

# A LOGIC OF ABSTRACTION

PROGRAMme spring workshop on machines, june 6-7, Lille.

Raymond Turner

June 11, 2019

# Abstraction

- ▶ Abstraction provides one of the central tools in mathematics and computer science.
- ▶ It moves us to new conceptual platforms where detail is suppressed or hidden.
- ▶ It enables us to attack and solve problems at the appropriate conceptual level.
- ▶ But what is abstraction?

## Objective

- ▶ But what is abstraction?
- ▶ In this talk we attempt to answer this question by developing a [Logic of Abstraction](#).
- ▶ In particular, we are concerned with uncovering the logical system that emerges from reflection on informal instances of computational and mathematical practice.

## Longer Term Goals

- ▶ To use the logic to provide an analysis and formalization (case studies) of abstraction in:
  - ▶ Mathematics - particularly constructive mathematics
  - ▶ Theoretical Computer Science.

# 1. THE CLASSICAL WAY OF ABSTRACTION

## Singular Terms

- ▶ Frege observes that many of the singular terms that appear to refer to abstract entities are formed by means of functional expressions.
- ▶ While many singular terms formed by means of functional expressions denote ordinary concrete objects: 'the father of Plato', 'the capital of France',...
- ▶ The functional terms that pick out *abstract entities* are distinctive.

## Abstract Objects

► *Abstract entities* are picked out by 'equations' of the following form.

*The direction of line A = The direction of line B iff A is parallel to B.*

*The set of F's = the set of G's iff for all x, x is F iff x is G.*

## Wright Abstraction Principle

Where ' $f(a)$ ' is such an expression, there is typically an equation of the form:

$$f(a) = f(b)$$

if and only if  $R(a, b)$  where  $R$  is an equivalence relation.



## Philosophical Basis

- ▶ The equivalence relation is given ahead of time.
- ▶ It is epistemologically and ontologically prior to the new term and notion of equality.
- ▶ The relation  $R$  supplies the semantic underpinnings of the new equality relation.

# Russell

- ▶ According to informal set theory, the functional expression 'set of' is indeed characterized by a putative abstraction principle.

The set of Fs = the set of Gs if and only if, for all  $x$ ,  $x$  is F if and only if  $x$  is G.

- ▶ But instances of this principle lead to Russell's paradox.

## Three Principles

One must be given up to avoid Russell.

$$(Comp) \quad \exists F . \forall x . F(x) \leftrightarrow \phi[x]$$

$$(Ext) \quad \forall F . \forall G . \hat{x}.F(x) = \hat{x}.G(x) \leftrightarrow \forall x . F(x) \leftrightarrow G(x)$$

$$(Object) \quad \forall F . \exists u . u = \hat{x}.F(x)$$

## Abstractionism

- ▶ Abstractionists give up *Ext* but explore other principles of abstraction to provide a foundation for classical mathematics.
- ▶ They begin with Hume's principle: a principle of abstraction that insists that two concepts have the same number if they can be put into one-to-one correspondence with each other.
- ▶ This facilitates the development of Peano arithmetic.
- ▶ But the program aims to abstract Zermelo-Fraenkel set theory, and beyond.
- ▶ Although, much work has been invested, and considerable progress has been made, and work is still continuing, the overall enterprise has not been a complete success.

## 2. KINDS and ABSTRACTION

## A New Kind of Thing

► Implicitly, the abstractionist views abstraction principles as introducing a new kind of thing.

*1. The direction of a line A = The direction of line B iff A is parallel to B.*

*2. The set of Fs = the set of Gs iff for all x, x is F if and only if x is G.*

► Kinds (or types) of thing are introduced as a result of abstraction.

► *Directions* are brought into existence in 1.

► *Sets* are brought into existence in 2.

