A FUNDAMENTAL FLAW IN AN INCOMPLETENESS PROOF BY STANISŁAW ŚWIERCZKOWSKI

James R Meyer
http://www.jamesrmeyer.com

v2 30 Apr 2015

Abstract

This paper examines a proof of incompleteness by Świerczkowski in a paper entitled "*Finite sets and Gödel's incompleteness theorems*".[10] Świerczkowski's proof is different to most other proofs of incompleteness because the formal system that is used for the proof is a system of hereditarily finite sets. Świerczkowski claims that this makes his proof simple and elegant, and that it has enabled his proof to be complete and without any gaps or omissions, nor relying on references to other publications. Świerczkowski claims that for this reason his proof his superior to most other proofs of incompleteness. However, the author makes an elementary error in his proof that renders the proof invalid.

Version History:

Version 2: After receiving comments to the effect that version 1 uses an incorrect interpretation of Świerczkowski's Lemma 6.1, Section 3 has been rewritten and expanded to include all possible interpretations of the lemma, with detailed explanations of the logical invalidity of the lemma, regardless of whatever interpretation is used. Also Section 4 has been expanded. The previous version can be accessed via *http://www.jamesrmeyer.com/sitemap.html* .

1 Introduction

Since Gödel published his original proof of incompleteness[1] over eighty years ago, there have been numerous attempts (see, for example Smullyan,[9] or Smith[8]) to provide proofs of incompleteness that are superior to Gödel's original proof. This paper examines one such attempt which is given by Świerczkowski in a paper entitled "*Finite sets and Gödel's incompleteness theorems*".[10]

2 Background to Świerczkowski's Proof

Most incompleteness proofs use a coding system that creates a unique one-to-one correspondence between the symbol combinations of a formal system and natural numbers. Given such coding, a correspondence is then established between relations between formulas of the formal system and relations between numbers. In Świerczkowski's proof the coding system creates a one-to-one correspondence between expressions of the formal system and certain finite sets, where these sets correspond to the notion of natural numbers. Świerczkowski's formal system will only be described briefly here, since most of the details are peripheral to the demonstration of the flaw in the proof. In any case, full details of the formal system can be found in Świerczkowski's paper.

2.1 Hereditarily Finite Sets

For convenience the Hereditarily Finite Set system that Świerczkowski uses as the formal system is referred to by the abbreviation HF. The objects of the formal system are hereditarily finite sets; HF uses a language that has 7 basic symbols and infinitely many variable symbols. The 7 basic symbols are 0, $\in$, $\triangleleft$, =, $\vee$, $\neg$ and $\exists$. Since there are infinitely many variables, a symbol system such as $x_1, x_2, x_3, \ldots$ can be used for the variables of the system.[a] Further details of the system will not be described here, as the details can be found in Świerczkowski's paper.

2.2 The Coding Function

One of the key ideas behind most incompleteness proofs is that we can form a correspondence between formulas of the formal system and certain objects; in this case the objects are finite sets. The coding is such so that for every relationship between formulas of the formal system that relationship can be mapped precisely to relationships between finite sets; so that if a certain relationship between formal system formulas applies, then there is corresponding relationship between the corresponding finite sets which also applies.

Świerczkowski refers to the system used in his proof to map the formal system formulas to finite sets as the 'coding'. This coding is a function which gives a one-to-one correspondence from symbol combinations of the formal system HF to finite sets, and this function is designated by $\ulcorner \quad \urcorner$. The function has one free variable, although Świerczkowski does not specify any symbol for this variable. For convenience we will refer to the function with the free variable Ω, which gives the function as $\ulcorner\Omega\urcorner$, where Ω is a variable of a meta-language to the system HF, and which has the domain of all symbol combinations of the formal system HF, including all variables of HF.

The proof of incompleteness involves the claim that there is a formula of the formal system that expresses the coding function in some way. However, in a consistent logical system there is no variable of the formal system that has the domain that includes all variables of the formal system, including itself. Hence, a claim that there is an expression of the formal system that contains the information of the definition of the coding function is a quite remarkable claim, since there is no variable of the formal system that is comparable to the free variable Ω of the function $\ulcorner\Omega\urcorner$. Because of this extraordinary claim, it is worth expending some effort examining the method by which the proof asserts that the system HF can express the coding function.

