ELSEVIER

Computers and Chemical Engineering 24 (2000) 1859–1864

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Combinatorial foundation for logical formulation in process network synthesis

M.H. Brendel [a], F. Friedler [a,b,*], L.T. Fan [b]

[a] *Department of Computer Science, University of Veszprém, Veszprém, Egyetem u. 10, H-8200, Hungary*
[b] *Department of Chemical Engineering, Kansas State University Manhattan, KS 66506, USA*

Received 6 April 1998; received in revised form 9 September 1999; accepted 20 April 2000

Abstract

A rigorous foundation is given for the logical formulation of process network synthesis on the basis of the combinatorial axioms of feasible processing networks. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Logical formulation; Process network synthesis; Combinatorial axioms

1. Introduction

In process synthesis, a process network can be represented in either a graph-theoretical or logical mode. A rigorous combinatorial foundation has been developed for the former by Friedler, Tarján, Huang and Fan (1992a), and the logical formulation with the conjunctive normal forms (CNF) and the disjunctive normal forms (DNF) has been introduced for the latter by Raman and Grossmann (1993). To facilitate the application of the logical formulation, however, its mathematical foundation need be rendered rigorous. The present note is intended to accomplish this by the rigorous combinatorial approach; the exact mathematical transformation from the combinatorial approach to the approach based on logical formulation will be identified. This transformation furnishes those who prefer logical formulation with all the effective devices and tools developed through the combinatorial approach.

The process structures can be represented unambiguously by P-graphs in process network synthesis (see Friedler et al., 1992a). The rigorously defined super-structure, i.e. the maximal structure, can be algorithmically generated (Friedler, Tarján, Huang & Fan, 1993), thus rendering it possible to take into account complex structures, such as those with a large number of interconnected loops, in process network synthesis. The risk of excluding feasible or potentially optimal networks from consideration and the unnecessary consideration of infeasible networks, therefore, are eliminated. All algorithms established for the combinatorial approach can be adopted; examples of such algorithms are those for accelerating the branch-and-bound search (Friedler, Varga, Fehér & Fan, 1996) and for integrating the synthesis of a process and its waste treatment system (Friedler, Varga & Fan, 1995b). In other words, this note will demonstrate that well-defined algorithms are available for rigorously generating both the CNF and DNF in logical formulation.

2. Generation of the logical forms based on the super-structure

The procedure of Raman and Grossmann (1991a,b, 1993, 1994) for generating the CNF assumes that the super-structure of the synthesis problem of concern is represented by a simple directed graph. The nodes in this graph are classified into various groups depending on the tasks to be performed. Interconnecting nodes

* Corresponding author. Tel.: +36-88-424483; fax: +36-88-428275.
E-mail address: friedler@dcs.vein.hu (F. Friedler).

0098-1354/00/$ - see front matter
PII: S0098-1354(00)00569-X

represent splitters, mixers, sources, and sinks; and the remaining nodes, the processing or operating units. A Boolean variable is assigned to each node in the graph; specifically, variables Z's are for the interconnecting nodes, and variables Y's are for the remaining nodes. This gives rise to the following relationships (Raman & Grossmann, 1993).

$Zm \Rightarrow Ya \vee Yb$ and $Zm \Rightarrow Yc$ correspond to the mixers, where Ya and Yb are the inputs and Yc is the output of mixer Zm; $Zs \Rightarrow Ya$ and $Zs \Rightarrow Yb \vee Yc$, the splitters, where Ya is the input and Yb and Yc are the outputs of splitter Zs; and $Yu \Rightarrow Ya \wedge Yb \wedge \ldots \wedge Yn$ and $Yu \Rightarrow Y1 \wedge Y2 \wedge \ldots \wedge Ym$, the operating units, where Ya, Yb,..., and Yn are the inputs and Y1, Y2,..., and Ym are the outputs of operating unit Yu.

3. Combinatorial theory for process network synthesis

For convenience, the combinatorial theory of process network synthesis is outlined below (Friedler et al., 1992a; Friedler, Fan & Imreh, 1998).

Let M be a given non-empty finite set of all materials to be involved in a process synthesis problem. This problem can then be defined structurally by triplet (P, R, O), where P is the set of products to be manufactured; R, the set of raw materials; and O, the set of operating units. The relationship among M, P, R, and O can be mathematically expressed as $P \subset M$; $R \subset M$; $P \cap R = \emptyset$; $P \neq \emptyset$; $M \cap O = \emptyset$; and $O \subseteq \wp(M) \times \wp(M)$ where $\wp$ denotes the power-set.

