Tarski's Theory of Truth

Richard G. Heck, Jr.

1 The Language of Arithmetic

We shall here look at an example of a Tarskian characterization of truth. The language for which we shall give the characterization is the *language of arithmetic*, understood in the ordinary way. We begin by saying what the language of arithmetic is, first describing its primitive vocabulary.

The language of arithmetic contains the usual collection of *logical expressions*:

- An infinite collection of variables: $x_1, x_2, x_3,$ etc.
- Four binary connectives: $\wedge$, $\rightarrow$, $\vee$, $\equiv$
- One unary connective: $\neg$
- Two quantifiers: $\exists$, $\forall$
- One binary relational expression: $=$
- Grouping devices: $(,)$

We have also certain *non-logical expressions*:

- One individual constant (i.e., name): 0
- One one-place function-symbol: S
- Two two-place function-symbols: $+$, $\times$
- One two-place relational expression: $<$

All of these symbols are supposed to have their obvious meanings: The only one which may be unfamiliar is 'S', which means 'successor', i.e., 'the next number after...' or '$+1$'.

The rules for forming complex expressions from these primitive expressions are what one would expect. We say that t is a *term* if it is an expression which is either:

1. a variable;
2. the individual constant '0';
3. the result of prefixing 'S' to some other term, itself written in parentheses;
4. the result of writing some term u, in parentheses, followed by '$+$', followed by some other term v, also in parentheses;
5. the result of writing some term u, in parentheses, followed by '$\times$', followed by some other term v, in parentheses.
6. Nothing except things you can form by using (1)–(5) is a term.

Thus, 'x_1' is a term; '0' is a term; '$S(0)$', '$S(S(0))$', and '$S(x_1)$' are terms; '$(x_1) + (S(0))$' is a term; and '$(x_3) \times (S(S(0)))$' and '$(x_6) \times ((x_1) + (S(0)))$' are also terms. On the other hand, 'SS' is not a term. Convince yourself of that.

The point of (6)—the so-called closure condition—is to validate a certain sort of argument called *induction on the complexity of expressions*. Suppose we can show—this is the basis case—that all variables have some property ϕ, and that we can also show that '0' has ϕ. Suppose further that we can show—this is the induction step—that, whenever t and u have ϕ, so do $\ulcorner S(t) \urcorner$, $\ulcorner (t) + (u) \urcorner$ and $\ulcorner (t) \times (u) \urcorner$. Then all terms have ϕ. This is because all terms are formed by repeatedly applying these three methods of formation to the basic terms.

We say that an expression A is a *formula* if it is either:

1. the result of writing a term u, in parentheses, followed by either '$=$' or '$<$', followed by a term v, also in parentheses (these are the *atomic* formulae);
2. the result of writing '$\neg$', followed by some formula B, in parentheses;
3. the result of writing some formula B, in parentheses, followed by either '$\wedge$', '$\vee$', '$\rightarrow$', or '$\equiv$', followed by some formula C, in parentheses;
4. the result of writing either '$\exists$' or '$\forall$', followed by a variable, followed by some formula B, in parentheses, which formula does not itself already contain bound occurences of that variable (it is possible to

give a precise definition of 'bound occurence', but we shall not do so).[1]

5. Nothing is a formula except things you can form by using (1)–(4).

Thus, among the atomic formulae are '$(x_1) = (x_3)$' and '$((x_2) + (S(0))) < ((x_6) \times ((x_1) + (S(0))))$'. But '$x_1(S) = (x_6)+$' is not a formula. Convince yourself of that, too. Finally, a formula is a *sentence* if it contains no free variables.

Here again, the point of the 'closure condition', (5), is to validate induction on the complexity of expressions. Suppose we can show that all atomic formulae have ϕ. And suppose we can also show that, whenever formulae A and B have ϕ, then so do $\ulcorner\neg(A)\urcorner$, $\ulcorner(A) \vee (B)\urcorner$, $\ulcorner(A) \wedge (B)\urcorner$, $\ulcorner(A) \rightarrow (B)\urcorner$, $\ulcorner\exists v(A)\urcorner$ and $\ulcorner\forall v(A)\urcorner$, for any variable v that does not already occur bound in A. Then, again, it would follow that all formulae have ϕ.

The insistence on using parentheses here is just to guarantee that the terms and formulae will not be ambiguous. We shall henceforth drop parentheses when convenient and group in the usual way.

Our goal, then, is a 'materially adequate' theory of truth for the language of arithmetic: We want a theory of truth that will allow us to prove, for each sentence A of this language, the 'T-sentence': $\mathrm{Tr}(\text{`}A\text{'}) \equiv A$.

2 Truth for the Propositional Fragment

For simplicity, let us begin by characterizing truth for the *propositional fragment* of this language, i.e., for sentences of the language containing neither variables nor quantifiers. Before we do so, we need to say a little bit about the meta-language in which we shall be giving our characterization. We shall assume that the meta-language contains all the vocabulary of the object-language, that is, that all the primitive expressions of the object-language—and therefore, all the terms and formulae of the object-language—are also contained in the meta-language. We shall also assume that meta-language contains the semantic expressions 'true', 'denotes', and the like, and that it contains the resources to talk

[1] OK, we'll do so in a footnote. We define free and bound occurences by simultaneous induction. All occurences of variables in atomic formulae are free. An occurrence of a variable is free in $\ulcorner\neg(A)\urcorner$, $\ulcorner(A) \vee (B)\urcorner$, $\ulcorner(A) \wedge (B)\urcorner$, or $\ulcorner(A) \rightarrow (B)\urcorner$ just in case it is free in A or B. An occurence of a variable in $\ulcorner\exists v(A)\urcorner$ or $\ulcorner\forall v(A)\urcorner$ is bound if it was already bound or is an occurrence of v; otherwise, it is free.

about object-language expressions, the forms of object-language expressions, and the like. Later, we shall need the meta-language to contain a device enabling us to speak of assignments of objects to variables. All of this can be made more precise, but we will not worry ourselves with making it so here. It should be fairly clear that everything we shall say about the syntax of the object-language uses only pretty basic sorts of mathematics.[2]

We begin by characterizing truth for atomic sentences, and then we show how to extend this characterization of truth to complex sentences. To characterize truth for the atomic sentences, we need to do two things: Say what the various terms in the language denote, and then say, assuming that we know the denotations of the terms appearing in it, when an atomic sentence is true. We begin by characterizing denotation.