2.3 Domains of Variables and Substitutability

As noted above, since there are infinitely many variables in HF, a symbol system such as $x_1, x_2, x_3, \ldots$ is required. Świerczkowski uses the term k to refer to the k^{th} suffix in a series of suffixes $1, 2, 3, \ldots$, so that x_k refers to the k^{th} variable in a series of variables $x_1, x_2, x_3, \ldots$.

It is important to maintain the distinction between the domain of a variable and the expressions that may be substituted for a variable. Substitutability of a variable by an expression does not imply that the expression is a member of the domain of the variable. For example, in conventional arithmetic,

[a] For convenience, Świerczkowski often uses other symbols such as x, y, z, etc, to indicate some such variable of the system HF.

given the formula $x+3>14$, it is valid to substitute the x by another variable, for example y, to give an equivalent formula $y+3>14$. One can also substitute an expression such as $y-2$ for the variable x to give another valid formula $y-2+3>14$ (where the set of values of y that satisfy the equation are different to the set of values of x that satisfy the original formula). In these cases, the variable is simply replaced by another variable expression which, like the original variable, has no specific value. In contrast, the domain of a variable necessarily consists only of expressions that are not variables; only specific values can be members of the domain of a variable.[b]

While Świerczkowski does not actually define the domain of the variables of HF, it is evident that the domain of these variables is all specific sets of HF.

2.4 Σ-formulas

Here we give a very brief mention of Σ-formulas as the details are peripheral for our purposes. Essentially Σ-formulas are formulas that are necessarily decidable, given any specific values of their free variables. The purpose of the definition is similar to the purpose of the use of primitive recursion in most incompleteness proofs.

2.5 Definitions of relations of HF

In Section 3, Świerczkowski introduces a series of relations, which are so defined that if the free variables of the relations correspond by the coding system to symbol combinations of the system HF, then there is a corresponding relation between those symbol combinations in a meta-language that can express such relations. For example, Świerczkowski's first relation is $\mathrm{Var}(x)$, which corresponds to the assertion that X is a variable of the system HF, so that if $x = \ulcorner X \urcorner$, and if X is a variable of the system HF, then the relation $\mathrm{Var}(x)$ is provable in the system HF. Note that here X is a variable of the meta-language.

It should be noted that Świerczkowski often uses the coding term within expressions which are meant to represent expressions of the system HF; for example, the definition of the relation $\mathrm{SeqTerm}(s,k,t)$ is:

$$\mathrm{LstSeq}(s,k,t) \wedge \forall(l \in k)\{s_l = 0 \vee \mathrm{Var}(s_l) \vee \exists(m,n \in l)[s_l = \langle \ulcorner \triangleleft \urcorner, s_m, s_n \rangle]\}$$

where $\ulcorner \triangleleft \urcorner$ is not an expression of the system HF, but is a term in the meta-language that ***represents*** a unique specific expression in the system HF. As long as this distinction is always maintained and recognized, then there should be no confusion of meta-language and object language.

Since $\ulcorner \Omega \urcorner$ is not an expression of HF, and since it does it represent any expression of HF, it should not appear in any expression that is claimed to represent an expression of HF; there is no expression of HF for which $\ulcorner \Omega \urcorner$ is a valid substitution. However, as shown above, certain terms that include the coding function $\ulcorner\ \urcorner$, such as the expression $\ulcorner \triangleleft \urcorner$ may be substituted, *in the meta-language*, for expressions of HF, provided certain conditions are met, such as recognizing that the expression only represents an expression of HF, and is not itself an expression of HF.

[b] Note that in a meta-language to an object language, the variables of the *object* language can be members of the domain of a variable of the meta-language.

2.6 *p-functions*

In Section 5, Świerczkowski introduces the concept of a *p-function.*[c]

If φ is a formula of HF with a free variable y and n other free variables, then if $\exists! y\varphi$ is a theorem of HF ($\exists! y$ means that there exists one and only one such y), then there is an associated function that is represented by the term F^y_φ. Such functions can have any finite number of free variables, and the function and its variables are represented by $F^y_\varphi(z_1,\ldots,z_n)$, and the number of free variables of this function F^y_φ is one less than for the formula φ.

By adding the symbols of such *p-functions* to the language of HF, an expanded language is created. Although they are not formulas of HF, for any such *p-function* in this language, there is a corresponding formula of HF.

Recalling what was said in Section 2.3 above, we note that no additional specific objects are defined that can be members of the domain of a variable of this expanded language. Hence the domain of the variables of this expanded language is precisely the same as the domain of the variables of HF.

