Necessary and Sufficient Conditions for a Phenotype System to Have a Factor-Union Representation

GEORGE MARKOWSKY
Computer Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598

Received 8 June 1982; revised 22 April 1983

ABSTRACT

This paper presents an algorithm for determining whether a phenotype system has a factor-union representation, i.e., whether sets of properties (factors) can be assigned to alleles so that the phenotype of each genotype is determined by the union of the factor sets of the alleles. If such a representation is possible, the algorithm can construct one using p or fewer factors, where p is the number of phenotypes. Refinements to the algorithm are presented that often construct representations using few factors. An example is presented to show that some phenotype systems can be represented only by using almost as many factors as there are phenotypes in the system. Finally, it is shown that the problem of finding factor-union representations for multilocus systems reduces to the problem for 1-locus systems.

1. INTRODUCTION

Cotterman [1] investigated the existence of factor-union representations for phenotype systems and derived some partial answers. This paper completely solves the problem and presents an algorithm which always determines whether a phenotype system has a factor-union representation. This algorithm is efficient enough to be used manually on some large examples.

If a factor-union representation is possible for a phenotype system, the algorithm can construct one using the phenotypes themselves as factors. In general, this may use many more factors than necessary, since it is possible to represent 2^k distinct phenotypes with k factors. Refinements to the algorithm can significantly reduce the number of factors required. As an example, two representations are constructed for an example left unresolved in [1]. The first one uses 12 factors, while the second one uses only 4.

An example in this paper shows that some phenotype systems require almost as many factors to represent them as they have phenotypes. This

52 Vanderbilt Ave., New York, NY 10017 0025-5564/83/$03.00

example is also of interest because the number of factors required grows quadratically in the number of alleles, which shows that simple representations are not always possible.

The representations constructed by the algorithm need not be based on actual physical factors. Determining a physical basis for a representation is a problem that cannot be solved purely mathematically, but the mathematical results can guide the search for suitable physical factors. Of course, if the algorithm shows that no representation is possible, much valuable time can be saved.

In Section 5, techniques are developed to translate multilocus systems into 1-locus systems so that finding a factor-union representation for one system finds it for the other. Thus, this algorithm can be used with any phenotype system.

2. DEFINITIONS AND NOTATION

Listed below are several simple definitions and concepts required for understanding this paper. Many of them come from [1], which should be consulted for general background on this topic. A few come from the theory of partial orders.

(1) Lowercase letters denote alleles, Greek letters denote genotypes, and uppercase letters denote phenotypes. Pairs of lowercase letters will be used when it is desirable to show the alleles contained in a genotype. Bracketed expressions such as $\langle ab, cd \rangle$ are used to show the genotypes in a phenotype.

(2) If the two alleles in a genotype are the same, it is called a *homozygote*. If they are different, it is called a *heterozygote*.

(3) The symbol $+$ denotes the union of individual sets, Σ the union of a family of sets, and Π the intersection of a family of sets. The symbols $\leqslant$ and $<$ denote set inclusion and proper set inclusion respectively.

(4) A *factor-union representation* of a phenotype system Φ consists of a set X of factors, called the *factor set*, and an assignment f of subsets of X to the alleles of Φ with the following properties:

(a) If ab and cd are two genotypes in the same phenotype,

$$f(a)+f(b)=f(c)+f(d).$$

The set $f(a)$ is called the *factor set* of a. If ab is in P then

$$f(a)+f(b)=\sum_{x \text{ in } P} f(x).$$

The quantity on the right side is abbreviated $f(P)$ and called the *factor set* of P. It is the union of the factor sets of all alleles in P.

(b) If P and Q are distinct phenotypes, $f(P) \neq f(Q)$. This condition states that phenotypes are determined by their factors.

(5) A *partially ordered set* (*poset*) is a set X together with a partial order $<$ defined on its elements. A *partial order* is any binary relation that satisfies the following two rules:

(a) *Reflexive property.* For all x in X, $x < x$ never holds.

(b) *Transitive property.* For all x, y, z in X, if $x < y$ and $y < z$, then $x < z$.

The string $x \leqslant y$ means that either $x < y$ or $x = y$. By convention, $y > x$ and $y \geqslant x$ mean the same as $x < y$ and $x \leqslant y$. The same symbols are used for partial order and set inclusion. This should cause no confusion, since set inclusion is a form of partial order.

(6) Let $(X, <)$ be a poset and S a subset of X. An element z of X is called an *upper bound for* S if it is greater than or equal to every element of S. An element is the sup or *least upper bound* of S if it is an upper bound that is less than or equal to every upper bound of S. In general, a subset of a poset need not have a sup. A poset in which every finite nonempty subset has a sup is called a *semilattice*.

