Implicit dynamic function introduction and its connections to the foundations of mathematics

Marcos Cramer
Mathematical Institute
University of Bonn
Endenicher Allee 60, 53115 Bonn, Germany
cramer@math.uni-bonn.de

Abstract: We discuss a feature of the natural language of mathematics – the implicit dynamic introduction of functions – that has, to our knowledge, not been captured in any formal system so far. If this feature is used without limitations, it yields a paradox analogous to Russell's paradox. Hence any formalism capturing it has to impose some limitations on it. We sketch two formalisms, both extensions of Dynamic Predicate Logic, that innovatively do capture this feature, and that differ only in the limitations they impose onto it. One of these systems is based on a novel theory of functions that interprets ZFC, and thus exhibits interesting connections to the foundations of mathematics.

Keywords: Dynamic Predicate Logic, function introduction, Ackermann set theory, function theory

1. Dynamic Predicate Logic

Dynamic Predicate Logic (DPL, Groenendijk & Stokhof, 1991) is a formalism whose syntax is identical to that of standard first-order predicate logic (PL), but whose semantics is defined in such a way that the dynamic nature of natural language quantification is captured in the formalism:

1. If a farmer owns a donkey, he beats it.
2. PL: $\forall x \forall y\ (farmer(x) \wedge donkey(y) \wedge owns(x,y) \rightarrow beats(x,y))$
3. DPL: $\exists x\ (farmer(x) \wedge \exists y\ (donkey(y) \wedge owns(x,y))) \rightarrow beats(x,y)$

In PL, 3 is not a sentence, since the rightmost occurences of x and y are free. In DPL, a variable may be bound by a quantifier even if it is outside its scope. The semantics is defined in such a way that 3 is equivalent to 2. So in DPL, 3 captures the meaning of 1 while being more faithful to its syntax than 2.

1.1. DPL semantics

We present DPL semantics in a way slightly different but logically equivalent to its definition in Groenendijk and Stokhof (1991). Structures and assignments are defined as for PL: A structure S specifies a domain $|S|$ and an interpretation a^S for every constant, function or relation symbol a in the language. An S-assignment is a function from variables to $|S|$. G_S is the set of S-assignments. Given two assignments g, h, we define $g[x]h$ to mean that g differs from h at most in what it assigns to the variable x. Given a DPL term t, we recursively define

$$[t]^g_S = \begin{cases} g(t) & \text{if } t \text{ is a variable,} \\ t^S & \text{if } t \text{ is a constant symbol,} \\ f^S([t_1]^g_S, \dots, [t_n]^g_S) & \text{if } t \text{ is of the form } f(t_1, \dots, t_n). \end{cases}$$

Groenendijk and Stokhof (1991) define an interpretation function $[\![\bullet]\!]_S$ from DPL formulae to subsets of $G_S \times G_S$. We instead recursively define for every $g \in G_S$ an interpretation function $[\![\bullet]\!]^g_S$ from DPL formulae to subsets of G_S:[1]

[1]This can be viewed as a different currying of the uncurried version of Groenendijk and Stokhof's interpretation function.

1. $[\![\top]\!]^g_S := \{g\}$
2. $[\![t_1 = t_2]\!]^g_S := \{h | h = g \text{ and } [t_1]^g_S = [t_2]^g_S\}$[2]
3. $[\![R(t_1, \ldots, t_2)]\!]^g_S := \{h | h = g \text{ and } ([t_1]^g_S, \ldots, [t_2]^g_S) \in R^S\}$
4. $[\![\neg\varphi]\!]^g_S := \{h | h = g \text{ and there is no } k \in [\![\varphi]\!]^h_S\}$
5. $[\![\varphi \wedge \psi]\!]^g_S := \{h | \text{there is a } k \text{ s.t. } k \in [\![\varphi]\!]^g_S \text{ and } h \in [\![\psi]\!]^k_S\}$
6. $[\![\varphi \to \psi]\!]^g_S := \{h | h = g \text{ and for all } k \text{ s.t. } k \in [\![\varphi]\!]^h_S, \text{ there is a } j \text{ s.t. } j \in [\![\psi]\!]^k_S\}$
7. $[\![\exists x\varphi]\!]^g_S := \{h | \text{there is a } k \text{ s.t. } k[x]g \text{ and } h \in [\![\varphi]\!]^k_S\}$

$\varphi \vee \psi$ and $\forall x\varphi$ are defined to be a shorthand for $\neg(\neg\varphi \wedge \neg\psi)$ and $\exists x\top \to \varphi$ respectively.

