Pergamon

Chemical Engineering Science, Vol. 50, No. 11, pp. 1755–1768, 1995
Copyright © 1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0009–2509/95 $9.50 + 0.00

0009-2509(95)00034-8

DECISION-MAPPING: A TOOL FOR CONSISTENT AND COMPLETE DECISIONS IN PROCESS SYNTHESIS

F. FRIEDLER[†,‡,§,¶], J. B. VARGA[†,||] and L. T. FAN[‡]

[†] Department of Systems Engineering, Research Institute of Chemical Engineering, Hungarian Academy of Sciences, Veszprém, Pf. 125, H-8201, Hungary

[‡] Department of Chemical Engineering, Kansas State University, Manhattan, KS 66506, U.S.A.

(*Received* 2 *February* 1994; *accepted in revised form* 2 *January* 1995)

Abstract—Decisions involved in process synthesis are often more complex than those involved in other disciplines. This arises from the fact that such decisions are concerned with specification or identification of highly interconnected systems, e.g. process structures, which may contain a multitude of recycling loops. It appears that no rigorous technique is available, which is capable of representing exactly and organizing efficiently the system of decisions for a process synthesis problem. A novel mathematical notion, decision-mapping, has been introduced in this work to render the complex decisions in process design and synthesis consistent and complete. The basic terminologies of decision-mapping, including extension, equivalence, completeness, complementariness, and active domain, have been defined based on rigorous set-theoretic formalism, and the most important properties of decision-mappings have been identified and proved. Decision-mapping, as a rigorously established technique, is directly applicable in developing efficient and exact process synthesis methods or improving existing methods. The applicability and meritorious features of this new technique are illustrated by synthesizing a large scale process.

1. INTRODUCTION

Efforts to apply mathematical programming methods to process synthesis have yielded encouraging results; nevertheless, a number of major issues remain unresolved. The principal ones are establishment of the mathematical foundation necessary to validate the algorithms for optimal process synthesis, and the reduction in the complexity of the mathematical programming algorithms for process synthesis.

A thorough understanding of the unique combinatorial properties of process structures enables us to validate rigorously the algorithms for process synthesis and to reduce drastically the complexity of these algorithms. In principle, two types of approaches are available for this purpose; they are logical formulation (Raman and Grossmann, 1993) and combinatorics (Friedler *et al.*, 1992c).

Combinatorics has been adopted in the present approach. The fundamental combinatorial properties of feasible process structures have been expressed as a set of axioms (Friedler *et al.*, 1992b, c). For the MINLP model of process synthesis, these axioms constrain the set of possible values of the integer variables to the set of combinatorially feasible values, thereby reducing the size of the search space by many orders of magnitude. Although these axioms constitute a rigorous foundation for the combinatorial segment of process synthesis, they do not directly give rise to the computational algorithms for process synthesis. The reason is that the axioms express self-evident facts instead of procedures, and thus, they are not in procedural form. The present work introduces a new combinatorial technique that is mathematically rigorous, i.e. decision-mapping, for direct application to developing and describing the algorithms for process synthesis. This technique is capable of representing process networks or structures of any type and size. For example, a structure containing any number of recyclings with arbitrary sizes can be represented in synthesizing a process by this technique. Decision-mapping has been developed with rigorous mathematical formalism and validated by resorting to the combinatorial axioms of process synthesis; therefore, any algorithm based on the decision-mapping can also be validated rigorously.

The focus of the present work is on the total flowsheet synthesis [see e.g. Siirola and Rudd (1971), Lu and Motard (1985) and Douglas (1988)]. Nevertheless, the results are applicable directly or can be extended to other classes of process synthesis.

2. UNAMBIGUOUS STRUCTURAL REPRESENTATION: P-GRAPH

For formally analyzing process structures in process synthesis, an unambiguous structural representation is required. Process graph or P-graph, which is a directed bipartite graph, has been introduced for this purpose [see Friedler *et al.* (1992c)]. A brief description of P-graph is given below.

[§] Also at the Department of Computer Science, University of Veszprém, Veszprém, Hungary.

[||] Also at the Department of Computer Science, A. József University, Szeged, Hungary.

[¶] Author to whom correspondence should be addressed at Research Institute of Chemical Engineering, Hungarian Academy of Sciences, Veszprém, Pf. 125, H-8201, Hungary.

Let M be a given finite nonempty set of objects, usually material species or materials, that can be transformed in the process under consideration. Transformation between two subsets of M occurs in an operating unit. It is necessary to link this operating unit to other operating units through the elements of these two subsets of M.

Let O be the set of operating units to be considered in synthesis; then, $O \subseteq \wp(M) \times \wp(M)$ where $O \cap M = \emptyset$. If (α, β) is an operating unit, i.e. $(\alpha, \beta) \in O$, then α is called the input set, and β, the output set of this operating unit. Pair (M, O) is termed a *process graph* or *P-graph* with the set of vertices $M \cup O$ and the set of arcs $\{(x,y) \mid y = (\alpha,\beta) \in O$ and $x \in \alpha\} \cup \{(y,x) \mid y = (\alpha,\beta) \in O$ and $x \in \beta\}$. P-graph (M,O) is defined to be a *subgraph* of (M',O'), i.e. $(M,O) \subseteq (M',O')$, if $M \subseteq M'$ and $O \subseteq O'$. The *union* of two P-graphs (M_1,O_1) and (M_2,O_2), $(M_1,O_1) \cup (M_2,O_2)$, is defined to be P-graph $(M_1 \cup M_2, O_1 \cup O_2)$. The *indegree* of vertex X, $d^-(X)$, is defined to be the number of arcs with endpoint X. If X is a material, then broadly speaking, $d^-(X)$ is the number of operating units producing material X.

Example 1. Let us suppose that set M_1 of materials and set O_1 of operating units of P-graph (M_1,O_1) be given as

$$M_1 = \{A,B,C,D,E,F,G,H,I,J,K,L\}$$

and

$$O_1 = \{(\{C\},\{A,F\}),(\{D\},\{A,B\}),(\{E,F\},\{C\}),(\{F,G\},\{C,D\}),(\{G,H\},\{D\}),(\{J\},\{F\}),(\{K,L\},\{H\})\}.$$

P-graph (M_1,O_1) is depicted in Fig. 1. Note that the input and output sets of operating unit 1, $(\{C\},\{A,F\})$, are $\{C\}$ and $\{A,F\}$, respectively, and that the indegree of vertex A, $d^-(A)$, is 2 since two operating units produce material A.

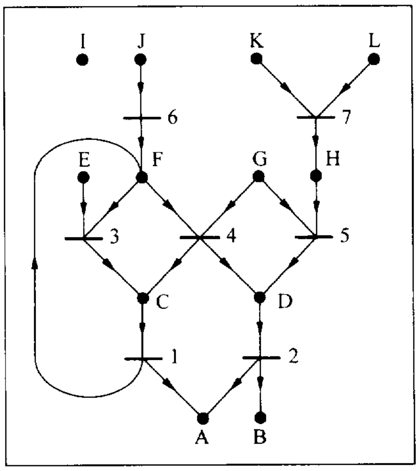

Fig. 1. P-graph (M_1,O_1) where A, B, C, D, E, F, G, H, I, J, K, and L are the materials, and 1, 2, 3, 4, 5, 6, and 7 are the operating units.

P-graph (M,O) contains the interconnections among the units of O. Each feasible process corresponds to a subgraph of (M,O); this subgraph can be generated by decision-mappings.

3. DECISION-MAPPING: DEFINITIONS AND SUPPORTING THEOREMS

Rigor and clarity demand that various terminologies introduced in this work be couched in the parlance of mathematical formalism. A mapping or function determines a unique "value" for an argument; the set of possible arguments, D, is called the *domain*; and the set of possible "values", R, is the *range* of the mapping or function. Mapping or function f determines a "value", i.e. $f(x) \in R$ for each $x \in D$. A mapping or function f, therefore, can be defined as a special subset of the Cartesian product of D and R, $D \times R$, i.e. f is a set of pairs (x,y) where $x \in D$ and $y = f(x) \in R$. This set of pairs is denoted by $f[D]$. Thus, while the name of a mapping followed by an element of its domain in parentheses represents the value determined by the mapping, e.g. $f(x)$, a mapping name followed by the domain in square brackets represents the mapping, e.g. $f[D]$. The notations of this type will appear throughout this paper.

Let us suppose that P-graph (M,O) represents the interconnections of the operating units of a synthesis problem. Then, for the set of materials M and the set of operating units O, $O \subseteq \wp(M) \times \wp(M)$ holds. Let us now induce mapping Δ from the set of materials to the set of subsets of the set of operating units, i.e. from M to $\wp(O)$. This mapping determines the set of operating units producing material X for any $X \in M$; consequently, $\Delta(X) = \{(\alpha,\beta) \mid (\alpha,\beta) \in O$ and $X \in \beta\}$.

Definition 1. Let m be a subset of M, and X be an element of m; moreover, let $\delta(X)$ be a subset of $\Delta(X)$. Then, mapping δ from set m to the set of subsets of set O, $\delta[m] = \{(X,\delta(X)) \mid X \in m\}$, is defined to be a *decision-mapping* on m, with m being the domain of the mapping. Decision-mapping $\delta_1[m_1]$ is defined to be the *restriction* of decision-mapping $\delta_2[m_2]$ to m_1 if $m_1 \subseteq m_2$ and $\delta_1[m_1] = \{(X,\delta_2(X)) \mid X \in m_1\}$. Thus, the "value" of δ_1 is equal to that of δ_2 for any element of m_1.

Mapping $\Delta[M] = \{(X,\Delta(X)) \mid X \in M\}$ can also be considered as a decision-mapping; it will be termed as the *maximal decision-mapping*.

