Chapter 2. Semantics

As in Part I we distinguish between informal semantics, which provides an explanation of use, and formal semantics, which provides truth conditions. On the (nominalistic version of the) informal semantics, the sentence letters of $\mathfrak{L}(\pi)$ stand in for (declarative) sentences, the predicate letters stand in for predicates; the variables are understood to range over a domain of objects; and the quantifier expression $\forall$**x** is understood in the same way as the quasi-English expression **for all x**. We take the term "object" in the broadest possible sense. Thus, ordinary physical things, actions, sets, classes, properties, and any things of the kind one might predicate something of in ordinary language count as objects.

The letters of the previous language $\mathfrak{L}$ were interpreted by (declarative) sentences $\mathfrak{L}(\pi)$. The letters of our present language $\mathfrak{L}(\pi)$ are interpreted by predicates. Let us make a few remarks as to what they are. Predicates are parts of language whose behavior can be understood either syntactically or semantically. In either case, the behavior of a predicate can be understood in terms of application or completion. In the syntactic case the application is to singular terms and in the semantic case to objects.

For the moment, let us suppose that we understood the notion of a singular term, i.e. of a term for an object. Then a predicate applies to a certain number of singular terms to form a sentence. So, for example, the predicate **is wise** applies to the singular term **Socrates** to form the sentence **Socrates is wise**, while the predicate **loves** applies to the singular terms **Anthony** and **Cleopatra** (in that order) to form the sentence **Anthony loves Cleopatra**; and, in general, with each predicate will be associated a positive number n, its <u>degree</u>, and the predicate will apply to n singular terms to form a sentence. As already indicated, we take sentences to be the predicates of degree 0; and each sentence, as so construed, will apply to 0 terms and result in that very sentence.

The result of applying a predicate to an appropriate number of terms will be called a <u>subject-predicate</u> or <u>predicational</u> sentence and the terms to which the predicate applies will be called the <u>subjects</u> of the sentence. One naturally takes a subject-predicate sentence to be saying something of some object or objects, with the predicate picking out what is said and the terms to which it is applied picking out the objects about which something is said.

Although the above examples seem very clear, there are general difficulties in identifying the predicates of ordinary language and in explaining how they apply to singular terms. Consider the sentence **Anthony loves Cleopatra**. What are its predicates? **Loves**, **loves Cleopatra**, or perhaps even **Anthony loves**? We shall not attempt to answer this question, which is a difficult question in linguistics, except to point out that the mere fact that one might construe a part of speech as a predicate does not mean that it is actually functioning as a predicate. For example, although one might construe the phrase **Anthony loves** in **Anthony loves Cleopatra** as picking out something that is said of Cleopatra, it is highly implausible to suppose that the phrase is functioning in this way.

Consider now the sentence **LA is between SD and SF**. It is natural to suppose that this is the result of applying a degree 3 predicate of betweenness to **LA**, **SD** and **SF**. But what exactly is the predicate? We could just say that it was the word **between**. But then the application of the predicate would require the insertion of the phrases **is** and **and**, which are special to this case. It would be desirable to have a syntactically uniform definition of application. So somehow the phrases **is** and **and** must be included in the predicate. But they cannot be included as a continuous part, as in **is between and**, since it is then not clear in general where the singular terms to which the predicate applies are to be inserted.

We solve this problem by taking a predicate to be a gappy sentence. We use the dash **-** as a gap-holder. A predicate may then be taken to be the result of replacing certain singular terms in a sentence with **-** (though not every gappy sentence will be a predicate. The degree of the predicate **P** is the number of gaps that it contains; and the result of applying **P** to the terms $\mathbf{t_1}$, .. , $\mathbf{t_n}$ will be the result of substituting $\mathbf{t_i}$ for the i-th gap, for i = 1, ... , n.

Even this solution is not quite faithful to ordinary language. Thus we would like the predicate **- loves -** to apply to **I** and **Cleopatra**, but this requires a grammatical change in the verb from the third person **loves** to the first person **love**. Thus our proposal represents a somewhat stylized version of ordinary language.

In natural language, the application of predicates to terms is not uniform. Thus in order to form a sentence from the predicate **is wise** the term is prefixed, while in order to form a term from **loves** one term is prefixed and the other postfixed. It is convenient to have a uniform procedure for applying predicates; and it is for this reason that we have adopted the convention whereby a predicate is always prefixed to the terms to which it applies.

Let us call a predicate **P** of degree n a <u>prefixor</u> if its application to any terms $\mathbf{t_1}$, ... , $\mathbf{t_n}$ (to which it applies) is always the result $\mathbf{Pt_1}$ ... $\mathbf{t_n}$ of prefixing **P** to the terms (in the given order). Then a consequence of our convention is that the predicate-letters can only properly be taken to stand in for prefixor predicates. Since the predicates of natural language are not usually prefixors, the application of our symbolism is therefore severely circumscribed.

However, there is a simple way in which it might be extended. For with each predicate **P** of degree n we may associate a prefixor predicate **[P]** whose prefixed application $\mathbf{[P]t_1...t_n}$ to the terms $\mathbf{t_1}$, ... , $\mathbf{t_n}$ is taken to be equivalent to the customary application of **P** to $\mathbf{x_1}$, ..., $\mathbf{x_n}$ (whatever that might be). This means that the substitution of ordinary language predicates for our predicate letters may not literally make sense. For example, if we substitute the predicate **is wise** for the predicate letter **P** in the formula **Pv** we obtain the piece of nonsense **is wise v**. In order to overcome this difficulty, we may substitute, instead of the original predicate **P**, the result **[P]** of enclosing it in square predicates, with the understanding that the prefixed application $\mathbf{[P]t_1...t_n}$ of **[P]** to the terms $\mathbf{t_1}$, ... , $\mathbf{t_n}$ is to be equivalent to the customary application of **P** to $\mathbf{x_1}$, ..., $\mathbf{x_n}$ (whatever that might be). Thus **[loves]Antony Cleopatra** is to be equivalent to **Anthony loves Cleopatra**.

Ordinary language is not very flexible in allowing the formation of predicates. There is, for example, no natural way of forming a binary predicate that is true of two objects just in case London is between the first and the second. Moreover, it is often not clear what the predicate is. **hurts**, for example, is ambiguous between being a predicate of degree 1 (the intransitive reading) or of a predicate of degree 2 (the transitive reading). We shall later consider a more flexible and accurate method for forming predicates from ordinary language.