(7) If X is a set, Power(X) denotes the *power set* of X, which is the set of all subsets of X. Set inclusion is a partial order on Power(X). With this order, Power(x) is a semilattice in which the sup of a family of sets is its union.

3. THE ALGORITHM

The algorithm attempts to construct a partial order on the set of phenotypes using necessary inclusion relations among the phenotypes. If the algorithm fails to construct a partial order, Theorem 2 shows that two distinct phenotypes will have identical factor sets in any factor-union representation. Thus no factor-union representation is possible. On the other hand, if the algorithm constructs a partial order, a factor-union representation can be constructed using Theorem 1.

The following describes the part of the algorithm that determines whether a factor-union system is possible. Theorem 2 proves that the algorithm is correct and shows how to construct a factor-union representation.

FACTOR-UNION REPRESENTATION ALGORITHM (*Existence Part*)

1. Represent each phenotype of Φ by a point.

2. For each ordered pair of distinct phenotypes (P_i, P_j), draw an arrow from P_i to P_j if and only if some genotype contained in P_i has all its alleles

belonging to some genotype in P_j. This step creates a directed graph having the phenotypes as its points and arrows as its edges. This graph is called Graph(Φ). If there is an arrow from P_i to P_j, then in any factor-union representation, the factor set of P_i is a subset of the factor set of P_j.

3. If Graph(Φ) has a *cycle*, i.e., a sequence of arrows from a point to itself, Φ does not have a factor-union representation. A cycle in Graph(Φ) implies that all the factor sets connected by the arrows must be equal.

4. If Graph(Φ) does not have a cycle, define a partial order $<$ on Graph(Φ) as follows: $P_i < P_j$ if and only if there is a sequence of arrows from P_i to P_j. A sequence of arrows from one point to another is called a *path*. [The condition that there are no cycles in Graph(Φ) implies that $<$ is reflexive. Clearly, $<$ is also transitive and so a partial order. If $P_i \leqslant P_j$, the factor set of P_i is a subset of the factor set of P_j in any factor-union representation.]

5. The next stage of the algorithm adds arrows to Graph(Φ) and extends the partial order $<$ until either a cycle appears in Graph(Φ) or there is no further way to carry out the following rule.

ARROW-ADDING RULE

Let *aa* be in P_1, *bb* in P_2, *ab* in P_3, and P_4 a phenotype such that $P_1 \leqslant P_4$ and $P_2 \leqslant P_4$ in Graph(Φ). Then add an arrow from P_3 to P_4 unless there already is such an arrow.

As arrows are added to Graph(Φ), $<$ must be updated. The partial order is always defined in the same way relative to Graph(Φ), i.e., $P < P^*$ if and only if there exists a path in Graph(Φ) from P to P^*.

To use the Arrow-Adding Rule, choose a heterozygote *ab*. Then find the phenotypes P_1 and P_2 that contain the homozygotes *aa* and *bb*. Next, find the upper bounds of P_1 and P_2. Finally, add the arrows from P_3 to the upper bounds. *The algorithm always halts*, since Graph(P) can have at most $p(p-1)/2$ arrows without having a cycle.

Instead of drawing all arrows in Graph(P), I will only draw an arrow from P to Q if $P < Q$ *and* there does not exist an R such that $P < R < Q$. In this case Q is said to *cover* P. This representation, called the *Hasse diagram*, contains all the information of the original graph, but is easier to draw and interpret. The original order can be reconstructed from this reduced order by transitivity. To further simplify the drawings, most arrow heads will be omitted. Line segments will point from lower to higher points.

Example 1 (System Φ_1). This example, from Cotterman [1, p. 8], has 6 alleles and 12 phenotypes. Cotterman was unable to determine whether this system had a factor-union representation.

Figure 1 shows the Hasse diagram of Graph(Φ_1) after the algorithm has finished with stage 2. There are no cycles in it, so that the algorithm proceeds