2.1 Denotation for Closed Terms

Terms that do not have variables in them come in four flavors (see the definition of 'term' above): We have the term '0', terms of the form 'Su', terms of the form '$t + u$', and terms of the form '$t \times u$'. It is clear what we want to say about each of these. '0' denotes zero; 'Su' denotes one more than the denotation of u; etc. We record all of this as follows:

Clause for '0': '0' denotes x iff $x = 0$.[3]

Clause for 'S': 'St' denotes x if, and only if, there is an object y such that t denotes y and $x = Sy$.

Clause for '+': '$t + u$' denotes x if, and only if, there are objects y and z such that t denotes y and u denotes z and $x = y + z$.

Clause for '$\times$': '$t \times u$' denotes x if, and only if, there are objects y and z such that t denotes y and u denotes z and $x = y \times z$.

[2] If the goal is a theory of truth that is 'materially adequate' and 'formally correct', in Tarski's sense—that is, if the goal is a theory capable of proving all the T-sentences—then the entire construction can be carried out in a theory that is interpretable in Robinson arithmetic, Q. The meta-*language* must be taken to contain not just $\in$ and semantic notions, like truth and denotation, but also the language of arithmetic; but the meta-theory need only contain the axioms of 'weak set theory'—null set and adjunction—and the semantic clauses. In particular, the meta-theory need contain no arithmetical axioms. Since weak set theory in its usual formulation is interpretable in Q, so is this new theory in the expanded language; and the new theory will then provide not only the syntax we need but also a theory of sequences, since weak set theory is sequential. (All of this is discussed in detail in my paper "The Strength of Truth-theories", *forthcoming*.)

[3] The point of stating the clause this way is to guarantee the uniqueness of denotation.

Note that our characterization of denotation is *recursive*, in the sense that we begin by characterizing denotation for the simplest case and then characterize it for more complex cases in terms of what we have said about the simpler cases.

This characterization of denotation is materially adequate in Tarski's sense. Let us first work through an example.

Consider the term '$S0 \times (0 + S0)$'. Since it is of the form $t \times u$, the clause for $\times$ applies:

> '$S0 \times (0 + S0)$' denotes x if, and only if, there are objects y and z such that '$S0$' denotes y and '$0 + S0$' denotes z and $x = y \times z$.

Since '$S0$' is of the form St, the clause for S applies:

> '$S0$' denotes y if, and only if, there is an object w such that '0' denotes w and $y = Sw$.

Of course, by the clause for '0':

> '0' denotes w iff $w = 0$.

Hence, there is an object w such that '0' denotes w, namely, 0; and $S0$ is certainly equal to $S0$. So we can conclude:

> There is an object w such that '0' denotes w and $S0 = Sw$.

And so:

> '$S0$' denotes $S0$.

Now, '$0 + S0$' is of the form $t + u$, so the clause for $+$ applies:

> '$0 + S0$' denotes v if, and only if, there are objects t and u such that '0' denotes t and '$S0$' denotes u and $v = t + u$.

But we have already seen that '0' denotes 0 and that '$S0$' denotes $S0$. Moreover, since $0 + S0$ is certainly equal to itself, we can conclude:

> There are objects t and u such that '0' denotes t and '$S0$' denotes u and $0 + S0 = t + u$.

And so:

> '$0 + S0$' denotes $0 + S0$.

But then there are objects y and z such that '$S0$' denotes y and '$0 + S0$' denotes z and $S0 \times (0 + S0) = y \times z$. And so:

'$S0 \times (0 + S0)$' denotes $S0 \times (0 + S0)$.

As wanted.

It should seem plausible that a similar sort of argument will, as claimed, produce such a result for each term. In fact, we can do better: We can argue by induction on the complexity of expressions that such a result can be produced for each term. There are two ways one can present this argument.

One is an argument given in the meta-language: We can use the clauses for '0', 'S', '$+$', and '$\times$' to argue that every expression has a unique denotation. The argument is straightforward. Clearly, '0' has a unique denotation. So assume that t has a unique denotation: That is, there is an object x such that t denotes z iff $z = x$. Then by the clause for S, 'St' denotes Sx. Suppose that 'St' denotes w. Then, by the clause for S, there must be an object v such that t denotes v and $w = Sv$. But by the induction hypothesis, $v = x$ and so $w = Sx$. That is: 'St' denotes w if, and only if, $w = Sx$, and so 'St' denotes uniquely. The arguments in the other two cases are similar and are left as Exercise 2. Taking those cases as read, however, we may be conclude that every term denotes uniquely.