Example 2. Let us revisit Example 1. According to the definition of $\Delta_1(X)$, sets $\Delta_1(A)$ through $\Delta_1(L)$ can be given for P-graph (M_1,O_1) as follows:

$$\Delta_1(A) = \{(\{C\},\{A,F\}),(\{D\},\{A,B\})\}$$

$$\Delta_1(B) = \{(\{D\},\{A,B\})\}$$

$$\Delta_1(C) = \{(\{E,F\},\{C\}),(\{F,G\},\{C,D\})\}$$

$$\Delta_1(D) = \{(\{F,G\},\{C,D\}),(\{G,H\},\{D\})\}$$

$$\Delta_1(E) = \emptyset$$

$$\Delta_1(F) = \{(\{C\},\{A,F\}),(\{J\},\{F\})\}$$

$\Delta_1(G) = \emptyset$

$\Delta_1(H) = \{(\{K, L\}, \{H\})\}$

$\Delta_1(I) = \Delta_1(J) = \Delta_1(K) = \Delta_1(L) = \emptyset.$

Let us now define decision-mapping δ_1 for P-graph (M_1, O_1) of this example. The domain, m_1, of δ_1 must be a subset of M_1, e.g., $m_1 = \{A, D, H\}$. Suppose that $\delta_1(A) = \{(\{D\}, \{A, B\})\}, \delta_1(D) = \{(\{F, G\}, \{C, D\}),$ $(\{G, H\}, \{D\})\}$, and $\delta_1(H) = \emptyset$. Obviously, $\delta_1(A) \subseteq \Delta_1(A)$, $\delta_1(D) \subseteq \Delta_1(D)$, and $\delta_1(H) \subseteq \Delta_1(H)$; therefore, $\delta_1[m_1] = \{(A, \delta_1(A)), (D, \delta_1(D)), (H, \delta_1(H))\} = \{(A,$ $\{(\{D\}, \{A, B\})\}), (D, \{(\{F, G\}, \{C, D\}), (\{G, H\}, \{D\})\}),$ $(H, \emptyset)\}$ is a decision-mapping of the example.

The rigorous definition of a P-graph of a decision-mapping requires additional tools other than those currently available as will be elaborated later. Nevertheless, for illustration, let us consider the operating units include in the range of a decision-mapping as its P-graph. The P-graph of decision-mapping $\delta_1[m_1]$ of Example 2 is illustrated in Fig. 2.

To examine the major properties of process structures, special class of decision-mapping must be introduced.

Definition 2. The *complement* of decision-mapping $\delta[m]$ is defined by $\bar{\delta}[m] = \{(X, Y) \mid X \in m$ and $Y = \Delta(X)/\delta(X)\}$; thus, for $X \in m$, $\bar{\delta}(X) = \Delta(X)/\delta(X)$. Since $\delta(X)$ is a set of operating units producing material X, $\bar{\delta}(X)$ is the set of operating units producing X, but are not included in $\delta(X)$.

The consistency of decisions is crucial in synthesizing a process. For example, we can decide independently that an operating unit producing materials X and Y be included in a process for the production of material X and be excluded from the production of material Y, thereby giving rise to a contradiction in the system of decisions. To avoid such a contradiction, consistent decision-mappings must be considered.

Definition 3. Decision-mapping $\delta[m]$ is said to be *consistent* if $|m| \leqslant 1$ or $(\delta(X) \cap \delta(Y)) \cup (\bar{\delta}(X) \cap \bar{\delta}(Y)) = \Delta(X) \cap \Delta(Y)$ for any $X, Y \in m$; otherwise, it is inconsistent. Let $m' \subseteq m$. We say that the decision-mapping $\delta[m]$ is *consistent on* m' if the decision-mapping $\delta[m']$ is consistent, where $\delta[m']$ denotes the restriction of $\delta[m]$ to m'. Obviously, if $\delta[m]$ is consistent, then it is consistent on any subset of m.

In other words, a decision-mapping is considered to be consistent if every operating unit producing materials X and Y is included in or excluded from both $\delta(X)$ and $\delta(Y)$. Another equivalent definition for consistency can be established by the following theorem.

Theorem 1. *Decision-mapping $\delta[m]$ for which $|m| \geqslant 1$ is consistent if and only if $\delta(X) \cap \bar{\delta}(Y) = \emptyset$ for all X, $Y \in m$.*

The proofs of the above theorem and all other theorems are given in Appendix A.

Example 3. Decision-mapping $\delta_1[m_1]$ in Example 2 is consistent because $\delta_1(A) \cap \bar{\delta}_1(D) = \emptyset$; $\delta_1(A) \cap \bar{\delta}_1(H) = \emptyset$; $\bar{\delta}_1(A) \cap \delta_1(D) = \emptyset$; $\bar{\delta}_1(A) \cap \delta_1(H) = \emptyset$; $\delta_1(D) \cap \bar{\delta}_1(H) = \emptyset$; and $\bar{\delta}_1(D) \cap \delta_1(H) = \emptyset$. Nevertheless, decision-mapping $\delta_2[m_2] = \delta_1[m_1] \cup \{(C, \{$ $(\{E, F\}, \{C\})\})\}$ is inconsistent since $\delta_2(D) \cap \bar{\delta}_2(C) =$ $\{(\{F, G\}, \{C, D\})\} \neq \emptyset$. In $\delta_2[m_2]$, operating unit $(\{F, G\}, \{C, D\})$ is simultaneously included and excluded from the consideration, thereby resulting in a contradiction in the system of decisions.

It is of importance to decide which decision-mappings be considered equivalent. For example, decision-mapping $\delta_3[m_1 \cup \{B\}] = \delta_1[m_1] \cup \{(B, \{(\{D\},$ $\{A, B\})\})\}$ is different from $\delta_1[m_1]$; nevertheless, both yield an identical structure, thus implying that these two decision-mappings have some equivalence; see Fig. 2. This type of equivalence will be established on the closure of the decision-mapping. For this purpose, let the set of operating units of decision-mapping $\delta[m]$ be denoted by op$(\delta[m])$, and the set of materials of set o of operating units, by mat(o); hence,

$$\text{op}(\delta[m]) = \bigcup_{X \in m} \delta(X) \quad \text{and} \quad \text{mat}(o) = \bigcup_{(\alpha, \beta) \in o} (\alpha \cup \beta).$$

Definition 4. For consistent decision-mapping $\delta[m]$, let $\mathbf{o} = \text{op}(\delta[m])$, $\mathbf{m} = \text{mat}(\mathbf{o}) \cup m$, and $\delta'[\mathbf{m}] =$ $\{(X, Y) \mid X \in \mathbf{m}$ and $Y = \{(\alpha, \beta) \mid (\alpha, \beta) \in \mathbf{o}$ and $X \in \beta\}\}$. Then, $\delta'[\mathbf{m}]$ is defined to be the *closure* of $\delta[m]$, and $\delta[m]$ is said to be *closed* if $\delta[m] = \delta'[\mathbf{m}]$. Naturally, the closure of a consistent decision-mapping is closed.

Example 4. Referring to Example 2, the closure of decision-mapping $\delta_1[m_1]$, $\delta'_1[\mathbf{m}_1]$, follows Definition 4. Domain $\mathbf{m}_1$ is the union of set m_1 and the set of materials involved in the operating units, i.e., $\mathbf{m}_1 = m_1 \cup \{B, C, F, G\}$. Hence $\delta'_1[\mathbf{m}_1] = \{(A, \{(\{D\},$ $\{A, B\})\}), (B, \{(\{D\}, \{A, B\})\}), (C, \{(\{F, G\}, \{C, D\})\}),$ $(D, \{(\{F, G\}, \{C, D\}), (\{G, H\}, \{D\})\}), (F, \emptyset), (G, \emptyset),$ $(H, \emptyset)\}$.

An important relation of a decision-mapping and its closure are expressed by the following theorem.

Theorem 2. *Let $\delta'[\mathbf{m}]$ be the closure of consistent decision-mapping $\delta[m]$; then, $\delta(X) = \delta'(X)$ for all $X \in m$, i.e. $\delta[m]$ is the restriction of $\delta'[\mathbf{m}]$ to m.*

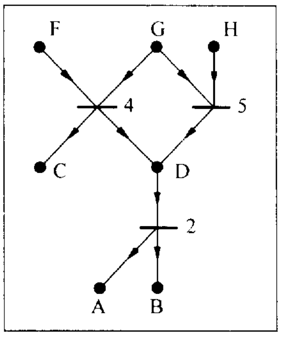

Fig. 2. P-graph of decision-mapping $\delta_1[m_1]$.

Theorem 2 implies that if $\delta'[\mathbf{m}]$ is the closure of consistent decision-mapping $\delta[m]$, then, $\delta[m] \subseteq \delta'[\mathbf{m}]$. It is essential to inquire if the closure preserves the consistency.

Theorem 3. *The closure of a consistent decision-mapping is consistent.*

The equivalence of decision-mappings can be established on their closure as stated in the following definition.

Definition 5. Two consistent decision-mappings are defined to be *equivalent if their closure is common.*

Naturally, a consistent decision-mapping is equivalent to its closure. The relation "equivalent" has the mathematically required properties of an equivalence relation: It is reflexive, symmetric, and transitive.

Example 5. Decision-mapping $\delta_4[\{B,D\}] = \{(B, \{(\{D\},\{A,B\})\}),(D,\{(\{F,G\},\{C,D\}),(\{G,H\},\{D\})\})\}$ has the same closure as $\delta_1[m_1]$ in Example 2; therefore, they are equivalent.

The restriction of a consistent decision-mapping is often equivalent to itself; nevertheless, this is not always the case. The definition given below serves to examine if a restriction preserves the equivalence.

Definition 6. m' is said to be an *active domain* of decision-mapping $\delta[m]$ if $m' \subseteq m$, $\text{op}(\delta[m']) = \text{op}(\delta[m])$, and $\text{op}(\bar{\delta}[m']) = \text{op}(\bar{\delta}[m])$.

Note that m is always an active domain of decision-mapping $\delta[m]$, and that a decision-mapping can have multiple active domains. It is sufficient, however, to define a decision-mapping on any of its active domains as stated in the following theorem.