A predicate applies to objects in the sense of being true or false of them. The exact way the application goes depends upon the degree of the predicate. A predicate of degree 1, such as **is wise**, is true or false of a single object; a predicate of degree 2, such as **loves**, is true or false of a pair of objects; and, in general, a predicate of degree n is true or false of an n-tuple of objects. As already indicated, we take sentences to be predicates of degree 0; they are true or false of no objects, i.e. true or false simpliciter.

When we apply a predicate to a variable or to variables, we obtain, not a sentence, but what we shall call an open sentence. This is like an ordinary sentence but contains variables where the sentence contains terms. So, for example, when we apply the predicate **[is wise]** to the variable **x** we obtain the open sentence **[is wise] x**, which can be obtained from the sentence **[is wise] Socrates** by the substitution of **x** for **Socrates**. In general, whenever a formula contains free variables, its interpretation will be given by an open rather than by a closed sentence.

But what in general is meant by an open sentence? It will not do to say that an open sentence is one that contains an occurrence of a variable. For **'x' is a variable** and **∃x[is wise] x** both contain a variable, yet neither is open as opposed to closed. There seems to be a distinction between those occurrences of variables that are free in the intuitive sense of its being appropriate to assign them a value and those that are not. An open sentence will then be one that contains, in this sense, a free occurrence of a variable.

Very roughly, the occurrence of a variable in an open sentence will be free when the evaluation of the sentence as true or false depends upon the value of the variable. In many cases, the truth-value of the sentence will depend upon the value of the variable (so, for example, **[is wise]x** will be true under the assignment of Socrates to **x** but false under the assignment of Bottom); and this is a sure sign that the truth-evaluation also depends upon the value of the variable. But the truth-evaluation may depend upon the value of the variable even when the truth-value is stable. For example, the variable **x** is free in the open sentence **[is self-identical]x** even though the sentence is true for all values of **x**. Open sentences are not true or false simpliciter but true or false relative to an assignment of values to the free occurrences of their variables. Suppose that **S** is an expression of some language that permits the use of variables and that $\mathbf{x_1}, \ldots , \mathbf{x_n}$ are the occurrences of various variables in **S**. Let α be the assignment of the objects (or "values") $o_1, \ldots , o_n$ to $\mathbf{x_1}, \ldots , \mathbf{x_n}$. Then the expression **S** is said to be satisfied by those assignments α that would render it true (and to be contra-satisfied by those assignments that would render it false). Thus the clause **x > 0** is satisfied by the assignment of 1 to **x** but contra-satisfied by the assignment of 0 to **x**.

Whether an assignment satisfies an expression does not depend upon the designation of the objects or the identity of the variables. So, given that the assignment of 9 to **v** satisfies **v > 0**, the assignment of the number of planets to **w** will satisfy **w > 0**. Note that the assignment of the values is to the occurrences of variables, in contrast to the variables themselves, as under the formal notion of satisfaction to be introduced later. The notion of satisfaction (or contra-satisfaction) may not have application in certain cases. We list three types of failure:

(1) **S** may not be sentential in form or even meaningful. It makes no sense, for example, to talk of the satisfaction of the expression **S = x >**.

(2) Even though **S** is sentential in form, some of its variable occurrences may be "idle", i.e. they require the assignment of a value but no assignment has been made. So, for example, it makes no sense to say that the assignment of 2 to **v** satisfies **v > w**.

(3) Even though **S** is sentential in form, some variables may have improperly been assigned a value. For example, where **S** is the expression **'v' has v letters**, we can meaningfully assign a value to the second occurrence of **v** (with the assignment of 1 satisfying the expression), but we cannot meaningfully assign a value to the first occurrence of **v**. Let us call an occurrence of a variable in an expression referential if it can meaningfully be assigned an object as a value. It is a matter of debate which occurrences of variables are to considered referential. For example, some philosophers have thought we may meaningfully talk of an object satisfying the expression

necessarily, v is the number of planets, while others have disagreed. However, this is a dispute on which we will not take sides.

We return now to the question of providing a rendition. We suppose given a language $\mathcal{L}_0$, such as English, which is already interpreted. In order to provide a rendition of the formulas of $\boldsymbol{\mathcal{L}(\pi)}$, we extend the language $\mathcal{L}_0$ to a hybrid language $\mathcal{L}_1$ by means of the following three devices:

(1) Addition of variables Suppose that $\mathbf{S} = \mathbf{S}(\mathbf{t}_1, \ldots, \mathbf{t}_m)$, for $m \geq 0$, is a sentence of $\mathcal{L}_1$, containing the distinct constituent expressions $\mathbf{t}_1, \ldots, \mathbf{t}_m$. Let $\mathbf{S}' = \mathrm{S}(\mathbf{u}_1, \ldots \mathbf{u}_m)$ be the result of replacing $\mathbf{t}_1, \ldots, \mathbf{t}_m$, respectively, with $\mathbf{u}_1, \ldots \mathbf{u}_m$. Then $\mathbf{S}'$ is admitted as what we call an <u>open sentence</u> into the extension $\mathcal{L}_1$ as long as all of the resulting occurrences of $\mathbf{u}_1, \ldots \mathbf{u}_m$ are referential. In the special case in which $\mathbf{S}'$ contains no referential occurrences of variables we say that it is a <u>closed sentence</u> or simply a <u>sentence</u>. Suppose, for example, that **S** is the sentence **Hesperus is identical to Phosphorus**, $\mathbf{t}_1$ = **Hesperus**, and $\mathbf{t}_2$ = **Phosphorus**. Then $\mathbf{S}' = \mathbf{v}_1$ **is identical to** $\mathbf{v}_2$ will be admitted as an open sentence since the occurrences of $\mathbf{v}_1$ and $\mathbf{v}_2$ are referential. On the other hand, when **S** = **Giorgione is so-called because of his size,** then $\mathbf{S}'$ = $\mathbf{v}_1$ **is so-called because of his size** will not be an open sentence since the occurrence of $\mathbf{v}_1$ is not referential.

(2) Formation of predicate terms.
Where **S** is an open sentence, we allow the formation of the predicate term $\mathbf{P} = \boldsymbol{\lambda}\mathbf{v}_1 \ldots \mathbf{v}_n\mathbf{S}$, where $\mathbf{v}_1 \ldots \mathbf{v}_n$ are distinct variables and $n \geq 0$. **P** is interpreted in the obvious way as a predicate. Suppose, for example, that **S** is the open sentence $\mathbf{v}_1 > \mathbf{v}_2$ and that **P** is the predicate $\boldsymbol{\lambda}\mathbf{v}_1(\mathbf{v}_1 > \mathbf{v}_2)$. Then when $\mathbf{v}_2$ is assigned a certain value, say 2, then **P** will be a predicate which is true of an object x just in case it is a number greater than 2.