## The Axis of Abstraction.

- ▶ Moreover, these abstractions not only generate new types of things, but also employ a type of thing on which the abstraction carves out the ones that are equivalent. *The axis of abstraction.*
- ▶ Lines are input
- ▶ Concepts are input

# Types

- ▶ In mathematical context, it seems more appropriate to use the word *type* instead of *kind*.
- ▶ This is also a better fit with our treatment of types characterized by abstraction.
- ▶ Specifically, it better fits the way their elements and their notions of equality are specified.



## Une Nouvelle Manière D'abstraction

The way of abstraction now takes the following shape.

$$\forall x : T. \forall y : T. f(x) =_{T/R} f(y) \leftrightarrow R(x, y)$$

where  $T$  is an existing type,  $R$  is a relation on the given type  $T$ ,  $T/R$  is the new type formed by elements of the form  $f(t)$  where  $t : T$ , and  $=_{T/R}$  the equality for the new type.

## The Julius Caesar Problem

- ▶ Notice that the abstraction principle only informs us about equality on the new type; it does not say anything outside of that domain.
- ▶ One response would have it that outside of the domain of directions any assertion of equality is **false**.
- ▶ Alternatively, we could insist that assertions of equality outside of the domain are **meaningless**.

### 3. A LOGIC OF ABSTRACTION

## Types and Their Elements

- ▶ Abstraction brings new types and type constructors into being. Consequently, we require a logical framework in which new types can be introduced as part of the deductive framework.
- ▶ We require judgements that allows us to *introduce new types* and their members.

$T$  type     $t : T$

- ▶ These are two of the *grammatical* judgements of the logic - they tell us what kinds of things there are, and provide their syntactic forms.

## Propositions

- ▶ Equality statements are restricted to objects of the same type. Consequently, the general notion of being a proposition is type dependent.
- ▶ Consequently, we require a further grammatical judgement to the effect that something is a proposition.

$\phi \text{ prop}$

- ▶ These three are the **grammatical judgements** of the system. The grammatical system replaces the standard context-free grammar that usually determines the syntax of logical systems.

# Truth

The final judgment is the standard logical one, namely the assertion that a proposition is true.

$\phi$

## Sequents

- ▶ Sequents in the theory have the following shape

$$\Gamma \vdash \Theta$$

where  $\Theta$  is one of our four judgment forms and  $\Gamma$  a context, where a context consists of propositional assumptions ( $\phi$ ) or type assignments ( $x : T$ ).

- ▶ Rules in the system will take the following form.

$$\frac{\Gamma_1 \vdash \Theta_1, \dots, \Gamma_m \vdash \Theta_m}{\Gamma \vdash \Theta}$$

## Equality

$$\frac{\Gamma \vdash t : T \quad \Gamma \vdash s : T}{\Gamma \vdash t =_T s \text{ prop}} \qquad \frac{\Gamma \vdash t : T}{\Gamma \vdash t =_T t}$$

$$\frac{\Gamma \vdash t =_T s}{\Gamma \vdash s =_T t} \qquad \frac{\Gamma \vdash t =_T s \quad \Gamma \vdash s =_T r}{\Gamma \vdash t =_T r}$$



## Structural Rules

$$\frac{\Gamma \vdash \phi \text{ prop}}{\Gamma, \phi \vdash \phi}$$

$$\frac{\Gamma \vdash T \text{ type}}{\Gamma, x : T \vdash x : T}$$

$$\frac{\Gamma, \Delta \vdash \Theta \quad \Gamma \vdash \phi \text{ prop}}{\Gamma, \phi, \Delta \vdash \Theta}$$

$$\frac{\Gamma, \Delta \vdash \Theta \quad \Gamma \vdash T \text{ type}}{\Gamma, x : T, \Delta \vdash \Theta}$$

$$\frac{\Gamma, y : S, \Delta \vdash \Theta \quad \Gamma \vdash s : S}{\Gamma, \Delta[s/y] \vdash \Theta[s/y]}$$

