Notre Dame Journal of Formal Logic
Volume 23, Number 4, October 1982

Expressibility in Two-Dimensional Languages for Presupposition

MERRIE BERGMANN

Formal two-dimensional languages were shown by Herzberger [4] to provide an interesting philosophical alternative to three-valued languages.[1] He claimed that these languages would probably prove valuable in the investigation of semantic presupposition, and indeed they have. (See the works cited in [1].) In this paper, I characterize the expressive power of certain propositional two-dimensional languages.

I have discussed in detail the motivation for a two-dimensional analysis of presupposition in [2], so I will give only a brief intuitive account of that analysis here.[2] Two-dimensional languages are four-valued languages, where each value assigned to a sentence is the result of two distinct valuations. The first valuation assigns a truth-value to each atomic sentence; and there are two truth-values: *true* and *false*. The second valuation assigns what I call a *security-value* to each atomic sentence. The security-value of a sentence intuitively registers information relevant to determining presuppositions. In English, the relevant information includes: whether names and definite descriptions denote, whether the complements of factive verbs are true, and whether predicates are sortally appropriate to the terms they combine with. Usually, if any of these fails for a sentence, that sentence will have a false presupposition. There are exceptions—for example, 'Santa Claus exists' does not have a false presupposition, although 'Santa Claus lives at the North Pole' does. With the exception of certain constructions such as those that make an explicit attribution of existence, then, a subject-predicate sentence is secure only if the conditions mentioned above are met.

*Some of the material in this paper was developed in my Ph.D. dissertation, *A Presuppositional Theory of Semantic Categories*, University of Toronto, 1976.

Received April 20, 1981

On the basis of the assignments to atomic sentences, the two valuations generate values for complex sentences. Presupposition is defined in terms of security, rather than in terms of truth-value:

A presupposes B ($A \gg B$) iff whenever A is secure B is true.

Thus, 'Santa Claus lives at the North Pole' presupposes both 'Santa Claus exists' and 'The North Pole exists' since the conditions under which the first sentence is secure will make the other two sentences true. And, if we have a policy that makes a disjunction secure iff its disjuncts are both secure, 'Either Santa Claus lives at the North Pole or two plus three equals five' will inherit the presuppositions of 'Santa Claus lives at the North Pole' (the disjunction may have other presuppositions as well). This example brings out an important point concerning the two-dimensional analysis of presupposition: since presupposition is defined independently of truth-value, a sentence may be true even though it has a false presupposition.[3] On the other hand, we may choose to stipulate for certain forms of sentences that their truth entails the truth of their presuppositions. An example is "internal negation": the internal negation of a sentence A is defined as true iff A is both false and secure; hence if the internal negation is true then the presuppositions of A are all true. With these introductory remarks, I now specify a two-dimensional language for presupposition.

The language L has the following primitive vocabulary (I use expressions of formal languages as their own names):

Atomic sentences: $P_1, P_2, P_3, \ldots$
Unary connectives: $\neg$, T, γ
Binary connective: $\vee$
Punctuation: (,)

Sentences are defined as usual. A valuation V of L assigns to each atomic sentence A an ordered pair of values $\langle V_t(A), V_s(A)\rangle$, where $V_t(A) \in \{1, 0\}$ and $V_s(A) \in \{1, 0\}$. A sentence A is true on V if $V_t(A) = 1$, false if $V_t(A) = 0$, secure if $V_s(A) = 1$, and nonsecure if $V_s(A) = 0$. I abbreviate the four values which V may assign to A by omitting the angle brackets and commas, e.g., '$\langle 1, 1\rangle$' is written as '11'. A valuation assigns values to complex sentences in accordance with the following matrices:

A	$\neg A$
11	01
01	11
10	00
00	10

$A \vee B$	11	01	10	00
11	11	11	10	10
01	11	01	10	00
10	10	10	10	10
00	10	00	10	00

A	TA
11	11
01	01
10	11
00	01

A	γA
11	11
01	11
10	01
00	01

The truth-conditions for the connectives $\neg$ and $\vee$ of L are defined as in classical bivalent logic. Thus sentences of L containing only these connectives that are logically true on the standard bivalent interpretation are also logically true in L, where a sentence A is logically true in L iff for every valuation V, $V_t(A) = 1$. (The converse also holds.) Nonsecurity is dominant for these connectives: a compound sentence is nonsecure whenever at least one of its components is.

