The history of the use of [[.]]-notation in natural language semantics

Article *in* Semantics and Pragmatics · April 2016

DOI: 10.3765/sp.9.12

Brian Rabern

The University of Edinburgh

The history of the use of ⟦.⟧-notation in natural language semantics[*]

Brian Rabern
University of Edinburgh

— forthcoming in *Semantics & Pragmatics*

Abstract This short note answers the following question: When was the ⟦.⟧-notation introduced to natural language semantics?

Keywords: denotation brackets, semantic evaluation, ⟦.⟧, semantic notation

In contemporary linguistics that concerns the semantics of natural languages (or in work in related fields such as philosophy of language and cognitive science) one will often see the use of special brackets to enclose a linguistic expression, e.g. ⟦**carrot**⟧. Current semantics textbooks—such as Heim & Kratzer (1998) or Chierchia & McConnell-Ginet (2000)—will include lexical entries such as the following:

$$[\![\textbf{Dana}]\!] = \text{Dana}$$

$$[\![\textbf{types}]\!] = \lambda x.\, x \text{ types}$$

The latest volume of any semantics journal, such as *Natural Language Semantics*, *Journal of Semantics*, *Semantics & Pragmatics*, or *Linguistics and Philosophy*, is sure to include heavy use of the notation.[1] These brackets—so-called *denotation* brackets or *semantic evaluation* brackets—stand for a function that maps a linguistic expression to its "denotation" or semantic value (perhaps relative to a model or other parameters). The question addressed in this short note is the following: *When was the ⟦.⟧-notation introduced to semantics*?

Often such facts are stored in the institutional memory transmitted by the way notation is referred to. For example, one has a good idea of where to look for the

[*] The information collected here stems from a joint "investigation" that took place in large part on social media—important references and suggestions were provided by Barbara Partee, Simon Charlow, Toby Meadows, Brian Buccola, and Seth Yalcin. Many thanks to Dana Scott, who provided some very helpful comments, which improved the accuracy of this note dramatically.

[1] The style guide for *Semantics & Pragmatics* has advice on the brackets and insists that one "use \sv{ } (provided by sp.cls) for semantic evaluation brackets".

history of Kronecker's delta, δ_{ij}, or the Halmos, □, just given the commonly used names. Those with a background in fields closely associated with computer science may have heard the $[\![.]\!]$-brackets referred to as "Strachey brackets", and thus think the genealogy can be traced to Christopher Strachey and the work emanating from Oxford's Programming Research Group. Other theorists with a familiarity in set theory and the notation used in relation to boolean-valued forcing might assume that the history somehow involves Dana Scott. Many in linguistics might think that the notation, which has come to be associated with "Montague grammar", must have been introduced in Richard Montague's series of groundbreaking papers on semantics (Montague 1968, 1970b, 1970a, 1973). The purpose of this note is to sort out these conflicting impressions and to provide the correct historical details concerning the use of $[\![.]\!]$-notation.

Given that formal semantics for natural language developed from mathematical logic and model theory, it wouldn't be surprising if this notation was borrowed or adapted from notation already in use. In particular, a relevant place to look would be work in the algebraic approach to logic stemming from Alfred Tarski.[2] And if the notation is not in Tarski, then an obvious place to look would be the early pioneer of natural language semantics Montague, who was Tarski's student. Yet, even though the notation has been used in one form or another since the early development of natural language semantics in the 1960s and 1970s, Montague didn't himself make use of the $[\![.]\!]$-notation in his papers on semantics (more on this claim below).

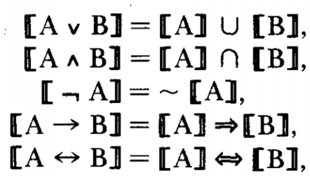
The earliest occurrence of the $[\![.]\!]$-notation where the brackets are clearly used to indicate *denotation* is, in fact, in a paper on the Continuum Hypothesis: Dana Scott's 1967, "A Proof of the Independence of the Continuum Hypothesis". Scott (1967) is concerned with *Boolean-valued models*, where a formula takes on values from "a