The other way to think of the argument is as one *about* what can be proven in a certain theory, namely, the meta-theory in which the previous argument was given. This argument would show, by induction on the complexity of expressions, that, for each term t, the following sentence is a theorem of the meta-theory:

(1) 't' denotes w iff $w = t$

To give this argument, we would need to formalize the meta-theory, which we shall not, but it is easy enough to see how the argument goes: Clearly, if t is the term '0', then we do have

(2) '0' denotes w iff $w = 0$

as a theorem of the meta-theory, since this is just the clause for '0'. Suppose, then, that the meta-theory proves:

(3) 't' denotes w iff $w = t$

Since the meta-theory contains the clause for S, it proves:

(4) 'St' denotes y if, and only if, there is an object z such that t denotes z and $y = Sz$

Now, the meta-theory contains some complete formalization of classical logic, so we may reason as follows: If the meta-theory proves some formulae $A_1, \ldots, A_n$, and if these together imply some formula B, then the meta-theory also proves B. But (3) and (4) together imply

(5) 'St' denotes *y* if, and only if, there is an object *x* such that $x = t$ and $y = Sx$

so the meta-theory proves as much. And this in turn implies

(6) 'St' denotes *w* if, and only if, $w = St$

as promised. Again, the arguments for '$+$' and '$\times$' are similar and are left as Exercise 3.

So, every term uniquely denotes, we may henceforth use a more compressed mode of expression: We can use the phrase 'the denotation of' when speaking of terms. Officially, this phrase is to be eliminated using Russell's theory of descriptions, but what we have just proven is that these descriptive phrases are always proper, so there are no concerns about scope-ambiguities and the like.

2.2 Truth for Atomic Sentences

We now complete the characterization of truth for atomic sentences.

All atomic sentences are of the forms '$t = u$' and '$t < u$', for some terms t and u. What we want to say about when such sentences will be true is fairly obvious. We record it in the

Clause for '=': A sentence of the form '$t = u$' is true if, and only if, there are objects x and y such that t denotes x and u denotes y and $x = y$.

Clause for <: A sentence of the form '$t < u$' is true if, and only if, there are objects x and y such that t denotes x and u denotes y and $x < y$.

Let us work through an example.

Consider the atomic sentence '$S0 = S0 \times (0 + S0)$' and work from outside in. Since this is a sentence of the form '$t = u$', the clause for '=' applies, telling us that

> '$S0 = S0 \times (0 + S0)$' is true iff there are objects x and y such that '$S0$' denotes x and '$S0 \times (0 + S0)$' denotes y and $x = y$

We know from our work in section 2.1 that

> the denotation of '$S0$' $= S0$
>
> the denotation of '$S0 \times (0 + S0)$' $= S0 \times (0 + S0)$

So we have:

> '$S0 = S0 \times (0 + S0)$' is true iff $S0 = S0 \times (0 + S0)$

As wanted. As it happens, it is the case that $S0 = S0 \times (0 + S0)$, since $1 = 1 \times (0 + 1)$, so it follows that '$S0 = S0 \times (0 + S0)$' is indeed true.

We can do better: We can prove by induction on the complexity of expressions that such a result can be proven for every atomic sentence. Here again, there are, in principle, two ways to think of this argument: Either as given in the meta-theory itself or as an argument about what can be proven in the meta-theory. The way we have set things up, however, no argument of the former sort is actually available: There is no 'property' on which we can run the inducton.[4]

The argument about the meta-theory proceeds as follows. We want to show that, for each atomic sentence A, the meta-theory proves a theorem of the form:

> 'A' is true if, and only if, A

Here again, making this argument precise would require us to formalize the meta-theory, which we shall not do, but the idea behind the argument is again straightforward. Let A be an atomic sentence: So it is either '$t = u$' or '$t < u$', for some terms t and u. By the results of section 2.1, the meta-theory proves both

> '*t*' denotes *t*
>
> '*u*' denotes *u*

So, since the meta-theory contains the clauses for $=$ and $<$, it also proves:

[4] There is, however, a different way to proceed, following Frege by taking sentences to denote their truth-values. We may take the truth-values to be 0 and $S0$, the former being False and the latter True, and restate the clause for $=$ as follows:

> A sentence of the form '$t = u$' denotes $S0$ if there are objects x and y such that t denotes x and u denotes y and $x = y$ and denotes 0 if there are objects x and y such that t denotes x and u denotes y and $x \neq y$.

Similar changes are then needed to the other clauses. Then we can argue by induction that every sentence denotes either 0 or $S0$.

'$t = u$' is true if, and only if, $t = u$

'$t < u$' is true if, and only if, $t < u$

Thus, our theory of truth for atomic sentences is materially adequate, in Tarski's sense.

2.3 Truth and the Propositional Connectives

We now extend our characterization of truth to complex sentences. These come in five flavors (see the definition of a formula above): '$A \wedge B$', '$A \vee B$', '$A \rightarrow B$' '$A \equiv B$', and '$\neg A$'. And it is clear what we want to say about these: E.g., that a sentence of the form '$A\&B$' is true if, and only if, ***A*** is true and ***B*** is true. This we record in the following five clauses.

Clause for '$\wedge$': A sentence of the form '$A \wedge B$' is true iff A is true $\wedge$ B is true

Clause for '$\vee$': A sentence of the form '$A \vee B$' is true iff A is true $\vee$ B is true

Clause for '$\rightarrow$': A sentence of the form '$A \rightarrow B$' is true iff A is true $\rightarrow$ B is true

Clause for '$\equiv$': A sentence of the form '$A \equiv B$' is true iff A is true $\equiv$ B is true

Clause for '$\neg$': A sentence of the form '$\neg A$' is true iff $\neg$(A is true)

That completes our characterization of truth for the propositional fragment of the language of arithmetic. Note that our characterization of truth is again *recursive*, in the sense that we have characterized truth first for atomic sentences, and then characterized truth for more complex sentences in terms of truth for simpler sentences.