On the other hand, sentences governed by T and γ are logically secure—secure on every valuation. Hence TA is not generally equivalent to A; although the two sentences always have the same truth-value, they cannot always be substituted one for the other *salva veritate.* These two connectives are connectives of "semantic ascent": the sentences TA and γA are intuitively *about* the sentence A. The connective T is the truth-connective; TA may be read as: A is true. The connective γ is the security-connective; γA may be read as: A is secure, or, better yet, A has no false presuppositions. For if A is secure, then by definition every sentence presupposed by A is true.[4] Using γ, we may define internal negation $\sim$:

$$\sim A =_{df} \neg(A \vee \neg\gamma A)$$

This connective has the following matrix:

A	$\sim A$
11	01
01	11
10	00
00	00

I shall now discuss the expressive powers of L and of several closely related languages. First, the definition of two-dimensional operations:

An n-ary *classical operation* θ is any function taking n members of $\{1, 0\}$ into $\{1, 0\}$.

An n-ary *two-dimensional operation* θ is any function taking n members of $\{11, 01, 10, 00\}$ into $\{11, 01, 10, 00\}$.

An n-ary two-dimensional operation θ is *expressible* in a language just in case there is a sentence B in that language containing exactly n atomic sentences $A_1, \ldots, A_n$ such that B has the value $\theta(t_1s_1, \ldots, t_ns_n)$ when $A_1, \ldots, A_n$ are assigned the values $t_1s_1, \ldots, t_ns_n$, respectively. In this case, θ is *expressed* by B. I shall also use connectives of L as the names of the operations specified in their matrices.[5]

An n-ary *product operation* θ is a two-dimensional operation

$$\theta(t_1s_1, \ldots, t_ns_n) = \langle\psi(t_1, \ldots, t_n), \delta(s_1, \ldots, s_n)\rangle$$

where ψ and δ are n-ary classical operations.

Let $L\{\neg, \vee\}$ be the sublanguage of L containing only $\neg$ and $\vee$ as connectives. Then

1. Every operation expressible in $L\{\neg, \vee\}$ is a product operation.

Proof: Straightforward.

However, not every product operation is expressible in $L\{\neg, \vee\}$. For example, the operation T is not. The reason is simple to see: an n-ary operation expressible in $L\{\neg, \vee\}$ will take n arguments into one of the values 11, 01 iff each argument is one of the values 11, 01. An operation of this sort is a weak product operation:

An n-ary *weak product operation* θ is a product operation
$$\theta(t_1s_1, \ldots, t_ns_n) = \langle\psi(t_1, \ldots, t_n), \min(s_1, \ldots, s_n)\rangle$$
where ψ is an n-ary classical operation and min is the n-ary classical operation that maps $a_1, \ldots, a_n$ into 1 iff $a_1, \ldots, a_n$ are all 1.

The expressive power of $L\{\neg, \vee\}$ is then circumscribed as:

2. All and only weak product operations are expressible in $L\{\neg, \vee\}$.

Proof: It is straightforward to show that only weak product operations are expressible in $L\{\neg, \vee\}$. To show that all weak product operations are expressible, I introduce the following matrix scheme for an n-ary two-dimensional operation:

Row	A_1	$\ldots$ A_n	B
1	$t_1^1s_1^1$	$\ldots$ $t_n^1s_n^1$	t_1s_1
2	$t_1^2s_1^2$	$\ldots$ $t_n^2s_n^2$	t_2s_2
.		.	.
.		.	.
.		.	.
4^n	$t_1^{4^n}s_1^{4^n}$	$\ldots$ $t_n^{4^n}s_n^{4^n}$	$t_{4^n}s_{4^n}$

The rows to the left of the vertical line represent the different combinations of values the atomic sentences $A_1, \ldots, A_n$ may have. If the matrix represents a weak product operation θ, then there is some n-ary classical operation ψ such that the value t_is_i in row i is $\langle\psi(t_1^i, \ldots, t_n^i), \min(s_1^i, \ldots, s_n^i)\rangle$. The task is to show that for any such ψ, there is a sentence B of $L\{\neg, \vee\}$ that expresses the operation θ. For each row i, let B_i be the sentence $(\ldots(A_1{}^* \vee A_2{}^*) \vee \ldots \vee A_n{}^*)$, where $A_j{}^*$ is $\neg A_j$ if t_j^i is 1, and $A_j{}^*$ is A_j otherwise. Then B_i is false on row i and true on all other rows; and B_i is secure on row j iff $s_1^j, \ldots, s_n^j$ are all 1. Then the desired sentence B is $(\ldots(B_1{}^* \vee B_2{}^*) \vee \ldots \vee B_{4^n}{}^*)$, where $B_i{}^*$ is $\neg B_i$ if t_i is 1 and $\neg(B_i \vee \neg B_i)$ otherwise.[6]