[2] One initially promising "ancestor" that people have mentioned in conversation is the square bracket notation for equivalence classes. For an equivalence relation R defined over some domain D the *equivalence class* of an element $a \in D$, is the set $[a] = \{x \mid Rxa\}$. At this level of abstraction there is no obvious connection to denotation, but specific equivalence relations on symbols of a formal language give rise to equivalence classes that can be construed as "semantic values". For example, the Lindenbaum algebra for propositional logic takes the equivalence class of sentences induced by the relation of provable equivalence: for a sentence ϕ, $[\phi] = \{\psi \mid \vdash \phi \leftrightarrow \psi\}$. Rasiowa and Sikorski's (1963) "Mathematics of Metamathematics" uses a form of this notation. They use the symbols $\sim$ and $\approx$ for different equivalence relations, and then use corresponding bars for the equivalence classes induced by the relations: $|x| = \{y \mid x \sim y\}$; $\|x\| = \{y \mid x \approx y\}$. On page 257 they let $\alpha \approx \beta$ if and only if both $(\alpha \to \beta)$ and $(\beta \to \alpha)$ are theorems of the propositional language. And thus they go on to write very contemporary looking semantic equations such as the following: $\|\alpha\| \cap \|\beta\| = \|\alpha \cap \beta\|$. This is all very suggestive of the current $[\![.]\!]$-notation but the connection is completely speculative. The somewhat related work by Scott (1967), discussed below, does cite Rasiowa and Sikorski, but here there is no plausible connection to the equivalence class notation. Another interesting speculation, which makes some sense in the context of denotational semantics, is that the notation evolved from parenthesised Quine corners: $(\ulcorner . \urcorner)$. Scott's own account (below) goes against both hypotheses.

system of generalized truth values"—values in a complete Boolean algebra beyond just *true* and *false* (or 1 and 0).[3] In particular, in this paper formulae take values from subsets of a probabilistic sample space Ω (up to sets of measure zero). Given this it would have proved convenient to introduce some new notation to indicate for a formula ϕ which subset of the sample space is ϕ's "truth value". For example, Scott has equations such as the following:

$$[\![\xi = \eta]\!] = \{\omega \in \Omega \mid \xi(\omega) = \eta(\omega)\}/[P = 0]$$

This notation also allows one to easily display of the truth values of various propositional combinations. For example, here is an excerpt from page 97 of Scott (1967):



$$\begin{aligned}
[\![\mathrm{A} \vee \mathrm{B}]\!] &= [\![\mathrm{A}]\!] \cup [\![\mathrm{B}]\!],\\
[\![\mathrm{A} \wedge \mathrm{B}]\!] &= [\![\mathrm{A}]\!] \cap [\![\mathrm{B}]\!],\\
[\![\neg \mathrm{A}]\!] &= \sim [\![\mathrm{A}]\!],\\
[\![\mathrm{A} \rightarrow \mathrm{B}]\!] &= [\![\mathrm{A}]\!] \Rightarrow [\![\mathrm{B}]\!],\\
[\![\mathrm{A} \leftrightarrow \mathrm{B}]\!] &= [\![\mathrm{A}]\!] \Leftrightarrow [\![\mathrm{B}]\!],
\end{aligned}$$

This use of double brackets (or double bars) is still commonly used in the literature on "forcing" in set theory (see Bell 2005 and Chow 2009). But *why* was this notation in particular introduced in the context of Boolean-valued models? It seems here there isn't any deep conceptual explanation. It was simply a convenient notation for "truth value", which was adapted from the $|.|$-notation for "absolute value"—or really a generalisation thereof for the norm of a vector. This is the account from Dana Scott, who writes,

> For Boolean-valued models, I first used $\|.\|$. In calculus (both real and complex) we write $|z|$ for the "absolute value of z". In linear space theory, we write $\|v\|$ for the "norm of the vector v". And in Euclidean spaces $\|v - w\|$ then gives us "the distance from v to w". I took the $\|$-notation to be "truth value" as a notation easy to type. But, as formulae became longer and longer, I changed to $[\![\Phi]\!]$ as being easier to read. (Also real double brackets—I seem to remember—became available on the IBM golf-ball typewriters, luckily.) That was in the mid-1960s. (Scott 2015, email)

[3] This method is connected to the unpublished paper Scott & Solovay 1967, "Boolean valued models of set theory" which sheds light on the method of forcing in set theory (cf. Scott 1969). See also the forward to Bell (2005) (written by Scott), and pp. 21-24 where the [[.]]-notation is introduced.

This account accords with what we take to be the earliest occurrences of the $[\![.]\!]$-notation in print in the mid-1960's.[4] Thus the "Scott bracket" terminology that shows up in some strands of the mathematical literature correctly encodes this aspect of the history (see, e.g., van Dalen 2012: 66). Although, it must immediately be pointed out that although the label is around, it is not terribly common in the literature, and it is even less common to indicate why the notation is associated with Scott.