It is should be clear that this characterization is also materially adequate in Tarski's sense. We work through a simple example (using some abbreviations for brevity).

'$(0 = S0) \vee (S0 = 0 + S0)$' is true iff '$0 = S0$' is true $\vee$
'$S0 = 0 + S0$' is true
iff the den of '0' = the den of '$S0$' $\vee$
the den of '$S0$' = the den of '$0 + S0$'
iff 0 = S(the den of '0') $\vee$
S(the den of '0') = the den of '0' + S(the den of '0')
iff (0 = S0) $\vee$ (S0 = 0 + S0)

And so:

$$\text{`}(0 = S0) \vee (S0 = 0 + S0)\text{' is true iff } (0 = S0) \vee (S0 = 0 + S0)$$

As wanted. We have thus given a *materially adequate* characterization of truth for the propositional fragment of the language of arithmetic.

The argument that this characterization is materially adequate is similar to the ones we have already seen and is left as Exercise 8.

3 Truth and the Quantifiers

3.1 Satisfaction

So far, all of this has been pretty simple. Things only get complicated when we introduce variables and quantification. The reason things get complicated is this. Consider the sentence

$$\exists x_1(x_1 = 0)$$

Up to this point, we have been characterizing the truth of a complex sentence in terms of the truth of its simpler parts. But the relevant part of this sentence is

$$x_1 = 0$$

and this is not a sentence at all. It does not make any obvious sense to talk about whether '$x_1 = 0$' is true. What we need to talk about, as should be familiar from introductory logic, is whether this formula is true under some assignment of an object (in the universe of discourse) to the free variable 'x_1'. More generally, if we are to give a characterization of truth which will apply to a sentence such as

$$\forall x_1 \exists x_2 \exists x_6(x_1 = x_2 + x_6)$$

we shall have to characterize truth under an assignment for such formulae as

$$x_1 = x_2 + x_6$$

We shall therefore need to modify our earlier characterization of truth for atomic sentences, so that it is a characterization not of truth simpliciter but of truth under an assignment of objects to free variables. Thus, we shall want to be able to say that '$x_1 = 0$' is true under the assignment of 0 to x_1; false under the assignment of 3 to x_1; and so forth.

What is an assignment? Well, an assignment associates an object with each variable. Mathematically, however, an 'association' is just a function, so an assignment is a function from variables to objects in the domain. We will assume that all such functions are total, that is, that they assign a value to each variable.[5] Our meta-language will therefore need to contain the resources to talk about such functions, and we will need to appeal to principles concerning them. As it happens, the following principle is sufficient:

$$\forall\alpha\forall v\forall x\exists\beta\,[\forall w\,(w \neq v \rightarrow val(\alpha, w) = val(\beta, w)) \wedge val(\beta, v) = x]$$

The Greek letters ares variables ranging over assignments, and $val(\alpha, v)$ means: the value α assigns to v. This principle thus says that, for any assignment, any variable, and any object, there is another assignment that (i) agrees with the original one on all variables other than the given one but (ii) assigns the object in question to the given variable. In short: You can change what gets assigned to the variables, pretty much at will.

Tarski proceeds in a different, but equivalent, way. Since the variables come in a natural order, we can just specify an infinite sequence[6] of objects and let the natural order of the variables pick out which objects get assigned to which variables: 'x_1' gets assigned the *first* member of the sequence; 'x_2' gets assigned the *second* member of the sequence; and, in general, the variable whose subscript is the numeral for n is assigned the n^{th} member of the sequence. It seems more natural, however, to work with functions from variables to objects.

Because the condition in the first conjunct of the principle governing assignments will be important to us, we introduce an abbreviation:

$$\alpha \underset{v}{\sim} \beta \stackrel{df}{\equiv} \forall w\,(w \neq v \rightarrow val(\alpha, w) = val(\beta, w))$$

[5] In fact, assignments can be taken to be finite and therefore partial functions; they do not need to assign a value to every variable. We would then need a relation $val(\alpha, v, x)$ and would formulate the crucial principle as:

$$\forall\alpha\forall v\forall x\exists\beta\,[\forall w\,(w \neq v \rightarrow val(\alpha, w, x) \equiv val(\beta, w, x)) \wedge val(\beta, w, x)]$$

We do not proceed in this way here as it introduces some complications below, which I shall mention in the footnotes. It is important that things *can* be done with finite functions, however: Infinitary objects, like functions that assign objects to all variables, are not needed in the definition of truth.

That said, it's fairly easy to see that the principle we are using gives us no additional power: The theory with the axiom we are using is interpretable in one that makes use only of finitary functions.

[6] For the reasons mentioned in footnote 5, finite sequences will suffice, in fact.

The principle governing assignments can then take the form:

$$\forall\alpha\forall v\forall x\exists\beta\left[\alpha \underset{v}{\sim} \beta \wedge val(\beta, v) = x\right]$$

To return to our problem, then, we need to characterize truth under assignments. But we shall follow Tarski in speaking, instead, of *satisfaction*. Instead of saying that a formula is *true under* a given assignment, we shall say that the formula is *satisfied by* the assignment, or that the assignment *satisfies* the formula.

3.2 Recasting the Characterization of Truth for the Propositional Fragment in Terms of Satisfaction

We now recast our characterization of truth for atomic sentences as a characterization of satisfaction for atomic formulas. Since we characterized truth in terms of denotation, we need to characterize something related to denotation in the way satisfaction is related to truth: We therefore characterize, not 't denotes x', but 't denotes x under σ', 'σ' being a variable for assignments. We will write this: t denotes$_\sigma x$. The changes to the old clauses should be pretty obvious. We have a new clause that characterizes the denotation of variables under an assignment.