It might be wondered why we do not confine our attention to ordinary predicate expressions, such as **loves** or **is wise**. But the present proposal allows us much more versatility in the construction of predicates. For example, we can construct the predicate $\boldsymbol{\lambda}\mathbf{v}_2\mathbf{v}_1$**(Othello is jealous of** $\mathbf{v}_1$**'s love for** $\mathbf{v}_2$**)**, which has no natural ordinary language counterpart.

The predicate $\boldsymbol{\lambda}\mathbf{v}_1 \ldots \mathbf{v}_n\mathbf{S}$ is said to be of degree n. Note that we allow the degree of a predicate to be 0. Also note that the "body" **S** of a predicate $\boldsymbol{\lambda}\mathbf{v}_1 \ldots \mathbf{v}_m\mathbf{S}$ may not contain all of the variables $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and that it may contain variables that are not among $\mathbf{v}_1, \ldots, \mathbf{v}_n$. We may want to "parameterize" certain predicates to a time, for example. The predicate of sitting would then be represented by $\boldsymbol{\lambda}\mathbf{v}_1$($v_1$ is sitting at time $\mathbf{v}_2$) rather than by $\boldsymbol{\lambda}\mathbf{v}_1$($v_1$ is sitting).

(3) Predicate Application. If **P** is the predicate term $\boldsymbol{\lambda}\mathbf{v}_1 \ldots \mathbf{v}_n\mathbf{S}(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ and $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are variables (not necessarily distinct), then the application $\mathbf{Pu}_1 \ldots \mathbf{u}_n$ of **P** to $\mathbf{u}_1, \ldots, \mathbf{u}_n$ is also admitted as an open sentence of the language $\mathcal{L}_1$.

(4) Application of logical operations.
We close the open sentences of $\mathcal{L}_1$ under the same formation rules as those of $\boldsymbol{\mathcal{L}(\pi)}$. Thus we allow the formation of negations, disjunctions and universal quantifications, with each of these understood in the manner that has been proposed.

By a (simple) <u>rendition</u> of $\boldsymbol{\mathcal{L}(\pi)}$ in $\mathcal{L}_1$ is meant an assignment of predicate terms from $\mathcal{L}_1$ to the predicate letters of $\boldsymbol{\mathcal{L}(\pi)}$, with the degree of the predicate letter matching the degree of the predicate term. A simple rendition of $\boldsymbol{\mathcal{L}(\pi)}$ in $\mathcal{L}_1$ can then be extended to arbitrary expressions **E** by simply replacing each predicate letter that occurs in **E** by the corresponding predicate term. Every formula of $\mathcal{L}(\pi)$ will then be rendered by an open sentence of $\mathcal{L}_1$.

The task of paraphrasing (closed) sentences of the hybrid language $\mathcal{L}_1$ in plain English, or at least into something closer to English, is not at all straightforward. The truth-functional connectives can be dealt with in the same way as before; and a predication (such as **[v_1 philosophizes]x**) can be reduced through substitution (to **it is the case that x philosophizes**, in the given example). However, the variables resist a uniform treatment. In particular cases, they can be made to disappear. Thus **for all x it is the case that x philosophizes** can be rendered **it is the case that all things philosophize**. But it is not clear, in general, how (if at all) variables are to be avoided.

The above interpretation rests upon certain important assumptions, which it will be important to disentangle. Let us call the domain of objects which the variables of quantification range over the <u>domain of quantification</u>. Then the first assumption, which we call <u>Universality</u>, is that the domain of quantification is universal; it consists of all objects whatever, without restriction. Let us call the domain of objects of which the predicates are true or false the <u>domain of predication</u>. Then the second assumption, which we call <u>Coincidence</u>, is that the domains of quantification and predication coincide, every object in the one domain is in the other. It is clear that the domain of quantification should be included in the domain of predication. If, for example, **v** can take a certain object as its value. Then the degree one predicate **P** must be true or false of that object if a quantified statement such as ∀**vPv** is to make sense. However, there is no necessity that the domain of predication should be included in the domain of quantification. Given that the sentence ∀**vPv** makes sense, the applicability of the predicate **P** to an object does not require that the variable should range over that object.

For certain purposes we may want to relinquish Universality and restrict the domain of discourse; we may, for example, wish to talk about numbers (and only numbers) or about people or steam locomotives. This may be done by relativizing the above account. We suppose that we are given a special degree-one predicate term **D**, which we take to indicate the domain of discourse. Thus in the first of the examples above, **D** would be the predicate term **λv(v is a number)**.

It is usual to suppose that each assigned predicate will apply (either truly for falsely) to any objects that conform to the domain predicate **D** (one might even require that they should necessarily have such application). Thus if the predicate **[v_1 sleeps]** does not apply to numbers, we cannot both assign this predicate to a predicate letter and take the domain predicate to be **[v_1 is a number]**. It is also usual to suppose that the domain predicate is true of some object (i.e., the domain itself should be nonempty). We shall later consider some of the consequences of dropping these two conditions.

The platonistic version of the informal semantics is likewise more complicated. There are at least two ways it can go - one Russellian and the other Fregean. We present the Russellian way here. We should understand the sentence letters as standing for propositions, as before. The domain must now be specified by a property P. The predicate letters of degree n should be understood as standing for relations of degree n that apply to any objects from the domain and the variables should be understood as standing for objects from the domain. Under an assignment of objects from the domain to the free variables of the formula, each formula can be taken to signify a proposition. Thus when Socrates is assigned to the variable **x** and the property of philosophizing to the predicate letter **F** the formula **Fx** will signify the proposition that Socrates philosophizes. Suppose that a proposition has been assigned to the formula **A** under an assignment of objects to the variables. We now suppose that upon selection of a variable **x** a corresponding property **F** can be formed. This is intuitively the property that is formed from the proposition by making a gap in the places occupied by the object assigned to **x**. We then take $\forall\mathbf{xA}$ to signify the proposition that whatever has P has F. We do not consider the question of how exactly the property F is to be obtained from the proposition.

We now provide a formal semantics for $L(\pi)$. A <u>model</u> is a pair (D,υ), where:

(i) D is a non-empty set

(ii) υ relates each predicate letter of degree n to n objects in the domain, i.e., υ consists of $(n+1)$-tuples $(F,d_1,...,d_n)$ in which F is an n-place predicate letter and $d_1,...,d_n$ are elements of D. We write $\mathbf{\upsilon Fd_1,...,d_n}$ for $\mathbf{<F,d_1,...,d_n>} \in\upsilon$. υ might be called a <u>predicate valuation</u>, in contrast to the <u>sentential valuations</u> of earlier chapters.

