5.3. Names and Predicates: Formal Semantics

1. Models. The semantics for sentence logic sharpened the intuitive idea of **possible situation** into the technical concept of **valuation**. With the language of sentence logic expanded to include names and predicates, we expand its semantics as well by extending valuations into **models**.

For those parts of the formal language inherited from previous chapters, the semantics will remain unchanged in models: a model assigns one (and only one) **truth value** to each sentence letter; and negations, conjunctions, disjunctions, and conditionals follow the familiar semantic rules.

This much semantics addresses A1, along with 2 through 6 of the construction rules.

Revised Construction Rules (*First Draft*)

Atomic Sentences:

A1. Sentence letters are atomic sentence
A2. A predicate letter followed by a name letter is an atomic sentence.

Formal Sentences:

1. Atomic sentences are formal sentences.
2. If ● is a formal sentence, then ~● is a formal sentence.
3. If ● and ▲ are formal sentences, then (● ∧ ▲) is a formal sentence.
4. If ● and ▲ are formal sentences, then (● ∨ ▲) is a formal sentence.
5. If ● and ▲ are formal sentences, then (● → ▲) is a formal sentence.
6. If ● and ▲ are formal sentences, then (● ↔ ▲) is a formal sentence.

Here we provide semantics for the one new type of sentence, introduced by A2: a predicate-letter-plus-name-letter.

2. Names and Reference. Note that neither an English predicate like "is Greek" nor name like "Aristotle" is a natural candidate for truth or falsehood. Yet in combination they form something which can be true or false: a sentence such as "Aristotle is Greek". And the same is true of predicate and name letters. So our formal semantics must provide each predicate letter and name letter with a non-1/0 value – but in a way that allows those values, combined, to yield a value of 1 or 0.

A proper name such as "Aristotle" serves to **refer** to some individual – and unlike a short-term, reusable pointer like "it," a proper name always refers to the **same** individual. A name letter, as the formal counterpart to a proper name likewise refers invariably to a particular individual.

In order to represent such reference in models, the expanded formal semantics will include a set of **objects** populating a **domain of discourse** – or "domain" for short. A little three-member domain would look like so.

D: {**Aristotle**, **Benjamin Franklin**, **William James**}

(Don't be misled by the need to depict things on this printed page by way of words: the three members of this domain aren't three names, but those three men themselves.)

And when we don't already have particular individuals (such as Aristotle or Benjamin Franklin) in mind, but just need some generic objects to populate a model, a quick way of meeting that need is to use numbers as the objects in the domain.

D: {**2, 3, 4**}

(We start with 2 to avoid confusion – because the numerals "0" and "1" are already used in the semantics to represent true and false.)

In order for the semantics to produce the desired results concerning validity, we insist that **the domain cannot be empty**. Every model must have a domain with at least one object.[1]

[1] Though there are 'non-classical' logics whose semantics allows an empty domain.

The semantics then specifies a **referent** – an object being referred to – for each name letter, drawing these from the domain of the model. While the semantics of previous chapters was governed by a single fundamental principle – the **Principle of Bivalence** – the expanded semantics imposes an additional principle, of equal importance: the **Principle of Reference**.[2]

> **Principle of Reference: each name letter refers to one and only one object in the domain.**

Just as a valuation assigns exactly one truth value to a sentence letter, a model assigns exactly one referent to each name letter under consideration – as in the following example.

D: {**Aristotle**, **Benjamin Franklin**, **William James**}

P: 1	**A**: **Aristotle**
Q: 0	**B**: **Benjamin Franklin**
R: 0	**C**: **William James**
	D: **Benjamin Franklin**

Note that the Principle of Reference *does* permit an object to have more than one name within the same model: here both "B" and "D" refer to the same object, Benjamin Franklin – in the same way that, e.g., the names "Benjamin Franklin" and "Silence Dogood" were different names for that same individual.[3] The Principle of Reference again parallels Bivalance: each sentence letter must have exactly one truth value, but different sentence letters can have the same truth value.