The brackets are also used in Scott and Christopher Strachey's work from the late 1960s on the semantics of programming languages (Scott & Strachey 1971). Some have speculated that the $[\![.]\!]$-notation used in model-theoretic semantics was adapted from notation already in use in computer science, but the influence is clearly in the other direction. Scott confirms this, he says

> My work with Strachey began in the fall of 1969 in Oxford. I discovered the lattice-theoretic models for lambda-calculus in November of that year. And in developing denotational semantics it was I who suggested the $[\![.]\!]$-notation to Strachey, which he liked [since it helped insulate the object language from the metalanguage]. (Scott 2015, email)

In this work, where $\mathcal{V}$ is a function mapping numerals to numbers they let $\mathcal{V}[\![n]\!]$ be the number denoted by the numeral n, e.g.:

$$\mathcal{V}[\![0]\!] = \mathit{0}$$

$$\mathcal{V}[\![1]\!] = \mathit{1}$$

They emphasise that it is important to keep the object language separate from the metalanguage (the symbol '1' versus the number 1), and state that "... in the semantic equations we have enclosed the object language expressions in the special brackets $[\![\ ]\!]$ merely as an aid to the eye" (*ibid.*: 3). There are echoes of Scott and Strachey in various textbooks, e.g. Tennent (1976) who says "the symbols $[\![$ and $]\!]$ are used to enclose syntactic elements in order to separate the object and metalanguages" (439) (cf. Schmidt 1986: 55 and Winskel 1993: 56). The current literature in this area—what is called "denotational semantics" for programming languages—still uses the $[\![.]\!]$-notation, where in some corners the brackets are called "Strachey

[4] IBM's revolutionary Selectric typewriter was originally released in 1961. One of its novel features was the "typeball" technology, which had interchangeable font elements for Greek letters, and elements for special symbols for science and mathematics. At some point in the 1960's there was a special typeball for use in scientific and technical writing called "Press Roman Symbol Technical" (PRX-T), which had the $[\![\ ]\!]$ brackets. One wonders what compelled IBM to include double brackets on this typeball.

brackets".[5] Though the brackets were originally used as a device to "insulate" the object language from the metalanguage, the brackets are sometimes used to stand for the evaluation function itself (or at least the evaluation is suppressed).

Thus, the ⟦.⟧-notation was introduced to both the semantics of mathematical languages and the semantics of programming languages by Dana Scott. And eventually the notation made its way into linguistics via the semantics of natural languages. Model-theoretic semantics was applied to natural languages in the 1960s by theorists in the Tarskian tradition, such as Richard Montague, and others. They would have, of course, been familiar with, if not actively involved with, the aforementioned work in model-theory and the algebraic approach to logic.

But as was noted at the outset, Montague didn't make use of the ⟦.⟧-notation in his series of papers.[6] In Montague's PTQ (1973) he does use single square brackets in a way that, on a first glance, might seem like a variant of the ⟦.⟧-notation. But here the square brackets are, in fact, part of the *syntax* of the object language—the brackets are included in the formation rules as is often done in first-order logic. For example, something of the form "$[\phi \wedge \psi]$" is a meaningful expression. Then for any meaningful expression α Montague defines $\alpha^{\mathfrak{A},i,j,g}$ as the extension of α with respect to $\mathfrak{A}, i, j$, and g. Thus, the work of denotation brackets is being done by the superscripting alone. But since some expressions of the language contain outer square brackets some semantic clauses involve strings of the form $[\ldots]^{\mathfrak{A},i,j,g}$, e.g. $[\phi \wedge \psi]^{\mathfrak{A},i,j,g}$. Yet, the fact that this looks like a variant of the ⟦.⟧-notation is clearly an accident.[7]

An early place where the notation is explicitly advised in connection with Montague's work is Scott (1970) "Advice on modal logic"—yes, once again Scott is promoting the notation. He introduces the double bar variant of the ⟦.⟧-notation as follows:

[5] Although internet searches will see the label "Strachey brackets" show up in the lecture notes of various computer science courses, it does not seem to show up in canonical textbooks—the label does nevertheless show up in certain pockets of the literature (e.g. early work associated with the Vienna Development Method). For example, Kneuper (1989): "Terms in the object language (or specification or programming language) are written in Strachey brackets ⟦...⟧, in order to distinguish them from terms in the (meta-) language used for describing the semantics of the term" (28). Boulton (1998) also speaks of "the Strachey (emphasized) brackets ⟦ and ⟧ commonly used in denotational semantics" (71).