If a formula contains free variables, the information in a model does not determine whether it is true or false: **Fx** is true for some values of **x** and false for others. Even the truth of a formula without free variables may depend on whether subformulas that do have free variables are true for various values of **x**: $\forall\mathbf{xFx}$ is true only if **Fx** is true for all values of **x**. If $M=(D,\upsilon)$ is a model then let us say that an <u>M-assignment</u> (or, if M is understood from context, an <u>assignment</u>) is any function from variables to members of D. The truth value of a formula **A** depends only on the values of the variables that occur in **A** so we do not require that assignments have all variables in their domain. Let us say that **A** is <u>covered by</u> an assignment δ, or that δ <u>covers</u> **A**, if every free variable of **A** is in the domain of δ. (The assignment ϕ with the empty domain covers only sentences with no free variables.) If δ and δ' are assignments then $\delta(u)$ is **strictly identical to** $\delta'(v)$ ($\delta(u)\simeq\delta'(v)$) if either $\delta(u)$ and $\delta'(v)$ are both undefined or else both are defined and $\delta(u)=\delta'(v)$. (Strict identity differs from ordinary identity in that $\delta(u)=\delta(v)$ is undefined when either $\delta(u)$ or $\delta(v)$ is undefined, whereas $\delta(u)\simeq\delta(v)$ is always either true or false.) If Δ is a set of variables then assignments δ and δ' <u>agree on Δ</u> if, for all variables $v\in\Delta$, $\delta'(v)\simeq\delta(v)$. If δ is an assignment then $\delta(^d/_x)$ is the assignment that agrees with δ on all variables other than x and that takes the value d at x. An assignment like $\delta(^d/_x)$ that is defined on x and that agrees with δ on variables other than x is said to be an <u>x-variant of δ</u>. Note that if x and y are distinct $\delta(^d/_x)(^e/_y)\simeq\delta(^e/_y)(^d/_x)$ and, more generally, if $x_1,...,x_n$ are distinct the assignments described by appending $(^{d_1}/_{x_1}),...,(^{d_n}/_{x_n})$ to δ in any order are all the same. We use the notation $\delta(^{d_1,...,d_n)}/_{x_1,...,x_n})$ to refer to this assignment.

If δ is an M-assignment that covers **A** we write $\vDash_M A[\delta]$ for **the formula A, under assignment δ is true in model M**. As usual the subscript can be dropped if the model is clear from context. A definition of truth and falsity can now be given.

Definition *a.* Let $M=(D,\upsilon)$ be a model and δ an M-assignment that covers **A**.

(i) If $\mathbf{A=P_ix_1...x_n}$ then $\vDash_M\mathbf{A}[\delta]$ if $\upsilon P\delta(\mathbf{x_1}),...,\delta(\mathbf{x_n})$, and $\nvDash_M\mathbf{A}[\delta]$ if not $\upsilon P\delta(\mathbf{x_1}),...,\delta(\mathbf{x_n})$;

(ii) If $\mathbf{A=(B\vee C)}$ then $\vDash_M\mathbf{A}[\delta]$ if $\vDash_M\mathbf{B[\delta]}$ or $\vDash_M\mathbf{C}[\delta]$, and $\nvDash_M\mathbf{A}[\delta]$ if both $\nvDash_M\mathbf{B}[\delta]$ and $\nvDash M\mathbf{C}[\delta]$;

(iii) If $\mathbf{A=\neg B}$ then $\vDash_M\mathbf{A}[\delta]$ if $\nvDash_M\mathbf{B}[\delta]$, and $\nvDash_M\mathbf{A}[\delta]$ if $\vDash_M\mathbf{B}[\delta]$;

(iv) If $\mathbf{A}=\forall\mathbf{xB}$ then $\vDash_M\mathbf{A}[\delta]$ if $\vDash_M\mathbf{B}[\delta']$ for all δ' that are x-variants of δ, and $\nvDash_M\mathbf{A}[\delta]$ if $\nvDash_M\mathbf{B}[\delta']$ for some x-variant δ' of δ.

As in Part I, the truth definition provides simultaneous inductive definitions of **verifies** ($\vDash$) and **falsifies** ($\nvDash$); and it may be shown, as before, that a model and an assignment covering **A** jointly determine a unique truth-value for **A**. Clause i shows how the truth value of atomic sentences under an interpretation is provided by the valuation. Clauses ii and iii are familiar from truth functional logic. Notice that the truth value under an assignment of a disjunction or a negation depends on the truth values of its immediate components under that assignment. The truth value under an assignment of a universal formula, however, depends on the truth value of its immediate component under other assignments. In particular, $\forall\mathbf{uPuv}$ is true under δ iff **Puv** is true any assignment that is defined on **u** and that agrees with δ on all other variables. This implies that $\forall\mathbf{uPuv}$ is true under δ only if, for all $d\in D$, $\upsilon\mathbf{P}d,\delta(\mathbf{v})$. Talking of truth *under other assignments* provides a way of talking of truth *about all objects*. The general result that our formal semantics respects the informal semantics in the sense that $\vDash_M A$ iff the rendition of **A** that naturally corresponds to **M** is true is best postponed until some basic results about the formal semantics have been proved.

Let us look at a couple of examples to see how the truth definition is applied.

Example 1

Let $M=(D,\upsilon)$, where D is the set of positive integers, $\upsilon\mathbf{P}m$ iff m is an even integer, and $\upsilon\mathbf{R}m,n$ iff $m=2$ and $n=2$. Let δ be the function with domain $\{\mathbf{u},\mathbf{v}\}$ such that $\delta(\mathbf{v})=2$ and $\delta(\mathbf{u})=3$. Then $\vDash_M\forall\mathbf{v}(\neg\mathbf{Rvv}\vee\mathbf{Pu})[\delta]$ iff $\vDash_M(\neg\mathbf{Rvv}\vee\mathbf{Pu})[\delta']$ for all δ' such that δ' is defined on **v**, $\delta'(\mathbf{u})=3$, and δ' is undefined elsewhere. This holds iff for every positive integer m either m is not 2 or 3 is even. When m is 2 this disjunction is false. Hence $\nvDash_M\forall\mathbf{v}(\neg\mathbf{Rvv}\vee\mathbf{Pu})[\delta]$.