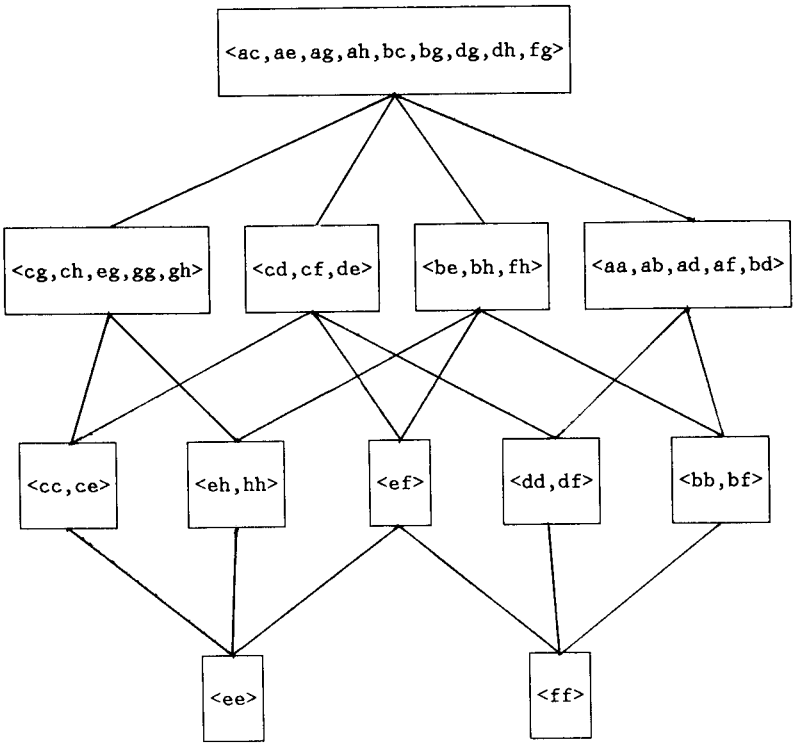

FIG. 1. The Hasse diagram of Graph(Φ_1) after being processed by the algorithm.

to stage 4. Stage 5 lasts a short time, since the Arrow-Adding Rule cannot be applied. Thus Φ_1 has a factor-union representation.

A factor-union representation with 12 factors can be constructed for Φ_1 using Theorem 2. This representation will not be displayed since a representation using only 4 factors is produced in Section 4.

Example 2 (System Φ_2). This example also has 6 alleles. The phenotypes are shown in Figure 2, which displays the results after the first stage of the algorithm is completed. Graph(Φ_2) does not have any cycles, and there are many ways to apply the Arrow-Adding Rule. Theorem 2 shows that the Arrow-Adding Rule can be applied to heterozygotes in any order. If, however, a factor-union representation does not exist, some sequences of heterozygotes will produce a cycle faster than others. For this example, the Arrow-Adding Rule will be applied to the heterozygotes gx, hx, hu, uy, and ty in order. The intermediate steps are detailed below and the results shown in Figure 3.

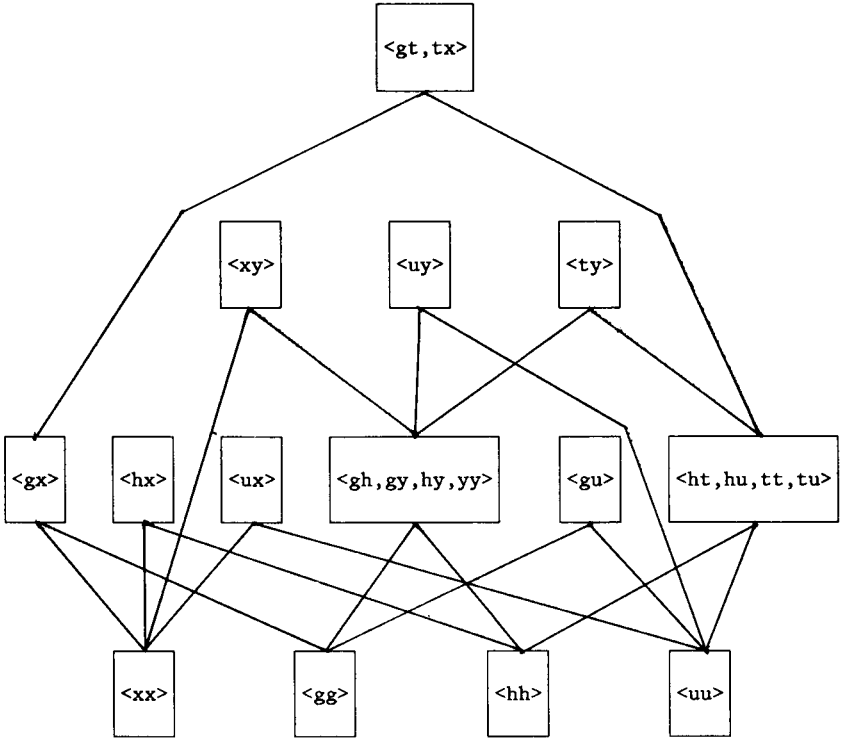

FIG. 2. The Hasse diagram of Graph(Φ_2) after the first stage of the algorithm.

Since $\langle xx \rangle$ and $\langle gg \rangle$ are both less than $\langle xy \rangle$, an arrow must be added from $\langle gx \rangle$ to $\langle xy \rangle$. The arrow from $\langle xx \rangle$ to $\langle xy \rangle$ can now be removed, since there is a path from $\langle xx \rangle$ to $\langle xy \rangle$ through $\langle gx \rangle$.