Theorem 4. *Let $\delta[m]$ be a consistent decision-mapping. Then, it is determined on its whole domain, m, if it is given only on one of its active domains.*

According to the next theorem, it is sufficient to examine the consistency of a decision-mapping on any of its active domains.

Theorem 5. *If a decision-mapping is consistent on one of its active domains, then, it is consistent.*

If a consistent system of decisions is incomplete in synthesizing a process, the additional decisions should be made by preserving its consistency. The following definition formalizes this requirement as an extension of a decision-mapping.

Definition 7. Let $\delta_1[m_1]$ and $\delta_2[m_2]$ be consistent decision-mappings, with their closures, $\delta'_1[\mathbf{m}_1]$ and $\delta'_2[\mathbf{m}_2]$, respectively. Then, $\delta_1[m_1]$ is defined to be an *extension* of $\delta_2[m_2]$ if

(i) $\mathbf{m}_1 \supseteq \mathbf{m}_2$,
(ii) $\delta_2[m_2]$ is the restriction of $\delta_1[m_1]$ to m_2, i.e. $m_1 \supseteq m_2$ and $\delta_1(X) = \delta_2(X)$ for $X \in m_2$, and
(iii) $\delta'_1(X) \supseteq \delta'_2(X)$ for $X \in \mathbf{m}_2/m_2$.

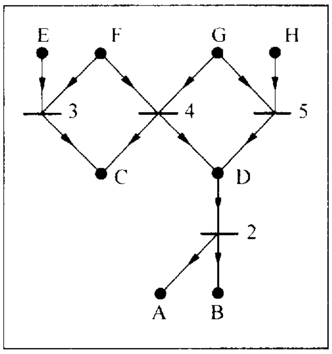

Fig. 3. P-graph of decision-mapping $\delta_5[m_1 \cup \{C\}]$.

That $\delta_1[m_1]$ is an extension of $\delta_2[m_2]$ is denoted by $\delta_1[m_1] > \delta_2[m_2]$. Naturally, the closure of a consistent decision-mapping is its extension.

Example 6. Decision-mapping $\delta_5[m_1 \cup \{C\}] = \delta_1[m_1] \cup \{(C,\{(\{E,F\},\{C\}),(\{F,G\},\{C,D\})\})\}$ is an extension of decision-mapping $\delta_1[m_1]$ because it satisfies every requirement stated in Definition 7. The P-graph of decision-mapping $\delta_5[m_1 \cup \{C\}]$ is given in Fig. 3.

A major property of relation extension is expressed by the following statement.

Theorem 6. *Relation extension is a partial order on the set of consistent decision-mappings.*

4. REPRESENTATION OF A PROCESS GRAPH BY DECISION-MAPPING

With the necessary tools in hand, the relationship between the P-graphs and decision-mappings can now be examined. Let P-graph (m,o) be a specific subgraph of P-graph (M,O); then, $m \subseteq M$, $o \subseteq O$, $O \subseteq \wp(M) \times \wp(M)$, and $o \subseteq \wp(m) \times \wp(m)$. Let us assume that $m = \text{mat}(o)$.

Definition 8. m' is an *active set* of P-graph (m,o), if $m' \subseteq m$ and $\beta \cap m' \neq \emptyset$ for any $(\alpha,\beta) \in o$.

Thus, at least one output material of each operating unit of P-graph (m,o) is represented in its active set. Naturally, set m is active if, for any $(\alpha,\beta) \in o$, $\beta \neq \emptyset$; conversely, P-graph (m,o) has no active set if there exists $(\alpha,\beta) \in o$ such that $\beta = \emptyset$. This type of operating units has no practical value; thus, we suppose that $\beta \neq \emptyset$ for any operating unit (α,β).

Definition 9. Let m' be an active set of P-graph (m,o); then, $\delta[m']$ is defined to be a *decision-mapping of P-graph* (m,o), if $\delta[m'] = \{(X,Y) \mid X \in m'$ and $Y = \{(\alpha,\beta) \mid (\alpha,\beta) \in o$ and $X \in \beta\}\}$, i.e. if $\delta(X) = \{(\alpha,\beta) \mid (\alpha,\beta) \in o$ and $X \in \beta\}$ for $X \in m'$. Since m' is an active set, $o = \text{op}(\delta[m'])$.

Example 7. Again referring to Example 2, both $\{A,D,H\}$ and $\{B,C,D,H\}$ are active sets of the

P-graph given in Fig. 2. For these active sets, $\delta'[\{A,D,H\}] = \{(A,\{(\{D\},\{A,B\})\}),(D,\{(\{F,G\},\{C,D\}),(\{G,H\},\{D\})\}),(H,\emptyset)\}$ and $\delta''[\{B,C,D,H\}] = \{(B,\{(\{D\},\{A,B\})\}),(C,\{(\{F,G\},\{C,D\})\}),(D,\{(\{F,G\},\{C,D\}),(\{G,H\},\{D\})\}),(H,\emptyset)\}$ are two decision-mappings of this P-graph. Since they have identical closure, these decision-mappings are equivalent. The following theorems need be proved, however, to demonstrate that it is the case in general, i.e. the different decision-mappings of the same P-graph are always equivalent.

Theorem 7. *The decision-mappings of P-graph* (m,o) *are consistent.*

If the decision-mapping of a P-graph is given on the entire set of materials, then it is closed as stated in the following theorem.

Theorem 8. *Decision-mapping* $\delta[m]$ *of P-graph* (m,o) *is closed.*

The connection between the active set and active domain is expressed as a theorem as follows:

Theorem 9. *Active set* m' *of P-graph* (m,o) *is an active domain of its decision-mapping* $\delta[m]$, *if* $\text{op}(\bar{\delta}[m']) = \text{op}(\bar{\delta}[m])$.

Since a P-graph may have multiple active sets, it may also have different decision-mappings. The principal question whether these decision-mappings are equivalent, is answered by the following theorem that ensures the validity of Definition 9.

Theorem 10. *The decision-mappings of P-graph* (m,o) *are equivalent provided that* $m = \text{mat}(o)$.

Let us suppose that part of a process has been designed or it is temporarily assumed when synthesizing a process, e.g. a substructure is given by a decision-mapping. Then, the remaining part should be described in accordance with the previous decisions. The definition given below formalizes this requirement.

Definition 10. Let P-graphs $\sigma_1 = (m_1,o_1)$ and $\sigma_2 = (m_2,o_2)$ be given, where $m_1 = \text{mat}(o_1)$, $m_2 = \text{mat}(o_2)$, $m_1 \subseteq M$, $m_2 \subseteq M$, $o_1 \subseteq O$, and $o_2 \subseteq O$. Let $\delta_1[m_1']$ be a decision-mapping of σ_1. Then, σ_2 is said to be an *extension of* σ_1 *relative to* $\delta_1[m_1']$ if there exists a decision-mapping $\delta_2[m_2']$ of σ_2 such that $\delta_2[m_2'] > \delta_1[m_1']$.

Since the same structure may have different decision-mappings, the extension of a structure may depend on the particular decision-mapping considered. For example, both $\delta_1[m_1]$ and $\delta_6[m_1 \cup \{C\}] = \delta_1[m_1] \cup \{(C,(\{F,G\},\{C,D\}))\}$ are decision-mappings of the P-graph given in Fig. 2; nevertheless, decision-mapping $\delta_5[m_1 \cup \{C\}]$ of Example 6 is the extension of $\delta_1[m_1]$, but it is not the extension of $\delta_6[m_1 \cup \{C\}]$.

Theorem 11. *Let* $\delta[m']$ *be a consistent decision-mapping;* $o = \text{op}(\delta[m'])$; *and* $m = \text{mat}(o) \cup m'$. *Then,* (i) (m,o) *is a P-graph,* (ii) m' *is an active set of P-graph* (m,o), *and* (iii) $\delta[m']$ *is a decision-mapping of P-graph* (m,o).

This theorem suggests the definition for the P-graph of a decision-mapping.

Definition 11. The *P-graph of consistent decision-mapping* $\delta[m']$ *is defined to be* (m,o) where $o = \text{op}(\delta[m'])$ and $m = \text{mat}(o) \cup m'$.

The following two theorems establish this definition.

Theorem 12. *Let* $\delta[m']$ *be a consistent decision-mapping and* (m,o) *be its P-graph. If* m'' *is an active domain of* $\delta[m']$, *then* m'' *is an active set of* (m,o).

Theorem 13. *Equivalent decision-mappings have the same P-graph.*

A path in a decision-mapping can analogously be defined as a path in a P-graph. Since this term has a special significance, it is explicitly defined below.

Definition 12. Let $Y_1 \in \text{op}(\delta[m])$ and $X_n \in \text{mat}(\text{op}(\delta[m]))$; then, there is a *path between operating unit* Y_1 *and material* X_n in decision-mapping $\delta[m]$ if and only if there exists a sequence $Y_1, X_1, Y_2, X_2, \dots, Y_n, X_n$ such that $Y_i \in \delta(X_i)$ $(i = 1,2,\dots,n)$ and $X_i \in \text{mat}^{\text{in}}(\{Y_{i+1}\})$ $(i = 1,2,\dots,n-1)$ where mat^{in} determines the set of input materials for a set of operating units.

Example 8. A path exists between operating unit $(\{G,H\},\{D\})$ and material A in decision-mapping $\delta_1[m_1]$ in Example 2, since we have sequence $(\{G,H\},\{D\})$, D, $(\{D\},\{A,B\})$, A, which satisfies the requirements of Definition 12 (see Fig. 2).

5. DECISION-MAPPINGS AND THE COMBINATORIAL AXIOMS OF PROCESS SYNTHESIS

The MINLP model of a process synthesis problem gives rise to difficulties of both combinatorial and continuous nature even though they are not totally independent of each other. While several methods are available for mitigating the difficulties of the continuous nature, this is not the case for the difficulties of the combinatorial nature. To develop an exact and efficient algorithmic or mathematical programming method for process synthesis, therefore, it is necessary to comprehend the major combinatorial properties of process structures; moreover, these properties should be taken into account in search for the optimal process structures.