Clause for '0': '0' denotes$_\sigma$ x iff $x = 0$.

Clause for variables: 'x_n' denotes$_\sigma$ y iff $y = val(\sigma, `x_n')$

Clause for 'S': 'St' denotes$_\sigma$ y iff there is an object x such that t denotes$_\sigma$ x and $y = Sx$.

Clause for '+': '$t + u$' denotes$_\sigma$ z iff there are objects x and y such that t denotes$_\sigma$ x and u denotes$_\sigma$ y and $z = x + y$.

Clause for '$\times$': '$t \times u$' denotes$_\sigma$ z iff there are objects x and y such that t denotes$_\sigma$ x and u denotes$_\sigma$ y and $z = x \times y$.

We now characterize satisfaction for atomic sentences:

Clause for '=': An assignment σ satisfies a sentence of the form '$t = u$' iff there are objects x and y such that t denotes$_\sigma$ x and u denotes$_\sigma$ y and $x = y$.

Clause for '<': An assignment σ satisfies a sentence of the form '$t < u$' iff there are objects x and y such that t denotes$_\sigma$ x and u denotes$_\sigma$ y and $x < y$.

The clauses for the truth-functional connectives are then re-cast to give a characterization of satisfaction for truth-functional compounds:

Clause for '$\wedge$': An assignment σ satisfies a sentence of the form '$A \wedge B$' iff (σ satisfies A $\wedge$ σ satisfies B)

Clause for '$\vee$': An assignment σ satisfies a sentence of the form '$A \vee B$' iff (σ satisfies A $\vee$ σ satisfies B)

Clause for '$\rightarrow$': An assignment σ satisfies a sentence of the form '$A \rightarrow B$' iff (σ satisfies $A \rightarrow \sigma$ satisfies B)

Clause for '$\equiv$': An assignment σ satisfies a sentence of the form '$A \equiv B$' is true iff (σ satisfies $A \equiv \sigma$ satisfies B)

Clause for '$\neg$': An assignment σ satisfies a sentence of the form '$\neg A$' is true iff $\neg$(σ satisfies A)

No surprises there.

3.3 Satisfaction for the Quantifiers

Finally, we need to give clauses for the quantifiers. To understand these clauses, consider the question under what circumstances we shall want to say that the sentence '$\exists x_1(x_1 = 0)$' is true. In basic logic, we learn that this sentence is true if, and only if, there is some assignment under which the formula '$x_1 = 0$' is true, i.e., if, and only if, there is some object we could assign to 'x_1' such that, under that assignment, '$x_1 = 0$' comes out true. But, for the reasons we discussed above, it is not enough to characterize the *truth* of quantified sentences: Sometimes quantified formulae are parts of other quantified formulae. For example, the formula '$\exists x_1(x_1 = Sx_2)$' is in the relevant sense part of '$\forall x_2 \exists x_1(x_1 = Sx_2)$'. We therefore need to characterize the notion of *satisfaction* for quantified formulae: We need to say when quantified formulae are satisfied by assignments. So consider the formula

$$\exists x_1(x_1 = Sx_2)$$

Is this formula satisfied by the assignment that assigns 0 to 'x_1', 1 to 'x_2', and (say) 2 to every other variable? What we want to say is that it is true relative to this assignment iff there is some object y such that '$x_1 = Sx_2$' would come out true under the assignment of y to 'x_1', with

all the other assignments—in particular, the assignment of 1 to 'x_2'—being left unchanged. Thus, what we are doing is fiddling around with what gets assigned to 'x_1' while leaving what gets assigned to all the other variables fixed: We are, that is, considering what happens if *we change what is assigned to the quantified varaible but leave the rest of the assignment fixed*.

It is thus that we arrive at Tarski's clause for the existential quantifier:

Clause for '$\exists$': An assignment σ satisfies a formula of the form $\ulcorner\exists v(A)\urcorner$ iff there is some assignment τ which differs from σ, if at all, in what it assigns to v, such that τ satisfies A.

Using the notation introduced above, we can write this as:[7]

Clause for '$\exists$': σ satisfies $\ulcorner\exists v(A)\urcorner$ iff $\exists\tau[\sigma \underset{v}{\sim} \tau \wedge \tau$ satisfies $A]$

The clause for the universal quantifier is similar:

Clause for '$\forall$': σ satisfies $\ulcorner\forall v(A)\urcorner$ iff $\forall\tau[\sigma \underset{v}{\sim} \tau \rightarrow \tau$ satisfies $A]$

That completes our characterization of satisfaction for the language of arithmetic.

3.4 A Characterization of Truth

To complete our characterization of *truth* for the language of arithmetic, we now need only say when a sentence is true. A *sentence*, recall, is a formula with no free variables, and the fact that a formula happens not to contain any free variables does not mean that it does not make sense to speak of it as being satisfied by a given assignment. But if a formula does not contain any free variables, then whether it is satisfied

[7] It is now usual in logic, as it was not when Tarski wrote "The Concept of Truth", to take the quantifiers in the object-language not to range over all the objects there are—or, more precisely, over all the objects over which the quantifiers of the meta-language range—but instead to range over some 'domain'. We can make allowance for this in a variety of ways. Perhaps the simplest is to take $\tau \underset{v}{\sim} \sigma$ instead to abbreviate:

$$\forall w[w \neq v \rightarrow val(\tau, w) = val(\sigma, w)) \wedge D(val(\tau, v)]$$

where the domain is given by the formula $D(x)$. Note that, if we do so, then the T-sentences for sentences containing quantifiers will not be fully 'homophonic' but will instead look like: '$\forall x A(x)$' is true iff $\forall x(D(x) \rightarrow A(x))$.

by a given assignment will not depend upon what objects are assigned to which variables: There aren't any free variables in the sentence, so the entire assignment is just garbage.[8] Thus, if a sentence is satisfied by one assignment, it is satisfied by all: More generally, a sentence is either satisfied by all assignments or by none. It is for this reason that Tarski can characterize truth as follows:

Truth: A sentence A is true iff $\forall\sigma$[σ satisfies A]

That then really *does* complete our characterization of truth for the full language of arithmetic.