Example 2

Let $M=(D,\upsilon)$ where D is the letters of the (Roman) alphabet and $\upsilon\mathbf{R}\alpha,ß$ iff ß comes after α in alphabetical order. Let δ be ϕ, the assignment function that is everywhere undefined. Then we have the following chain of equivalences: $\vDash_M\forall\mathbf{u}\forall\mathbf{v}(\mathbf{Ruv}\supset\neg\mathbf{Rvu})[\delta]$ iff for all letters d $\vDash_M\forall\mathbf{v}(\mathbf{Ruv}\supset\neg\mathbf{Rvu})[\delta(^{d}/_{u})]$ iff for all letters d and e $\vDash_M(\mathbf{Ruv}\supset\neg\mathbf{Rvu})[\delta(^{d}/_{u})(^{e}/_{v})]$ iff for all letters d and e $\nvDash_M\mathbf{Ruv}[\delta(^{d}/_{u})(^{e}/_{v})]$ or $\vDash_M\neg\mathbf{Rvu}[\delta(^{d}/_{u})(^{e}/_{v})]$ iff for all letters d and e either e does not come after d in alphabetical order or d does not come after e in alphabetical order. Since the last condition is true, $\vDash_M\forall\mathbf{u}\forall\mathbf{v}(\mathbf{Ruv}\supset\neg\mathbf{Rvu})[\delta]$.

The truth definition yields the appropriate truth conditions for the existential quantifier.

Theorem *b.* With respect to any model M and any M-assignment δ covering $\exists\mathbf{xA}$, $\vDash_M\exists\mathbf{xA}[\delta]$ iff $\vDash_M\mathbf{A}[\delta(^{d}/_{x})]$ for some d in the domain of M.

The proof is straightforward and is left as an exercise.

The notion of truth under an assignment is naturally extended to sets: $\vDash_M\Gamma[\delta]$ iff $\vDash_M\mathbf{A}[\delta]$ for all $\mathbf{A}\in\Gamma$. Γ is said to be satisfiable if it is true under some assignment in some model. A sentence of $\mathcal{L}(\pi)$ is true if it is true under the empty assignment.

The notation $\vDash_M\mathbf{A}[\delta]$ suggests that assignments can be thought of as combining with formulas to form some entities to which a model assigns truth values. Let us take this idea seriously and say that a formula $\mathbf{A}$ (or set Γ) and an M-assignment δ that covers it together comprise a reified formula $\mathbf{A}[\delta]$ (or a reified set $\Gamma[\delta]$). We may identify the reified formula $\mathbf{A}[\delta]$ with the ordered pair $<\mathbf{A}, \delta>$. A reified formula is a kind of hybrid, containing both linguistic elements and elements of the model. It specifies both a formula and the objects of which this formula is said to hold. Remember that $\mathbf{A}[\delta]$, unlike $\mathbf{A}(\mathbf{x})$ or $\mathbf{A}[\mathbf{x}]$, is not just another way of writing $\mathbf{A}$, but a new entity of a different kind. Reified formulas are essentially the same as David Kaplan's "valuated formulas" (Kaplan, 'Opacity', in L. Hahn and P Schilpp eds., The Philosophy of W.V.Quine La Salle, Ill., Open Court, 230-293, 1984?). A valuated formula is "an open formula under an assignment of values to its free variables." It is, according to Kaplan, a linguistic analogue of Russell's "singular proposition"--a construction containing objects and properties as constituents.

Alternatively, assignments can be thought of as combining with models to provide a means of assigning truth values to formulas. Let us take this idea seriously and say that if δ is an assignment covering $\mathbf{A}$ (covering every formula in Γ) then $I=(M,\delta)$ is an **interpretation for A (Γ)**. The notation $\vDash_{\mathbf{I}}\mathbf{A}$ or $\vDash_{\mathbf{I}}\Gamma$ indicates that $\mathbf{A}$ or Γ is true under interpretation I.

We would like to define notions of validity and consequence in $\mathscr{L}(\pi)$ that correspond to the informal notions of logical truth and logically valid argument-pair. For sentences the appropriate definitions clearly follow the pattern of truth-functional logic: $\mathbf{A}$ is valid ($\vDash\mathbf{A}$) if $\mathbf{A}$ is true in all models; $\mathbf{A}$ is a consequence of Γ ($\Gamma\vDash\mathbf{A}$) if, for every M, $\vDash_M\Gamma$ implies $\vDash_M\mathbf{A}$. Since formulas containing free variables do not correspond to English sentences, it is not obvious how to extend these definitions to arbitrary formulas. The following definition provides a natural generalization of the above concepts that preserves the form of the previous definitions.

Definition c.
(i) $\vDash\mathbf{A}$ if $\vDash_I\mathbf{A}$ for all interpretations I for $\mathbf{A}$.
(ii) $\Gamma\vDash\mathbf{A}$ if $\vDash_I\Gamma$ implies $\vDash_I\mathbf{A}$ for all interpretations I for $\Gamma\cup\{A\}$.

Let us illustrate these notions with some examples.

1. $\vDash\forall\mathbf{uPuv}\supset\exists\mathbf{uPuv}$. For suppose there is (M,δ) such that $\vDash_M\forall\mathbf{uPuv}[\delta]$. Choose any $d\in D$ and let $\delta'=\delta({}^{d}/_{u})$. By clause (iv) of the truth definition $\vDash_M\mathbf{Puv}[\delta']$. By theorem 2.2 $\vDash_M\exists\mathbf{uPuv}[\delta]$, which establishes the desired result.

2. $\exists\mathbf{u}\forall\mathbf{vPuv}\vDash\forall\mathbf{v}\exists\mathbf{uPuv}$. For $\vDash_M\exists\mathbf{u}\forall\mathbf{vPuv}[\delta]$ implies $\vDash_M\forall\mathbf{vPuv}[\delta({}^{d}/_{u})]$ for some d in the domain of M, which implies that for some d $\vDash_M\mathbf{Puv}[\delta({}^{d}/_{u})({}^{e}/_{v})]$ for all e in the domain of M. In particular, if f is an arbitrary member of the domain of M, then for some d $\vDash_M\mathbf{Puv}[\delta({}^{d}/_{u})({}^{f}/_{v})]$. This implies $\vDash_M\exists\mathbf{uPuv}[\delta({}^{f}/_{v})]$ and, since f is arbitrary, that $\vDash_M\forall\mathbf{v}\exists\mathbf{uPuv}[\delta]$.

3. $\mathbf{Pu}\nvDash\forall\mathbf{uPu}$. For suppose $M=(D,\upsilon)$ where $D=\{d_1,d_2\}$ and υPd_1 but not υPd_2. Let $\delta = \{(u,d_1)\}$. Then $\vDash_M\mathbf{Pu}[\delta]$ but $\nvDash_M\mathbf{Pu}[\delta({}^{d_2}/_{u})]$ and consequently $\nvDash_M\forall\mathbf{uPu}[\delta]$.