## Propositional Rules

$$\frac{\Gamma \vdash \phi \text{ prop} \quad \Gamma \vdash \psi \text{ prop}}{\Gamma \vdash \phi \wedge \psi \text{ prop}}$$

$$\frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}$$

$$\frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi}$$

$$\frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}$$

$$\frac{\Gamma \vdash \phi \text{ prop} \quad \Gamma \vdash \psi \text{ prop}}{\Gamma \vdash \phi \vee \psi \text{ prop}}$$

$$\frac{\Gamma \vdash \phi \vee \psi \quad \Gamma, \phi \vdash \eta \quad \Gamma, \psi \vdash \eta}{\Gamma \vdash \eta}$$

$$\frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi \text{ prop}}{\Gamma \vdash \phi \vee \psi}$$

$$\frac{\Gamma \vdash \psi \quad \Gamma \vdash \phi \text{ prop}}{\Gamma \vdash \phi \vee \psi}$$

$$\Gamma \vdash \Omega \text{ prop} \quad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \neg \phi}{\Gamma \vdash \Omega} \quad \frac{\Gamma \vdash \phi \text{ prop} \quad \Gamma \vdash \Omega}{\Gamma \vdash \phi}$$

$$\frac{\Gamma \vdash \phi \text{ prop} \quad \Gamma \vdash \psi \text{ prop}}{\Gamma \vdash \phi \rightarrow \psi \text{ prop}} \quad \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \rightarrow \psi} \quad \frac{\Gamma \vdash \phi \rightarrow \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}$$

$$\frac{\Gamma \vdash \phi \text{ prop}}{\Gamma \vdash \neg \phi \text{ prop}} \quad \frac{\Gamma, \phi \vdash \Omega}{\Gamma \vdash \neg \phi}$$

## Quantifiers

$$\frac{\Gamma, x : T \vdash \phi \text{ prop}}{\Gamma \vdash \exists x : T \cdot \phi \text{ prop}} \qquad \frac{\Gamma, x : T \vdash \phi \text{ prop}}{\Gamma \vdash \forall x : T \cdot \phi \text{ prop}}$$

$$\frac{\Gamma \vdash \phi[t/x] \qquad \Gamma \vdash t : T}{\Gamma \vdash \exists x : T \cdot \phi}$$

$$\frac{\Gamma \vdash \exists x : T \cdot \phi \qquad \Gamma, x : T, \phi \vdash \eta}{\Gamma \vdash \eta}$$

$$\frac{\Gamma, x : T \vdash \phi}{\Gamma \vdash \forall x : T \cdot \phi} \qquad \frac{\Gamma \vdash \forall x : T \cdot \phi \qquad \Gamma \vdash t : T}{\Gamma \vdash \phi[t/x]}$$

## The Logic

- ▶ We shall write

$$\Gamma \vdash_{LOA} \Theta$$

if the sequent  $\Gamma \vdash \Theta$  is derivable using the rules of *LOA*.

- ▶ The grammatical system *G* is that subsystem of *LOA* whose contexts are restricted to type assignments and whose rules are those with premises and conclusions of the *T* type,  $t : T$  and  $\phi$  *prop*. We shall write

$$\Gamma \vdash_G \Theta$$

If this follows from rules whose premises, contexts and conclusions are grammatical.

## Logic and Grammar

- ▶ **Theorem** The Logic is conservative over the grammar for grammatical judgements
- ▶ **Theorem** If  $\Gamma \vdash_{\mathbf{LOA}} \phi$  then  $\Gamma \vdash_{\mathbf{G}} \phi$  *prop*

## Summary

- ▶ Different to predicate logic in that the grammar is given by rules rather than context-free grammar
- ▶ If a grammatical judgement is provable in the logic, it is provable in the grammar
- ▶ Only grammatical propositions are provable

## Relations and Functions in LOA

- ▶ In parallel with our treatment of equality, relations and functions must declare their types.
- ▶ More explicitly, relations must be given relative to their types by grammatical assumptions of the following form.

$$w_1 : T_1, \dots, w_n : T_n \vdash R(w_1, \dots, w_n) \text{ prop}$$

- ▶ *Where*  $R$  is binary, with  $A_1 = A_2 = A$ , then we shall refer to  $R$  as a binary relation of type  $A$ .

