MATH 318/Fall, 2007
Notes on reading formulas and Skolem functions

1. The Tarski machine

When we are given a sentence Φ in predicate logic, and a structure M interpreting the relation and operation symbols in Φ, we can ask whether or not M satisfies Φ; in symbols $M \vDash \Phi$? In case M is a finite structure, that is, its underlying set can be listed in a finite sequence, the question can be decided in an algorithmic manner. The algorithm that does the job of deciding if $M \vDash \Phi$ is true is based on Alfred Tarski's 1935 elucidation of the notion of truth, that is, of the notion of an arbitrary (not necessarily finite) structure M satisfying a sentence Φ of predicate logic; for this reason, we call it the *Tarski machine*.

Briefly put, the work of the algorithm consists of making up *tables*, one for each *subformula* of the given sentence Φ. A *subformula* of a formula is a consecutive part of the formula which itself is a formula; the formula itself is regarded as a subformula of itself. Each subformula has a set of free variables; this set may be empty (when the subformula in question is a sentence; for instance, when it is Φ itself). The *table* corresponding to the subformula gives the truth-value ($\top$ or $\bot$) of the subformula corresponding to each assignment of values of the free variables in the underlying set of the structure. The construction of the tables is *recursive*: for a subformula other than an atomic one, the calculation of the corresponding table relies on one or more earlier tables.

Consider the following example:

$$\Phi :=: \qquad \forall x \exists y \forall z (Rzy \longleftrightarrow \forall u (Ruz \rightarrow Rux)) . \tag{1}$$

We list the subformulas, with their free variables:

1	Rux	u, x
2	Ruz	u, z
3	$Ruz \rightarrow Rux$	u, x, z
4	$\forall u(Ruz \rightarrow Rux)$	x, z
5	Rzy	z, y
6	$Rzy \longleftrightarrow \forall u(Ruz \rightarrow Rux)$	x, z, y
7	$\forall z(Rzy \longleftrightarrow \forall u(Ruz \rightarrow Rux))$	x, y
8	$\exists y \forall z(Rzy \longleftrightarrow \forall u(Ruz \rightarrow Rux))$	x
9	$\forall x \exists y \forall z(Rzy \longleftrightarrow \forall u(Ruz \rightarrow Rux))$	$\varnothing$

With any given structure $M=(A; R)$, the calculation of the truth-value of $M \vDash \Phi$ will consist of making up nine tables, one for each of the nine formulas above. Here is the calculation when $A=\{1, 2\}$, and $R=\{(1, 1), (1, 2), (2, 2)\}$:

$$R : \qquad \begin{tikzcd} 1 \arrow[loop left] \arrow[r] & 2 \arrow[loop right] \end{tikzcd}$$

1 Rux u x

Rux	u	x
$\top$	1	1
$\top$	1	2
$\bot$	2	1
$\top$	2	2

2

Ruz	u	z
$\top$	1	1
$\top$	1	2
$\bot$	2	1
$\top$	2	2

3

$Ruz\rightarrow Rux$	u	x	z
$\top$	1	1	1
$\top$	1	1	2
$\top$	1	2	1
$\top$	1	2	2
$\top$	2	1	1
$\bot$	2	1	2
$\top$	2	2	1
$\top$	2	2	2

4

$\forall u(Ruz\rightarrow Rux)$	x	z
$\top$	1	1
$\bot$	1	2
$\top$	2	1
$\top$	2	2

5

Rzy	z	y
$\top$	1	1
$\top$	1	2
$\bot$	2	1
$\top$	2	2

6

$Rzy\longleftrightarrow\forall u(Ruz\rightarrow Rux)$	x	z	y
$\top$	1	1	1
$\top$	1	1	2
$\top$	1	2	1
$\bot$	1	2	2
$\top$	2	1	1
$\top$	2	1	2
$\bot$	2	2	1
$\top$	2	2	2

7

$\forall z(Rzy\longleftrightarrow\forall u(Ruz\rightarrow Rux))$	x	y
$\top$	1	1
$\bot$	1	2
$\bot$	2	1
$\top$	2	2

8

$\exists y\forall z(Rzy\longleftrightarrow\forall u(Ruz\rightarrow Rux))$	x
$\top$	1
$\top$	2

9 $\forall x\exists y\forall z(Rzy\longleftrightarrow\forall u(Ruz\rightarrow Rux)) = \top$.