Since $\langle xx \rangle$ and $\langle hh \rangle$ are both less than $\langle xy \rangle$ and $\langle gt, tx \rangle$, arrows must be added from $\langle hx \rangle$ to $\langle xy \rangle$ and $\langle gt, tx \rangle$.

Since $\langle hh \rangle$ and $\langle uu \rangle$ are both less than $\langle uy \rangle$, add an arrow from $\langle ht, hu, tt, tu \rangle$ to $\langle uy \rangle$. Now the arrow from $\langle uu \rangle$ to $\langle uy \rangle$ can be removed, since there is a path from $\langle uu \rangle$ to $\langle uy \rangle$ through $\langle ht, hu, tt, tu \rangle$.

Again, $\langle ty \rangle$ is greater than $\langle uu \rangle$ and $\langle gh, gy, hy, yy \rangle$, so an arrow is added from $\langle uy \rangle$ to $\langle ty \rangle$. Since $\langle uy \rangle$ is greater than $\langle ht, hu, tt, tu \rangle$ and $\langle gh, gy, hy, yy \rangle$, we add an arrow from $\langle ty \rangle$ to $\langle uy \rangle$. This arrow combined with the previous one creates a cycle involving the two phenotypes $\langle ty \rangle$ and $\langle uy \rangle$. Thus Φ_2 does not have a factor-union representation.

Based on the results just obtained, there is a direct way to show that Φ_2 cannot have a factor-union representation. Suppose, to the contrary, that it did have a factor-union representation f. Then the factors associated with

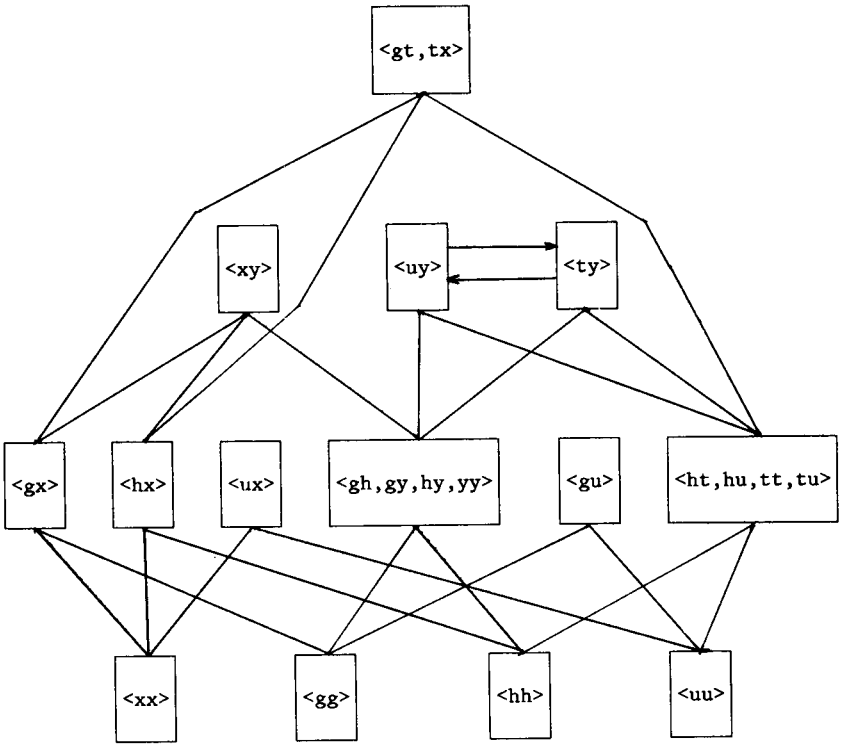

FIG. 3. The Hasse diagram of Graph(Φ_2) when a cycle is first detected.

$\langle uy \rangle$ would be $f(u)+f(y)$. Similarly, the factors associated with $\langle ty \rangle$ would be $f(t)+f(y)$. Since yy is in the same phenotype as gh, it would follow that $f(y)=f(y)+f(y)=f(g)+f(h)$. Similarly, since tt is in the same phenotype as hu, it follows that $f(t)=f(h)+f(u)$. Thus $f(t)+f(y)=f(h)+f(u)+f(g)+f(h)=f(u)+f(g)+f(h)=f(u)+f(y)$. This is impossible, since distinct phenotypes must have distinct factor sets associated with them.

Theorem 1 is a version of Theorem 2.1 in Markowsky [1973], and contains the basic facts about the representation of posets needed to prove that the algorithm is correct. Theorem 2 gives the proof that the algorithm is correct. For additional references and a more complete treatment of these ideas, see Markowsky [1980].