Suppose that sets P, R, and O are known for set M of materials given, where P is the set of products, R is the set of raw materials, and O is the set of operating units. The relation among these sets can be expressed as $P \subset M$, $R \subset M$, $P \cap R = \emptyset$, $M \cap O = \emptyset$, and $O \subseteq \wp(M) \times \wp(M)$. Triplet (P,R,O) defines a synthesis problem, if none of sets P, R, and O is empty. The process structures for synthesis problem (P,R,O) are the subgraphs of P-graph (M,O); however, the P-graph of a feasible process must always conform to certain combinatorial properties.

These properties have been expressed as a set of axioms; moreover, P-graphs satisfying these axioms are defined to be the solution-structures of the synthesis problem [Friedler *et al.* (1992c), also see Appendix B for a brief summary].

The axioms of process synthesis can also be expressed by the help of the decision-mapping. This form is advantageous in developing algorithms for process synthesis.

Definition 13. Let m be a subset of M for P-graph (M, O) and let $\delta[m]$ be a consistent decision-mapping. Then, $\delta[m]$ is defined to be a *combinatorially feasible* decision-mapping of synthesis problem (P, R, O) if it satisfies the following axioms.

$(D1)$ $P \subseteq \text{mat}(\text{op}(\delta[m]))$.

$(D2)$ For any $x \in \text{mat}(\text{op}(\delta[m])), \delta(x) = \emptyset$ if and only if $x \in R$.

$(D3)$ If $o \in \text{op}(\delta[m])$, then an $x \in P$ exists such that there is a path from o to x in $\delta[m]$.

It can be proved that a decision-mapping is combinatorially feasible if and only if its P-graph is a solution-structure of (P, R, O). On the other hand, for any solution-structure (m, o), its decision-mapping $\delta[m]$ is combinatorially feasible. Thus, Definition 13, together with Definition 11, results in a description of the solution-structures of synthesis problem (P, R, O).

6. APPLICATION OF DECISION-MAPPINGS FOR THE COMBINATORIAL ALGORITHMS OF PROCESS SYNTHESIS

Fundamental combinatorial algorithms have been developed through the decision-mapping. Such algorithms include those for the generation of the super-structure (maximal structure) and the set of

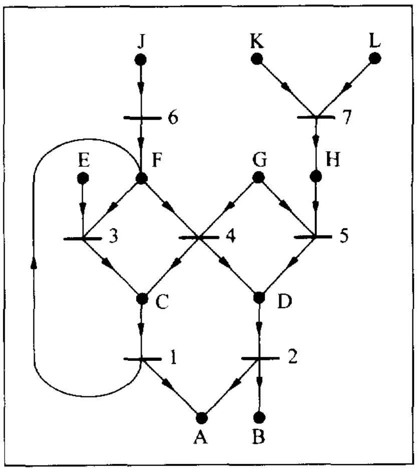

Fig. 5. Maximal structure of Example 1.

```
input: M, P, R, $\Delta[M]$;
comment: P, R, $\Delta[M]$ belong to synthesis problem (P, R, O), where
$P \subset M$, $R \subset M$, $P \cap R = \emptyset$, $\Delta(x) = \{(\alpha, \beta) | (\alpha, \beta) \in O \;\&\; x \in \beta\}$, $\Delta(x) = \emptyset \Leftrightarrow x \in R$,
$\Delta[M] = \{(x, \Delta(x)) | x \in M\}$, $\delta[m]$ is a decision-mapping on (M, O);
output: all solution-structures of synthesis problem (P, R, O);
global variables: R, $\Delta[M]$;

begin
if $P = \emptyset$ then stop;
SSG(P, $\emptyset$, $\emptyset$)
end

procedure SSG(p, m, $\delta[m]$):
begin
if $p = \emptyset$ then begin write $\delta[m]$; comment: $\delta[m]$ defines a solution-structure;
                  return end
let $x \in p$;
C:= $\wp(\Delta(x)) \setminus \{\emptyset\}$;
for all $c \in C$ do
      begin
      if $\forall y \in m$, $c \cap \bar{\delta}(y) = \emptyset \;\&\; (\Delta(x) \setminus c) \cap \delta(y) = \emptyset$
        then
          begin
          $\delta[m \cup \{x\}] := \delta[m] \cup \{(x, c)\}$;
          SSG($(p \cup \text{mat}^{in}(c)) \setminus (R \cup m \cup \{x\})$, $m \cup \{x\}$, $\delta[m \cup \{x\}]$)
          end
      end
return
end
```

Fig. 4. Algorithm SSG for generating the solution-structures of a synthesis problem.

Table 1. Recursive steps of algorithm SSG in generating the solution-structures of Example 9

Number of call	Depth of recursion	Parameter p	Parameter m	Parameter δ[m]	Remark
1	0	$\{A\}$	$\emptyset$	$\emptyset$	Initial call
2	1	$\{C\}$	$\{A\}$	$\{(A,\{1\})\}$	
3	2	$\{F\}$	$\{A,C\}$	$\{(A,\{1\}),(C,\{3\})\}$	
4	3	$\emptyset$	$\{A,C,F\}$	$\{(A,\{1\}),(C,\{3\}),(F,\{1\})\}$	Solution #1
5	3	$\emptyset$	$\{A,C,F\}$	$\{(A,\{1\}),(C,\{3\}),(F,\{1,6\})\}$	Solution #2
6	2	$\{F\}$	$\{A,C\}$	$\{(A,\{1\}),(C,\{4\})\}$	
7	3	$\emptyset$	$\{A,C,F\}$	$\{(A,\{1\}),(C,\{4\}),(F,\{1\})\}$	Solution #3
8	3	$\emptyset$	$\{A,C,F\}$	$\{(A,\{1\}),(C,\{4\}),(F,\{1,6\})\}$	Solution #4
9	2	$\{F\}$	$\{A,C\}$	$\{(A,\{1\}),(C,\{3,4\})\}$	
10	3	$\emptyset$	$\{A,C,F\}$	$\{(A,\{1\}),(C,\{3,4\}),(F,\{1\})\}$	Solution #5
11	3	$\emptyset$	$\{A,C,F\}$	$\{(A,\{1\}),(C,\{3,4\}),(F,\{1,6\})\}$	Solution #6
12	1	$\{D\}$	$\{A\}$	$\{(A,\{2\})\}$	
13	2	$\{F\}$	$\{A,D\}$	$\{(A,\{2\}),(D,\{4\})\}$	
14	3	$\emptyset$	$\{A,D,F\}$	$\{(A,\{2\}),(D,\{4\}),(F,\{6\})\}$	Solution #7
15	2	$\{H\}$	$\{A,D\}$	$\{(A,\{2\}),(D,\{5\})\}$	
16	3	$\emptyset$	$\{A,D,H\}$	$\{(A,\{2\}),(D,\{5\}),(H,\{7\})\}$	Solution #8
17	2	$\{F,H\}$	$\{A,D\}$	$\{(A,\{2\}),(D,\{4,5\})\}$	
18	3	$\{H\}$	$\{A,D,F\}$	$\{(A,\{2\}),(D,\{4,5\}),(F,\{6\})\}$	
19	4	$\emptyset$	$\{A,D,F,H\}$	$\{(A,\{2\}),(D,\{4,5\}),(F,\{6\}),(H,\{7\})\}$	Solution #9
20	1	$\{C,D\}$	$\{A\}$	$\{(A,\{1,2\})\}$	
21	2	$\{D,F\}$	$\{A,C\}$	$\{(A,\{1,2\}),(C,\{3\})\}$	
22	3	$\{F,H\}$	$\{A,C,D\}$	$\{(A,\{1,2\}),(C,\{3\}),(D,\{5\})\}$	
23	4	$\{H\}$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{3\}),(D,\{5\}),(F,\{1\})\}$	
24	5	$\emptyset$	$\{A,C,D,F,H\}$	$\{(A,\{1,2\}),(C,\{3\}),(D,\{5\}),(F,\{1\}),(H,\{7\})\}$	Solution #10
25	4	$\{H\}$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{3\}),(D,\{5\}),(F,\{1,6\})\}$	
26	5	$\emptyset$	$\{A,C,D,F,H\}$	$\{(A,\{1,2\}),(C,\{3\}),(D,\{5\}),(F,\{1,6\}),(H,\{7\})\}$	Solution #11
27	2	$\{D,F\}$	$\{A,C\}$	$\{(A,\{1,2\}),(C,\{4\})\}$	
28	3	$\{F\}$	$\{A,C,D\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4\})\}$	
29	4	$\emptyset$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4\}),(F,\{1\})\}$	Solution #12
30	4	$\emptyset$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4\}),(F,\{1,6\})\}$	Solution #13
31	3	$\{F,H\}$	$\{A,C,D\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4,5\})\}$	
32	4	$\{H\}$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4,5\}),(F,\{1\})\}$	
33	5	$\emptyset$	$\{A,C,D,F,H\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4,5\}),(F,\{1\}),(H,\{7\})\}$	Solution #14
34	4	$\{H\}$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4,5\}),(F,\{1,6\})\}$	
35	5	$\emptyset$	$\{A,C,D,F,H\}$	$\{(A,\{1,2\}),(C,\{4\}),(D,\{4,5\}),(F,\{1,6\}),(H,\{7\})\}$	Solution #15
36	2	$\{D,F\}$	$\{A,C\}$	$\{(A,\{1,2\}),(C,\{3,4\})\}$	
37	3	$\{F\}$	$\{A,C,D\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4\})\}$	
38	4	$\emptyset$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4\}),(F,\{1\})\}$	Solution #16
39	4	$\emptyset$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4\}),(F,\{1,6\})\}$	Solution #17
40	3	$\{F,H\}$	$\{A,C,D\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4,5\})\}$	
41	4	$\{H\}$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4,5\}),(F,\{1\})\}$	
42	5	$\emptyset$	$\{A,C,D,F,H\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4,5\}),(F,\{1\}),(H,\{7\})\}$	Solution #18
43	4	$\{H\}$	$\{A,C,D,F\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4,5\}),(F,\{1,6\})\}$	
44	5	$\emptyset$	$\{A,C,D,F,H\}$	$\{(A,\{1,2\}),(C,\{3,4\}),(D,\{4,5\}),(F,\{1,6\}),(H,\{7\})\}$	Solution #19

solution-structures (Friedler *et al.*, 1992a, 1993). These algorithms, in turn, have given rise to the so-called accelerated branch and bound algorithm of process synthesis that is highly efficient (Friedler *et al.*, 1990; Friedler and Fan 1993a, b). This accelerated branch and bound algorithm not only is mathematically verifiable but also effectively minimizes the number and sizes of the subproblems to be solved for generating the optimal solution. The decision-mapping and the resultant algorithms manipulate process structures explicitly; hence, the user's decisions can be incorporated interactively into the algorithmic decision procedure of process synthesis.