The tables **1** , **2** and **5** correspond to atomic formulas, and they are directly drawn from the data for R . Table **3** is based on tables **1** and **2** . To express the dependence, we may write

$$\mathbf{3}(u, x, z) = \mathbf{2}(u, z) \rightarrow \mathbf{1}(u, x) \ ;$$

$\rightarrow$ is the usual operation on the truth-values $\top$ and $\bot$.

Table **4** depends on table **3** ; $\mathbf{4}(x, z) = \forall u\mathbf{3}(u, x, z)$. The calculation of the value of the latter expression, for any given values of x and z , is a *search* for some u for which $\mathbf{3}(u, x, z)=\bot=False$. If this "search for $\bot$ " is *successful*, that is, we do find some u for which $\mathbf{3}(u, x, z)=\bot=False$, then we put

$$\mathbf{4}(x, z) = \forall u\mathbf{3}(u, x, z) = \bot \quad (!);$$

otherwise,

$$\mathbf{4}(x, z) = \forall u\mathbf{3}(u, x, z) = \top .$$

In our example, the "search for $\perp$ " is successful just in case $x=1$ and $z=2$; the u for which $\mathbf{3}(u, 1, 2)=\perp$ is $u=2$: $\mathbf{3}(2, 1, 2)=\perp$. This says that one of the four lines in table **4** , the one for $x=1$, $z=2$ has the entry $\perp$, all other lines contain $\top$.

Table **6** is obtained like table **3** , with the operation $\longleftrightarrow$ in place of $\rightarrow$:

$$\mathbf{6}(x, z, y) = \mathbf{5}(z, y) \longleftrightarrow \mathbf{4}(z, x) .$$

Table **7** is another application of the universal quantifier:

$$\mathbf{7}(x,y) = \forall z\mathbf{6}(x,z,y) \ ;$$

its calculation is like that of table **4** .

Table **8** uses the *existential quantifier*:

$$\mathbf{8}(x) = \exists y\mathbf{7}(x, y) .$$

The calculation, for any given value of x . is again a *search*, but now for a value $\top$, rather than $\perp$: we look for some y for which $\mathbf{7}(x, y) = \top$. In our case, for all (both) values of x , that is, $x=1$ and $x=2$, such y is indeed found; in fact, $y=1$ when $x=1$ and $y=2$ when $x=2$. We get that $\mathbf{8}(1) = \mathbf{8}(2) = \top$.

Finally, $\mathbf{9} = \forall x\mathbf{8}(x)$; and since the search for x for which $\mathbf{8}(x)=\perp$ is unsuccessful, we get $\mathbf{9}=\top$, which is to say that, in this case, $M\vDash\Phi$.

In some ideal sense, we may say that the calculation of the truth-value of $M\vDash\Phi$ in case of an *infinite* structure $M=(A;R)$ is similar to the one above, with "infinite tables" corresponding to subformulas, with entries given by the values of some free variables ranging over the infinite set A (e.g., $A=\mathbb{N}$, or even $A=\mathbb{R}$). However, when the set A is infinite, it cannot be fully searched for appropriate values, to calculate the tables like **4** , **7** , **8** , **9** above, although special tricks, or specific knowledge may enable us to "fill" the tables. In fact, the difficulty of carrying out the Tarski algorithm obviously increases with the size (cardinality) of A even if that size remains finite.

§2. Skolem functions

Consider the simple sentence $\forall x \exists y Rxy$, and the structure $(\mathbb{N}; <)$, with $<$ the usual irreflexive order to interpret R. Of course, we have $(\mathbb{N}; <) \models \forall x \exists y Rxy$. When asked why, we will point out that, for each $x \in \mathbb{N}$, $y = x+1$ will work, since $x < x+1$. What we have done here is pointing out a *Skolem function* $y = f(x) = x+1$ for the existential quantifier $\exists y$ in the structure $(\mathbb{N}; <)$ for $\forall x \exists y Rxy$.

We may introduce the Skolem function formally. From the sentence $\forall x \exists y Rxy$, we pass to the sentence $\forall x R(x, fx)$, with a new unary operation symbol f. We are arguing for the fact that $(\mathbb{N}; <) \models \forall x \exists y Rxy$ by saying that $(\mathbb{N}; <, f) \models \forall x R(x, fx)$, where the function $f : \mathbb{N} \longrightarrow \mathbb{N}$ is defined by putting $f(x) = x+1$.