And as was noted in Section 2.3 above, under certain conditions, an expanded set of expressions can be validly substituted for a variable within this expanded language. A brief description of such substitutions as described by Świerczkowski follows below.

For the purposes of the expanded language, F^y_φ is called a *p-symbol* and any formula that includes the term F^y_φ is called a *p-formula*. A *p-term* is any term of the expanded language; *p-formulas* contain *p-terms*, and *p-formulas*, while not expressions of HF, represent expressions of HF.

Given an expression $F^y_\varphi(\tau_1,\ldots,\tau_n)$ of the expanded language, if $\tau_1,\ldots,\tau_n$ are substitutable only by members of the domain of variables of HF, then $\tau_1,\ldots,\tau_n$ are called simple *p-terms*; otherwise they are non-simple *p-terms*.

Świerczkowski describes a process that he calls '*reduction*', by which the number of non-simple *p-terms* in a *p-formula* can be reduced until there are only simple *p-terms* remaining in the expression.

For example, following Świerczkowski's description, suppose that α is a *p-formula* expression which has only one non-simple *p-term*. There is a corresponding formula φ of HF with the free variables x and y, which we write as $\varphi(x,y)$.

The simple *p-term* of the *p-formula* α occurs in some atomic sub-formula of α, and we call this atomic sub-formula $\beta(z)$. Then there is a simple *p-term* τ such that $\beta(F^y_\varphi(\tau))$ is a sub-formula of α. The reduction process is the replacement of this atomic sub-formula $\beta(F^y_\varphi(\tau))$ by:

$$\exists w\beta(w) \wedge \widetilde{\varphi}(\tau,w),$$

where w is a variable that does not occur in α nor τ, and where $\widetilde{\varphi}$ is equivalent to φ but such that τ and w are *substitutable* for x and y in $\widetilde{\varphi}$.[d] The result of this reduction is a formula with one free variable which is *substitutable* only by members of the domain of variables of HF.

As noted above, this process of reduction does not affect the domain of the variables in the formulas, which remains as all specific sets of HF.

[c] Świerczkowski also gives more details of *p-functions* in his Appendix 3: *p-functions.*

[d] This means that τ and w can be substituted in $\widetilde{\varphi}$ because there are no instances already of τ and w in $\widetilde{\varphi}$ such that the substitution would result in additional instances of τ or w that would alter the logical content of the formula φ.

3 Świerczkowski's Proof of Incompleteness

Throughout his paper, Świerczkowski exhibits a careless disregard for the distinction between meta-language and object-language. The point at which this culminates in a logical absurdity occurs in Świerczkowski's Section 6. Świerczkowski's Lemma 6.1 asserts:

There is a p-function W such that
$\vdash W(\ulcorner x_k \urcorner) = \ulcorner(\ulcorner x_k \urcorner)\urcorner$
for every variable x_k.

It has been suggested to me that one could interpret Świerczkowski's Lemma 6.1 in different ways. It seems that the reason why different interpretations are suggested is that in Świerczkowski's proof of the lemma, the entities $x_1, x_2, x_3, \ldots$ are, on the one hand, constants to which the coding function applies, but, on the other hand, the entities $x_1, x_2, x_3, \ldots$ are variables of *p-functions* and the system HF. This anomaly gives rise to attempts to provide a viable interpretation of the lemma. While some of these interpretations may appear to the reader as obviously invalid, what may appear obvious to one reader may not appear so to another, and so, for the sake of completeness, an analysis of each interpretation will be given in the following pages. Świerczkowski's proof of his Lemma 6.1 will be considered in Section 3.7.

N.B: in the following, for clarity, variables of the meta-language are represented in upper case letters, while variables of the system HF are represented by lower case letters.

Interpretation 1:

There is a p-function W such that, $\forall X_K$, (3.1.a)
$\vdash W(\ulcorner X_K \urcorner) = \ulcorner(\ulcorner X_K \urcorner)\urcorner$ (3.1.b)

where the X_K is a variable in the meta-language, and whose domain is the variables $x_1, x_2, x_3, \ldots$ of the HF system.