Although $L\{\neg, \vee\}$ is not functionally complete in the sense that every two-dimensional operation is expressible, it is *truth-functionally* complete. A two-dimensional language is truth-functionally complete iff for each n-ary classical operation ψ, there is an n-ary two-dimensional operation θ expressible in that language such that

$$\theta(t_1s_1, \ldots, t_ns_n) = \langle\psi(t_1, \ldots, t_n), \delta(t_1, \ldots, t_n, s_1, \ldots, s_n)\rangle$$

for some $2n$-ary classical operation δ.

3. $L\{\neg, \vee\}$ is truth-functionally complete.

Proof: Straightforward, as all weak product operations are expressible in $L\{\neg, \vee\}$.

The expressive power of a two-dimensional language has important connections with the presupposition relation generated by that language. It may determine which presuppositions of the components of a complex sentence can be inherited by the sentence itself, and whether a complex sentence can have presuppositions not shared by its components. And it may determine that the presupposition relation is significant for a language only under certain conditions.

If we take all valuations of $L\{\neg, \vee\}$ to be admissible, then the presupposition relation is trivial, for each sentence presupposes all and only those sentences that are logically true. When all valuations for a language are admissible, I shall say that the A-policy has been adopted. Given the A-policy, the feature of $L\{\neg, \vee\}$ that is responsible for the trivialness of presupposition is that truth-values and security-values are defined independently for all sentences. This follows from result 2. And if we add the connective T (which does not express a weak product operation) to $L\{\neg, \vee\}$, the expanded language will still have a trivial presupposition relation if the A-policy is adopted. More generally,

4. Any two-dimensional language in which only product operations are expressible has a trivial presupposition relation if the A-policy is adopted.

Proof: (i) Let B be a sentence that is logically true. Then, trivially, for every sentence A, $A \gg B$. (ii) Let B be a sentence that is false on some valuation V, and let A be a sentence that is secure on some valuation V'. Let V'' be the valuation such that for every atomic sentence C, $V_t''(C) = V_t(C)$ and $V_s''(C) = V_s'(C)$. Then $V_s''(A) = 1$ and $V_t''(B) = 0$; hence $A \not\gg B$. (iii) Let A be a sentence that is logically nonsecure. Then trivially, for every sentence B, $A \gg B$.

A sentence expressing a product operation may be logically nonsecure, hence the third clause. In this case the relation is still trivial, for such a sentence presupposes every sentence.

Given result 4, a language in which only product operations are expressible will have a significant presupposition relation only if the A-policy is dropped. Dropping the A-policy is tantamount to dropping the independence of the values assigned to atomic sentences. For example, let us restrict the admissible valuations to those on which either P_1 is nonsecure or P_2 is true (which we may choose to do if P_1 symbolizes 'Santa Claus is jolly' and P_2 symbolizes 'Santa Claus exists'). Then it follows that $P_1 \gg P_2$, and no other atomic sentence presupposes P_2. But P_2 is not logically true. Whatever policy as to admissible valuations is adopted, we have:

5. A complex sentence of $L\{\neg, \vee\}$ presupposes all sentences presupposed by at least one of its components.[7]

Proof: A straightforward mathematical induction.

This is the so-called "cumulative hypothesis" regarding the projection of presuppositions.

However, the full language L has a nontrivial presupposition relation even when the A-policy is adopted. For γA is not in general logically true, yet

6. In L, each sentence A presupposes γA.

It is the presence of γ, then, that makes the presupposition relation significant in L with the A-policy, since sentences containing only the other connectives express product operations.

The operation γ is a semi-product operation:

An n-ary *semi-product operation* θ is a two-dimensional operation

$$\theta(t_1s_1, \ldots, t_ns_n) = \langle\psi(t_1, \ldots, t_n, s_1, \ldots, s_n), \delta(s_1, \ldots, s_n)\rangle$$

where ψ is a $2n$-ary classical operation and δ is an n-ary classical operation.