Still staying with the sentence $\forall x \exists y Rxy$ and its *Skolem form* $\forall x R(x, fx)$, we have that for any structure $M = (A; R)$,

(2) $(A; R) \models \forall x \exists y Rxy$ if and only if
there exists function $f : A \to A$ for which $(A; R, f) \models \forall x R(x, fx)$.

Having a particular function $f : A \to A$ with the property that $(A; R, f) \models \forall x R(x, fx)$ is to have more information than merely knowing the fact that $(A; R) \models \forall x \exists y Rxy$.

The Skolem function is by no means unique. In our example, we could have taken the function $f : \mathbb{N} \to \mathbb{N}$ for which $f(x) = x+2$, or indeed, a large number of other functions as well.

When one *reads* a sentence Φ with some existential quantifiers in a structure M, and one finds that $M \models \Phi$ is indeed the case, one usually comes up with particular Skolem functions corresponding to the existential quantifiers in Φ. We have to clarify a general formalism for Skolem functions, to make this precise.

We first start with defining a particular class of formulas, the ones that are in *Negation Normal Form* (NNF). A formula is in NNF, if two conditions are met. **One** is that it is built up using the logical operators $\wedge$, $\vee$, $\neg$, $\forall$ and $\exists$ only, without using $\rightarrow$ and $\longleftrightarrow$; the other is that negation, $\neg$, is used only in a constrained manner only, namely only in front of atomic subformulas in the formula.

The formula Φ in (1) is certainly not in NNF, since it violates the first condition. However, there is a simple way of producing a formula in NNF which is logically equivalent to any given formula Φ; we call the result the Negation Normal Form of Φ, and denote it by NNF(Φ). We show this on the example (1). We have

$$\forall x \exists y \forall z (Rzy \longleftrightarrow \forall u (Ruz \rightarrow Rux)) \equiv$$

$$\forall x \exists y \forall z [(\neg Rzy \vee \forall u (Ruz \rightarrow Rux)) \wedge (Rzy \vee \neg \forall u (Ruz \rightarrow Rux))]; \qquad (3)$$

for this logical equivalence, we used that

$$\begin{aligned} A \longleftrightarrow B &\equiv (A \rightarrow B) \wedge (B \rightarrow A) \\ &\equiv (\neg A \vee B) \wedge (\neg B \vee A) \equiv (\neg A \vee B) \wedge (A \vee \neg B) . \end{aligned}$$

In the formula in (3), we have a negation, $\neg$, which is in front of a non-atomic formula, namely $\forall u(Ruz \rightarrow Rux)$. When one negates a formula that starts with a universal quantifier, one obtains one that starts with an existential quantifier, according to the logical equivalence $\neg\forall x\Psi \equiv \exists x\neg\Psi$. Accordingly, from (3) we can continue as in

$$\equiv \forall x\exists y\forall z[(\bar{R}zy \vee \forall u(\bar{R}uz \vee Rux)) \wedge (Rzy \vee \exists u\neg(\bar{R}uz \vee Rux))]$$

where we have also used that $A \rightarrow B \equiv \bar{A} \vee B$, and re-denoted $\neg Rzy$ as $\bar{R}zy$, and similarly in other cases with a negation in front of an atomic formula. Finally, we use the De Morgan law $\neg(A \vee B) \equiv \neg A \wedge \neg B$, as well as $\neg\neg A \equiv A$ (double negation) to conclude that for

$$\mathtt{NNF}(\Phi) :=: \forall x\exists y\forall z[(\bar{R}zy \vee \forall u(\bar{R}uz \vee Rux)) \wedge (Rzy \vee \exists u(Ruz \wedge \bar{R}ux))] \quad (4)$$

we have $\Phi \equiv \mathtt{NNF}(\Phi)$.

We note that the NNF of our example Φ in (1) has an additional existential quantifier, $\exists u$, that did not appear explicitly in the original formula.

In general, $NNF(\Phi)$ is obtained by, first, rewriting the formula by eliminating the uses of $\rightarrow$ and $\longleftrightarrow$ by the appropriate identities of Boolean logic used above; and, second, in case there are negation signs in front of non-atomic formulas, by pushing $\neg$ in front of atomic subformulas, via the use of the De Morgan identities, the law of double negation, and the logical equivalences $\neg\forall x\Psi \equiv \exists x\neg\Psi$, $\neg\exists x\Psi \equiv \forall x\neg\Psi$.