Interpretation 2:

There is a p-function W such that (3.2.a)
$\vdash \forall x_k, W(\ulcorner x_k \urcorner) = \ulcorner(\ulcorner x_k \urcorner)\urcorner$ (3.2.b)

which asserts that the system HF can prove the proposition:

$$\forall x_k, W(\ulcorner x_k \urcorner) = \ulcorner(\ulcorner x_k \urcorner)\urcorner$$

Interpretation 3:

There is a p-function W such that, $\forall K$, (3.3.a)
$\vdash W(\ulcorner x_K \urcorner) = \ulcorner(\ulcorner x_K \urcorner)\urcorner$ (3.3.b)

where the K is a variable in the meta-language, and whose domain is $1, 2, 3, \ldots$.

Interpretation 4

There is a p-function W such that, $\forall K$, (3.4.a)

$$\vdash \forall x_K, W(\ulcorner x_K \urcorner) \;=\; \ulcorner(\ulcorner x_K \urcorner)\urcorner \tag{3.4.b}$$

where K is a variable in the meta-language whose domain is $1, 2, 3, \ldots$, hence the values that x_K can take are the variables $x_1, x_2, x_3, \ldots$ of the HF system.

Interpretation 5:

There is a p-function W such that, $\forall X_K$, (3.5.a)

$$\vdash \forall X_K, Condition(X_K) \implies W(X_K) \;=\; \ulcorner X_K \urcorner \tag{3.5.b}$$

where $Condition(X_K)$ is the condition that X_K is a certain value, and that value is $X_K \;=\; \ulcorner X_K \urcorner$.

Here X_K is a variable in the meta-language, and whose domain is the variables $x_1, x_2, x_3, \ldots$ of the HF system.

Interpretation 6:

There is a p-function W such that, $\forall V, \exists U, \exists K$ (3.6.a)

$$\vdash W(V) \;=\; U \tag{3.6.b}$$

if V is the code of a variable of HF, i.e., if $V \;=\; \ulcorner x_K \urcorner$ *for some* K, *and where* $U \;=\; \ulcorner V \urcorner$ (3.6.c)

We shall now analyze each interpretation. We note that in each of the interpretations apart from Interpretation 6, in the expression for which the assertion is that the system HF can prove that expression, the right-hand side of the expression is the coding function. The coding function $\ulcorner \Omega \urcorner$ is defined as a function of the meta-language, where its free variable Ω has the domain of all expressions of the system HF, and there is no variable of the system HF that has this domain. It follows straightaway that each of the interpretations 1-5 is logically invalid, since the system HF cannot prove an expression that cannot be an expression in the system HF. However, we shall nonetheless analyze these interpretations in order to demonstrate how this confusion of language results in logical contradiction.

3.1 Interpretation 1

We now consider the interpretation given by 3.1.a and 3.1.b above. This asserts that the system HF can prove the proposition:

$$W(\ulcorner X_K \urcorner) = \ulcorner(\ulcorner X_K \urcorner)\urcorner$$

for any instance of X_K, that is for any one of $x_1, x_2, x_3, \ldots$.

As noted above, the right-hand side of the expression is the coding function, which is an expression of the meta-language, and hence the system HF cannot prove the expression. Besides this flaw, we note that under this interpretation, for any instance of X_K, the Lemma 6.1 results in an expression, from 3.1.b above, with a free variable, since there is no quantifier on the variable. Expressions of the system HF that have free variables cannot be provable in the system HF, since such expressions are not propositions in the system HF. It follows that this proposed interpretation is not logically valid.

e.g. for X_K as x_3, we have $W(\ulcorner x_3 \urcorner) = \ulcorner(\ulcorner x_3 \urcorner)\urcorner$, which is not a proposition. When a valid value is substituted for x_3, the result is a proposition.

Alternatively, one might suppose that the intention was:

$\vdash W(\ulcorner x_1 \urcorner) = \ulcorner(\ulcorner x_1 \urcorner)\urcorner$
and
$\vdash W(\ulcorner x_2 \urcorner) = \ulcorner(\ulcorner x_2 \urcorner)\urcorner$
and
$\vdash W(\ulcorner x_3 \urcorner) = \ulcorner(\ulcorner x_3 \urcorner)\urcorner$
and ...

where $x_1, x_2, x_3, \ldots$ are not variables in the above expressions.

But the problem with this is that if $x_1, x_2, x_3, \ldots$ are not variables, then any expressions that include $x_1, x_2, x_3, \ldots$ as non-variables cannot be expressions of HF or of *p-functions*, since, by definition, wherever $x_1, x_2, x_3, \ldots$ occur in expressions of HF or of *p-functions*, they are necessarily variables; they are not members of the domain of any of the variables of HF or of *p-functions*. And if an expression cannot be an expression of HF or of *p-functions*, then it cannot be proved within HF.