THEOREM 1

Let Δ *be a poset and* $\theta: \Delta \to$ Power(Δ) *a function given by the following rule*:

$$\theta(x) = \{y \text{ in } \Delta | y \text{ is not } \geqslant x\}$$

The function θ has the following properties:

(1) *For all x and y in Δ,*

$$x \leqslant y \quad \text{if and only if} \quad \theta(x) \leqslant \theta(y).$$

Thus distinct elements are mapped into distinct elements.

(2) *Let S be a subset of Δ, and assume that S has a sup, s. Then*

$$\theta(s) = \sum_{t \text{ in } S} \theta(t).$$

Proof. If $x \leqslant y$, and t is not $\geqslant x$, then t is not $\geqslant y$, since $\geqslant$ is transitive. Thus $\theta(x)$ is a subset of $\theta(y)$. Suppose that $\theta(x)$ is a subset of $\theta(y)$. Further, suppose y is not $\geqslant x$. Then y must be an element of $\theta(x)$ and hence of $\theta(y)$. But if y belongs to $\theta(y)$, then y is not $\geqslant y$, which is impossible. Thus $y \geqslant x$.

If s is the sup of S, then $s \geqslant t$ for all t in S. From the first part of this theorem, it follows that $\theta(s)$ contains the union of the $\theta(t)$'s, since it contains each individual $\theta(t)$. Now suppose that there exists an x which belongs to $\theta(s)$ but not to any $\theta(t)$ where t is an element of S. Thus x is not $\geqslant s$, but $x \geqslant t$ for all t in S. Since s is the sup of S, it follows that $x \geqslant s$. This contradicts the fact that x is not in $\theta(s)$. ■

THEOREM 2

If the algorithm described above halts without finding a cycle, Φ has a factor-union representation. If a cycle is found, Φ cannot have a factor-union representation. Furthermore, the heterozygotes in stage 5 of the algorithm can be chosen arbitrarily.

Proof. If Φ has a factor-union representation f, it is impossible for the algorithm to construct a cycle when processing Graph(Φ), since all the arrows introduced by the algorithm correspond to set inclusions that hold in any a factor-union representation.

If a cycle exists in Graph(Φ), at least two distinct phenotypes have the same factor sets, contradicting the assumption that f was a factor-union representation.

If the algorithm halts without finding a cycle, the relation $<$ is a partial order on Graph(Φ). Theorem 1 produces a representation θ assigning subsets of Graph(Φ) to each point in Graph(Φ).

For each allele a in Φ, let $f(a) = \theta(\langle aa \rangle)$. I claim that f is a factor-union representation of Φ. Note that $\theta(\langle ab \rangle) = f(a) + f(b)$, since the Arrow-Adding Rule ensures that $\langle ab \rangle$ is the sup of $\langle aa \rangle$ and $\langle bb \rangle$ in Graph(Φ), and the representation of Theorem 1 transforms sups into unions. If α is a genotype, define $f(\alpha)$ to be the union of the factor sets of the alleles in the genotype. It follows that $f(\alpha) = \theta(P)$, where P is the phenotype containing α.

Since θ maps distinct phenotypes into distinct subsets, f is a factor-union representation of Φ.

The order in which heterozygotes are considered in stage 4 is immaterial, because either a phenotype system has a factor-union representation or it doesn't. If it does, it is impossible for the algorithm to find a cycle, since such a cycle would imply the equality of two distinct phenotypes. Similarly, if a system doesn't have a factor-union representation, the algorithm must eventually discover a cycle, else a factor-union representation could be constructed for the system. ■

4. THE SIZE OF A REPRESENTATION

Example 3 (System Φ_3). This example has $2k$ alleles: $a_1, a_2, \cdots, a_k$ and $x_1, x_2, \cdots, x_k$, and k^2+2k+1 phenotypes:

(1) $\langle a_i a_i \rangle$ for $i = 1,2,\ldots$ (k of this type).
(2) $\langle x_i x_i \rangle$ for $i = 1,2,\ldots$ (k of this type).
(3) $\langle a_i x_j \rangle$ for $i, j = 1,2,\ldots$ (k^2 of this type).
(4) $\langle a_i a_j, x_i x_j | i \neq j$ and $i, j = 1,2,\ldots,k \rangle$ (1 of this type).

I claim that any factor-union representation of Φ_3 must have at least k^2 factors in it. To see this, assume a factor-union representation f for Φ_3 has been given. Let S be factor set of the phenotype $\langle a_1 a_2, \ldots \rangle$, and $S_{i,j}$ the factor set of the phenotype $\langle a_i x_j \rangle$. Here i and j both range over $1,\ldots,k$. Clearly, $S_{i,j}$ is a subset of S for all i and j. Let $C_{i,j}$ be $S - S_{i,j}$. Each $C_{i,j}$ is not empty, since each $S_{i,j}$ must be a proper subset of S.