2. Implicit dynamic introduction of function symbols

Functions are often dynamically introduced in an implicit way in mathematical texts. For example, Trench (2003) introduces the additive inverse function on the reals as follows:

(1) For each a there is a real number $-a$ such that $a + (-a) = 0$.

Here the natural language quantification "there is a real number $-a$" *locally* (i.e. inside the scope of "For each a") introduces a new real number to the discourse. But since the choice of this real number depends on a and we are universally quantifying over a, it *globally* (i.e. outside the scope of "For each a") introduces a function "$-$" to the discourse.

The most common form of implicitly introduced functions are functions whose argument is written as a subscript, as in the following example:

(2) Since f is continuous at t, there is an open interval I_t containing t such that $|f(x) - f(t)| < 1$ if $x \in I_t \cap [a, b]$. (Trench, 2003)

If one wants to later explicitly call the implicitly introduced function a function, the standard notation with a bracketed argument is prefered:

(3) Suppose that, for each vertex v of K, there is a vertex $g(v)$ of L such that $f(st_K(v)) \subset st_L(g(v))$. Then g is a simplicial map $V(K) \to V(L)$, and $|g| \simeq f$. (Lackenby, 2008)

When no uniqueness claims are made about the object locally introduced to the discourse, implicit function introduction presupposes the existence of a choice function, i.e. presupposes the Axiom of Choice. We hypothesise that the naturalness of such implicit function introduction in mathematical texts contributes to the wide-spread feeling that the Axiom of Choice must be true.

Implicitly introduced functions generally have a restricted domain and are not defined on the whole universe of the discourse. For example in (3), g is only defined on vertices of K and not on vertices of L. Implicit function introduction can also be used to introduce multi-argument functions, but for the sake of simplicity and brevity, we restrict ourselves to unary functions in this paper.

If the implicit introduction of functions is allowed without limitations, one can derive a contradiction:

[2]The condition $h = g$ in cases 2, 3, 4 and 6 implies that the defined set is either $\emptyset$ or $\{g\}$.

(4) For every function f, there is a natural number $g(f)$ such that

$$g(f) = \begin{cases} 0 & \text{if } f \in dom(f) \text{ and } f(f) \neq 0, \\ 1 & \text{if } f \notin dom(f) \text{ or } f(f) = 0. \end{cases}$$

Then g is defined on every function, i.e. $g(g)$ is defined. But from the definition of g, $g(g) = 0$ iff $g(g) \neq 0$.

This contradiction is due to the *unrestricted function comprehension* that is implicitly assumed when allowing implicit introductions of functions without limitations. Unrestricted function comprehension could be formalised as an axiom schema as follows:

Unrestricted function comprehension
For every formula $\varphi(x, y)$, the following is an axiom:

$$\forall x\ \exists y\ \varphi(x, y) \rightarrow \exists f\ \forall x\ \varphi(x, f(x))$$

The inconsistency of unrestricted function comprehension is analogous to the inconsistency of unrestricted set comprehension, i.e. Russell's paradox.

Russell's paradox led to the abandonment of unrestricted comprehension in set theory. Two radically different approaches have been undertaken for restricting set comprehension: Russell himself restricted it through his Ramified Theory of Types, which was later simplified to Simple Type Theory (STT), mainly known via Church's formalisation in his simply typed lambda calculus (Church, 1940). On the other hand, the risk of paradoxes like Russell's paradox also contributed to the development of ZFC (Zermel-Fraenkel set theory with the Axiom of Choice), which allows for a much richer set theoretic universe than the universe of simply typed sets. Since all the axioms of ZFC apart from the Axiom of Extensionality, the Axiom of Foundation and the Axiom of Choice are special cases of comprehension, one can view ZFC as an alternative way to restrict set comprehension.

Similarly, the above paradox must lead to the abandonment of unrestricted function comprehension. The type-theoretic approach is easily adapted to functions, so we will first sketch the system that formalises this approach, *Typed Higher-Order Dynamic Predicate Logic*. For an untyped approach, there is no clear way to transfer the limitations that ZFC puts onto set comprehension to the case of function comprehension. However, there is an axiomatization of set theory called *Ackermann set theory* that is a conservative extension of ZFC. It turns out that the limitations that Ackermann set theory poses on set comprehension can be transfered to the case of function comprehension, and hence to the case of implicit dynamic function introduction.