For illustration, let us consider algorithm SSG for generating the solution-structures of synthesis problem (P, R, O). This procedure is given in Fig. 4

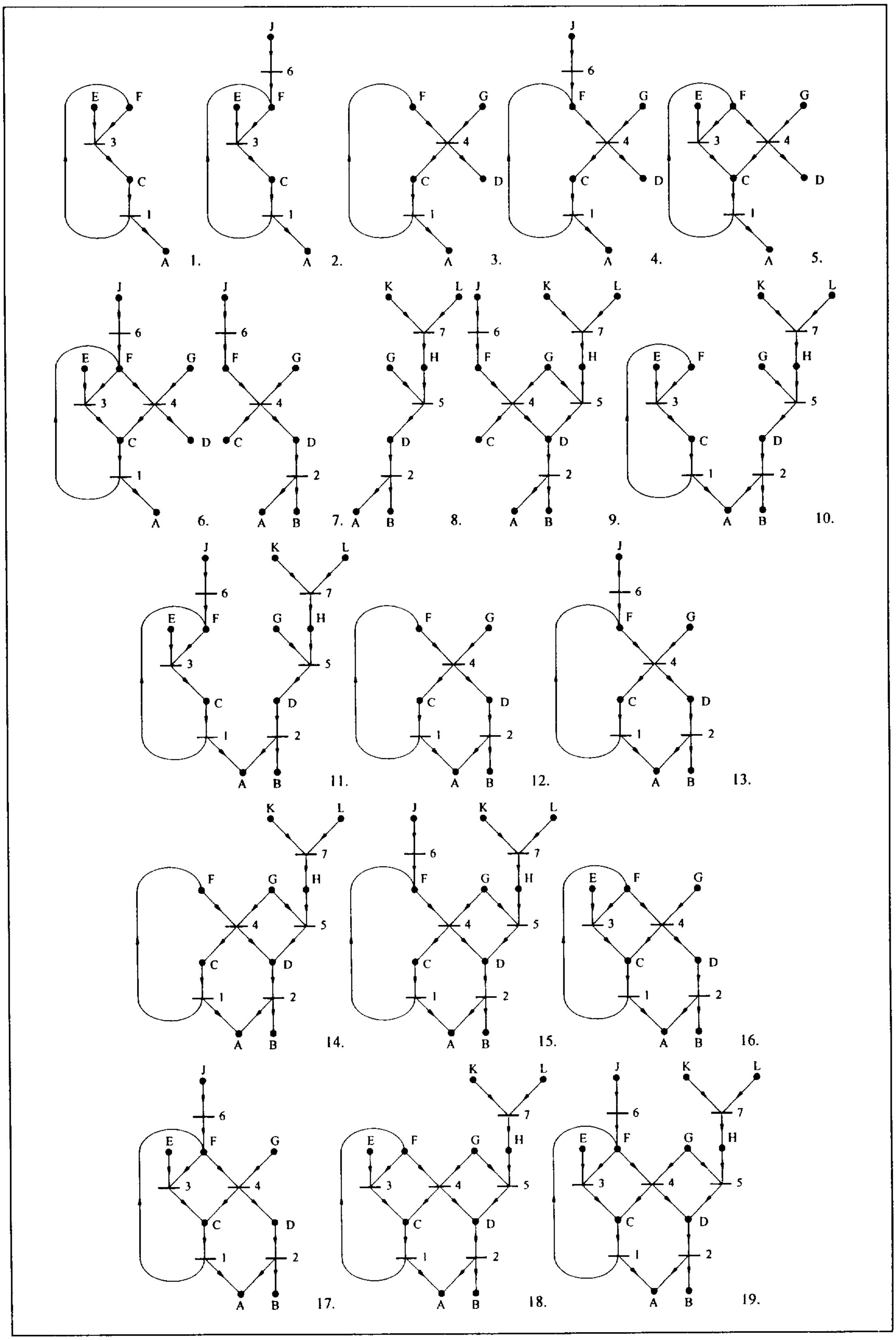

Fig. 6. Solution-structures of Example 9 generated by algorithm SSG.

Table 2. Operating units of Example 10

No.	Type	Inputs	Outputs
1.	Feeder	*A*1	*A*5
2.	Reactor	*A*2, *A*3, *A*4	*A*9
3.	Reactor	*A*3, *A*4, *A*6, *A*11	*A*10
4.	Reactor	*A*3, *A*4, *A*5	*A*12
5.	Reactor	*A*3, *A*4, *A*5	*A*13
6.	Reactor	*A*7, *A*8, *A*14	*A*16
7.	Reactor	*A*8, *A*14, *A*18	*A*16
8.	Separator	*A*9, *A*11	*A*21, *A*22, *A*24
9.	Separator	*A*10, *A*11	*A*22, *A*24, *A*37
10.	Separator	*A*12	*A*25, *A*26
11.	Separator	*A*13	*A*25, *A*31
12.	Dissolver	*A*15, *A*16	*A*32
13.	Reactor	*A*14, *A*17, *A*18, *A*19, *A*20	*A*33
14.	Reactor	*A*6, *A*21	*A*35
15.	Washer	*A*22, *A*23	*A*48
16.	Washer	*A*5, *A*24	*A*36
17.	Separator	*A*5, *A*11, *A*25	*A*37, *A*38, *A*39
18.	Separator	*A*11, *A*26	*A*40, *A*42
19.	Reactor	*A*14, *A*27, *A*28, *A*29, *A*30	*A*41
20.	Separator	*A*11, *A*31	*A*40, *A*42
21.	Centrifuge	*A*32	*A*44, *A*45
22.	Washer	*A*33, *A*34	*A*46
23.	Separator	*A*36	*A*14, *A*48
24.	Separator	*A*38	*A*14, *A*48
25.	Filter	*A*41	*A*50, *A*51
26.	Washer	*A*43, *A*44	*A*53
27.	Filter	*A*46	*A*55, *A*56
28.	Separator	*A*47, *A*48	*A*5, *A*57
29.	Separator	*A*48, *A*49	*A*5, *A*58
30.	Separator	*A*50	*A*59, *A*60
31.	Dryer	*A*51, *A*54	*A*61
32.	Dryer	*A*52, *A*53	*A*61
33.	Dryer	*A*54, *A*55	*A*61
34.	Distillation	*A*59	*A*62, *A*63
35.	Separator	*A*60	*A*64, *A*65

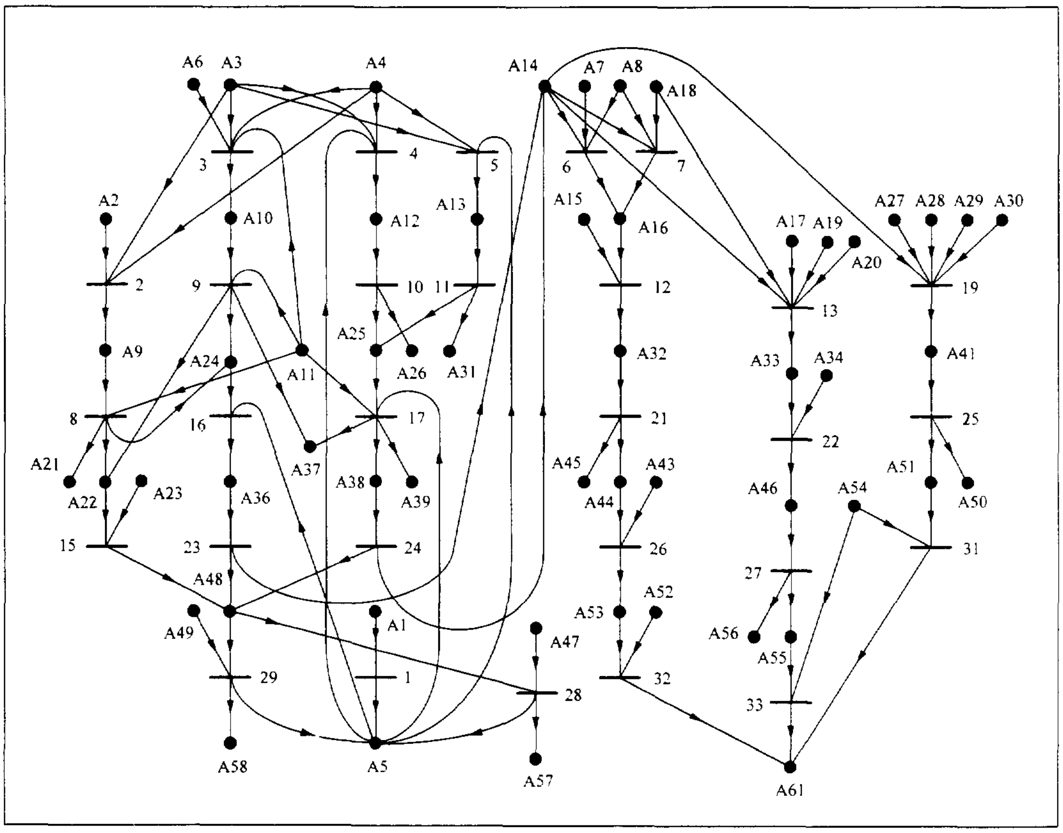

Fig. 7. Maximal structure of Example 10.