3.2 Interpretation 2

Next we consider the interpretation given by 3.2.a and 3.2.b above. This asserts that the system HF can prove the proposition:

$$\forall x_k, W(\ulcorner x_k \urcorner) = \ulcorner(\ulcorner x_k \urcorner)\urcorner$$

As noted above, the right-hand side of the expression is the coding function, which is an expression of the meta-language, and hence the system HF cannot prove the expression. Besides this obvious flaw, we note that while one could define a function W to be a *p-function*, the function $W(\ulcorner x_k \urcorner)$ is a *composite* of the W function and the $\ulcorner\ \urcorner$ function. But $\ulcorner x_k \urcorner$ cannot be a *p-function*, since its free variable x_k has the domain of all variables of HF, that is, $x_1, x_2, x_3, \ldots$, and as explained above in Section 2.6, no variable of HF or of the expanded language of *p-functions* has this domain, otherwise it would include itself in its own domain, and it would be a variable and an object of the language at the same time, which is a logical absurdity. Hence the free variable x_k is not a variable of HF, nor can it be a variable of the expanded language of *p-functions*. It follows that $\ulcorner x_k \urcorner$ cannot be a *p-function*, and the composite function $W(\ulcorner x_k \urcorner)$ also cannot be a *p-function*. The objective in the definition of *p-functions* is that, given any *p-function*, there is a corresponding formula of the system HF. But the expression $W(\ulcorner x_k \urcorner) \;=\; \ulcorner(\ulcorner x_k \urcorner)\urcorner$ includes expressions that are not *p-functions*. Hence there cannot be any expression in HF corresponding to $\ulcorner x_k \urcorner$ or $\ulcorner(\ulcorner x_k \urcorner)\urcorner$, and hence there cannot be a proof in HF regarding such expressions.

3.3 Interpretation 3

Next we consider the interpretation given by 3.3.a and 3.3.b above. This asserts that the system HF can prove the proposition:

$$W(\ulcorner x_K \urcorner) \;=\; \ulcorner(\ulcorner x_K \urcorner)\urcorner$$

for any instance of K.

It can be seen that this interpretation gives the same result as Interpretation 1, and similarly is logically invalid.

3.4 Interpretation 4

We now consider the interpretation given by 3.4.a and 3.4.b above. This asserts that the system HF can prove, for any instance of K, the proposition:

$$\forall x_K, W(\ulcorner x_K \urcorner) \;=\; \ulcorner(\ulcorner x_K \urcorner)\urcorner$$

This interpretation is similar to the preceding two interpretations, except that it is assumed that the expression to be proved includes a quantifier. Again, as noted above, the right-hand side of the expression is the coding function, which is an expression of the meta-language, and hence the system HF cannot prove the expression.

Besides this flaw, in this interpretation, the $x_1, x_2, x_3, \ldots$ are variables of *p-functions* and as such have the domain of HF sets (since, as previously noted, the variables of *p-functions* have the same domain as the variables of HF). However, according to the definition of the coding function $\ulcorner\;\urcorner$, if the variable takes a value that is an expression of the HF language, then the coding function gives a value that is dependent on that expression of HF. And since $x_1, x_2, x_3, \ldots$ are variables of HF, then the coding function $\ulcorner x_1 \urcorner$ or $\ulcorner x_2 \urcorner$ or $\ulcorner x_3 \urcorner, \ldots$ by definition must give a singular value that corresponds to the variable x_1 or x_2 or $x_3, \ldots$.

But, on the other hand, if any one of $x_1, x_2, x_3, \ldots$ is a variable of the coding function $\ulcorner\;\urcorner$, then it must be the case that the variable can take any value of the domain of the variable, and the value of the coding function depends on that value, which contradicts the fact that the value of $\ulcorner x_1 \urcorner$ or $\ulcorner x_2 \urcorner$ or $\ulcorner x_3 \urcorner, \ldots$ is a singular value determined by x_1 or x_2 or $x_3, \ldots$.

3.5 Interpretation 5

Now we consider the interpretation given by 3.5.a and 3.5.b above. This asserts that the system HF can prove the proposition:

$$\forall X_K, Condition(X_K) \implies W(X_K) = \ulcorner X_K \urcorner$$

where $Condition(X_K)$ is the condition that X_K is a certain value.