The need to deal with implicit function introduction arose for us in the context of the *Naproche project*, a project aiming at automatic formalisation of natural language mathematics (Cramer, Fisseni, et al., 2010). It has been implemented in the Naproche system using type restrictions as in Typed Higher-Order Dynamic Predicate Logic, and we plan to implement it using the less strict restrictions of the untyped Higher-Order Dynamic Predicate Logic in a future version of the system.

3. Typed Higher-Order Dynamic Predicate Logic

In this section, we extend DPL to a system called *Typed Higher-Order Dynamic Predicate Logic* (THODPL), which formalises implicit dynamic function introduction, and also allows for explicit quantification over functions. THODPL has variables typed by the types of STT. In the below examples we use x and y as variables of the basic type i, and f as a variable of the function type $i \to i$. A complex term is built by well-typed application of a function-type variable to an already built term, e.g. $f(x)$ or $f(f(x))$.

The distinctive feature of THODPL syntax is that it allows not only variables but any well-formed terms to come after quantifiers. So (5) is a well-formed formula:

(5) $\forall x\ \exists f(x)\ R(x, f(x))$

(6) $\forall x\ \exists y\ R(x, y)$

(7) $\exists f\ (\forall x\ R(x, f(x)))$

The semantics of THODPL is to be defined in such a way that (5) has the same truth conditions as (6). But unlike (6), (5) dynamically introduces the function symbol f to the context, and hence turns out to be equivalent to (7).

We now sketch how these desired properties of the semantics can be achieved. In THODPL semantics, an assignment assigns elements of $|S|$ to variables of type i, functions from $|S|$ to $|S|$ to variables of type $i \to i$ etc. Additionally, an assignment can also assign an object (or function) to a complex term. For example, any assignment in the interpretation of $\exists f(x)\ R(x, f(x))$ has to assign some object to $f(x)$. The definition of $g[x]h$ can now naturally be extended to a definition of $g[t]h$ for terms t. The definition of $[t]^g_S$ has to be adapted in the natural way to account for function variables.

Just as in the case of DPL semantics, we recursively define an interpretation $[\![\bullet]\!]^g_S$ from DPL formulae to subsets of G_S (the cases 1-5 of the recursive definition are as before):

6. $[\![\varphi \to \psi]\!]^g_S := \{h|h$ differs from g in at most some function variables $f_1, \ldots, f_n$ (where this choice of function variables is maximal), and there is a variable x such that for all $k \in [\![\varphi]\!]^g_S$, there is an assignment $j \in [\![\psi]\!]^k_S$ such that $j(f_i(x)) = h(f_i)(k(x))$ for $1 \leq i \leq n$, and if $n > 0$ then $k[x]g$ $\}$

7. $[\![\exists t \varphi]\!]^g_S := \{h|$there is a k s.t. $k[t]g$ and $h \in [\![\varphi]\!]^k_S\}$

In order to make case 6 of the definition more comprehensible, let us consider its role in determining the semantics of (5), i.e. of $\exists x \top \to \exists f(x)\ R(x, f(x))$: $[\![\exists f(x)\ R(x, f(x))]\!]^k_S$ is the set of assignments j satisfying $R(x, f(x))$ (i.e. for which $[\![R(x, f(x))]\!]^j_S$ is non-empty) such that $j[f(x)]k$. $[\![\exists x \top]\!]^g_S$ is the set of assignments k such that $k[x]g$. So by case 6 with $n = 1$,

$[\![\exists x \top \to \exists f(x)\ R(x, f(x))]\!]^g_S = \{h|h[f]g$ and there is a variable x such that for all k such that $k[x]g$, there is an assignment j satisfying $R(x, f(x))$ such that $j[f(x)]k$ and $j(f(x)) = h(f)(k(x))$, and $k[x]g\}$

$= \{h|h[f]g$ and for all k such that $k[x]g$, there is an assignment j satisfying $R(x, f(x))$ such that $j[f(x)]k$ and $j(f(x)) = h(f)(k(x))\}$

$= \{h|h[f]g$ and for all k such that $k[x]h$, k satisfies $R(x, f(x))\}$

$= [\![\exists f\ (\forall x\ R(x, f(x)))]\!]^g_S$

The type restrictions THODPL imposes may be too strict for some applications: Mathematicians sometimes do make use of functions that do not fit into the corset of strict typing, e.g. a function defined on both real numbers and real functions. To overcome this restriction, we will introduce an untyped variant HODPL in section 6. But for this we require some foundational preliminaries.