Let us make one additional correction that may be advisable. This is that we should make sure that no two quantifiers bind occurrences of the same variable. That is, we do not want to have two distinct occurrences of $\forall x$ with the same variable x ; the same for $\exists x$; and we do not want to have both $\forall x$ and $\exists x$ in our formula with the same variable x . This desirable state of affairs can always be achieved by an appropriate change of bound variables, without changing the binding pattern of the formula.

An example would be the change of $\forall x\exists yRxy \wedge \forall x\exists ySxy$ to the logically equivalent sentence $\forall x\exists yRxy \wedge \forall u\exists vSuv$.

The *Skolem form* of a formula Φ , denoted $\mathtt{Sk}(\Phi)$, is obtained from the NNF of Φ ; if Φ is already in NNF , then, of course, $\mathtt{NNF}(\Phi) = \Phi$, and we work on Φ itself.

$\mathtt{Sk}(\Phi)$ is obtained from $\mathtt{NNF}(\Phi)$ by two operations: first, we delete the existential quantifiers in the formula; second, we replace the variables that were originally bound by existential quantifiers with functional expressions of the form $f(x, \ldots)$ where $x, \ldots$ are certain of the variables bound by universal quantifiers. The list $x, \ldots$ may be empty; in this case, the expression we substitute is a constant c (nullary operation).

More precisely, if $\exists y$ is an existential quantifier in $\mathtt{NNF}(\Phi)$, we look at all those universal quantifiers $\forall x$ for which $\exists y$ is in the *scope* of $\forall x$, the scope of $\forall x$ being the part of the formula within the pair of brackets opening immediately after $\forall x$; we choose a new operation symbol f . a *Skolem function* (symbol), not yet used in the formula, and replace each occurrence of y originally bound by $\exists y$ with the expression $f(x, \ldots)$ where $x, \ldots$ is the list of the variables corresponding to the $\forall x$ which have our $\exists y$ in their scope. We do this for each existential quantifier $\exists y$ in $\mathtt{NNF}(\Phi)$, making sure that we use different Skolem functions for different existential quantifiers.

Note that we are introducing exactly as many different Skolem functions as there are existential quantifiers $\exists y$ in $\mathtt{NNF}(\Phi)$.

In the case of our initial example Φ , and its NNF in (4), we have, in $\mathtt{NNF}(\Phi)$, the two existential quantifiers $\exists y$ and $\exists u$. The first one is in the scope of $\forall x$, the second is in the scope of both $\forall x$ and $\forall z$. Accordingly, we employ the substitutions $y{=}f(x)$ and $u{=}g(x, z)$, and obtain

$$\mathtt{Sk}(\Phi) :=: \forall x \forall z[(\bar{R}(z, fx) \vee \forall u(\bar{R}uz \vee Rux)) \\ \wedge (R(z, fx) \vee (R(g(x, z), z) \wedge \bar{R}(g(x, z), x))] \; .$$

The general fact is as follows.

Theorem $M \vDash \Phi$ if and only if
there exist Skolem functions $f, \ldots$ such that $(M, f, \ldots) \vDash \mathtt{Sk}(\Phi)$.

The rigorous proof of this fact uses the basic fact exhibited in **(2)** (which is essentially the same as what is called the **Axiom of Choice**), together with a "structural induction" on the formula $\mathtt{NNF}(\Phi)$.

Let me point out how appropriate Skolem functions can be obtained from the run of of the Tarski machine on the formula. In fact, this procedure contains the essence of the proof of the "only if" (left-to-right implication) of the theorem in the general case, even for infinite structures, when tables are understood in an idealized sense.

We assume that we have found that $M \vDash \Phi$ by an application of the Tarski algorithm; we want to extract the additional information of certain specific Skolem functions $f, \ldots$ for which $(M, f, \ldots) \vDash \mathtt{Sk}(\Phi)$.

First of all, we take the application of the Tarski algorithm to $\mathtt{NNF}(\Phi)$, rather than Φ itself. However, there is a relatively minor difference only between the runs of the Tarski machine on Φ and on $\mathtt{NNF}(\Phi)$. We concentrate on the tables for each of the subformulas of the form $\exists y \Psi$ of $\mathtt{NNF}(\Phi)$, as well as the preceding table for Ψ (with one more free variable, namely y). In other words, we look at each subformula $\exists y \Psi$ in $\mathtt{NNF}(\Phi)$ headed by an existential quantifier $\exists y$, take off the $\exists y$, and consider the table of the remaining formula Ψ , as well as the table for $\exists y \Psi$. The information in these tables is sufficient to produce the Skolem functions.