To represent a process structure in synthesis, P-graphs can be conveniently adopted; a P-graph is defined as pair (m,σ) of finite sets of materials m and operating units σ provided that $\sigma \subseteq \wp(m) \times \wp(m)$ is satisfied. The nodes of the graph are in set $N = m \cup \sigma$ and the arcs in $A = \{(x, y): x = (\alpha, \beta) \in \sigma, y \in m, y \in \beta\} \cup \{(y, x): x = (\alpha, \beta) \in \sigma, y \in m, y \in \alpha\}$. Moreover, $d^-(\chi)$ denotes the in-degree of material χ ($\chi \in m$), i.e. $d^-(\chi) = |\Delta(\chi)|$, where $\Delta(\chi) = \{(\alpha, \beta) \in \sigma: \chi \in \beta\}$; and [x, y] a path from node x to node y in the graph.

The P-graph is a general representation tool implementable with facility and rigor for solving a wide variety of synthesis problems ranging from the synthesis of networks of elementary reactions to the synthesis of processing systems. Nevertheless, this paper is intended neither to contrast the domains of applicability of the combinatorial and logical formulations nor to compare the efficacy of the two formulations in their applications.

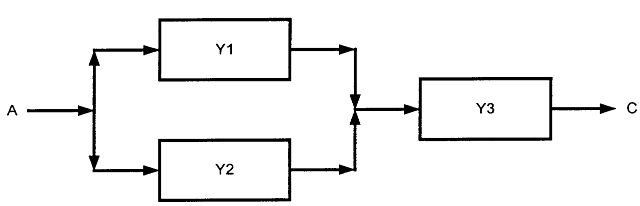

Fig. 1. Simple synthesis problem for illustration (Raman & Grossmann, 1993).

The following set of axioms has been constructed to express the combinatorial properties to which a feasible process structure, P-graph (m, σ), should conform (see, e.g. Friedler et al., 1992a).

3.1. Axioms of combinatorially feasible networks or structures

P-graph (m, σ) is combinatorially feasible for synthesis problem (P, R, O) if the following axioms are satisfied.

(S1) Every final product is represented in the graph, i.e. $P \subseteq m$.
(S2) A vertex representing a material in m has no input if and only if it represents a raw material, i.e. $(\forall \chi \in m)(d^-(\chi) = 0 \Leftrightarrow \chi \in R)$.
(S3) Every operating unit represented in the graph is defined in the synthesis problem, i.e. $\sigma \subseteq O$.
(S4) Every vertex of the graph representing an operating unit has at least one path leading to a vertex representing a final product, i.e. $(\forall y_0 \in \sigma)((\exists \text{ path } [y_0, y_1])(y_1 \in P))$.
(S5) Every vertex of the graph representing a material must be an input to or an output from at least one vertex representing an operating unit in the graph, i.e. $(\forall \chi \in m)((\exists (\alpha, \beta) \in \sigma)(\chi \in (\alpha \cup \beta)))$.

Any feasible network of a synthesis problem must satisfy the above five axioms; moreover, any network satisfying these axioms can be a feasible structure for the problem. The set of axioms, therefore, can not be extended to reduce the set of solutions further without the risk of eliminating the optimal solution.

It is worth noting that two classes of information pertaining to process structures need be strictly distinguished in developing a process synthesis procedure so that it is verifiable whenever necessary. The first is the structural properties to be satisfied by all solutions because of the inherent characteristics of process structures they represent, and the second is additional properties reflecting the designer's wishes, e.g. to exclude the parallel production of a certain material, even though they may lead to the elimination of some feasible structures.

Obviously, the axioms of combinatorial feasible structures belong to the first class of information; hence, they can serve as the basis of a synthesis method or algorithm. The space of feasible structures may be reduced by resorting to the second class of information, thereby giving rise to a parameter or parameters of the

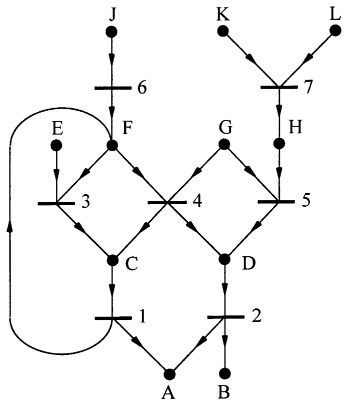

Fig. 2. P-graph of the example (Friedler et al., 1995a).

synthesis algorithm controllable externally (with the risk of excluding the optimal solution).