This characterization of truth is materially adequate, though seeing that it is is a lot more difficult than in the previous case. We shall again work through a simple example, this time using a lot of additional notation which should be pretty much self-explanatory.

Consider the sentence: $\forall x_1 \exists x_3 (x_3 = Sx_1)$. According to the characterization of truth:

(1) $\mathbf{True}(\text{`}\forall x_1 \exists x_3 (x_3 = Sx_1)\text{'})$ iff $\forall\sigma(\mathbf{Sat}_\sigma(\text{`}\forall x_1 \exists x_3 (x_3 = Sx_1)\text{'})$

According to the clause for '$\forall$', we have that

(2) $\mathbf{Sat}_\sigma(\text{`}\forall x_1 \exists x_3 (x_3 = Sx_1)\text{'})$ iff $\forall\tau[\tau \underset{x_1}{\sim} \sigma \rightarrow \mathbf{Sat}_\tau(\text{`}\exists x_3 (x_3 = Sx_1)\text{'})]$

By the clause for '$\exists$':

(3) $\mathbf{Sat}_\tau(\text{`}\exists x_3 (x_3 = Sx_1)\text{'})$ iff $\exists\upsilon[\upsilon \underset{x_3}{\sim} \tau \wedge \mathbf{Sat}_\upsilon(\text{`}x_3 = Sx_1\text{'})]$

And then:

(4) $\mathbf{Sat}_\upsilon(\text{`}x_3 = Sx_1\text{'})$ iff $\mathbf{Den}_\upsilon(\text{`}x_3\text{'}) = \mathbf{Den}_\upsilon(\text{`}Sx_1\text{'})$

(5) $\mathbf{Sat}_\upsilon(\text{`}x_3 = Sx_1\text{'})$ iff $\mathbf{Den}_\upsilon(\text{`}x_3\text{'}) = S(\mathbf{Den}_\upsilon(\text{`}x_1\text{'})$

(6) $\mathbf{Sat}_\upsilon(\text{`}x_3 = Sx_1\text{'})$ iff $val(\upsilon, \text{`}x_3\text{'}) = S(val(\upsilon, \text{`}x_1\text{'}))$

Substituting into (3), then:

(7) $\mathbf{Sat}_\tau(\text{`}\exists x_3 (x_3 = Sx_1)\text{'})$ iff $\exists\upsilon[\upsilon \underset{x_3}{\sim} \tau \wedge val(\upsilon, \text{`}x_3\text{'}) = S(val(\upsilon, \text{`}x_1\text{'}))]$

[8] This is actually a very deep and very important *theorem*, though not one we shall try to prove here. To prove it, one must prove the more general claim that, if σ and τ are assignments that agree on all free variables in A, then $\mathrm{Sat}_\sigma(\text{`}A\text{'}) \equiv \mathrm{Sat}_\tau(\text{`}A\text{'})$. The argument is by induction on the complexity of formulas and requires, as a lemma, a similar result for terms.

We now have to get rid of the quantifier '$\exists \upsilon$', which certainly had better not still be hanging around when we get to our T-sentence at the end of all of this. To do so, we prove in the meta-theory that[9]

(8) $\exists \upsilon[\upsilon \underset{x_3}{\sim} \tau \wedge val(\upsilon, \text{`}x_3\text{'}) = S(val(\upsilon, \text{`}x_1\text{'}))]$ iff $\exists x_3[x_3 = S(val(\tau, \text{`}x_1\text{'}))]$

Left-to-right: Suppose that there is some such assignment υ. Then $val(\upsilon, \text{`}x_3\text{'}) = S(val(\upsilon, \text{`}x_1\text{'}))$, so if we let $x_3 = val(\upsilon, \text{`}x_3\text{'})$, then certainly $x_3 = S(val(\upsilon, \text{`}x_1\text{'}))$. But since '$x_1$' $\neq$ 'x_3', we must have that $val(\upsilon, \text{`}x_1\text{'}) = val(\tau, \text{`}x_1\text{'})$, so $x_3 = S(val(\tau, \text{`}x_1\text{'}))$, as well..

Right-to-left: Suppose there is an x_3 such that $x_3 = S(val(\tau, \text{`}x_1\text{'}))$. By the principle governing assignments (see page 3.1), there is therefore an assignment $\upsilon$ such that $\upsilon \underset{x_3}{\sim} \tau$ and $val(\upsilon, \text{`}x_3\text{'}) = x_3$. So $val(\upsilon, \text{`}x_3\text{'}) = S(val(\upsilon, \text{`}x_1\text{'}))$, and we are done.

Note where there are quotes and where there are not! What makes arguments like this one hard to understand is the way in which we are moving between talking about variables and talking about their values—that is, the way we are moving between mention and use. That is what the theory of truth is about: about connecting talk of expressions with talk about the world.