Every product operation is a semi-product operation, but the converse does not hold. The defined connective $\sim$, like γ, forms sentences expressing semi-product operations that are not product operations. The general idea motivating the addition of these operations is that the truth-value of a sentence may be determined, in part, by whether the presuppositions of its components are true. Such is the case with "internal negation". On the other hand, the security-value of a sentence expressing a semi-product operation (registering the status of its presuppositions) does not depend on the truth-values of its components.

The operations γ and $\sim$ are not definable in terms of the other connectives of L. It is also true that

7. The operation T is not definable in terms of $\neg$, $\vee$, and γ.

Proof: By mathematical induction on the length of sentences of L with one atomic component, which do not contain T. A property all such sentences have is that the operation expressed by each:

i. takes 10 into 11 iff it takes 00 into 11, and
ii. takes 10 into 01 iff it takes 00 into 01.

Clearly, the operation T does not have properties i and ii; hence it is not definable in terms of the remaining connectives of L.

None of the four primitive operations of L can be defined in terms of the other three.

L, like $L\{\neg, \vee\}$, is truth-functionally complete but not functionally complete. The operations $\circledast$ and $\otimes$, for example, are not expressible in L:

A	$\circledast A$
11	11
01	11
10	11
00	00

A	$\otimes A$
11	00
01	11
10	11
00	00

It can be verified that every unary operation expressible in L maps 10 into 11 or 01 iff it maps 00 into 11 or 01, and the operations considered here do not have that property.

The operations $\circledast$ and $\otimes$ are not semi-product operations (and hence not product operations). The operation expressed by $\otimes$ is not security-preserving, but $\circledast$ and all operations expressible in L are:

An n-ary *security-preserving operation* θ is a two-dimensional operation
$$\theta(t_1s_1, \ldots, t_ns_n) = \langle\psi(t_1, \ldots, t_n, s_1, \ldots, s_n), \delta(t_1, \ldots, t_n, s_1, \ldots, s_n)\rangle$$
where ψ is a $2n$-ary classical operation and δ is a $2n$-ary classical operation which maps arguments $a_1, \ldots, a_{2n}$ into 1 if $a_{n+1}, \ldots, a_{2n}$ are all 1.

Since each sentence A of L presupposes γA, and γA is true only if A is secure, the fact that all operations of L are security-preserving guarantees that a complex sentence has a false presupposition only if at least one of its components has a false presupposition. This seems correct. Many expressions of English may be responsible for generating semantic presuppositions but no one, to my knowledge, has suggested that expressions such as 'and', 'or', and the like are so responsible.[8]

But we have seen that $\circledast$ also has a property that those operations expressible in L do not have. Namely, the operation takes some, but not all, nonsecure sentences into secure ones. Every operation expressible in L is, as it were, security-functional; that is, the security-value of a sentence of L is a function of the security-values of its components. The semi-product two-dimensional operations are exactly the security-functional ones.

However, not all security-preserving semi-product operations are expressible in L. Every operation expressible in L is a definitive operation.

An n-ary *definitive operation* θ is a semi-product operation
$$\theta(t_1s_1, \ldots, t_ns_n) = \langle\psi(t_1, \ldots, t_n, s_1, \ldots, s_n), \delta(s_1, \ldots, s_n)\rangle$$
which ψ is a $2n$-ary classical operation and δ is an n-ary classical operation which maps arguments $a_1, \ldots, a_n$ into 0 iff for some i $(1 \leqslant i \leqslant n)$, $a_i = 0$ and δ maps the sequence of n arguments consisting of 1's except in the i^{th} place into 0.

Every definitive operation is a security-preserving operation, but the converse does not hold. For example, the security-preserving operation in the scheme

A	
11	11
01	01
10	11
00	00

is not definitive. A moment's reflection shows why this operation is not expressible in L: If A occurs only in the scope of T or γ in a sentence of L which has A as its only atomic component, then that sentence can never be nonsecure. If A has an occurrence not in the scope of T or γ in a sentence of L, then that sentence will be nonsecure whenever A is.