Table 3. Recursive steps of algorithm SSG in generating the solution-structures of Example 10

Number of call	Depth of recursion	Parameter p	Parameter m	Parameter $\delta[m]$	Remark
1	0	$\{A61\}$	$\emptyset$	$\emptyset$	Initial call
2	1	$\{A51\}$	$\{A61\}$	$\{(A61,\{31\})\}$	
3	2	$\{A41\}$	$\{A51, A61\}$	$\{(A51,\{25\}),(A61,\{31\})\}$	
4	3	$\{A14\}$	$\{A41, A51, A61\}$	$\{(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
5	4	$\{A36\}$	$\{A14, A41, A51, A61\}$	$\{(A14,\{23\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
6	5	$\{A5, A24\}$	$\{A14, A36, A41, A51, A61\}$	$\{(A14,\{23\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
7	6	$\{A5, A9\}$	$\{A14, A24, A36, A41, A51, A61\}$	$\{(A14,\{23\}),(A24,\{8\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
8	7	$\{A5\}$	$\{A9, A14, A24, A36, A41, A51, A61\}$	$\{(A9,\{2\}),(A14,\{23\}),(A24,\{8\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
9	8	$\emptyset$	$\{A5, A9, A14, A24, A36, A41, A51, A61\}$	$\{(A5,\{1\}),(A9,\{2\}),(A14,\{23\}),(A24,\{8\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	Solution #1
10	8	$\{A48\}$	$\{A5, A9, A14, A24, A36, A41, A51, A61\}$	$\{(A5,\{28\}),(A9,\{2\}),(A14,\{23\}),(A24,\{8\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
11	9	$\emptyset$	$\{A5, A9, A14, A24, A36, A41, A48, A51, A61\}$	$\{(A5,\{28\}),(A9,\{2\}),(A14,\{23\}),(A24,\{8\}),(A36,\{16\}),(A41,\{19\}),(A48,\{23\}),(A51,\{25\}),(A61,\{31\})\}$	Solution #2
12	9	$\{A22\}$	$\{A5, A9, A14, A24, A36, A41, A48, A51, A61\}$	$\{(A5,\{28\}),(A9,\{2\}),(A14,\{23\}),(A24,\{8\}),(A36,\{16\}),(A41,\{19\}),(A48,\{15,23\}),(A51,\{25\}),(A61,\{31\})\}$	$24 \notin \delta(A14)$
13	10	$\emptyset$	$\{A5, A9, A14, A22, A24, A36, A41, A48, A51, A61\}$	$\{(A5,\{28\}),(A9,\{2\}),(A14,\{23\}),(A22,\{8\}),(A24,\{8\}),(A36,\{16\}),(A41,\{19\}),(A48,\{15,23\}),(A51,\{25\}),(A61,\{31\})\}$	Solution #3
⋮	⋮	⋮	⋮	⋮	⋮
34	6	$\{A5, A10\}$	$\{A14, A24, A36, A41, A51, A61\}$	$\{(A14,\{23\}),(A24,\{9\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
35	7	$\{A5\}$	$\{A10, A14, A24, A36, A41, A51, A61\}$	$\{(A10,\{3\}),(A14,\{23\}),(A24,\{9\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	

Table 3. Contd.

Number of call	Depth of recursion	Parameter p	Parameter m	Parameter $\delta[m]$	Remark
36	8	$\emptyset$	$\{A5, A10, A14, A24, A36, A41, A51, A61\}$	$\{(A5,\{1\}),(A10,\{3\}),(A14,\{23\}),(A24,\{9\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	Solution #14
37	8	$\{A48\}$	$\{A5, A10, A14, A24, A36, A41, A51, A61\}$	$\{(A5,\{28\}),(A10,\{3\}),(A14,\{23\}),(A24,\{9\}),(A36,\{16\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
38	9	$\emptyset$	$\{A5, A10, A14, A24, A36, A41, A48, A51, A61\}$	$\{(A5,\{28\}),(A10,\{3\}),(A14,\{23\}),(A24,\{9\}),(A36,\{16\}),(A41,\{19\}),(A48,\{23\}),(A51,\{25\}),(A61,\{31\})\}$	Solution #15
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
89	4	$\{A38\}$	$\{A14, A41, A51, A61\}$	$\{(A14,\{24\}),(A41,\{19\}),(A51,\{25\}),(A61,\{31\})\}$	
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
537	1	$\{A53\}$	$\{A61\}$	$\{(A61,\{32\})\}$	
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
8007	21	$\{48\}$	$\{A5, A9, A10, A12, A13, A14, A16, A22, A24, A25, A32, A33, A36, A38, A41, A44, A46, A51, A53, A55, A61\}$	$\{(A5,\{1,28,29\}),(A9,\{2\}),(A10,\{3\}),(A12,\{4\}),(A13,\{5\}),(A14,\{23,24\}),(A16,\{6,7\}),(A22,\{8,9\}),(A24,\{8,9\}),(A25,\{10,11\}),(A32,\{12\}),(A33,\{13\}),(A36,\{16\}),(A38,\{17\}),(A41,\{19\}),(A44,\{21\}),(A46,\{22\}),(A51,\{25\}),(A53,\{26\}),(A55,\{27\}),(A61,\{31,32,33\})\}$	Last but one call
8008	22	$\emptyset$	$\{A5, A9, A10, A12, A13, A14, A16, A22, A24, A25, A32, A33, A36, A38, A41, A44, A46, A48, A51, A53, A55, A61\}$	$\{(A5,\{1,28,29\}),(A9,\{2\}),(A10,\{3\}),(A12,\{4\}),(A13,\{5\}),(A14,\{23,24\}),(A16,\{6,7\}),(A22,\{8,9\}),(A24,\{8,9\}),(A25,\{10,11\}),(A32,\{12\}),(A33,\{13\}),(A36,\{16\}),(A38,\{17\}),(A41,\{19\}),(A44,\{21\}),(A46,\{22\}),(A48,\{15,23,24\}),(A51,\{25\}),(A53,\{26\}),(A55,\{27\}),(A61,\{31,32,33\})\}$	Solution #3465

(Friedler *et al.*, 1992a), it generates each solution-structure of a synthesis problem exactly once; moreover, it generates only the solution-structures. This method can be validated rigorously by resorting to the theory of decision-mapping.

Example 9. For set M_1 of materials and set O_1 of operating units given in Example 1, let $P_1 = \{A\}$ be the set of products and $R_1 = \{E, G, J, K, L\}$ be the set of raw materials. Algorithm SSG requires the so-called maximal structure of the problem as its input (see Appendix B). This maximal structure, given in Fig. 5, has been created by algorithm MSG (Friedler *et al.*, 1993). Algorithm SSG generates the decision-mappings of the solution-structures recursively; the "values" of the parameters for each recursive step is listed in Table 1. Note that the number and order of calls and the order of the generation of the individual solution-structures may be effected by the implementation of algorithm SSG, nevertheless, the set of solution-structures is obviously invariant for the implementation. During the generation of solution-structures, the decision-mapping in the third parameter of algorithm SSG represents a solution-structure if the first parameter of algorithm SSG, set p, is empty. It occurs nineteen times in solving this example, thereby resulting in 19 solution-structures, as illustrated in Fig. 6.

Example 10. Let us now consider solution-structures of an industrial process synthesis problem. For this purpose the process synthesis problem introduced in Friedler *et al.* (1992a) is re-examined again. In this problem set M of materials has 65 elements, $M = \{A1, A2, \ldots, A65\}$, where $R = \{A1, A2, A3, A4, A6, A7, A8, A11, A15, A17, A18, A19, A20, A23, A27, A28, A29, A30, A34, A43, A47, A49, A52, A54\}$ is the set of raw materials. Moreover, 35 operating units are available for producing the product, material $A61$; these operating units are listed in Table 2. The maximal structure of the problem is given in Fig. 7. The recursive steps of algorithm SSG in generating the solution-structures of this example are listed in Table 3. For instance, the first solution-structure in Table 3 is composed of operating units $1, 2, 8, 16, 19, 23, 25$, and 31 of the maximal structure.

In addition to being a fundamental algorithm of the accelerated branch and bound algorithm, algorithm SSG is also capable of directly generating the mathematically valid disjunctive normal form for synthesis problems to serve as the inputs of synthesis methods based on logical formulation [see e.g. Raman and Grossmann (1993)].

7. CONCLUDING REMARKS

A novel mathematical notion, decision-mapping, has been introduced to render the complex decision systems of process synthesis consistent and complete. The major properties of decision-mapping have been identified and proved; moreover, its relationship to P-graphs has been established. The application of decision-mapping is illustrated by generating the solution-structures of an industrial process synthesis problem. The result indicates that it is highly efficient, exact, and useful.

Acknowledgements—The authors thank Professor B. Imreh of Department of Computer Science, University of Szeged, Hungary, for critically reviewing our manuscript. This research was partially supported by the Hungarian Academy of Sciences. Although the research described in this article has been funded in part by the United States Environmental Protection Agency under assistance agreement R-819653 to the Great Plains-Rocky Mountain Hazardous Substance Research Center for U.S. EPA Regions 7 and 8 with headquarters at Kansas State University, it has not been subjected to the Agency's peer and administrative review and, therefore, may not necessarily reflect the views of the Agency. No official endorsement should be inferred. This research was partially supported by the Kansas State University Center for Hazardous Substance Research.