To return to the argument, then, using (8) and substituting into (2) we have:

(9) $\text{Sat}_\sigma(\text{`}\forall x_1 \exists x_3(x_3 = Sx_1)\text{'})$ iff $\forall \tau[\tau \underset{x_1}{\sim} \sigma \rightarrow \exists x_3(x_3 = S(val(\tau, \text{`}x_1\text{'}))]$

Now, we get rid of the quantifier '$\forall \tau$', as we got rid of '$\exists \upsilon$' before. By an argument like that given for (8), we establish in the meta-theory that[10]

[9] If we have restricted the domain, as discussed in note 7, then what we will need to prove is instead:

$$\exists \upsilon[\upsilon \underset{x_3}{\sim} \tau \wedge val(\upsilon, \text{`}x_3\text{'}) = S(val(\upsilon, \text{`}x_1\text{'}))] \text{ iff } \exists x_3[D(x_3) \wedge x_3 = S(val(\tau, \text{`}x_1\text{'}))]$$

Left-to-right, $D(x_3)$ will follow from the altered definition of $\upsilon \underset{x_3}{\sim} \tau$; right-to-left, $\upsilon \underset{x_3}{\sim} \tau$, so defined, requires that $D(x_3)$.

[10] And here, if the domain is restricted, what we will prove is:

$$\forall \tau[\tau \underset{x_1}{\sim} \sigma \rightarrow \exists x_3(D(x_3) \wedge x_3 = S(val(\tau, \text{`}x_1\text{'}))] \text{ iff } \forall x_1[D(x_1) \rightarrow \exists x_3(D(x_3) \wedge x_3 = Sx_1)]$$

Left-to-right, we will assume that x_1 is in the domain, and this will be needed to show that there is an assignment τ such that $\tau \underset{x_1}{\sim} \sigma$ and $val(\tau, \text{`}x_1\text{'}) = x_1$. Right-to-left, the fact that $\tau \underset{x_1}{\sim} \sigma$ will guarantee that $D(val(\tau, \text{`}x_1\text{'})$ and so that $D(x_1)$.

(10) $\forall\tau[\tau \underset{x_1}{\sim} \sigma \rightarrow \exists x_3(x_3 = S(val(\tau, \text{`}x_1\text{'}))]$ iff $\forall x_1 \exists x_3[x_3 = Sx_1]$

Left-to-right: Let x_1 be an arbitrary object. We want to show that $\exists x_3[x_3 = Sx_1]$.[11] By the principle governing assignments, there is an assignment τ such that $\tau \underset{x_1}{\sim} \sigma$ and $val(\tau, \text{`}x_1\text{'}) = x_1$. Hence, by the left-hand side, $\exists x_3(x_3 = S(val(\tau, \text{`}x_1\text{'}))$, and so $\exists x_3[x_3 = Sx_1]$, and we are done.

Right-to-left: Let τ be an arbitrary sequence such that $\tau \underset{x_1}{\sim} \sigma$. We want to show that $\exists x_3(x_3 = S(val(\tau, \text{`}x_1\text{'}))]$. But this is immediate from the right-hand side, since, instantiating '$x_1$' with '$val(\tau, \text{`}x_1\text{'})$, we have immediately that $\exists x_3(x_3 = S(val(\tau, \text{`}x_1\text{'}))]$.

So, using (10) and substituting into (9), we have:

(11) $\text{Sat}_\sigma(\text{`}\forall x_1 \exists x_3(x_3 = Sx_1)\text{'})$ iff $\forall x_1 \exists x_3[x_3 = Sx_1]$

Finally, using (11) and substituting into (1), we have:

(12) $\text{Tr}(\text{`}\forall x_1 \exists x_3(x_3 = Sx_1)\text{'})$ iff $\forall\sigma\forall x_1 \exists x_3[x_3 = Sx_1]$

But now '$\forall\sigma$' is a vacuous quantifier, since 'σ' does not occur within the scope of '$\forall\sigma$'. So it can just be dropped, leaving us with:

(13) $\text{Tr}(\text{`}\forall x_1 \exists x_3(x_3 = Sx_1)\text{'})$ iff $\forall x_1 \exists x_3(x_3 = Sx_1)$

Voilá!

Derivation of the instances of schema (T) for sentences containing quantifiers are thus much more complicated than derivations for variable-free sentences: We shall always need to prove something like (8) or (10) in order to get rid of the quantifiers ranging over assignments which the clauses for '$\exists$' and '$\forall$' introduce. So one might wonder whether it always *will* be possible to prove the relevant results. To show that it will, we need to prove all instances of the following theorem schemata:

$\forall\tau[\tau \underset{v}{\sim} \sigma \rightarrow \phi^*]$ iff $\forall v(\phi)$

$\exists\tau[\tau \underset{v}{\sim} \sigma \wedge \phi^*]$ iff $\exists v(\phi)$

where ϕ^* is the result of replacing all occurences of v in ϕ by $val(\tau, v)$. That all instances of these two schemata can be proven in an appropriate meta-theory can itself be proven. We can thus show—though, again, the proof is far more complex than those at which we looked earlier—that the characterization of truth given above is indeed materially adequate.

[11] Given the right logical resources, there is an obviously shorter proof. But the present proof generalizes.

4 Definitions of Truth and Theories of Truth

There are two ways to think of what has gone on above. More specifically, there are two ways to think of the various 'clauses' that occur in the characterizations of denotation and truth, in Section 2, and of denotation under a sequence and satisfaction, in Section 3. On the one hand, one can think of these clauses as axioms that make substantive claims about the meanings of expressions in the object-language.[12] For example, the axiom "'0' denotes x iff $x = 0$" makes the non-trivial, substantive, but correct claim that the term '0', in the language of arithmetic, denotes the natural number zero. If we read the axioms this way, then what we did above was to give an axiomatic theory of truth for the object-language, treating the expressions 'denotes' and the like as primitive. This theory makes claims about the denotations, etc., of various expressions in the language of arithmetic. Tarski would not have regarded such a theory of truth as 'formally correct', since it uses semantic notions, such as denotation and satisfaction.