The fact that all operations of L are definitive entails that for any sentence A of L, there is a (possibly empty) subset $s(A)$ of its atomic components such that A is secure iff all the members of $s(A)$ are secure. So a sentence A of L has a false presupposition iff some member of $s(A)$ has a false presupposition. Consequently, we can explain the success of A's presuppositions in terms of the success of the presuppositions of each of $s(A)$'s member. Whether or not the presuppositions of atomic components of A that are not in $s(A)$ are true is irrelevant to the presupposition success or failure of A. In L, it is easy to see which atomic components will not be in $s(A)$: exactly those components which occur only in the scope of T or γ in A.

The expressive power of L is limited to definitive operations:

8. All and only definitive operations are expressible in L.

Proof: To prove that all definitive operations are expressible in L, I make use of the matrix scheme in the proof of result 2. I will characterize the relations between values in the matrix scheme under the assumption that it represents a definitive operation θ. For any row i, let $N(i) = \{k: s^i_k = 0\}$. Let $\leqslant$ be the following partial ordering:

Row i $\leqslant$ row j (abbreviated ($i \leqslant j$) iff $N(i) \subseteq N(j)$.

So if $i \leqslant j$, row j has security value 0 in at least every column to the left of the vertical line in which row i has security value 0. Since θ is a definitive operation, it follows that for arbitrary row i, $s_i = 0$ iff for some row j such that $j \leqslant i$ and $N(j)$ has exactly one member, $s_j = 0$. (And hence it follows that $s_i = 1$ if $s^i_1, \ldots, s^i_n$ are all 1.)

Now we specify for each row i a sentence Q_i that has the value 11 on row i, and 01 on every other row. Let Q_i be $(\ldots (A_1{}^* \ \& \ A_2{}^*) \ \& \ldots \& \ A_n{}^*)$, where

$$(B \ \& \ C) =_{df} \neg(\neg B \vee \neg C)$$

and

$$A_j{}^* \text{ is} \begin{cases} TA_j \ \& \ \ \gamma A_j \text{ if } t^i_j s^i_j \text{ is } 11 \\ \neg TA_j \ \& \ \ \gamma A_j \text{ if } t^i_j s^i_j \text{ is } 01 \\ TA_j \ \& \ \neg\gamma A_j \text{ if } t^i_j s^i_j \text{ is } 10 \\ \neg TA_j \ \& \ \neg\gamma A_j \text{ if } t^i_j s^i_j \text{ is } 00. \end{cases}$$

Next we specify for each row i a sentence R_i:

If $t_i s_i = 11$, let R_i be Q_i
If $t_i s_i = 01$, let R_i be $Q_i \ \& \ (TA_1 \ \& \ \neg TA_1)$.

In these cases, R_i will have value $t_i s_i$ on row i and value 01 on all other rows.

Let row i be such that $s_i = 0$. Here the definition of R_i is more complicated. Let $\leqslant!(i)$ be the collection of rows j such that $j \leqslant i$, $N(j)$ has exactly one member, and $s_j = 0$. Let $n(i)$ be $\{A_k: s^j_k = 0$ for some row $j \in \leqslant!(i)\}$. Because θ is a definitive operation, both $N(i)$ and $n(i)$ are nonempty. Moreover, if $j \in \leqslant!(i)$

then every row m such that $j \leqslant m$ will also have $s_m = 0$. Let $\leqslant!!(i)$ be the collection of these rows m, i.e., $\leqslant!!(i) = \{\text{row } m\text{: there is a } j \in \leqslant!(i) \text{ such that } j \leqslant m\}$. Now define R_i for the remaining cases:

If $t_i s_i = 10$, let R_i be Q_i & $(\ldots(A_1+ \,\&\, A_2+) \,\&\, \ldots \,\&\, A_n+)$, where

$$A_j+ \text{ is } \begin{cases} A_j \vee \neg A_j \text{ if } A_j \in n(i) \\ T(A_j \vee \neg A_j) \text{ otherwise.} \end{cases}$$

In this case R_i has value 10 on row i, 00 on all rows $j \in \leqslant!!(i)$ other than i, and value 01 on all other rows.

If $t_i s_i = 00$, let R_i be Q_i & $(\ldots(A_1 0 \,\&\, A_2 0) \,\&\, \ldots \,\&\, A_n 0)$, where

$$A_j 0 \text{ is } \begin{cases} A_j \,\&\, \neg A_j \text{ if } A_j \in n(i) \\ T(A_j \,\&\, \neg A_j) \text{ otherwise.} \end{cases}$$

In this case, R_i will have value 00 on row i and on all other rows $j \in \leqslant!!(i)$, and value 01 on all other rows.