In the example of (4) , we have two existential quantifiers: $\exists y$ and $\exists u$. We are going to determine the Skolem functions $y{=}f(x)$ and $u{=}g(x, z)$ in this order. In general, the Skolem functions "replacing" the existential quantifiers are determined "from the left to the right"; in our case, f first, g second. We will return to the point of this below; in our example, this discipline about the order turns out not to be essential.

The tables to consider are listed next. Note that the tables are not being computed now, rather, they are copied down from the Tarski algorithm on $\mathtt{NNF}(\Phi)$ in an order that is convenient for us now):

10 $\exists y \forall z[(\bar{R}zy \vee \forall u(\bar{R}uz \vee Rux)) \wedge (Rzy \vee \exists u(Ruz \wedge \bar{R}ux))]$

	x
$\top$	1
$\top$	2

11 $\forall z[(Rzy \vee \forall u(Ruz \vee Rux)) \wedge (Rzy \vee \exists u(Ruz \wedge Rux))]$

	x	y
$\top$	1	1
$\bot$	1	2
$\bot$	2	1
$\top$	2	2

Although we did not actually perform the Tarski algorithm on $\mathrm{NNF}(\Phi)$, in fact, table **10** is the same as table **8** for Φ above: $\mathbf{10}(x) \equiv \mathbf{8}(x)$; and table **11** is the same as table **7** for Φ : $\mathbf{11}(x, y) = \mathbf{7}(x, y)$.

From table **10** , we know that for all (both) x's, we have some y such that $\mathbf{11}(x, y)=\top$ (since $\mathbf{10}(x) = \exists y \mathbf{11}(x, y)$). We have to look at table **11** itself to see which y values should actually be chosen. We get: for $x=1$, $y=1$ works; for $x=2$, $y=2$ works. This says that $f: A \rightarrow A$ may be defined by $f(1)=1$, $f(2)=2$; in brief,

$$f=\begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} . \qquad (5)$$

Next we determine $g: A \times A \longrightarrow A$; for this we need the following tables from run of the Tarski machine for $\mathrm{NNF}(\Phi)$:

12 $\exists u(Ruz \wedge \bar{R}ux)$

	x	z
$\bot$	1	1
$\top$	1	2
$\bot$	2	1
$\bot$	2	2

13 $Ruz \wedge \bar{R}ux$

	u	x	z
$\bot$	1	1	1
$\bot$	1	1	2
$\bot$	1	2	1
$\bot$	1	2	2
$\bot$	2	1	1
$\top$	2	1	2
$\bot$	2	2	1
$\bot$	2	2	2

In fact, these tables are easy variants of ones for Φ ; $\mathbf{12}(x, z) = \neg \mathbf{4}(x, z)$, and $\mathbf{13}(u, x, z) = \neg \mathbf{3}(u, x, z)$.

From table **12** , we see there is exactly one value of the pair (x, z) , namely $(x=2, z=1)$, for which $\mathbf{12}(x, z)=\top$. This means that for $(x=2, z=1)$, there is some u for which $\mathbf{13}(u, x, z) = \top$; for the other pairs (x, z) , there is no such u . We do our best: we look at table **12** , to see which u actually works for $(x=2, z=1)$: we find that $u=2$ works. On the basis of this, we define

$$g(x, z) = \begin{cases} 2 & \text{when } x=2 \ \& \ z=1 \\ * & (\text{any value; e.g., } =1) \text{ otherwise} \end{cases}$$

(6)

This completes the determination of the skolem functions f and g ; with f and g given in (5) and (6) , we have that

$$(A; R, f, g) \vDash \mathrm{Sk}(\Phi) \ .$$

We consider another example in which it is clear that the stated order ("left to right") of the specifications of the Skolem functions may be important.

Let us assume that we have a structure $M=(A; \ldots)$ with the underlying set $A=\{1, 2\}$, and we have have a formula $\Psi(x, y, u, v)$ without quantifiers, having exactly the indicated free variables, and in NNF already, so that, when we put

$$\Phi :=: \forall x\exists y\forall u\exists v\Psi(x, y, u, v) \ ,$$

we have that $M\vDash\Phi$ holds true, and in fact, the table for $\Psi(x, y, u, v)$ in M , presented in two halves, is as follows:

14

$\Psi(x, y, u, v)$	x	y	u	v	$\Psi(x, y, u, v)$	x	y	u	v
$\top$	1	1	1	1	$\top$	2	1	1	1
$\bot$	1	1	1	2	$\bot$	2	1	1	2
$\bot$	1	1	2	1	$\bot$	2	1	2	1
$\top$	1	1	2	2	$\top$	2	1	2	2
$\bot$	1	2	1	1	$\bot$	2	2	1	1
$\top$	1	2	1	2	$\bot$	2	2	1	2
$\top$	1	2	2	1	$\bot$	2	2	2	1
$\bot$	1	2	2	2	$\bot$	2	2	2	2

Note that the Skolem function symbols are $y=f(x)$ and $v=g(x, u)$, and the Skolem form of Φ is:

$$Sk(\Phi) :=: \forall x\forall u\Psi(x, fx, u, g(x, u)) \ .$$

In order to get Skolem functions f and g witnessing the fact that $M\vDash\Phi$, let us calculate the successive tables for the subformulas $\exists v\Psi(x, y, u, v)$, $\forall u\exists v\Psi(x, y, u, v)$, $\exists y\forall u\exists v\Psi(x, y, u, v)$ and $\forall x\exists y\forall u\exists v\Psi(x, y, u, v)$:

15

$\exists v\Psi(x, y, u, v)$	x	y	u
$\top$	1	1	1
$\top$	1	1	2
$\top$	1	2	1
$\top$	1	2	2
$\top$	2	1	1
$\top$	2	1	2
$\bot$	2	2	1
$\bot$	2	2	2

16

$\forall u\exists v\Psi(x, y, u, v)$	x	y
$\top$	1	1
$\top$	1	2
$\top$	2	1
$\bot$	2	2

17

$\exists y\forall u\exists v\Psi(x, y, u, v)$	x
$\top$	1
$\top$	2

18 $\forall x\exists y\forall u\exists v\Psi(x, y, u, v) = \top$.

For $\exists y$, that is, $y=f(x)$, we use tables **17** and **16** , and we find the following two possible choices for $y=f(x)$: $f_1 = \binom{1\ 2}{1\ 1}$, and $f_2 = \binom{1\ 2}{2\ 1}$.

Next, we treat $\exists v$, and determine $v=g(x, y)$. Now, we have to consult tables **15** and **14** . We see however that these tables contain the free variable y , whereas the arguments of g are only x and u , y not among them. But of course, we already have a determination of the value of y as $y=f(x)$. Thus, we substitute $y=f(x)$ into these tables, obtaining ones which have no y as free variable; first, we use $f=f_1$:

18$_1$	$\Psi(x, f_1(x), u, v)$	x	u	v
	⊤	1	1	1
	⊥	1	1	2
	⊥	1	2	1
	⊤	1	2	2
	⊤	2	1	1
	⊥	2	1	2
	⊥	2	2	1
	⊤	2	2	2

19$_1$	$\exists v\Psi(x, f_1(x), u, v)$	x	u
	⊤	1	1
	⊤	1	2
	⊤	2	1
	⊤	2	2

On the basis of these, we get

$$v = g_1(x, u) = \begin{cases} 1 & \text{if } x=1 \ \&\ u=1 \\ 2 & \text{if } x=1 \ \&\ u=2 \\ 1 & \text{if } x=2 \ \&\ u=1 \\ 2 & \text{if } x=2 \ \&\ u=2 \end{cases}$$

Thus, with f_1 and g_1 so determined, we have

$$(M, f_1, g_1) \models \Phi .$$

On the other hand, when we use $f=f_2$ instead of f_1 , we get

18$_2$	$\Psi(x, f_2(x), u, v)$	x	u	v
	⊥	1	1	1
	⊤	1	1	2
	⊤	1	2	1
	⊥	1	2	2
	⊤	2	1	1
	⊥	2	1	2
	⊥	2	2	1
	⊤	2	2	2

19$_2$	$\exists v\Psi(x, f_2(x), u, v)$	x	u
	⊤	1	1
	⊤	1	2
	⊤	2	1
	⊤	2	2

and so

$$g_2(x,u) = \begin{cases} 2 & \text{if } x=1 \,\&\, u=1 \\ 1 & \text{if } x=1 \,\&\, u=2 \\ 1 & \text{if } x=2 \,\&\, u=1 \\ 2 & \text{if } x=2 \,\&\, u=2 \end{cases} .$$

Thus, $g_2(x, u)$ is different from $g_1(x, u)$; the pairs (f_1, g_1) , (f_2, g_2) are, however, equally good for $(M, f, g) \models \mathrm{Sk}(\Phi)$.