Here X_K is a variable of the meta-language. As noted previously, here the left-hand side represents an expression of HF, but the right-hand side does not; the coding function is defined as a function of the meta-language and hence the expression $\forall X_K, Condition(X_K) \implies W(X_K) = \ulcorner X_K \urcorner$ is not an expression of the system HF for any value of X_K, and so cannot be proved in the system HF.

Besides this logical flaw, in this interpretation, if we now consider the right-hand side of the above expression, which is $\ulcorner X_K \urcorner$, we see that since the domain of X_K is a variable of HF, then it must be one of $x_1, x_2, x_3, \ldots$. Now, according to the definition of the coding function, since $x_1, x_2, x_3, \ldots$ are symbol strings of HF, each $\ulcorner x_1 \urcorner$, $\ulcorner x_2 \urcorner$, $\ulcorner x_3 \urcorner$, ... is a fixed value that is completely determined by the definition of the coding function. For example, for the instance of the variable X_K as x_1, we have:

$$\vdash \forall x_1, Condition(x_1) \implies W(x_1) = \ulcorner x_1 \urcorner$$

Hence the value of the right-hand side of the proposition is a constant that is dependent on whichever one of $x_1, x_2, x_3, \ldots$ is the free variable X_K in the purported *p-function* $W(X_K)$. Since there is no valid *p-function* whose value is dependent only on the choice of the symbol used for its free variable, it follows that there can be no *p-function* W that satisfies this interpretation.

3.6 Interpretation 6

Finally, we consider the interpretation given by 3.6.a, 3.6.b and 3.6.c above. This asserts that the system HF can prove the proposition:

$$W(V) = U$$

whenever there is some x_K such that $V = \ulcorner x_K \urcorner$, and where $U = \ulcorner V \urcorner$, i.e., $U = \ulcorner(\ulcorner x_K \urcorner)\urcorner$.

Here V and U are variables of the meta-language. In this interpretation, it is the only the meta-language that refers to the coding function. The system HF does not itself actually refer to the coding function, and hence in this interpretation the system HF cannot be asserted to state anything whatsoever regarding the coding function. All that this interpretation is asserting is that the system HF proves that the value of $W(V)$ is some value of the system HF; the system HF is not proving anything regarding the value of $W(V)$ in relation to the coding function. It is the *meta-language* that is making assertions regarding the value of $W(V)$ and the coding function. Since Świerczkowski's subsequent Lemma 6.2, Lemma 6.3, Theorem 6.4, Theorem 6.5 all rely on Lemma 6.1's assertion that the system HF proves a relationship between the value of $W(V)$ and the coding function, this interpretation renders all these subsequent lemmas and theorems invalid. So while this interpretation, unlike the previous interpretations, is not logically invalid, it is so far removed from the actual wording of the lemma that it fails to provide the desired result - that the system HF can prove expressions that include the coding function of the meta-language.

The above analyses establish that none of the above interpretations of Świerczkowski's Lemma 6.1 can demonstrate that the system HF can prove an assertion regarding the coding function of the meta-language.

3.7 Świerczkowski's proof of Lemma 6.1

Świerczkowski's proof of Lemma 6.1 is a proof by induction on the value of K.[e] In this proof, the entities $x_1, x_2, x_3, \ldots$ are taken as not being variables, since in the proof the value of the coding function $\ulcorner\ \urcorner$ is taken as the value for the expressions of HF that are $x_1, x_2, x_3, \ldots$. However, if $W(\ulcorner x_1 \urcorner)$, $W(\ulcorner x_2 \urcorner)$, $W(\ulcorner x_3 \urcorner)$, ... are *p-functions*, then (as referred to in Section 2.6) the domain of the variables of *p-functions* must be the same as the domain of the variables of HF. Since $x_1, x_2, x_3, \ldots$ are variables of HF, they cannot be members of the domain of any variable of a *p-function*, and must be variables of *p-functions*. But if that is the case, then for any member of the domain of any such $x_1, x_2, x_3, \ldots$ that is substituted for the variable, the coding function must then give a different value, which contradicts the assumption in the proof that the value is determined by the entity $x_1, x_2, x_3, \ldots$ itself, and not on a substituted value of the entity $x_1, x_2, x_3, \ldots$.