NOTATION

$d^{-}(X)$	indegree of vertex X, i.e. the number of arcs with endpoint X
$f(x)$	"value" determined by the mapping or function for an element of the domain, x
iff	if and only if
$(m, o), (M, O)$	P-graph
m, M	set of materials
mat	set of materials involved in a set of operating units
mat^{in}	set of input materials for a set of operating units
o, O	set of operating units
op	operating units of decision-mapping
P	set of products
R	set of raw materials
(P, R, O)	synthesis problem defined by the specific set of products (P), raw materials (R), and operating units (O)
$S(P, R, O)$	set of solution-structures for synthesis problem (P, R, O)
$[y_i, y_j]$	path in P-graph
$f[D]$	mapping or function, where f is the name, D is the domain of the mapping or function
Greek letters	
$\mu(P, R, O)$	maximal structure for the synthesis problem (P, R, O)
σ	P-graph
Δ	maximal decision-mapping
δ	decision-mapping
$\bar{\delta}$	complement of decision-mapping δ
Mathematical symbols	
$\emptyset$	empty set
$\wp$	power set
$\forall$	for any
$\exists$	there exists
$\times$	Cartesian product
$\{\ \}$	set
$\|\ \|$	cardinality of a set
$\setminus$	set difference
$(\subset)\subseteq$	(proper) subset or subgraph
$\in$	element
$\notin$	not an element
$\cap$	intersection of sets or graphs
$\cup$	union of sets or graphs
$>$	extension defined on decision-mappings

REFERENCES

Douglas, J. M., 1988, *Conceptual Design of Chemical Processes*. McGraw-Hill, New York.

Friedler, F. and Fan, L. T., 1993a, Combinatorial acceleration of the branch and bound search for process network synthesis. *Proceedings of Symposium on Applied Mathematical Programming and Modeling*, Budapest, Hungary, pp. 192–200.

Friedler, F. and Fan, L. T., 1993b, Reduction of subproblem size for the accelerated branch and bound method of process synthesis. *AIChE National Meeting*, Houston, TX, U.S.A., March 28–April 1.

Friedler, F., Tarjan, K., Huang, Y. W. and Fan, L. T., 1990, Combinatorial acceleration of branch and bound search in process synthesis. *AIChE Annual Meeting*, Chicago, IL, U.S.A., November 11–16.

Friedler, F., Tarjan, K., Huang, Y. W. and Fan, L. T., 1992a, Combinatorial algorithms for process synthesis. *Comput. chem. Engng* **16**, supplement, S313–320.

Friedler, F., Tarjan, K., Huang, Y. W. and Fan, L. T., 1992b Combinatorial structure of process network synthesis. *Sixth SIAM Conference on Discrete Mathematics*, Vancouver, Canada, June 8–11.

Friedler, F., Tarjan, K., Huang, Y. W. and Fan, L. T., 1992c, Graph-theoretic approach to process synthesis: axioms and theorems. *Chem. Engng Sci.* **47**, 1973–1988.

Friedler, F., Tarjan, K., Hunag, Y. W. and Fan, L. T., 1993, Graph-theoretic approach to process synthesis: polynomial algorithm for maximal structure generation. *Comput. chem. Engng* **17**, 929–942.

Lu, M. D. and Motard, R. L., 1985, Computer-aided total flowsheet synthesis. *Comput. chem. Engng* **9**, 431–445.

Raman, R. and Grossmann, I. E., 1993, Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis. *Comput. chem. Engng* **13**, 909–927.

Siirola, J. J. and Rudd, D. F., 1971, Computer-aided synthesis of chemical process designs. *Ind. Engng Chem. Fundam.* **10**, 353–362.

APPENDIX A: PROOFS OF THEOREMS

Proof of Theorem 1. Obviously, $\Delta(X) = \delta(X) \cup \bar{\delta}(X)$ and $\Delta(Y) = \delta(Y) \cup \bar{\delta}(Y)$; hence,

$$\Delta(X) \cap \Delta(Y) = (\delta(X) \cap \delta(Y)) \cup (\delta(X) \cap \bar{\delta}(Y)) \cup (\bar{\delta}(X) \cap \delta(Y)) \cup (\bar{\delta}(X) \cap \bar{\delta}(Y)). \quad \text{(A1)}$$

(i) If $\delta(X) \cap \bar{\delta}(Y) = \emptyset$ for all $X, Y \in m$, i.e. $\bar{\delta}(X) \cap \delta(Y) = \emptyset$, then, $\Delta(X) \cap \Delta(Y) = (\delta(X) \cap \delta(Y)) \cup (\bar{\delta}(X) \cap \bar{\delta}(Y))$ from expression (A1).

(ii) Conversely, let us suppose that $(\delta(X) \cap \delta(Y)) \cup (\bar{\delta}(X) \cap \bar{\delta}(Y)) = \Delta(X) \cap \Delta(Y)$ for any $X, Y \in m$. Since $((\delta(X) \cap \delta(Y)) \cup (\bar{\delta}(X) \cap \bar{\delta}(Y)) \cap (\delta(X) \cap \bar{\delta}(Y)) = \emptyset$ and $((\delta(X) \cap \delta(Y)) \cup (\bar{\delta}(X) \cap \bar{\delta}(Y))) \cap (\bar{\delta}(X) \cap \delta(Y)) = \emptyset$, we have $\delta(X) \cap \bar{\delta}(Y) = \emptyset$ and $\bar{\delta}(X) \cap \delta(Y) = \emptyset$ from expression (A1).

Proof of Theorem 2.

(i) For $X \in m$, $\delta(X) = \{(\alpha, \beta) | (\alpha, \beta) \in \delta(X) \text{ and } X \in \beta\} \subseteq \{(\alpha, \beta) | (\alpha, \beta) \in \mathbf{o} \text{ and } X \in \beta\} = \delta'(X)$.

(ii) Suppose that there exists an (α, β) for $X \in m$ such that $(\alpha, \beta) \in \mathbf{o}$, $X \in \beta$, and $(\alpha, \beta) \notin \delta(X)$. Consequently, $(\alpha, \beta) \in \bar{\delta}(X)$, and there exists $Y \in m$ such that $(\alpha, \beta) \in \delta(Y)$, i.e. $\bar{\delta}(X) \cap \delta(Y) \neq \emptyset$ in contradiction to $\delta[m]$ being consistent. Thus, $(\alpha, \beta) \in \delta(X)$ and $\delta(X) = \{(\alpha, \beta) | (\alpha, \beta) \in \mathbf{o} \text{ and } X \in \beta\} = \delta'(X)$.

Proof of Theorem 3. Let $\delta[\mathbf{m}]$ be the closure of consistent decision-mapping $\delta[m]$. Naturally, $m \subseteq \mathbf{m}$.

(i) For $X \in m$, $\delta'(X) = \delta(X)$; thus, if $|m| \geqslant 1$ and $X, Y \in m$, then, $\bar{\delta}'(X) \cap \delta'(Y) = \emptyset$.

(ii) Suppose that $\mathbf{m}/m \neq \emptyset$, and let $X \in m$ and $Y \in \mathbf{m}/m$. Then, $\bar{\delta}'(X) \cap \delta'(Y) = \bar{\delta}(X) \cap \{(\alpha, \beta) | (\alpha, \beta) \in \mathbf{o}$ and $Y \in \beta\} \subseteq \bar{\delta}(X) \cap \mathbf{o} = \bar{\delta}(X) \cap (\bigcup_{Z \in m} \delta(Z)) = \bigcup_{Z \in m} (\bar{\delta}(X) \cap \delta(Z)) = \emptyset$; thus, $\bar{\delta}(X) \cap \delta'(Y) = \emptyset$. On the other hand, since $\delta'(Y) \subseteq \Delta(Y)$, $\delta'(X) \cap \bar{\delta}'(Y) = \delta(X) \cap (\Delta(Y)/\delta'(Y)) = \delta(X) \cap \Delta(Y)/\delta(X) \cap \delta'(Y)$. Nevertheless, $\delta(X) \cap \Delta(Y) = \delta(X) \cap \{(\alpha, \beta) | (\alpha, \beta) \in O$ and $Y \in \beta\} = \{(\alpha, \beta) | (\alpha, \beta) \in \delta(X)$ and $Y \in \beta\}$, $\delta(X) \cap \delta'(Y) = \delta(X) \cap \{(\alpha, \beta) | (\alpha, \beta) \in \mathbf{o}$ and $Y \in \beta\} = \{(\alpha, \beta) | (\alpha, \beta) \in \delta(X)$ and $Y \in \beta\}$; consequently, $\delta(X) \cap \Delta(Y) = \delta(X) \cap \delta'(Y)$, and eventually, $\delta'(X) \cap \bar{\delta}'(Y) = \emptyset$.

(iii) For $X, Y \in \mathbf{m}/m$, it can be similarly proved that $\delta'(X) \cap \bar{\delta}'(Y) = \emptyset$ provided that $|\mathbf{m}/m| \geqslant 1$.

Proof of Theorem 4. Let m' be an active domain of consistent decision-mapping $\delta[m]$, and also let $Y \in m/m'$.

(i) Obviously, $\delta(Y) \subseteq (\bigcup_{X \in m} \delta(X)) = (\bigcup_{X \in m'} \delta(X))$ and $\delta(Y) \subseteq \Delta(Y)$; thus,

$$\delta(Y) \subseteq \Delta(Y) \cap \left(\bigcup_{X \in m'} \delta(X) \right) = \bigcup_{X \in m'} (\Delta(Y) \cap \delta(X)).$$

(ii) Since $\delta[m]$ is consistent, $(\delta(X) \cap \delta(Y)) \cup (\bar{\delta}(X) \cap \bar{\delta}(Y)) = \Delta(X) \cap \Delta(Y)$ for any $X \in m'$. The intersections of both sides of this equation with set $\delta(X)$ result in

$$\delta(X) \cap \delta(Y) = (\Delta(X) \cap \delta(X)) \cap \Delta(Y).$$

This can be rewritten as $\delta(X) \cap \delta(Y) = \delta(X) \cap \Delta(Y)$. Obviously, $\delta(Y) \supseteq \delta(X) \cap \delta(Y) = \delta(X) \cap \Delta(Y)$ for any $X \in m'$; hence,

$$\delta(Y) \supseteq \bigcup_{X \in m'} (\delta(X) \cap \Delta(Y)).$$

(iii) From (i) and (ii), we have

$$\delta(Y) = \bigcup_{X \in m'} (\Delta(Y) \cap \delta(X)) = \mathrm{op}(\delta[m']) \cap \Delta(Y).$$

Note that this is the formula for determining $\delta(Y)$ for any $Y \in m/m'$.