$\mathscr{L}(\pi)$ is a first order language. But it is also a truth-functional language in the sense of Part I. For the atomic formulas and universal formulas may be regarded as the truth-functional constituents from which all of the other formulas of $\mathscr{L}(\pi)$ can be obtained by means of truth-functional composition. From this perspective it is as if the formulas of the form $\forall\mathbf{xA}$ were themselves sentence-letters, without any significant internal structure. As a consequence, all of the definitions and results of Part I will apply to the language $\mathscr{L}(\pi)$ under an appropriate realignment of the set of constituents. We may say, in particular, that α is a valuation for $\mathscr{L}(\pi)$, that $\mathbf{A}$ is true under the valuation α, that $\mathbf{A}$ is a truth-functional consequence of Γ or that $\mathbf{A}$ is truth-functionally valid. The symbol $\vDash$, of course, cannot be used to express these ideas in $\mathscr{L}(\pi)$, since it has been appropriated to express concepts which are somewhat different in this context. Let us therefore use $\alpha\vDash_{\mathbf{tf}}\mathbf{A}$, $\vDash_{\mathbf{tf}}\mathbf{A}$ and $\Gamma\vDash_{\mathbf{tf}}\mathbf{A}$ to indicate that $\mathbf{A}$ is true under the valuation α, that $\mathbf{A}$ is truth-functionally valid and that $\mathbf{A}$ is a truth-functional consequence of Γ. When it is necessary to distinguish valuations for $\mathscr{L}(\pi)$ (viewed as a truth-functional language) from valuations for $\mathscr{L}$, we call the former *extended (sentential) valuations* and the latter *simple (sentential) valuations*. When it is necessary to distinguish the truth-functionally valid formulas that belong to the language $\mathscr{L}$ from other truth-functionally valid formulas, we shall call the former *tautologies;* and when it is necessary to convey that formulas and sets of formulas *of* $\mathscr{L}$ are related by truth-functional consequence, we shall use the term *tautological consequence*.

Any truth-functionally valid formula is valid. For if it were not, there would be a model M and an assignment δ under which it was false. But the set of all constituents that are true with respect to M and δ forms a valuation that makes exactly the same formulas true as M does. So there would then be a valuation under which the formula is false, and it would not be truth-functionally valid after all. The converse of this observation, however, is false. The valid formulas of $\mathscr{L}(\pi)$ are not all truth-functionally valid. For example the formula $\forall\mathbf{uPuv}\supset\exists\mathbf{uPuv}$ of example 1 is valid. But it is not truth functionally valid. By letting $\forall\mathbf{uPuv}$ and $\forall\mathbf{u}\neg\mathbf{Puv}$ both be true (letting $\alpha=\{\forall\mathbf{uPuv},\forall\mathbf{u}\neg\mathbf{Puv}\}$, say) the formula comes out false. For quantifier-free formulas, however, validity and truth-functional consequence do coincide. For if $\mathbf{A}$ is not truth-functionally valid then $\alpha \nvDash_{\mathbf{tf}}\mathbf{A}$ for some assignment α. It is not difficult to find a model $M=(D,\upsilon)$ and M-assignment δ that assigns the same truth value to $\mathbf{A}$ as α does. One such interpretation can be constructed by taking as objects the variables of $\mathbf{A}$ themselves. Let D be a non-empty set that includes every variable that occurs in $\mathbf{A}$, let $\delta(x)=x$ for every variable in $\mathbf{A}$, and let $\upsilon Fx_1,\ldots,x_n$ iff $Fx_1\ldots x_n \in \alpha$. This ensures that α and (M,δ) assign the same truth values to the constituents of $\mathbf{A}$ and, since $\mathbf{A}$ is quantifier-free, to $\mathbf{A}$ itself. Hence $\mathbf{A}$ is not valid, and we have proved the following result.

Theorem d. If $\mathbf{A}$ is quantifier-free then $\vDash_{\mathbf{tf}}\mathbf{A}$ iff $\vDash\mathbf{A}$.

It was noted above that bound variables serve primarily to indicate linkage and independence among predicate places and quantifiers. Free variables serve an additional function. They indicate the sequence of objects that a reified formula is about. The identity of the bound variables that indicate a particular pattern of links in $\mathbf{A}$ is unimportant. Similarly, the identity of the free variables that indicate a particular sequence of objects to which $\mathbf{A}$ applies under δ is unimportant. Accordingly, let us say that $\mathbf{A}[\delta]$ is *content-identical* to $\mathbf{B}[\gamma]$ if the bound variable occurrences in $\mathbf{A}$ and $\mathbf{B}$ exhibit the same pattern of linkage and corresponding free variable occurrences in $\mathbf{A}[\delta]$ and $\mathbf{B}[\gamma]$ are assigned the same object. More precisely, suppose $\mathbf{B}$ is obtained by simultaneously replacing each occurrence of $\mathbf{x_1},\ldots,\mathbf{x_n}$ in $\mathbf{A}$ by $\mathbf{y_1},\ldots,\mathbf{y_n}$ where $\mathbf{x_1},\ldots,\mathbf{x_n}$ is a list, in order and including repetitions, of all the variables occurring in A. Then $\mathbf{A}[\delta]$ is content-identical to $\mathbf{B}[\gamma]$ iff for all i, $1\le i\le n$, either x_i is bound and, for $1\le j\le n$, x_i is linked to x_j iff y_i is linked to y_j or x_i is free and $\delta(x_i)=\gamma(y_i)$. It follows from this definition that content identity is an equivalence relation on reified formulas. We call the equivalence class of $\mathbf{A[\delta]}$ under this relation the <u>reified statement</u> expressed by $\mathbf{A[\delta]}$, and write it as $\mathbf{[A,\delta]}$. If we imagine replacing the free variables in $\mathbf{A[x_1,\ldots,x_n][\delta]}$ by the objects $\delta(x_1),\ldots,\delta(x_n)$, then content-identical reified formulas would express the same "statements".

We now prove a basic result stating that content-identical reified formulas have the same truth value, so that truth and falsity can be meaningfully applied to reified statements. The truth-value of a formula under an assignment is a function of the elements assigned to that formula's free variable occurrences. It does not depend on either the identity of the free variables that occur in the formula or the objects, if any, that are assigned to other variables.

Theorem e. If $\mathbf{A[\delta]}$ is content-identical to $\mathbf{B[\gamma]}$ then, for any model M for which δ and γ are defined, $\vDash_M \mathbf{A[\delta]}$ iff $\vDash_M \mathbf{B[\gamma]}$.