Finally, let B be the sentence $(\ldots(R_1 \vee R_2) \vee \ldots \vee R_{4^n})$. This sentence has value $t_i s_i$ on each row i.

Now we prove that only definitive operations are expressible in L. It is straightforward to prove that only security-preserving operations are expressible in L. If a non-definitive operation were expressed by a sentence of L, then, the matrix for the sentence would fall under one of two cases:

Case a. For some row i, $s_i = 0$ and for every row j such that $j \leqslant i$ and $N(j)$ has one member, $s_j = 1$.

Case b. For some row i, $s_i = 1$ and for some row j such that $j \leqslant i$ and $N(j)$ has one member, $s_j = 0$.

I will consider only Case a, for Case b is handled similarly. By examining the matrices for the primitive connectives of L, we see that if $s_i = 0$ for some row i in the matrix scheme, then for some A_j, $s_j^i = 0$ and A_j has at least one occurrence not within the scope of T or γ in B. But then on every row k such that $s_j^k = 0$, $s_k = 0$ as well. In particular, for the row m such that $s_j^m = 0$ and $N(m)$ has one member, we will also have $s_m = 0$. But then Case a cannot describe an operation expressible in L.[9]

Adding the operation $\circledast$ to L will generate a language $L^{+\circledast}$ in which every security-preserving operation is expressible:

9. All and only security-preserving operations are expressible in $L^{+\circledast}$.

Proof: It is straightforward to prove that only security-preserving operations are expressible in $L^{+\circledast}$. The proof that all such operations are expressible borrows from the proof of 8. For each row i of the matrix of a security-preserving operation, let Q_i be defined as in 8. We now define for each row i a sentence R_i which is assigned the value $t_i s_i$ in row i, and 01 in all other rows. If $t_i s_i$ is 11 or 01, R_i is defined as in 8.

If $t_i s_i$ is 10, let R_i be $\neg\circledast\neg(Q_i \,\&\, (\ldots((A_1 \vee \neg A_1) \,\&\, (A_2 \vee \neg A_2)) \,\&\, \ldots \,\&\, (A_n \vee \neg A_n))$

If t_is_i is 00, let R_i be $\sim\circledast\neg(Q_i \vee (\ldots((A_1 \& \neg A_1) \& (A_2 \& \neg A_2)) \& \ldots \& (A_n \& \neg A_n))$.

($\sim$ is definable in terms of the primitive connectives of $L^{+\circledast}$.) In verifying that these latter sentences have the desired values, note that security-preservation entails that at least one of $A_1, \ldots, A_n$ has security-value 0 on row i if $s_i = 0$.

Finally, let B be the sentence $(\ldots(R_1 \vee R_2) \vee \ldots \vee R_{4^n})$. This sentence has value t_is_i on each row i.[10]

And adding the operation $\otimes$ to L will generate a language $L^{+\otimes}$ in which every two-dimensional operation is expressible:

10. $L^{+\otimes}$ is functionally complete.

Proof: Here we borrow again from the proof of 8. For each row i, let Q_i be defined as in 8. We now define for each row i a sentence R_i which is assigned the value t_is_i in row i, and 01 in every other row. If t_is_i is 11 or 01, R_i is defined as in 8.

If t_is_i is 10, let R_i be $Q_i \& \neg\otimes Q_i$
If t_is_i is 00, let R_i be $\sim\otimes Q_i$.

These latter two cases are not borrowed from 9 (even though $\circledast$ is definable in $L^{+\otimes}$) because in 9 security-preservation guaranteed that t_is_i is 10 or 00 only if for some j, $t_j^is_j^i = 0$. The restriction of security-preservation has been dropped.

Finally, let B be the sentence $(\ldots(R_1 \vee R_2) \vee \ldots \vee R_{4^n})$. This sentence has value t_is_i on each row i.

NOTES

1. Despite the common name, two-dimensional languages in the sense of Herzberger should not be confused with two-dimensional modal languages in the sense of Segerberg [8].

2. My interpretation of the semantics differs from the interpretation in Herzberger [4]; I discussed this difference in [2].

3. In a trivalent semantics, with the definition

   A presupposes *B* iff whenever *A* is true or false *B* is true,

   it is impossible for a true sentence to have a false presupposition.