## Defined Relations

- ▶ As usual, new relations and function maybe conservatively added by definition.
- ▶ But again, we have to ensure the type discipline is preserved. More explicitly, where

$$w_1 : T_1, \dots, w_n : T_n \vdash \phi[w_1, \dots, w_n] \text{ prop}$$

we may may introduce a new relation  $R$  by the following implicit definition.

$$R \doteq [x_1 : A_1, \dots, x_n : A_n | \phi[x_1, \dots, x_n]].$$

- ▶ This is taken to introduce a new relation  $R$  of type  $A_1, \dots, A_n$  that satisfies the following axiomatic demand.

$$\forall x_1 : A_1, \dots, \forall x_n : A_n. R(x_1, \dots, x_n) \leftrightarrow \phi[x_1, \dots, x_n].$$



## 4. ABSTRACTION IN LOA

## Dynamic Abstraction

- ▶ With this machinery in place we are able to give a formal account of abstraction inside *LOA*.
- ▶ On the assumption that  $R$  is an equivalence relation of type  $T$ , we may introduce a new type via the following abstraction principle.

$$(Abstraction) \quad \forall x : T. \forall y : T. f(x) =_{T/R} f(y) \leftrightarrow R(x, y)$$

- ▶ This expresses the logical content of abstraction.
- ▶ However, to **ensure that this is a proposition**, we require some grammatical rules.

## Grammar Rules

- ▶ The first ensures that  $T/R$  is a type of the extended theory.

$$(Type) \quad \frac{T \text{ type}}{T/R \text{ type}}$$

- ▶ The next rule explicitly provides the membership of the new type.

$$(Mem) \quad \frac{t : T}{f(t) : T/R}$$

- ▶ These two *grammatical rules* inform us that  $T/R$  is the new type that has members of the form  $f(t)$  where  $t : T$ .

## Types in Constructive Mathematics

- ▶ This fits the form of types in constructive mathematics exactly.

“A set is not an entity which has an ideal existence: a set exists only when it has been defined. To define a set we prescribe, at least implicitly, what we (the constructing intelligence) must do in order to construct an element of the set, and what we must do to show that two elements are equal” [Bishop]

- ▶ This is precisely what the rules of abstraction do.

## Completeness

- ▶ Intuitively, directions are precisely the things that arise by abstraction from lines but the abstraction principle does not guarantee this.
- ▶ Generally, to complete the picture we must ensure that the only objects of type  $T/R$  are of the form  $f(t)$  for some  $t : T$ .

$$(Com) \quad \forall x : T/R . \exists y : T . x =_{T/R} f(y)$$

## Congruence

- ▶ Assume that

$$x : A \vdash S(x) \text{ prop}$$

- ▶ Assume that  $R$  is an equivalence relation on  $A$ . Then  $S$  is a *Congruence Relative to  $R$*  iff the following is provable.

$$\forall x : A. \forall y : A. R(x, y) \rightarrow (S(x) \rightarrow S(y))$$

- ▶ Where  $S$  is a congruence relative to  $R$  we may safely extend  $S$  to the abstracted type.

$$[S] = [y : A/R. \exists x : A. f(x) =_{A/R} y \wedge S(x)]$$

## Abstraction is Conservative

### **Theorem** (Conservative Extension)

*If  $\Gamma \vdash \Theta$  in the abstracted theory and  $\Gamma \vdash \Theta$  is a judgement expressible in the source theory, then  $\Gamma \vdash \Theta$  is derivable in the source theory.*

## Scheme of Type Introduction

- ▶ Not a simple type constructor.
- ▶ Introduces a new scheme for the introduction of new types and type constructors.
- ▶ This is its foundational role.