[6] Lewis (1970) also doesn't use the notation, although he does in later papers such as Lewis (1973: 47), where he calls them "proposition brackets".

[7] Notice, however, that Montague does sometimes add gratuitous square brackets to an expression. He admits, "In the presentation of actual expressions of intensional logic square brackets will sometimes for perspicuity be omitted, an sometimes gratuitously inserted" (Montague 1973: 230). Brackets are gratuitously inserted when stating certain semantic rules. For example, on complex expressions such as $\neg\phi$—presumably as a means of disambiguating what the superscript applies to. Instead of $\neg\phi^{\mathfrak{A},i,j,g}$ he writes $[\neg\phi]^{\mathfrak{A},i,j,g}$. This is a perhaps an unconscious step toward Scott's convention.

> In order to state in a convenient way the connections between statements and their parts some notation is in order. Let us first make truth-values visible: we write 1 for *true* and 0 for *false*. The reason for this choice of notation is that $2 = \{0, 1\}$ is a simple and readily available symbol for the set of the two truth-values. Next associated with a statement Φ will be a *function*, call it $\|\Phi\|$, the *value* of Φ in the interpretation, defined on [the set of indices] I with values in 2. In other words we shall write the equation
>
> $\|\Phi\|_i = 1$
>
> to mean the Φ *is true at i*. Other notations are possible, and some variants are discussed later. (Scott 1970: 150-151)

Toward the end of the paper he notes that he has been suppressing the interpretation $\mathfrak{A}$ and comments on why he prefers this notation over Montague's:

> The notation I would use is:
>
> $\|\tau\|^{\mathfrak{A}}$ and $\|\Phi\|^{\mathfrak{A}}$
>
> whereas Montague has recommended:
>
> $\tau_{\mathfrak{A}}$ and $\Phi_{\mathfrak{A}}$.
>
> His is shorter—too short it seems to me. The notation leaves us nothing to write when mention of $\mathfrak{A}$ is suppressed. (Of course to some, suppression is evil, and they would never consider doing it.) Thus I prefer the writing of the double bars as forcing me to remember the distinction between the *expression* and its *value*. That clearly is the kind of advice that one can either take or leave: all I ask is that you be reasonably clear about what you are doing. (Scott 1970: 163-164)

Notice that here, just as in the Scott & Strachey (1971) work on programming languages, Scott is using the notation to aid in keeping separate the object language expression from its value.

Scott's advice was taken: We find others in the early 1970s using denotation brackets, e.g. in Hans Kamp's (1971) work on double indexing in tense logic[8], and in Lewis' work on counterfactuals (Lewis 1971, Lewis 1973). Also Kaplan

[8] The ancestor of Kamp's famous 1971 paper on 'now' is Kamp (1967) "The treatment of 'now' as a 1-place sentential operator". This document consists of eight pages of hand-written notes that Kamp presented to Montague's seminar on pragmatics at UCLA in 1967 (see Blackburn & Jørgensen forthcoming for the history of the notes and how they influenced A.N. Prior). Interestingly, Kamp doesn't use double brackets in these early notes (he uses Montagovian conventions throughout, e.g. "ϕ is true$_{\langle i,j\rangle,\mathfrak{A}}$"), but does use double brackets for the 1971 publication. The notes are kept in the Prior archives (Box 15) in the Bodleian library, Oxford. Thanks to Klaus Frovin Jørgensen for providing the relevant archival work.

(1989)—published much later but which was originally presented as lectures in 1971—uses single bars, $|\alpha|^{\mathfrak{A}}_{cftw}$, for the denotation of terms (see also Partee 1975). There are many semantics papers throughout the 1970s that use some version of the double brackets (or bars). And then, the influential textbook Dowty et al. (1981) *Introduction to Montague Semantics* states the following notational convention:

> Notational Convention 2: For any expression α, we use $[\![\alpha]\!]$ to indicate the semantic value of α.

As noted already, current semantics textbooks, at least textbooks in the formal semantics tradition, tend to follow this convention (e.g. Heim & Kratzer 1998, Chierchia & McConnell-Ginet 2000, and Jacobson 2014).[9]

References

Bell, John L. 2005. *Set theory: Boolean-valued models and independence proofs*. Oxford University Press.

Blackburn, Patrick Rowan & Klaus Frovin Jørgensen. forthcoming. Arthur Prior and 'Now'. *Synthese* .