4. Ackermann set theory

Ackermann set theory (Ackermann, 1956) postulates not only sets, but also proper classes which are not sets. The sets are distinguished from the proper classes by a unary predicate M (from the German word "'Menge"' for "'set"').

Ackermann presented a pure version of his theory without urelements, and a separate version with urelements, which we will present here. The language of Ackermann set theory contains three predicates: A binary predicate $\in$, a unary predicate M and a unary predicate U for urelements. We introduce $L(x)$ ("x is limited") as an abbreviation for $M(x) \vee U(x)$. The idea is that sets and urelements are objects of limited size, and are distinguished from the more problematic classes of unlimited size.

The axioms of Ackermann set theory with urelements are as follows:

- *Extensionality axiom:* $\forall x \forall y(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$
- *Class comprehension axiom schema:* Given a formula $F(y)$ (possibly with parameters) that does not have x among its free variables, the following is an axiom:
  $\forall y(F(y) \rightarrow L(y)) \rightarrow \exists x \forall y(y \in x \leftrightarrow F(y))$
- *Set comprehension axiom schema:* Given a formula $F(y)$ (possibly with parameters that are limited) that does not have x among its free variables and does not contain the symbol M, the following is an axiom:
  $\forall y(F(y) \rightarrow L(y)) \rightarrow \exists x(M(x) \wedge \forall y(y \in x \leftrightarrow F(y)))$
- *Elements and subsets of sets are limited:*
  $\forall x \forall y(M(y) \wedge (x \in y \vee \forall z(z \in x \rightarrow z \in y)) \rightarrow L(y))$

So unlimited set comprehension is replaced by two separate comprehension schemata, one for class comprehension and one for set comprehension. In both cases, the comprehension is restricted by the constraint that only limited objects satisfy the property that we are applying comprehension to. But for set comprehension, we have the additional constraint that the property may not be defined using the setness predicate or using a proper class as parameter. Ackermann justified this approach by appeal to a definition of "set" from Cantor's work (Ackermann, 1956).

If an Axiom of Foundation for sets is added, Ackermann set theory turns out to be – in what it says about sets – precisely equivalent to ZF (Reinhardt, 1970). But this equivalence is not a triviality: It is especially hard to establish Replacement for the sets of Ackermann set theory.

5. Ackermann-like function theory

Now we transfer the ideas of a comprehension limited in this way from set comprehension to function comprehension. For this a dichotomy similar to that between sets and classes has to be imposed on functions. We propose the terms *function* and *map* respectively for this dichotomy, and call the theory resulting from these limitations on function comprehension *Ackermann-like Function Theory* (AFT). AFT can be shown to be equiconsistent with Ackermann set theory and hence with ZFC.

The language of Ackermann-like function theory contains

- a unary predicate F for functions,
- a unary predicate U for urelements,

- a constant symbol u for undefinedness, and
- a binary function symbol a for function application.

Instead of $a(f,t)$ we usually simply write $f(t)$. We write $L(x)$ instead of $U(x) \vee F(x)$. The undefinedness constant u is needed for formalising the idea that a function is only defined for certain values and undefined for others. In this language, the unrestricted function comprehension schema would be as follows:

> Given a variable z and formulae $P(z)$ and $R(z,x)$ (possibly with parameters), the following is an axiom: $\forall z\ (P(z) \rightarrow \exists x\ R(z,x)) \rightarrow \exists f\ (\neg U(f) \wedge \forall z\ ((P(z) \rightarrow R(z,f(z))) \wedge (\neg P(z) \rightarrow f(z) = u)))$

Analogously to the case of Ackermann set theory, AFT has separate comprehension schemata for maps and functions. The restriction that is imposed on both schemata now is $\forall z\ \forall x\ (R(z,x) \rightarrow L(z) \wedge L(x))$. In the function comprehension schema, in which $F(f)$ appears among the conclusions we may draw about f, the additional restriction is that the formula $R(z,x)$ may not contain the symbol F and may not have unlimited objects as parameters.