4. The semantics for the connectives of L are discussed more fully in [2].

   Insofar as they form sentences that are logically secure, connectives of semantic ascent bear an affinity to the trivalent Frege-Bochvar horizontal which eliminates truth-value gaps (truth-value gaps playing the role in the trivalent definition of presupposition that nonsecurity plays in the two-dimensional definition). See Herzberger [4] and [6] on the Frege-Bochvar horizontal. The stipulation of logical security for sentences governed by the connectives T and γ is closely related to the view concerning trivalent or supervaluational semantics, that although a language may be nonbivalent, our semantic assessments of statements in that language should be bivalent. Cf. van Fraassen [10], p. 168.

5. The expression t_is_i stands for one of the ordered pairs 11, 01, 10, 00, where t_i is the first member of the pair and s_i is the second member.

The terminology 'classical operation', 'product operation', 'weak product operation', and 'semi-product operation' is from Herzberger [5]. In those notes, a result equivalent to my result 1 is stated and a result equivalent to my result 2 is conjectured. Comments in those notes first suggested to me how the operations expressible could be circumscribed, and that led to my result 8.

In [4], Herzberger shows that his languages L_1 and L_2 are truth-functionally complete; each of his languages contains the operations $\neg$ and $\vee$ (see my result 3).

6. The language $C \times K_W^2$ in Martin [7] is equivalent to $L\{\neg, \vee\}$, insofar as exactly the weak product operations are expressible in $C \times K_W^2$.

7. But it need not presuppose only those sentences. E.g., let the admissible valuations include all and only those which make one of the following two combinations of assignments to P_1, P_2, and P_3:

P_1	P_2	P_3
11	10	01
10	11	01

Then $(P_1 \vee P_2) \gg P_3$, but $P_1 \not\gg P_3$ and $P_2 \not\gg P_3$.

8. In [3], Grice argued that certain logical connectives in English generate "implicatures", and implicatures are often taken to be presuppositions. It might be thought that this refutes the claim just made. But it is common practice now to distinguish between semantic and pragmatic presuppositions (see Stalnaker [9]), and I would argue that the implicatures generated by 'and', 'or', 'but', and 'if-then' are pragmatic rather than semantic.

9. The expressive powers of L_1 and L_2 in Herzberger [4] are stronger than $L\{\neg, \vee\}$ but weaker than L. Semi-product operations which are not product operations are expressible in both L_1 and L_2, but some semi-product operations are expressible in neither language. There are no logically secure sentences in L_1, and γ but not T is definable in L_2.

10. The language $C \times K_S^2$ in Martin [7] is expressively weaker than $L^{+\circledast}$, and neither weaker nor stronger than L. Some operations in $C \times K_S^2$ are not semi-product operations, but there are no logically secure sentences in $C \times K_S^2$.

REFERENCES

[1] Bergmann, M., "*Only*, *even*, and clefts in two-dimensional logic," in *Proceedings of the 1981 International Symposium on Multiple-Valued Logic,* The Institute of Electrical and Electronics Engineers, pp. 117-123.

[2] Bergmann, M., "Presupposition and two-dimensional logic," *Journal of Philosophical Logic,* vol. 10 (1981), pp. 27-53.

[3] Grice, H. P., "Logic and conversation," in *Syntax and Semantics: Speech Acts,* Vol. 3, eds. P. Cole and J. H. Morgan, Academic Press, New York, 1975.

[4] Herzberger, H. G., "Dimensions of truth," *Journal of Philosophical Logic,* vol. 2 (1973), pp. 535-556.

[5] Herzberger, H. G., "Product and semi-product logics," typescript, University of Toronto, 1975.

[6] Herzberger, H. G., "Truth and modality in semantically closed languages," in *The Paradox of the Liar,* ed. R. L. Martin, Yale University Press, New Haven, 1970.

[7] Martin, J., "A many-valued semantics for category-mistakes," *Synthese,* vol. 31 (1975), pp. 63-83.

[8] Segerberg, K., "Two-dimensional modal logic," *Journal of Philosophical Logic,* vol. 2 (1973), pp. 77-96.

[9] Stalnaker, R., "Presuppositions," *Journal of Philosophical Logic,* vol. 2 (1973), pp. 447-457.

[10] van Fraassen, B. C., *Formal Semantics and Logic,* Macmillan, New York, 1971.

Department of Philosophy
Dartmouth College
Hanover, New Hampshire 03755