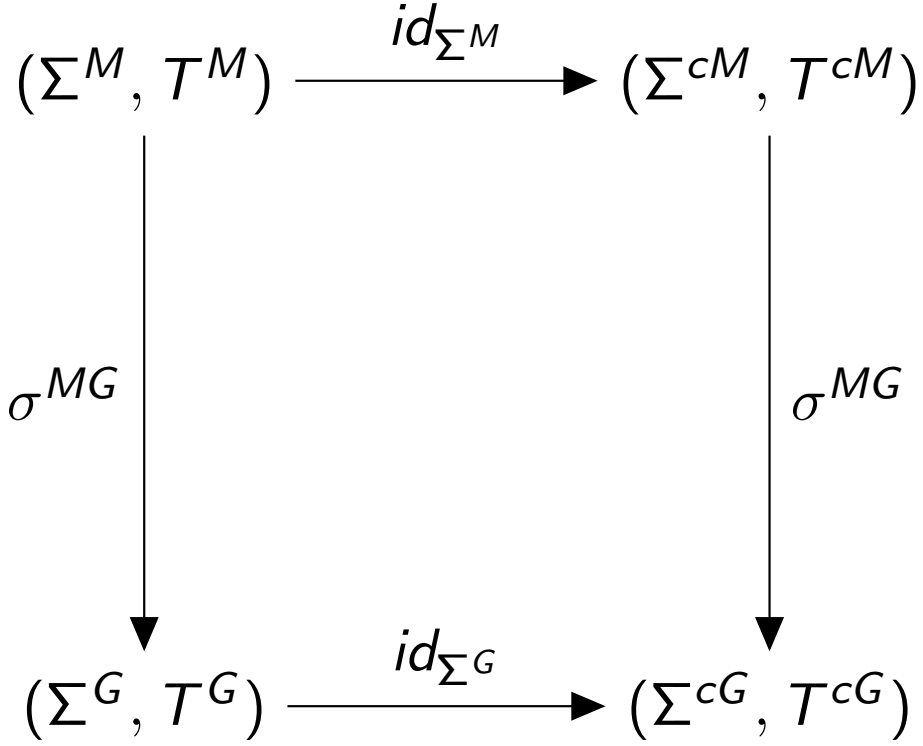


and if for every commuting (O, o_1, o_2) as in the left diagram, there is a unique h such that the right diagram commutes.



Pushouts: Intuition

- Pushouts generalize the concept of union with sharing
- P is the union of B_1 and B_2 with shared structure A
- There are a several pushouts in the running example, e.g.,



Pushouts: Lemmas

- So far, we have only defined flattening for $FOL^{=}$.
- Pushouts are crucial for the structured theory development because we can use them to define in general what flattening means.
- Lemma: Th has pushouts for all (f_1, f_2) if Sig does.
- Lemma: $Sig^{FOL^{=}}$ has pushouts.