The other way to think about the clauses used above is as parts of a *definition* of the notions of denotation, satisfaction, and the like. The problem with thinking of them that way is that they simply don't *look* like parts of a definition. After all, a definition of denotation, say, would be expected to have the form:

the denotation of $t = \ldots t \ldots$,

where the right-hand side of the definition specifies, in terms of t, what its denotation is. Definitions of this familiar sort are called *explicit* definitions. But there are other kinds of definitions, among which are so-called *recursive* definitions. A recursive definition is one which proceeds by defining something first for a range of basic cases, and then giving a definition for more complex cases in terms of what has been said about the simpler cases. Indeed, the definitions of 'term' and 'formula' given above are recursive definitions: We said what the 'basic' terms and 'atomic' formulae were, then what would count as a 'non-basic' term or as a 'complex' formula.

Plainly, the clauses used above may be thought of in this way: We have said what denotation is for the most basic cases—of '0' and of variables—and then we have said what the denotation of a complex term

[12] Or stipulations about what their meanings are to be, if they do not have meaning already.

is in terms of the denotations of its simpler parts. Recursive definitions do not always work, but in the case of giving a recursive definition applying to expressions of a language (and in many other cases, such as recursive definitions of properties of natural numbers), they do. That they work in such cases may be proved by showing that, given sufficiently powerful mathematical machinery, one can convert the recursive definitions into explicit ones. The methods for doing this were developed (independently) by Dedekind and Frege, and the details of how it is done need not detain us. It suffices to note that it can be done.

How we think of the T-sentences we proved will depend dramatically upon how we think of the characterization of truth we gave. If we think of it in the first way, then we will think of the sentences as theorems of a theory that makes *substantive*, even *empirical* claims about what certain expressions mean in a particular language. The T-sentences then make substantive empirical claims themselves about that same language. What the theory, so construed, will do is make such claims as it does about the truth and falsity of sentences on the basis of claims about denotation and satisfaction. In that sense, the axiomatic theory of truth might be said to explain the semantic properties of complex expressions in terms of the semantic properties of their parts, and it is this sort of 'theory' of truth that is employed in actual semantic theorizing. In so far as Tarski contributed to the foundations of semantics, it was by showing how to formulate theories of truth, in this sense.

On the other hand, if we think of the characterization of truth in the second way, then the T-sentences will be proved from *definitions* and will, in fact, themselves turn out to be definitionally equivalent to theorems of pure mathematics and so will *be* theorems of pure mathematics. The T-sentences then *most certainly will not* make any substantive, empirical claims about anything, let alone the truth and falsity of sentences of some actual language.

Putting things this way makes it sound as if Tarski gave two completely different characterizations of truth—one an axiomatic theory, the other a definition—ran them together, and generally confused things completely. But that impression is mistaken. For the two characterizations are not as different as all that. The relationship between them is this: What the *definition* of truth shows is that it is possible to translate the *theory* of truth—let us call it $\mathcal{T}$—into the theory in which the definition of truth is given—let us call that theory $\mathcal{D}$—in such a way that all the axioms of $\mathcal{T}$—all the various clauses—become theorems of $\mathcal{D}$. This shows (to use the technical term) that it is possible to 'interpret' $\mathcal{T}$ in

$\mathcal{D}$, and it follows that $\mathcal{T}$ is consistent if $\mathcal{D}$ is: If it were possible to prove a contradiction in $\mathcal{T}$, the proof could be 'mimicked' or 'reproduced' in $\mathcal{D}$ by means of the translation. Hence, if the definition of truth is given in some mathematical theory of whose consistency we are reasonably certain, we can be equally certain of the consistency of $\mathcal{T}$. So one might say that what Tarski has done is, first, to produce a materially adequate *theory* of truth and then to produce a materially adequate and formally correct *definition* of truth, and the relation between these shows that the theory in question is consistent if the mathematical theory in which the definition of truth is stated is itself consistent.

And that's no mean feat!

5 Exercises

Exercise 1. Prove that '$S(SS0 + SS0)$' denotes $S(SS0 + SS0)$.

Exercise 2. Complete the argument on page 6, that every expression has a unique denotation, by showing that, if t and u have unique denotations, then so do '$t + u$' and '$t \times u$'.

Exercise 3. Complete the argument on page 7 by showing that, if the meta-theory proves

't' denotes w iff $w = t$

'u' denotes w iff $w = u$

then it also proves:

'$t + u$' denotes w iff $w = t + u$

'$t \times u$' denotes w iff $w = t \times u$

Exercise 4. Prove that '$S(SS0+SS0) < SSS0$' is true iff $S(SS0+SS0) < SSS0$.

Exercise 5. Fill in the details of the argument begun on page 8.

Exercise 6. (Optional) Prove the result mentioned in footnote 4 on page 8.

Exercise 7. Prove that '$(S(SS0 + SS0) < SSS0) \wedge S0 = S0$' is true iff $(S(SS0 + SS0) < SSS0) \wedge S0 = S0$.

Exercise 8. Prove that the characterization of truth for the propositional fragment of the language of arithmetic, given in 2.3, is materially adequate. (The argument will be similar to those in Exercises 2, 3, and 5, but will, of course, concern the propositional connectives. You can do only two of the cases for the binary connectives.)

Exercise 9. Prove that '$\forall x[\exists y(x = Sy) \rightarrow \exists y(y = Sx)]$' is true iff $\forall x[\exists y(x = Sy) \rightarrow \exists y(y = Sx)]$.

Exercise 10. (Optional) Prove that the characterization of truth for the langauge of arithmetic is materially adequate by proving the result mentioned at the end of Section 3 on page 17.