## 5. INCIDENCE GEOMETRY

## A Fragment of Geometry

- ▶ To formulate matters we require a background theory - a fragment of geometry that concerns points and lines.
- ▶ We require two types: lines and points.

*L type*      *P type*

- ▶ These are linked by the following relation that states that a point  $p$  is on a line.

$$\frac{l : L \quad p : P}{\text{On}(p, l) \text{ prop}}$$

## Logical Axioms.

- ▶ Given any two distinct points, there is precisely one line that contains both of them.

$$\forall p : P. \forall q : P. p \neq q \rightarrow \exists ! l : L. On(p, l) \wedge On(q, l)$$

- ▶ Given any line, there are at least two distinct points that lie on it.

$$\forall l : L. \exists p : P. \exists q : P. p \neq q \wedge On(p, l) \wedge On(q, l)$$

## Parallel Postulate

- ▶ Two parallel non equal lines have no points in common.

$$\parallel \doteq [l : L, k : L \mid l = k \vee \sim \exists p : P . On(p, l) \wedge On(p, k)]$$

- ▶ The parallel postulate then asserts that given any line and point not on it, there is a unique line through that point which is parallel to it.

$$\forall l : L . \forall p : P . \neg On(p, l) \rightarrow \exists ! k : L . On(p, k) \wedge k \parallel l$$

## Abstracting Directions

- ▶ We are able to form the type of directions via the following abstraction principle.

$$\textit{Abstraction} \quad \forall x : L. \forall y : L. d(x) =_D d(y) \leftrightarrow x \parallel y$$

- ▶ This introduces a new type given by the following grammar rules.

$$(\textit{Type}) \quad \frac{L \textit{ type}}{D \textit{ type}}$$

$$(\textit{Mem}) \quad \frac{l : L}{d(l) : D}$$

where we have written  $L/R$  as  $D$ . We are able to extend the relations of being parallel to directions.

$$\llbracket \rrbracket \doteq [x : D, y : D \mid \exists u : L. \exists v : L. u \parallel v \wedge d(u) =_D x \wedge d(v) =_D y]$$

## 6. SETS FROM CONCEPTS

## Concepts and Sets

- ▶ Our second case study concerns concepts and the abstraction of a theory of sets.

*The set of Fs = the set of Gs iff for all x, x is F if and only if x is G.*

- ▶ Our intention is to derive a theory of sets from a theory of concepts by abstraction.

## Theory of Concepts

- ▶ The notion of concept satisfies the following grammatical introduction rule for concepts.

$$\frac{T \text{ type}}{\text{Concept}(T) \text{ type}}$$

- ▶ There is one basic operation on concepts namely that of predication whereby a concept of type  $T$  may be applied to an object of type  $T$  to form a proposition.

$$\frac{a : T \quad f : \text{Concept}(T)}{\text{pred}(f, a) \text{ prop}}$$



## Comprehension

- Concepts are constituted by a simple axiom scheme of comprehension. Given a formula  $\phi[x]$  such that

$$x : T \vdash \phi \text{ prop}$$

comprehension guarantees the following.

$$\exists z : \text{Concept}(T). \forall x : T. \text{pred}(z, x) \leftrightarrow \phi[x]$$

- This is the basic theory of concepts. Equality of concepts is taken as primitive.

## Extensional Equivalence of Concepts

However, we may define the extensional equivalence of concepts as follows.

$$\equiv \doteq [f : \text{Concept}(T), g : \text{Concept}(T). \forall u : T. \text{pred}(f, u) \leftrightarrow \text{pred}(g, u)]$$

This is an equivalence relation over concepts.