Boulton, Richard J. 1998. Generating embeddings from denotational descriptions. In Jim Grundy & Malcolm Newey (eds.), *Theorem proving in higher order logics* Lecture Notes in Computer Science, 67–86. Springer, Berlin: Heidelberg.

Chierchia, G. & S. McConnell-Ginet. 2000. *Meaning and grammar: An introduction to semantics*. The MIT Press.

Chow, Timothy. 2009. A beginner's guide to forcing. In T. Chow & D. Isaksen (eds.), *Communicating mathematics*, 25–40. AMS.

van Dalen, D. 2012. *Logic and structure*. Springer, London.

Dowty, D.R., R.E. Wall & S. Peters. 1981. *Introduction to Montague semantics*. Springer.

Heim, I. & A. Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell Publishers.

Jacobson, Pauline. 2014. *Compositional semantics: An introduction to the syntax/semantics interface*. Oxford University Press.

Kamp, Hans. 1967. The treatment of 'now' as a 1-place sentential operator. multilith University of California in Los Angeles.

Kamp, Hans. 1971. Formal properties of 'now'. *Theoria* 37. 227–274.

Kaplan, David. 1989. Demonstratives. In J. Almog, J. Perry & H. Wettstein (eds.), *Themes from kaplan*, 481–563. Oxford University Press.

[9] A notable exception is the Davidsonian text Larson & Segal (1995) where they write clauses like "Val(1, **Sam runs**) iff Sam runs" instead of, e.g.,"$[\![$**Sam runs**$]\!] = [\![$**runs**$]\!]([\![$**Sam**$]\!])$".

Kneuper, Ralf. 1989. *Symbolic execution as a tool for validation of specifications*: Department of Computer Science, University of Manchester dissertation.

Larson, R. & G. Segal. 1995. *Knowledge of meaning: An introduction to semantic theory*. MIT Press.

Lewis, D. 1970. General semantics. *Synthese* 22(1). 18–67.

Lewis, David. 1971. Completeness and decidability of three logics of counterfactual conditionals. *Theoria* 37(1). 74–85.

Lewis, David. 1973. *Counterfactuals*. Basil Blackwell.

Montague, R. 1968. Pragmatics. In R. Klibansky (ed.), *Contemporary philosophy: A survey*, vol. 1, 102–22. Florence, La Nuova Italia Editrice. Reprinted in *Formal philosophy: selected papers of Richard Montague*, 1974.

Montague, R. 1970a. Universal grammar. *Theoria* 36(3). 373–398.

Montague, R. 1973. The proper treatment of quantification in ordinary English. *Approaches to natural language* 49. 221–242.

Montague, Richard. 1970b. English as a formal language. In B. Visentini (ed.), *Linguaggi nella societá e nella tecnica*, 188–221. Edizioni di Communita.

Partee, Barbara. 1975. Montague grammar and transformational grammar. *Linguistic inquiry* 203–300.

Rasiowa, Helena & R. Sikorski. 1963. *The mathematics of metamathematics*. Pafistwowe Wydawnictwo Naukowe, Warsaw. Monografie matematyczne, Vol. 41.

Schmidt, David. 1986. *Denotational Semantics: A Methodology for Language Development*. Allyn and Bacon.

Scott, Dana. 1967. A proof of the independence of the continuum hypothesis. *Theory of Computing Systems* 1(2). 89–111.

Scott, Dana. 1969. Boolean models and nonstandard analysis. In W. A. J. Luxemburg (ed.), *Applications of model theory to algebra, analysis, and probability*, 87–92. New York, Holt, Rinehart and Winston.

Scott, Dana. 1970. Advice on modal logic. In K. Lambert (ed.), *Philosophical problems in logic: Some recent developments*, 143–173. D. Reidel.

Scott, Dana. 2015. ‘denotation brackets’. Personal email to Brian Rabern, June 5 2015.

Scott, Dana & Robert Solovay. 1967. Boolean valued models of set theory. *unpublished* .

Scott, Dana & Christopher Strachey. 1971. *Toward a mathematical semantics for computer languages*. Oxford University Computing Laboratory, Programming Research Group,.

Tennent, Robert D. 1976. The denotational semantics of programming languages. *Communications of the ACM* 19(8). 437–453.

Winskel, Glynn. 1993. *The formal semantics of programming languages: an introduction*. MIT press.

Brian Rabern
University of Edinburgh, PPLS
3 Charles St.
Edinburgh, EH8 9AD , United Kingdom
brian.rabern@gmail.com