The proof is by induction on $\mathbf{A}$. We prove the quantifier case, leaving the remaining cases to the reader. Suppose $\mathbf{(\forall xC)[x_1\ldots x_n]}[\delta]$ and $\mathbf{(\forall yD)[y_1\ldots y_n][\gamma]}$ are content-identical. $\vDash_\mathbf{M}\mathbf{(\forall xC)[x_1\ldots x_n]}[\delta]$ iff, for all d in the domain of M, $\vDash_\mathbf{M}\mathbf{C[z_1\ldots z_m]}[\delta(^d/_x)]$, where $(\mathbf{z_1},\ldots,\mathbf{z_m})$ is obtained by inserting occurrences of $\mathbf{x}$ in appropriate places in $(\mathbf{x_1},\ldots,\mathbf{x_n})$. Now let $(\mathbf{z'_1},\ldots,\mathbf{z'_m})$ be the list obtained by inserting occurrences of $\mathbf{y}$ in the corresponding places of $(\mathbf{y_1},\ldots,\mathbf{y_n})$. Then, for any d in M, $\mathbf{C[z_1\ldots z_m]}[\delta(^d/_x)]$ is content-identical to $\mathbf{D[z'_1\ldots z'_m]}[\gamma(^d/_y)]$. By induction hypothesis, $\vDash_\mathbf{M}\mathbf{C[z_1\ldots z_m]}[\delta(^d/_x)]$ iff $\vDash_\mathbf{M}\mathbf{D[z'_1\ldots z'_m]}[\gamma(^d/_y)]$ for any such d and hence $\vDash_\mathbf{M}\mathbf{(\forall xC)[x_1\ldots x_n]}[\delta]$ iff $\vDash_\mathbf{M}\mathbf{(\forall xD)[y_1\ldots y_n]}[\delta]$.

In view of theorem 2.5, we can write $\vDash_M\alpha$ where α is a reified statement to mean $\vDash_M\mathbf{A[\delta]}$ for $\mathbf{A[\delta]}$ in α. Note that the conditions of the truth definition continue to apply, in the sense that, for example, $\vDash_M\mathbf{[A\vee B,\delta]}$ iff $\vDash_M\mathbf{[A,\delta]}$ or $\vDash_M\mathbf{[B,\delta]}$. Three special cases of the theorem are of independent interest. The first is the case in which $\mathbf{A}$ and $\mathbf{B}$ are identical and the second two are cases in which δ and γ are identical.

Corollary 1 (irrelevance of assignments to extraneous variables) If δ and γ agree on the variables that occur free in A, then $\vDash_M\mathbf{A}[\delta]$ iff $\vDash_M\mathbf{A}[\gamma]$.

Corollary 2 (irrelevance of bound variable identity) If $\mathbf{A}\approx\mathbf{B}$ then $\vDash_M\mathbf{A}[\delta]$ iff $\vDash_M\mathbf{B}[\delta]$.

Corollary 3 (irrelevance of free variable identity) If $\delta(\mathbf{x_i})=\delta(\mathbf{y_i})$ for $1\leq i\leq n$ then $\vDash_M\mathbf{A[x_1,\ldots,x_n]}[\delta]$ iff $\vDash_M\mathbf{A[y_1,\ldots,y_n]}[\delta]$.

Corollary 1 implies, for example, that a sentence of $\mathscr{L}(\pi)$ is true iff it is true under all assignments. More generally, it allows us to omit irrelevant information in our notation for assignments. If all the variables that occur free in $\mathbf{A}$ are among $\mathbf{x_1},\ldots,\mathbf{x_n}$ we write $\vDash_M\mathbf{A(^{d_1,\ldots,d_n}/_{x_1,\ldots,x_n})}$ to indicate that $\mathbf{A}$ is true in M under some (and therefore any) assignment that takes the values $d_1,\ldots,d_n$ for arguments $\mathbf{x_1},\ldots,\mathbf{x_n}$, respectively. We can, in effect, regard $\mathbf{A(^{d_1,\ldots,d_n}/_{x_1,\ldots,x_n})}$ as being formed by replacing $\mathbf{A}$'s variables by the very objects that $\mathbf{A}[\delta]$ concerns. Note that from this perspective content-identical reified formulas are just alphabetically variant formulas. We do not define content identity in this way because of problems of ambiguity that arise when the expressions of the language are among the objects of the model.

Corollary 2 allows us to refer unambiguously to the truth value under an assignment of a statement as well as that of a formula.

As will be seen in the exercises, these three corollaries jointly imply the theorem, and can therefore be seen as providing an analysis of it.

One more special case of theorem 2.5 will be used in the completeness proof. A simple way of obtaining a content-identical copy of a reified formula is to uniformly reletter its variables. By a <u>relettering of variables</u> we mean any one-one function from one set of variables to another. If r is a relettering that is defined on all the variables (both free and bound) in **A**, then we write **r(A)** to indicate the formula that results from replacing each variable **x** in **A** by r(x) and, if **r** is defined on all the variables in the formulas of Γ we write $\mathbf{r(\Gamma)}$ to indicate $\{\mathbf{r(A)}:\mathbf{A}\in\Gamma\}$. If **r** is a relettering then two occurrences of **x** are linked in **A** iff the corresponding occurrences of **r(x)** are linked in **r(A)**. Hence, if δ and γ are assignments satisfying the condition $\delta(x)=\gamma(r(x))$ then $\mathbf{A[\delta]}$ and $\mathbf{r(A)[\gamma]}$ are content-identical. Thus theorem 2.5 implies the following result.

Corollary 3 Suppose **r** is a relettering of variables defined on the variables occurring in **A** and δ and γ are M-assignments satisfying the condition $\delta(r(x))=\gamma(x)$ for all variables x. Then $\vDash_M\mathbf{A}[\delta]$ iff $\vDash_M\mathbf{r(A)}[\gamma]$.