I claim that the intersection of two distinct $C_{i,j}$'s must be empty. If $(i,j) \neq (i',j')$, then either $i \neq i'$ or $j \neq j'$. In the first case, $S_{i,j} + S_{i',j'}$ contains S, since $S = f(a_i a_{i'})$. The same result holds in the second case. Since the union of $S_{i,j}$ and $S_{i',j'}$ is S, the intersection of $C_{i,j}$ and $C_{i',j'}$ must be empty. Thus the $C_{i,j}$'s are a family of k^2 mutually disjoint, nonempty subsets of S. This implies that S has at least k^2 factors.

Φ_3 has k^2+2k+1 phenotypes. As k increases, the minimum number of factors necessary to represent it asymptotically approaches the number of phenotypes, and is quadratic in the number of alleles.

The following theorem is useful for finding minimal factor-union representations when the poset turns out to be a semilattice. It is an immediate consequence of Proposition 2.6 in [2]. For additional details on this result see [3].

THEOREM 3

Let Δ be a finite semilattice and $\chi: \Delta \to \text{Power}(X)$ a function which is injective (1-*to*-1), *preserves order, and preserves* sups. *Then the number of elements in X is $\geq$ the number of elements of Δ that have exactly one element*

covering them in the Hasse diagram of Δ. *These elements are called meet-irreducibles. Furthermore, if* M *is the set of all meet-irreducible elements in* P, *then* $\theta : \Delta \to \text{Power}(M)$ *given by*

$$\theta(x) = \{y \text{ in } M | y \text{ is not } \geqslant x\}$$

maps distinct elements into distinct elements, preserves order, and preserves sups.

If Graph(Φ) is a semilattice such that in any factor-union representation all sups correspond to unions, Theorem 3 shows how to easily find a factor-union representation using the smallest possible number of factors. Checking whether Graph(Φ) is a semilattice only requires that you check that every pair of elements has an upper bound. Checking whether unions would always correspond to sups is more involved, and is illustrated in the following examples.

It is easy to check that Graph(Φ_1) is a semilattice, with sups being unions. For example, $P = \langle cd, cf, de \rangle$ and $Q = \langle be, bh, fh \rangle$ have $R = \langle ac, ae, \ldots, fg \rangle$ as a sup in the partial order constructed. In any factor-union representation f of Φ_1, $f(R)$ must contain the union of $f(P)$ and $f(Q)$. To apply Theorem 3, it must be the union of $f(P)$ and $f(Q)$. This follows because $f(R) = f(b) + f(c)$, but $f(P) + f(Q) = f(c) + f(d) + f(b) + f(e)$. The argument for all other sups is similar. Thus any factor-union representation of Φ_1 must be a mapping of the type described in Theorem 3.

Graph(Φ_1) has 4 meet-irreducible elements: $\langle aa, \ldots \rangle$, $\langle cg, \ldots \rangle$, $\langle be, \ldots \rangle$ and $\langle cd, \ldots \rangle$. Let these be represented by the symbols w, x, y, and z for brevity. The 4-factor representation of Φ_1 derived from Theorem 3 is illustrated in Figure 4.

The phenotypes corresponding to the meet-irreducible elements have both a label (one of w, x, y, or z) and a factor set of labels. It is important to distinguish between the two. The phenotype $\langle cd, cf, de \rangle$ has the label z, but receives the factor set $\{w, x, y\}$ because the other three meet-irreducible elements, w, x, and y, are not greater than or equal to it. The phenotypes that are not meet-irreducibles only have a factor set associated with them, so there is no confusion.

Example 3 is analyzed similarly. The meet-irreducible elements turn out to be the $\langle a_i x_j \rangle$'s where $i, j = 1, 2, \ldots, k$. Here Theorem 3 shows that k^2 is a possible number of factors for a factor-union representation and that it is also the smallest possible number.