Finally, purely for purposes of convenience, we require that **every object in the domain must have at least one name**. There is no deep metaphysical point to this stipulation – our logic is not committed to some claim that there *couldn't* be an unnamed object (such as the Tao is perhaps said to be, in Chapter One of the *Tao Te Ching*). The requirement is only a time-saving measure, useful later in the semantics of quantifiers. While we could develop quantifier semantics without employing this assumption, it would only be a more complicated way of achieving the same results.

[2] This is sometimes called the *Principle of Denotation*.
[3] "Silence Dogood" being a pen name used by the young Benjamin Franklin

3. Predicates and Extensions. Turning to predicate letters, note that while a predicate can't be true, it can be **true *of*** something. For instance, in a domain containing just Aristotle, Benjamin Franklin, and William James, the predicate "is American" is true of two objects in the domain, and not true of one of them.

The set of things which a predicate holds true of is the **extension** of that predicate. In the last example, the extension of "is American" is the set containing just Benjamin Franklin and William James.

In formal models, the extension of a predicate letter is likewise the set of objects in the model's domain which that predicate letter holds true of. Our semantics must therefore specify an extension for every predicate letter listed in the model – each such extension being populated by objects in the domain of the model.[4] We extend our earlier example to include extensions for predicate letters "G" through "J".

D: {**Aristotle**, **Benjamin Franklin**, **William James**}

P: 1	**A**: **Aristotle**
Q: 0	**B**: **Benjamin Franklin**
R: 0	**C**: **William James**
	D: **Benjamin Franklin**
	G: {**Benjamin Franklin**, **William James**}
	H: {**Aristotle**, **Benjamin Franklin**, **William James**}
	I: {**Aristotle**}
	J: { }

For instance, "G" might stand for "is American"; "H" for "is human; "I" for "is Greek"; and "J" for "is a unicorn". Then our model works out sensibly enough: in this little three-member domain, James and Franklin are American; Aristotle is Greek; all three are human; and none are unicorns.

Note that while every proper name must refer to an object, a predicate letter need not have any objects in its extension. In this model the extension of "J"

[4] The extension of each predicate letter will thus be some (proper or improper) subset of the domain.

is empty. Exactly right: for reading "J" as "is a unicorn," in a universe containing just Aristotle, Benjamin Franklin, and William James, that predicate should indeed fail to apply to anything in the universe.

We do require, however, that **each predicate letter** have **only one extension** in a given model, so that name and predicate semantics in combination yields truth or falsehood without violating Bivalence.

Securing truth or falsehood for a predicate-letter-plus-name-letter is straightforward: such a sentence is true exactly when the object *referred to* by the name letter is contained in the *extension* of the predicate letter.

In this model, name letter "A" refers to Aristotle, who is indeed in the extension of "H" ("is human"). So the sentence "HA" ("Aristotle is human") is **true** in this model. But Aristotle is not contained within the extension of "G" ("is American"); so "GA" ("Aristotle is American") is **false**.

D: {**Aristotle**, **Benjamin Franklin**, **William James**}

P: 1
Q: 0
R: 0

A: **Aristotle**
B: **Benjamin Franklin**
C: **William James**
D: **Benjamin Franklin**

G: {**Benjamin Franklin**, **William James**}
H: {**Aristotle**, **Benjamin Franklin**, **William James**}
I: {**Aristotle**}
J: { }

HA: 1
GA: 0

Once the model assigns truth values to such atomic sentences, the Negation, Conjunction, Disjunction, Conditional, and Biconditional Rules assign values to *molecular sentences* built out of these atoms – as the following examples illustrate.

D: {**Aristotle**, **Benjamin Franklin**, **William James**}

P: 1
Q: 0
R: 0

A: **Aristotle**
B: **Benjamin Franklin**
C: **William James**
D: **Benjamin Franklin**

G: {**Benjamin Franklin**, **William James**}
H: {**Aristotle**, **Benjamin Franklin**, **William James**}
I: {**Aristotle**}
J: { }

~GA: 1
(GA ∧ HA): 0
(GB ∧ GD): 1
(GB ∧ HB): 1
(JA ∨ HA): 1
(GA → HA): 1
(HC → GA): 0

(HD ∧ P): 1
(GA ∨ Q): 0
(HC ↔ GA): 0
(GA ↔ JA): 1
(HB ↔ GB): 1