We close the chapter by showing that the formal semantics for PL respects the informal. Consider a rendition ρ that assigns the predicate term π_i to the predicate letter $\mathbf{F_i}$ for all $i\in I$ (and is undefined on other predicate letters). A model $M=(D,\upsilon)$ is said to be the <u>intended</u> model with respect to ρ if D is ρ's domain and, for all $i\in I$ and $d_1,...,d_n$ in D, $\pi_i d_1...d_n$ iff $\upsilon\mathbf{F_i}d_1...d_n$. We want to show that if $M=(D,\upsilon)$ is the intended model with respect to ρ then, for all formulas **A** on which ρ is defined, $\vDash_M\mathbf{A}[\delta]$ iff the open sentence that ρ assigns to **A** is true relative to the objects that δ assigns to its variables. The argument is a straightforward formula induction. The basis case established by the chain: $\vDash_M\mathbf{F_i x_1...x_n}[\delta]$ iff $\upsilon\mathbf{F_i},\delta(\mathbf{x_1}),...,\delta(\mathbf{x_n})$ iff $\pi_i\delta(\mathbf{x_1})...\delta(\mathbf{x_n})$ iff $\pi_i x_1...x_n$ is true relative to $\delta(x_1),...,\delta(x_n)$. The formal and informal interpretations of the truth functional connectives remain the same they were in Part I, so there is nothing new to prove in these cases. The remaining case is established by the chain: $\vDash_M\forall\mathbf{xB}[\delta]$ iff $\vDash_M\mathbf{B}[\delta(^{\mathbf{d}}/_{x})]$ for all $d\in D$ iff, for all $d\in D$, the open sentence that ρ assigns to **B** is true relative to $\delta(^{\mathbf{d}}/_{x})$ iff the open sentence that ρ assigns to $\forall\mathbf{xB}$ is true relative to δ. This argument, it should be noted, only shows that a rendition agrees with the model that is intended with respect to it, *if* there is such a model. In one sense, the formal semantics falls short of the informal. For if the domain predicate associated with a rendition does not have a set as its extension, there is no intended model for the rendition.

Drills, Exercises and Problems

0**[e]**.Find formulas of $\mathscr{L}(\pi)$ that represent forms of the following English sentences under the given key given below:

Key:

D is λx(x is an object)
$\mathbf{P}_1$ is assigned λx(x is a person)
$\mathbf{P}_2$ is assigned λx(x is rice)
$\mathbf{P}_3$ is assigned λx(x is in China)
$\mathbf{P}_4$ is assigned λxy(x owns y)
$\mathbf{P}_5$ is assigned λx(x is Meat)
$\mathbf{P}_6$ is assigned λx(x is Vegetables)
$\mathbf{P}_7$ is assigned λxy(x eats y)
$\mathbf{P}_8$ is assigned λxy(x is a fool of y)
$\mathbf{P}_9$ is assigned λxy(x shaves y)

a. Nobody owns all the rice in China.
b. Some eat only meat, while some eat meat and vegetables.
c. Everybody's somebody's fool.
d. Nobody shaves all and only those who do not shave themselves.

1**[p]**.Suppose we regard the reified formula $\mathbf{A}[\delta]$, not as the ordered pair $<\mathbf{A}, \delta>$, but as the result $\mathbf{A}({}^{d_1,\ldots,d_n}/_{\mathbf{x_1},\ldots,\mathbf{x_n}})$ of substituting the objects $d_1 = \delta(\mathbf{x}_1)$, ... , $d_n = \delta(\mathbf{x}_n)$ for the variables $\mathbf{x_1}$, ... ,$\mathbf{x_n}$ that occur free in **A**. Let d_1 be Mont Blanc, d_2 be the expression $\mathbf{v_1}$, d_3 the expression $\mathbf{v_1v_2}$, d_4 the expression $\mathbf{v_2v_3}$, and d_5 the expression $\mathbf{v_3}$.
a. How might one make sense of the idea of substituting an object for an expression?
b. What are $\mathbf{P^1v_1}({}^{d_1}/_{\mathbf{v_1}})$, $\mathbf{P^1v_1}({}^{d_2}/_{\mathbf{v_1}})$, $\mathbf{P^2v_1v_2}({}^{d_2,d_4}/_{\mathbf{v_1},\mathbf{v_2}})$, and $\mathbf{P^2v_1v_2}({}^{d_3,d_5}/_{\mathbf{v_1},\mathbf{v_2}})$?
c. Show how these examples give rise to difficulties for the proposed account of reified formulas.

*2. By an **x**-formula **A** is meant a formula whose only variable is **x**. By a <u>generalized link map for</u> **A** is meant a pair <F,≡>, where F is the set of occurrences of variables in **A** and ≡ is an equivalence on F which is subject only to the requirement that if an occurrence $\mathbf{x}_1$ of **x** is related by ≡ to an occurrence $\mathbf{x}_2$ of **x** that immediately follows a quantifier occurrence then $\mathbf{x}_1$ must occur in the formula that begins with that quantifier occurrence. Note that the linkage map, as so defined, need not respect the links between variable occurrences as normally conceived. Thus in the formula $\forall\mathbf{x}\forall\mathbf{x}\mathbf{Fxx}$, ≡ might relate only the first and third occurrences of **x** and the second and fourth.

A <u>linked formula</u> is an ordered pair <**A**,f> consisting of an **x**-formula **A** and a generalized link map for **A**. We might think of linked formulas as providing a way of making precise the idea that the links between variable occurrences should be indicated by lines going from the position of the one occurrence to the position of the other. Thus the type of the variable occurrence is irrelevant to its role; all that matters is its position. Redo the results of this section using linked formulas in place of ordinary formulas.

3. a.(indiscernability) Let M=(D,υ) and N=(E,μ) be two models. An <u>indiscernability</u> relation on M and N is a relation R between elements of D and elements of E such that, for any n, and any degree-n predicate **F**, if $d_i R e_i$ for $1 \leq i \leq n$ then $\upsilon\mathbf{F}d_1 \ldots d_n$ iff $\mu\mathbf{F}e_1 \ldots e_n$. We may say that M and N are <u>qualitatively identical</u> if every element in each model's domain is indiscernible from one in the other's, i.e., if there is some indiscernability relation R that satisfies $\forall d \in D\ \exists e \in E$ such that dRe and $\forall e \in E\ \exists d \in D$ such that dRe.

a. Show that qualitatively identical models verify exactly the same formulas. [Hint: Extend the indiscernability relation to assignments by saying that $\delta R\delta'$ if $\delta(\mathbf{x})R\delta'(\mathbf{x})$ for all variables **x**. Then prove by induction on **A** that $\vDash_M\mathbf{A}[\delta]$ iff $\vDash_N A[\delta']$.

b. (Upward Skolem-Loewenheim). Use a to prove that if Γ is satisfiable by a model of cardinality Λ then for any $\Lambda'>\Lambda$, it is satisfiable by a model of cardinality Λ'. [Hint: Let $M=(D,\upsilon)$ be the model that satisfies Γ. Take any object $d\in D$ and any set S disjoint from D of cardinality Λ'. Add the members of S to the domain of M as "duplicates" of d by stipulating that $\mu Fe_1\ldots e_n$ iff $\upsilon Fd_1\ldots d_n$ where $d_i=e_i$ if $e_i\in D$ and $d_i=d$ if $e_i\in S$. Show that the resulting model is qualitatively identical to M.]