5. MULTILOCUS SYSTEMS

Finding factor-union representations for multilocus systems reduces in a straightforward way to finding factor-union representations for 1-locus sys-

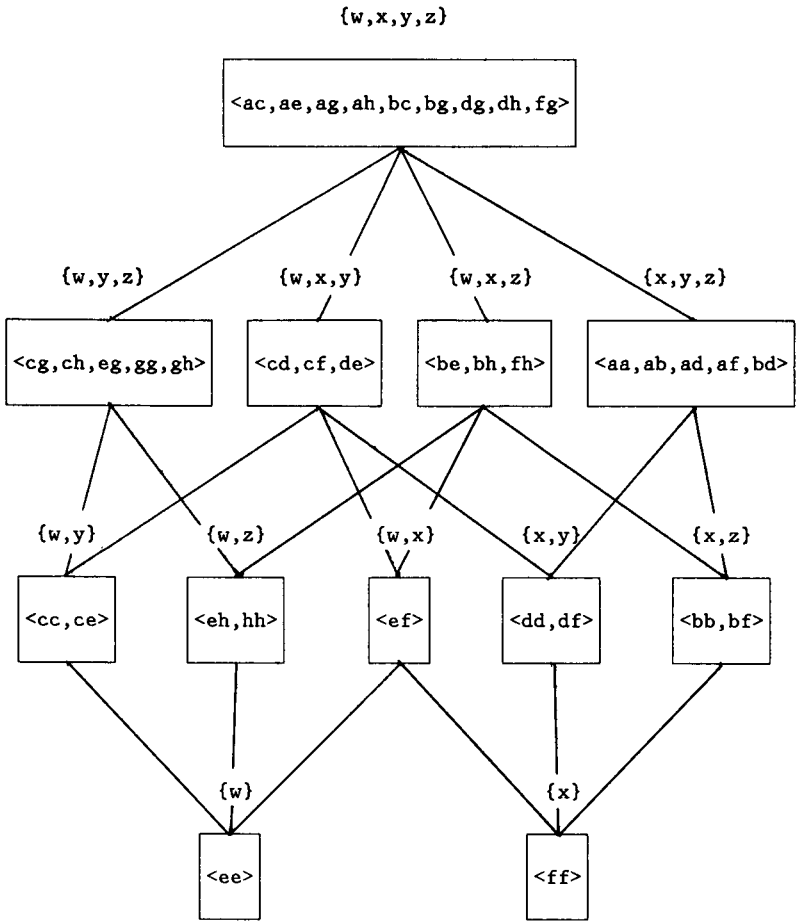

FIG. 4. A 4-factor representation of Φ_1.

tems. A computer is probably required for determining whether even small multilocus systems have a factor-union representation.

DEFINITIONS AND NOTATION

(1) The conventions introduced earlier for alleles and phenotypes are still in effect. A new notation, however, is required to represent genotypes. Thus *aa-bc-dd* represents an individual homozygous at the first and third loci and heterozygous at the second locus. For simplicity, it will be assumed that an allele can occur only at a particular locus. Because of this assumption, distinct genotypes must have different sets of alleles.

(2) The genotype *ab* can be considered the union of *aa* and *bb*. Similarly,

ab-cd-ee can be considered the union of *aa-cd-ee* and *bb-cd-ee*. In general, the union of two or more genotypes is possible if and only if there are no more than two different alleles at each locus.

(3) In the remainder of this paper, the term *homozygote* will refer to a genotype which is homozygous at each locus. The term *heterozygote* will refer to a genotype which is heterozygous at some locus.

(4) Given a multilocus system Φ, a 1-locus system, called the *1-locus model of* Φ and denoted by Trans(Φ), can be derived from it as follows:
   (a) The alleles of Trans(Φ) are derived from the homozygotes of Φ, i.e., for each homozygote of Φ, use a new symbol to represent an allele in Trans(Φ). If α is a homozygote in Φ, let $A(\alpha)$ represent the corresponding allele of Trans(Φ).
   (b) Since each genotype α in Φ can be written as the union of two homozygotes and the union of any two homozygotes is possible, every genotype in Φ corresponds to some genotype in Trans(Φ). Any such genotype is denoted by Trans(α).
   (c) Define the phenotypes as follows. Trans(α) and Trans(β) are in the same phenotype if and only if α and β are in the same phenotype in Φ. For each phenotype P in Φ, Trans(P) denotes the corresponding phenotype in Trans(Φ). [In general, Trans(Φ) has more genotypes than Φ, but the same number of phenotypes.]

(5) A *factor-union representation* for a multilocus system Φ consists of a set of factors and an assignment of subsets of factors to the alleles of Φ, such that the factor sets associated with all the genotypes within a given phenotype are equal, and distinct phenotypes have distinct factor sets. The factor set of a genotype is computed by taking the union of factor sets associated with the alleles at each locus. This definition is essentially the same as the definition for 1-locus systems.