Proof of Theorem 5. Let m' be an active domain of decision-mapping $\delta[m]$, and $\bar{\delta}(X) \cap \delta(Y) = \emptyset$ for all $X, Y \in m'$.

(i) If $X \in m'$ and $Y \in m/m'$, then by $\mathrm{op}(\delta[m']) = \mathrm{op}(\delta[m])$, $\bar{\delta}(X) \cap \delta(Y) \subseteq \bar{\delta}(X) \cap \left(\Delta(Y) \cap \left(\bigcup_{Z \in m'} \delta(Z) \right) \right) = \Delta(Y) \cap \left(\bigcup_{Z \in m'} (\delta(Z) \cap \bar{\delta}(X)) \right) = \Delta(Y) \cap \emptyset = \emptyset$.

Similarly, by

$$\mathrm{op}(\bar{\delta}[m']) = \mathrm{op}(\bar{\delta}[m]),$$

$$\delta(X) \cap \bar{\delta}(Y) \subseteq \delta(X) \cap \left(\Delta(Y) \cap \left(\bigcup_{Z \in m'} \bar{\delta}(Z) \right) \right)$$

$$= \Delta(Y) \cap \left(\bigcup_{Z \in m'} (\delta(X) \cap \bar{\delta}(Z)) \right) = \Delta(Y) \cap \emptyset = \emptyset.$$

(ii) Suppose that $|m/m'| \geqslant 2$ and let $X, Y \in m/m'$. Then,

$$\delta(X) \cap \bar{\delta}(Y) \subseteq \Delta(X) \cap \left(\bigcup_{Z \in m'} \delta(Z) \right)$$

$$\cap \Delta(Y) \cap \left(\bigcup_{U \in m'} \bar{\delta}(U) \right) = \Delta(X) \cap \Delta(Y)$$

$$\cap \left(\bigcup_{Z, U \in m'} (\delta(Z) \cap \bar{\delta}(U)) \right) = \Delta(X) \cap \Delta(Y) \cap \emptyset = \emptyset.$$

Proof of Theorem 6. The reflexivity of relation extension is satisfied trivially. The antisymmetry will be proved at the outset. Let $\delta_1[m_1]$ and $\delta_2[m_2]$ be consistent decision-mappings with closures $\delta'_1[\mathbf{m}_1]$ and $\delta'_2[\mathbf{m}_2]$, respectively. Moreover, let us suppose that $\delta_1[m_1] > \delta_2[m_2]$ and $\delta_2[m_2] > \delta_1[m_1]$. Then, from the definition of extension, we have $\mathbf{m}_1 \supseteq \mathbf{m}_2$, $\mathbf{m}_2 \supseteq \mathbf{m}_1$, $m_1 \supseteq m_2$, and $m_2 \supseteq m_1$, i.e. $\mathbf{m}_1 = \mathbf{m}_2$ and $m_1 = m_2$; therefore, $\delta_1(X) = \delta_2(X)$ for $X \in m_2$.

The proof of transitivity follows. Let us suppose that $\delta_1[m_1] > \delta_2[m_2]$ and $\delta_2[m_2] > \delta_3[m_3]$; then, we need to prove that $\delta_1[m_1] > \delta_3[m_3]$. The first and second conditions of extension are satisfied, since the relations "subset" and "equal to" are also transitive. From $\delta_1[m_1] > \delta_2[m_2]$, $\delta'_1(X) = \delta_2(X)$, $\delta_2(X) = \delta'_2(X)$ for $X \in m_2$, and $\delta'_1(X) \supseteq \delta'_2(X)$ for $X \in \mathbf{m}_2/m_2$. Hence, $\delta'_1(X) \supseteq \delta'_2(X)$ for $X \in \mathbf{m}_2$. $\delta_2[m_2] > \delta_3[m_3]$; hence, $\delta'_2(X) \supseteq \delta'_3(X)$ for $X \in \mathbf{m}_3/m_3$. Thus, we have $\delta'_1(X) \supseteq \delta'_3(X)$ for $X \in \mathbf{m}_3/m_3$.

Proof of Theorem 7. Let m' be an active set of P-graph (m, o), and suppose that $|m'| \geqslant 2$; moreover, let X and Y be elements of m'. If $\Delta(X) \cap \Delta(Y) = \emptyset$, then, $\bar{\delta}(X) \cap \delta(Y) = \emptyset$. Let us suppose that $\Delta(X) \cap \Delta(Y) \neq \emptyset$ and let (α, β) be an element of $\Delta(X) \cap \Delta(Y)$; then, $X, Y \in \beta$. If $(\alpha, \beta) \in o$, then, $(\alpha, \beta) \in \delta(X)$ and $(\alpha, \beta) \in \delta(Y)$, i.e. $(\alpha, \beta) \in \delta(X) \cap \delta(Y)$. If $(\alpha, \beta) \notin o$, then, $(\alpha, \beta) \notin \delta(X)$, $(\alpha, \beta) \in \bar{\delta}(X)$, and $(\alpha, \beta) \in \bar{\delta}(Y)$; thus, $(\alpha, \beta) \in \bar{\delta}(X) \cap \bar{\delta}(Y)$.

Proof of Theorem 8. Let $\delta[m] = \{(X, Y) | X \in m$ and $Y = \{(\alpha, \beta) | (\alpha, \beta) \in o$ and $X \in \beta\}\}$ be the decision-mapping of (m, o) on m. Since m is an active set, $\bigcup_{X \in m} \delta(X) = o$, $o = \mathbf{o}$, and for P-graph (m, o), $\bigcup_{(\alpha, \beta) \in o} (\alpha \cup \beta) \subseteq m$; thus, $\mathbf{m} = m$ and $\delta'[\mathbf{m}] = \delta[m]$.

Proof of Theorem 9. Let $\delta[m]$ be a decision-mapping and m' be an active set of P-graph (m, o). Then, for any $(\alpha, \beta) \in o$, there exists an $X \in m'$ such that $(\alpha, \beta) \in \delta(X)$; thus, $o = \bigcup_{X \in m'} \delta(X)$. Since $m' \subseteq m$, we have $\bigcup_{X \in m'} \delta(X) \subseteq \bigcup_{X \in m} \delta(X) = o$. Thus, $\bigcup_{X \in m'} \delta(X) = \bigcup_{X \in m} \delta(X)$. The assumption, $\text{op}(\bar{\delta}[m']) = \text{op}(\bar{\delta}[m])$, implies that m' is an active domain of $\delta[m]$.

Proof of Theorem 10. By Theorems 7 and 8, decision-mapping $\delta[m]$ of P-graph (m, o) is consistent and it is also closed. Let us suppose that there exists another active set m_1 of P-graph (m, o); then, by Theorem 7, decision-mapping $\delta_1[m_1]$ of P-graph (m, o) is consistent. Since m_1 is an active set of (m, o), $o = \text{op}(\delta_1[m_1])$. Then, by the assumption that $m = \text{mat}(o)$, the closure of $\delta_1[m_1]$ is equal to the closure of $\delta[m]$; thus, they are equivalent.

Proof of Theorem 11.

(i) $m \cap o = \emptyset$ and $o \subseteq \wp(m) \times \wp(m)$ are trivially satisfied.

(ii) From the construction of o, it follows that m' is an active set of P-graph (m, o).

(iii) $\delta[m']$ is a decision-mapping of P-graph (m, o): For all $X \in m', \delta(X) \subseteq \{(\alpha, \beta) \in o$ and $X \in \beta\}$ follows from the construction of o; the equality also holds, since $\delta[m']$ is consistent.

Proof of Theorem 12. Let $\delta[m']$ be a consistent decision-mapping; m'' be an active domain of $\delta[m']$; and (m, o) be the P-graph of $\delta[m']$. Since m'' is active domain of $\delta[m']$, $o = \bigcup_{X \in m'} \delta(X) = \bigcup_{X \in m''} \delta(X)$. Thus, for all $(\alpha, \beta) \in o$, there exists $X \in m''$ such that $(\alpha, \beta) \in \delta(X)$, i.e., $\beta \cap m''$ is not empty.

Proof of Theorem 13. Let $\delta_1[m_1]$ and $\delta_2[m_2]$ be two equivalent consistent decision-mappings. By definition, they share a closure; thus $o_1 = \bigcup_{X1 \in m1} \delta_1(X_1) = \bigcup_{X2 \in m2} \delta_2(X_2) = o_2$. It follows that the material sets constructed according to Definition 11 are also identical.

APPENDIX B: FORMAL DEFINITIONS OF PROCESS SYNTHESIS AND COMBINATORIAL PROPERTIES OF FEASIBLE PROCESS STRUCTURES

Let M be a finite nonempty set of objects, usually materials. A *synthesis problem* is defined to be a triplet (P, R, O) where $P(\subset M)$ is a set of final products; $R(\subset M)$ is a set of raw materials $(P \cap R = \emptyset)$; and $O \subseteq (\wp(M) \times \wp(M))$ is a set of operating units. This triplet determines a P-graph which is defined by pair (M, O) in the usual fashion. If (y_{i-1}, y_i) is an arc of the P-graph for $i = 1, 2, \ldots, n$, then $[y_0, y_n]$ is a *path* in P-graph (M, O).

P-graph (m, o) is a *solution-structure* of synthesis problem (P, R, O) if it satisfies the following axioms:

(S1) $P \subseteq m$;

(S2) $\forall X \in m$, $d^-(X) = 0$ iff $X \in R$;

(S3) $o \subseteq O$;

(S4) $\forall y_0 \in o$, $\exists$ path $[y_0, y_n]$, where $y_n \in P$;

(S5) $\forall X \in m$, $\exists(\alpha, \beta) \in o$ such that $X \in (\alpha \cup \beta)$.

The set of solution-structures of synthesis problem (P, R, O) is denoted by $S(P, R, O)$. P-graph $\mu(P, R, O)$ is defined to be the *maximal structure* of synthesis problem (P, R, O) by the following equation: $\mu(P, R, O) = \bigcup_{\sigma \in S(P, R, O)} \sigma$.