This is no trivial error. It is a fundamental error for which there is no simple resolution, since, as shown above, there is no interpretation of the Lemma 6.1 that is logically valid and which gives the desired result. Since the remainder of Świerczkowski's paper relies on the Lemma 6.1, Świerczkowski's 'proof' of incompleteness fails and his paper is fatally flawed.

It might be noted that it is a simple matter to unearth the reason why Świerczkowski's Lemma 6.1 (and Świerczkowski's Lemma 6.3) is logically untenable in any interpretation. If one considers the entire basis for coding, where a relation between HF sets can be made to correspond to a relation between expressions of HF, this is done by the correspondence being enabled through the coding function, so that if U and V are expressions of HF, and u and v are hereditary finite sets, then if $u = \ulcorner U \urcorner$ and $v = \ulcorner V \urcorner$, then there can be a corresponding relation between the hereditary finite sets u and v. In the case of Lemma 6.1, this means that a relation is asserted to exist between, on the left-hand side, a u that corresponds to some HF expression U by $u = \ulcorner U \urcorner$, and on the right-hand side, a v which corresponds to some HF expression V by $v = \ulcorner V \urcorner$.

It can readily be seen that $u = \ulcorner U \urcorner = \ulcorner x_k \urcorner$, and hence $U = x_k$ but on the right-hand side we have that $v = \ulcorner V \urcorner = \ulcorner(\ulcorner x_k \urcorner)\urcorner$, which gives that $V = \ulcorner x_k \urcorner$.[f] However, for the correct correspondence through coding, both U and V must be expressions of HF. But $\ulcorner x_k \urcorner$ is an expression of the meta-language, not of HF. It is this fundamental confusion of language that determines that Świerczkowski's Lemma 6.1 cannot be a logically valid assertion. The same erroneous assumption that the system HF can prove expressions that include the meta-language coding function occurs in several other places in Świerczkowski's paper.

For example, Świerczkowski's Proposition 5.1 is an assertion that the system HF can prove the expression $\mathrm{REPL}(\ulcorner x_i \urcorner, \ulcorner \tau \urcorner, \ulcorner \phi \urcorner) = \ulcorner x_i/\tau \urcorner$, but this expression includes several instances of the coding function, which is a function of the meta-language. The correct assertion is that, given a formula ϕ of HF with one free variable x_1, and a constant term of HF τ, then if ψ is the formula that results from the substitution of the free variable x_1 by τ, then that defines a relationship between x_1, ϕ, τ and ψ, and it can be asserted *in the meta-language* that by the coding function, there is a corresponding relationship between the hereditary finite sets u, v, w, and x, where $u = \ulcorner x_1 \urcorner$, $v = \ulcorner \tau \urcorner$, $w = \ulcorner \phi \urcorner$ and $x = \ulcorner psi \urcorner$. The values u, v, w, and x are values in the meta-language, and hence the relationship between u, v, w, and x is a relationship in the meta-language, but it can be asserted that there is a formula in the system HF that *corresponds* to that meta-language relationship. It might be noted that Świerczkowski's REPL function has essentially the same role as the Sb function[g] in Gödel's proof.[1]

[e] Note that K is not a variable of HF.

[f] Or that $U = x_1$, and $v = \ulcorner(\ulcorner x_1 \urcorner)\urcorner$, which gives that $U = x_1$ and $V = \ulcorner x_1 \urcorner$, which demonstrates the same confusion of levels of language.

[g] Relation 31 in Gödel's paper.

3.8 Common Notation for the meta-language and the object-language

In Świerczkowski' proof (and many other similar proofs), it is asserted that there is an expression of HF that proclaims the impossibility of its own proof, and of its negation. If that is so, then one might expect that that expression of HF would be completely independent of any meta-language. However, it is observed that in Świerczkowski's proof, the symbols used for the notation of hereditary finite sets in the meta-language are precisely the same as those used for HF (i.e., 0, $\in$, $\triangleleft$). In fact, if different symbols were to be used for the notation of hereditary finite sets in the meta-language, Świerczkowski's Lemma 6.1 is immediately seen to be a confusion of levels of language, since the coding function is a function whose free variable has the domain only of expressions of HF, but its value is always a hereditary finite set of the meta-language - hence in $\ulcorner(\ulcorner\;\;\urcorner)\urcorner$ the value of the 'inner' coding function cannot be a valid value for the 'outer' coding function.

The same use of a common notation applies to many other incompleteness proofs, where the same notation is assumed for numbers in both the object-language and the meta-language.