Example 4. Let Φ be a 2-locus system with two alleles at each locus. Suppose the alleles at the first locus are w and x, while the alleles at the second locus are y and z. Φ has nine genotypes: four homozygotes and five heterozygotes. Further, suppose each genotype of Φ forms a distinct phenotype. To construct Trans(Φ), pick four new symbols corresponding to the homozygotes of Φ. I will use the following correspondence:

$$\langle ww\text{-}yy \rangle \to a, \quad \langle ww\text{-}zz \rangle \to b, \quad \langle xx\text{-}yy \rangle \to c, \quad \langle xx\text{-}zz \rangle \to d.$$

Trans(Φ) has ten genotypes and nine phenotypes. Each phenotype, except for $\langle ad, bc \rangle$, contains just a single genotype. The phenotype $\langle ad, bc \rangle$ corresponds to the phenotype $\langle wx\text{-}yz \rangle$ in Φ.

THEOREM 4

Let Φ be a multilocus system, and Trans(Φ) the 1-locus model of Φ. Then Φ has a factor-union representation if and only if Trans(Φ) does.

Proof. Suppose that Φ has a factor-union representation. Let f be the function that assigns sets of factors from a set S to the alleles in Φ. The factor-union representation of Trans(Φ) will use the same set S and a function g similar to f. For each allele $A(\alpha)$ in Trans(Φ), define

$$g(A(\alpha)) = f(\alpha).$$

Since all unions in Trans(Φ) are derived from unions in Φ, it is easy to see that we have constructed a factor-union representation for Trans(Φ).

Deriving a factor-union representation of Φ from one of Trans(Φ) is more involved, since none of the entities of Trans(Φ) correspond directly to the alleles of Φ.

Some additional concepts are needed to conclude the proof. If X is a set of alleles of Φ, let Hom(X) be the set of homozygotes that contain all the elements in X. If X has more than one allele from the same locus, Hom(X) will be empty. Now define

$$F(X) = \prod_{\alpha \text{ in } \mathrm{Hom}(X)} g(A(\alpha)).$$

I claim that if X, Y, and Z are sets of alleles of Φ such that $Z = X + Y$ and Hom(Z) is nonempty,

$$F(Z) = F(X) + F(Y) \qquad (*)$$

Since Hom(Z) is a subset of Hom(X) and Hom(Y), both of them are also nonempty. Pick arbitrary χ in X and υ in Y. There is a ζ in Hom(Z) and another homozygote λ such that $\zeta + \lambda$ are the same genotype as $\chi + \upsilon$. For brevity, use μ to denote this genotype. Because g is a factor-union representation,

$$g(A(\zeta)) \leqslant g(A(\mu)) = g(A(\chi)) + g(A(\upsilon))$$

Furthermore,

$$F(Z) \leqslant g(A(\zeta)).$$

Thus

$$F(Z) \leqslant g(A(\chi)) + g(A(\upsilon))$$

for all χ in X and υ in Y. By fixing χ, letting υ vary through all of Y, and taking the intersection of all terms, we get

$$F(Z) \leqslant g(A(\chi)) + F(Y).$$

Now by letting χ vary through all of X and taking the intersection of all

terms produced, we get

$$F(Z) \leqslant F(X) + F(Y)$$

Since $F(Z)$ is $\geqslant F(X)$ and $F(Y)$, we have $F(Z) \geqslant F(X) + F(Y)$. Consequently, (*) follows.

For each allele a in Φ define

$$f(a) = F(\{a\}).$$

Let Alleles(α) be the set of all alleles in the homozygote α. By induction and (*) it follows that

$$F(\text{Alleles}(\alpha)) = \sum_{a \text{ in } \alpha} f(a).$$

But

$$\text{Hom}(\text{Alleles}(\alpha)) = \{\alpha\},$$

so

$$F(\text{Alleles}(\alpha)) = g(A(\alpha)).$$

This means that the "factor sets" assigned to the alleles by f are such that when the appropriate unions are taken, each homozygote receives the same set of factors g assigns to its image in Trans(Φ). Because heterozygotes are unions of homozygotes, it follows that f is a factor-union representation of Φ. ■

I gratefully thank Dr. Kenneth Lange for introducing me to this problem, supplying me with various references, and some very helpful comments on an early draft of this paper. I would also like to thank Dr. Joshua Lederburg for an interesting discussion that led to my cleaning up the section on multilocus systems. At a seminar where I presented these results he indicated that my reduction of 2-locus systems should also hold for multilocus systems. I thought the matter through and produced the proof that this is correct. Finally, I would like to thank the referee for making some suggestions that improved the exposition of the paper.

REFERENCES

1 C. W. Cotterman Factor-union phenotype systems, in *Computer Applications in Genetics* (N. E. Morton, Ed.), Univ. of Hawaii Press 1969, 1-19.

2 G. Markowsky, Combinatorial aspects of lattice theory with applications to the enumeration of free distributive lattices, Ph.D thesis, Harvard Univ., 1973.

3 G. Markowsky, The representation of posets and lattices by sets, *Algebra Universalis* 11:173–192 (1980).