## Abstracting Sets

- ▶ We abstract a notion of *Set* as follows where we write  $\text{Concept}(T)_{/\equiv}$  as  $\text{Set}(T)$ .

$$\forall f : \text{Concept}(T). \forall g : \text{Concept}(T). \text{set}(f) =_{\text{Set}(T)} \text{set}(g) \leftrightarrow f \equiv g$$

- ▶ This is taken to introduce a new type whose grammatical conditions are given by the following two rules:

$$\frac{\text{Concept}(T) \text{ type}}{\text{Set}(T) \text{ type}} \quad \frac{p : \text{Concept}(T)}{\text{set}(p) : \text{Set}(T)}$$

where the logical content of the theory is given by the above principle of abstraction.

## Membership

- ▶ We may then define the membership relation for sets in terms of predication.

$$\in = [x : T, y : \text{Set}(T) . \exists v : \text{Concept}(T) . \text{set}(v) =_{\text{Set}(T)} y \wedge \text{pred}(v, x)]$$

- ▶ Immediately, we have:

$$\frac{a : T \quad b : \text{Set}[T]}{a \in b \text{ prop}}$$

- ▶ Finally, we define extensional equivalence for sets.

$$\equiv = [x : \text{Set}(T), y : \text{Set}(T) . \forall u : T . u \in x \leftrightarrow u \in y]$$

## Separation

► Using completeness, we may now kick away the ladder of concepts to obtain a pure theory of sets that is governed by the following rules.

$$\frac{T \text{ type}}{\text{Set}(T) \text{ type}}$$

$$(Ext) \quad \forall x : \text{Set}(T). \forall y : \text{Set}(T). x =_{\text{Set}(T)} y \leftrightarrow \forall u : T. u \in x \leftrightarrow u \in y$$

$$(Comp) \quad \exists z : \text{Set}(T). \forall x : T. x \in z \leftrightarrow \phi[x]$$

where  $\phi$  is any proposition excluding the propositions of the concept theory. This makes no reference to the concept theory.

## 7. THE PHILOSOPHY OF ABSTRACTION

## Axiomatic Abstraction

- ▶ Given a type theory  $\Pi$  we may extend the theory by the addition of a new type constructor given by the following rules.

$$(Type) \quad \frac{T \text{ type}}{New(T) \text{ type}}$$

$$(Mem) \quad \frac{t : T}{f(t) : New(T)}$$

$$(Com) \quad \forall x : New(T). \exists y : T. x =_{New(T)} f(y)$$

- ▶ This theory  $New(\Pi)$  is the bare bones of the theory generated by abstraction. What is missing is the principle of abstraction.

## Abstraction

- ▶ However, this can be supplied by the standard relationship between functions and equivalence relations.

$$R \doteq [x : T, y : T | f(x) =_{New(T)} f(y)]$$

- ▶ This encodes the principle of abstraction that follows by definition of  $R$ .

$$\forall x : T. \forall y : T. R(x, y) \leftrightarrow f(x) =_{New(T)} f(y)$$

- ▶ So, formally we have all the ingredients of an abstraction theory. However, it has little substance.



## Impredicative Theory

- ▶ Let  $R$  be any equivalence relation of  $New(\mathbb{I})$ . Then we may impose a further axiomatic constraint on the given notion of equality via the principle of abstraction.

$$\forall x : T. \forall y : T. f(x) =_{New(T)} f(y) \leftrightarrow R(x, y)$$

- ▶ But this is now (possibly) impredicative.
- ▶ Even when the equivalence relation is not defined in terms of the  $New$  constructs, it is fundamentally different.
- ▶ It is an axiomatic theory: the **abstraction statement is a new axiom** imposed upon an notion of equality for an existing type.

## Semantic, Ontological and Epistemological Priority.

- ▶ In the original account the equivalence relation is given ahead of time and is stated within the source theory.
- ▶ It is epistemologically and ontologically prior to the notion of equality for the new type.
- ▶ The relation  $R$  supplies the semantic underpinnings of the equality relation for the new type.
- ▶ None of this applies to the above

## Much to do

To provide an analysis of some aspects of abstraction in:

- ▶ Constructive Mathematics.
- ▶ Theoretical Computer Science.