Additionally to these comprehension schemata, AFT has

- a function extensionality axiom,
- an axiom stating that any value a function takes and any value a function is defined at is limited, and
- an axiom stating that submaps of functions are functions.

In AFT one can interpret Ackermann set theory with Foundation, and hence ZFC. Since the map and function comprehension schemata presuppose the existence of choice maps and choice functions, the Axiom of Choice naturally comes out true in these interpretations.

6. Higher-order dynamic predicate logic

Now we are ready to sketch the untyped *Higher-Order Dynamic Predicate Logic* (HODPL). The restriction we impose on implicit function introduction are those imposed by AFT. AFT gives us untyped maps, which always have a restricted domain. So instead of using types to syntactically restrict the possible arguments for a given function term, we implement a semantic restriction on function application by integrating a formal account of presuppositions into the HODPL.[3] HODPL syntax thus allows for any term to be applied to any number of arguments to form a new term.

Besides the binary "=", HODPL has two unary logical relation symbols, U for urelements and F for functions. HODPL syntax does not depend on a signature, as we do not allow for constant, function and relation symbols other than "=", U and F. These can be mimicked by variables that respectively denote a non-function, denote a normal function or denote a function that only takes two predesignated urelements ("booleans") as values.

The domain of a structure always has to be a model of AFT. The possibility of presupposition failure is implemented in HODPL semantics by making the interpretation function partial rather than total. For conveniently talking about partial functions, we use the notation *def*$(f(x))$ to abbreviate that f is defined on x.

We define the partial interpretation function $[\![\bullet]\!]_S^g \subseteq G_S \times G_S$ by specifying its domain and its values trough a simultaneous recursion (the cases 3-8 of the second part are as in THODPL):

[3] See Cramer, Kühlwein, and Schröder (2010) for an introduction to presuppositions in mathematical texts.

- Domain of $[\![\bullet]\!]^g_S$:
  1. *def*($[\![U(t)]\!]^g_S$) iff $[t]^g_S \neq u^S$.
  2. *def*($[\![F(t)]\!]^g_S$) iff $[t]^g_S \neq u^S$.
  3. *def*($[\![\top]\!]^g_S$).
  4. *def*($[\![t_1 = t_2]\!]^g_S$) iff $[t_1]^g_S \neq u^S$ and $[t_2]^g_M \neq u^S$.
  5. *def*($[\![\neg\varphi]\!]^g_S$) iff *def*($[\![\varphi]\!]^g_S$).
  6. *def*($[\![\varphi \wedge \psi]\!]^g_S$) iff *def*($[\![\varphi]\!]^g_S$) and for all $h \in [\![\varphi]\!]^g_S$, *def*($[\![\psi]\!]^h_S$).
  7. *def*($[\![\varphi \rightarrow \psi]\!]^g_S$) iff *def*($[\![\varphi]\!]^g_S$) and for all $h \in [\![\varphi]\!]^g_S$, *def*($[\![\psi]\!]^h_S$).
  8. *def*($[\![\exists t\varphi]\!]^g_S$) iff for all h s.t. $h[t]g$, *def*($[\![\varphi]\!]^h_S$).
- Values of $[\![\bullet]\!]^g_S$:
  1. $[\![U(t)]\!]^g_S := \{h|g = h \text{ and } [t]^g_S \in U^S\}$
  2. $[\![F(t)]\!]^g_S := \{h|g = h \text{ and } [t]^g_S \in F^S\}$

One can define a sound proof system for HODPL that can prove everything provable in AFT. The details of this proof system are beyond the scope of this paper.

7. Philosophical discussion and conclusion

ZFC is nowadays the most widely accepted formal foundation of mathematics. From a philosophical point of view, it is therefore very natural to ask for a justification of the axioms of ZFC. Sometimes one hears the opinion that these axioms are self-evident or follow from intuition, and indeed they are often felt to be self-evident by people who have their first contact with axiomatic set theory (Shelah, 1991, pp.4-5). However, we contend that this feeling, at least in the case of novices to set theory, comes from the following facts:

- All ZFC axioms apart from Extensionality, Choice and Foundation are instances of set comprehension, which people naturally feel as being correct. This feeling does not disappear when encountering Russell's paradox: Indeed the feeling about instances of set comprehension other than Russell's paradox is almost untouched by the recognition of a contradiction in the special instance of set comprehension constituting Russell's paradox.