4 Conclusions

Świerczkowski claims that his paper is unique in that it can "*present all arguments completely, without omissions . . . such a degree of completeness in proving Gödel's results has never been attained before. . . . all [other] published proofs of Gödel's incompleteness theorems contain gaps, omissions or references to key results in other publications.*" But as in so many other incompleteness proofs[2] [5] [8] [9] the facade of detail hides a very simple unfounded crucial assumption, which invariably results in the formal system appearing to self-reference when a fully detailed logical analysis shows nothing of the sort.

It can be observed that, in common with Gödel's original proof, and most other incompleteness proofs, Świerczkowski's coding function maps expressions of the formal system to hereditary finite sets which ostensibly are objects of the meta-language, but which are, rather conveniently, expressions that are precisely in the same format in this meta-language as the sets of the formal system HF. And in common with the authors of the aforementioned proofs, Świerczkowski offers no explanation as to why this convenient identical format is used, rather than some other format being used for hereditary finite sets in the meta-language.

Lawrence Paulson has published some papers claiming a machine assisted proof of incompleteness that is based on Świerczkowski's paper. In one paper Paulson gives an overview[6] and in another the computer code.[7] Errors in other claims of computer checked proofs of incompleteness have been observed.[3] [4] As with Świerczkowski's proof, Paulson's proof fails to make a clear distinction between when an expression is intended to *represent* an expression of HF, and when it is itself actually an expression of HF; the result, as for Świerczkowski's proof, is an illogical confusion of meta-language and the language of HF. And, as for Świerczkowski's proof, Paulson's computer proof also relies on assuming the same notation for HF sets and for hereditary finite sets of the meta-language. Paulson does not include any computer code that proves that this cannot result in an illogical confusion of meta-language and object-language. Of course, it would be a simple matter to change the computer code to apply a different notation for HF sets and meta-language sets, but clearly the reason that was not done was because the proof could not proceed without that common notation for HF and the meta-language.

Paulson's computer assisted proof follows Świerczkowski's proof in including the assumption that such common notation cannot result in a loss of independence of the meta-language and the object-language, and that it cannot result in an illogical confusion of meta-language and object-language. This assumption is not stated anywhere either in Paulson's description or in his computer code, but if it is a fully formal proof (as Paulson claims his proof to be) then that assumption should not be implicit but should be an explicit statement, and included as part of the computer code.

References

[1] GÖDEL, K. *Über formal unentsceidebare Sätze der Principia Mathematica und verwandter Systeme I.* Monatshefte für Mathematik und Physik *38* (1931), 173–198.

[2] LINDSTROM, P. *Lecture Notes: Aspects of Incompleteness.* Springer-Verlag, 1997. ISBN: 3540632131.

[3] MEYER, J. R. *An Error in a Computer Verified Proof of Incompleteness by John Harrison*, 2011. http://www.jamesrmeyer.com/pdfs/ff_harrison.pdf.

[4] MEYER, J. R. *An Error in a Computer Verified Proof of Incompleteness by Russell O'Connor*, 2011. http://www.jamesrmeyer.com/pdfs/ff_oconnor.pdf.

[5] MOSTOWSKI, A. *Sentences Undecidable in Formalized Arithmetic.* Greenwood Press, 1982. ISBN: 9780313231513.

[6] PAULSON, L. C. *A Machine Assisted Proof of Gödel's Incompleteness Theorems for the Theory of Hereditarily Finite Sets.* The Review of Symbolic Logic (2013), 1–15. http://www.cl.cam.ac.uk/~lp15/papers/Isabelle/Goedel-logic.pdf.

[7] PAULSON, L. C. *Gödel's Incompleteness Theorems (Machine Code)*, Nov, 2013. http://afp.sourceforge.net/browser_info/current/AFP/Incompleteness/document.pdf.

[8] SMITH, P. *An introduction to Gödel's theorems*, Second ed. Cambridge University Press, 2013. ISBN: 9781107022843.

[9] SMULLYAN, R. M. *Gödel's Incompleteness Theorems.* Oxford University Press, 1992. ISBN: 0195046722.

[10] ŚWIERCZKOWSKI, S. *Finite sets and Gödel's incompleteness theorems*, vol. 422. Polska Akademia Nauk, Instytut Matematyczny, 2003. http://journals.impan.gov.pl/cgi-bin/dm/pdf?dm422-0-01.