- The Axiom of Extensionality constitutes – besides set comprehension – the second fundamental part of the intuitive feeling about the concept of set: It is the defining characteristic of set identity.

- The Axiom of Choice, we contend, is intuitively accepted because people implicitly accept function comprehension as presented in section 2. as part of their natural understanding of functions, and this function comprehension implies the existence of choice functions and hence the Axiom of Choice.

- The Axiom of Foundation is, of all axioms of ZFC, least naturally accepted by people new to set theory, but is quickly accepted as a limitation of the concept of set to the sets in the cumulative hierarchy, once this hierarchy has been introduced.

Given that this feeling is thus based on limiting the problematic principle of set comprehension to some of its instances (the axioms of Empty Set, Pairing, Union, Infinity, Separation and Replacement), one can further ask for a justification that these instances will not turn out to be just as contradictory as the now eliminated instances that yielded Russell's and Burali-Forti's paradoxes. Apart from the inductive justification that mathematicians have worked for a long time with these axioms without encountering a contradiction, a common justification is the iterative conception of the cumulative hierarchy (e.g. in Shoenfield (1977)). But as pointed out by Kanamori (2012):

> "When Replacement has been justified according to the iterative conception, the reasoning has in fact been circular as it was in Zermelo (1930), with some feature of the cumulative hierarchy picture newly adduced solely for this purpose."

Ackermann set theory starts from a completely different approach to limiting set comprehension, but nevertheless turns out to be equivalent to ZF. The fact that two such completely different approaches to limiting set comprehension yield essentially the same result can be taken to suggest that the "right" limitation has been found, similarly to the case of the Church-Turing thesis, where the fact that different formalisations of computability turned out to be equivalent has been taken to suggest that the "right" notion of computatability has been found.

In the case of Ackermann set theory, we have to add the Axiom of Choice explicitly to get the full strength of ZFC. If we interpret Ackermann set theory in AFT, on the other hand, the Axiom of Choice naturally appears. Thus AFT constitutes an alternative foundation of mathematics, with function as its primitive concept, and naturally including ZFC. Having functions as primitive concept is especially useful for formalising implicit dynamic function introduction, as it is done in the system HODPL, and this has indeed been our original motivation for developing AFT.

References

Ackermann, W. (1956). Zur Axiomatik der Mengenlehre. *Mathematische Annalen*, *131*, 336–345.

Church, A. (1940). A formulation of the simple theory of types. *Journal of Symbolic Logic*, *5*, 56–68.

Cramer, M., Fisseni, B., Koepke, P., Kühlwein, D., Schröder, B., & Veldman, J. (2010). The Naproche project – controlled natural language proof checking of mathematical texts. In *CNL 2009 Workshop, LNAI 5972* (pp. 170–186).

Cramer, M., Kühlwein, D., & Schröder, B. (2010). Presupposition projection and accommodation in mathematical texts. In *Semantic approaches in natural language processing: Proceedings of the Conference on Natural Language Processing 2010 (KONVENS)* (pp. 29–36). Universaar.

Groenendijk, J., & Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, *14*(1), 39–100.

Kanamori, A. (2012). In praise of replacement. *Bulletin of Symbolic Logic*, *18*(1), 46–90.

Lackenby, M. (2008). *Topology and groups* [Lecture Notes]. Available from `http://people.maths.ox.ac.uk/lackenby/tg050908.pdf`

Reinhardt, W. (1970). Ackermann's set theory equals ZF. *Annals of Mathematical Logic*, *2*, 189–249.

Shelah, S. (1991). The future of set theory. In H. Judah (Ed.), *Set theory of the reals.* (Proceedings of the Winter Institute held at Bar-Ilan University, Ramat Gan)

Shoenfield, J. (1977). Axioms of set theory. In J. Barwise (Ed.), *Handbook of mathematical logic* (pp. 312–344). Elsevier.

Trench, W. (2003). *Introduction to real analysis*. Prentice Hall.

Zermelo, E. (1930). Über das mengentheoretische Modell. In A. Kanamori & H.-D. Ebbinghaus (Eds.), *Collected works.* Springer.