The simple synthesis problem depicted in Fig. 1 (see Raman & Grossmann, 1993) yields three solutions: one containing operating units 1 and 3; one containing operating units 2 and 3; and one containing operating units 1, 2, and 3. Even though the co-existence of operating units 1 and 2 is not optimal in most cases, it should not be excluded in general. Nevertheless, the designer can furnish an extra constraint to exclude all or certain parallel processes. Algorithm SSG (Friedler, Tarján, Huang & Fan, 1992b) generates all networks by taking into account every subset of the set of operating units producing a material, i.e. its power-set; however, algorithm SSG contains a control parameter for the designer to exclude certain networks by exploring a subset of this power-set, e.g. the set of singletons, to exclude parallel production.

4. CNF based on the combinatorial axioms

The valid CNF of a process-network synthesis problem can be generated algorithmically on the basis of the axioms presented. If P-graph (m, σ) satisfies axiom (S5), then it is determined by the set of operating units, σ (Friedler et al., 1992a). If axiom (S3) is also satisfied, the set of materials can be specified as $m = \cup_{(\alpha, \beta)\in\sigma} (\alpha \cup \beta)$. Hence, axiom (S5) can be replaced by this expression if axiom (S3) is satisfied. To represent a process structure described by $\sigma(\subseteq O)$, logical variables are associated with every operating unit in set O. Let $O = \{(\alpha_1, \beta_1),(\alpha_2, \beta_2),\ldots,(\alpha_k, \beta_k)\}$; then, logical variable Y_i is associated with operating unit $o_i = (\alpha_i, \beta_i)$; its value is true if and only if $o_i \in \sigma$. As a result, Boolean variables $Y_1, Y_2,\ldots,$ and Y_k define a process structure. In the current treatment, the set of materials is given by $M = \{m_1, m_2,\ldots, m_n\}$, and for convenience, the two index sets, I and J, are defined as $I = \{1, 2,\ldots, k\}$ and $J = \{1, 2,\ldots, n\}$, respectively.

Let us define the sets of indices of the products, $MP = \{j\in J\colon m_j \in P\}$; the operating units producing material m_j $(j\in J)$, $OM\_j = \{i\in I\colon m_j \in \beta_i\}$; the operating units producing any raw material, $OR = \{i\in I\colon R \cap \beta_i \neq \emptyset\}$; the input materials to operating unit $o_i(i\in I)$ excluding raw materials, $MN\_i = \{j\in J\colon m_j \in \alpha_i \backslash R\}$; the operating units producing no product, $OP = \{i\in I\colon P \cap \beta_i = \emptyset\}$; and the operating units, each receiving an output from operating unit $o_i(i\in I)$ as an input, $OS\_i = \{h\in I\colon \beta_i \cap \alpha_h \neq \emptyset\}$.

The CNF of the process network synthesis problem can be generated algorithmically as $(O1) \wedge (O2) \wedge (O3) \wedge (O4)$, where (O1) through (O4) are the logical forms of the axioms of combinatorially feasible structures as given below.

(O1) $\bigwedge_{j\in MP} (\bigvee_{i\in OM\_j} Y_i)$

(O2) $\bigwedge_{i\in OR} \neg Y_i$

(O3) $\bigwedge_{i\in I,\, j\in MN\_i} (\neg Y_i \vee (\bigvee_{h\in OM\_j} Y_h))$

(O4) $\bigwedge_{i\in OP} (\neg Y_i \vee (\bigvee_{h\in OS\_i} Y_h))$

In the following, only the derivation of (O1) is given in detail because of space limitation. Axiom (S1) states that each final product is represented on a feasible process structure. Since a product cannot be a raw material, it must be an output from at least one operating unit represented in the structure according to axiom (S2). In other words, for any $j\in MP$, for which m_j is a product, there is at least one $i\in OM\_j$, where operating unit o_i produces m_j, such that Y_i is true; consequently, $\wedge_{j\in MP} (\vee_{i\in OM\_j} Y_i)$.

Note that expressions (O3) and (O4) are derived from the expressions, $\wedge_{i\in I,\, j\in MN\_i}(Y_i \Rightarrow (\vee_{h\in OM\_j} Y_h))$ and $\wedge_{i\in OP}(Y_i \Rightarrow (\vee_{h\in OS\_i} Y_h))$, respectively. These forms are similar to the predecessor and successor premiere cuts implemented by Hooker, Yan, Grossmann and Raman (1994); however, expressions (O3) and (O4) are more exact, they have been constructed from the axioms of the combinatorial theory. Expressions (O2) and (O3) are collectively equivalent to axiom (S2). Note that axiom (S3) is implicitly included in the system in defining the Boolean variables. Expression (O4) corresponds to axiom (S4); nevertheless, the latter is more rigorous than the former, i.e. the latter may exclude more infeasible networks than the former.

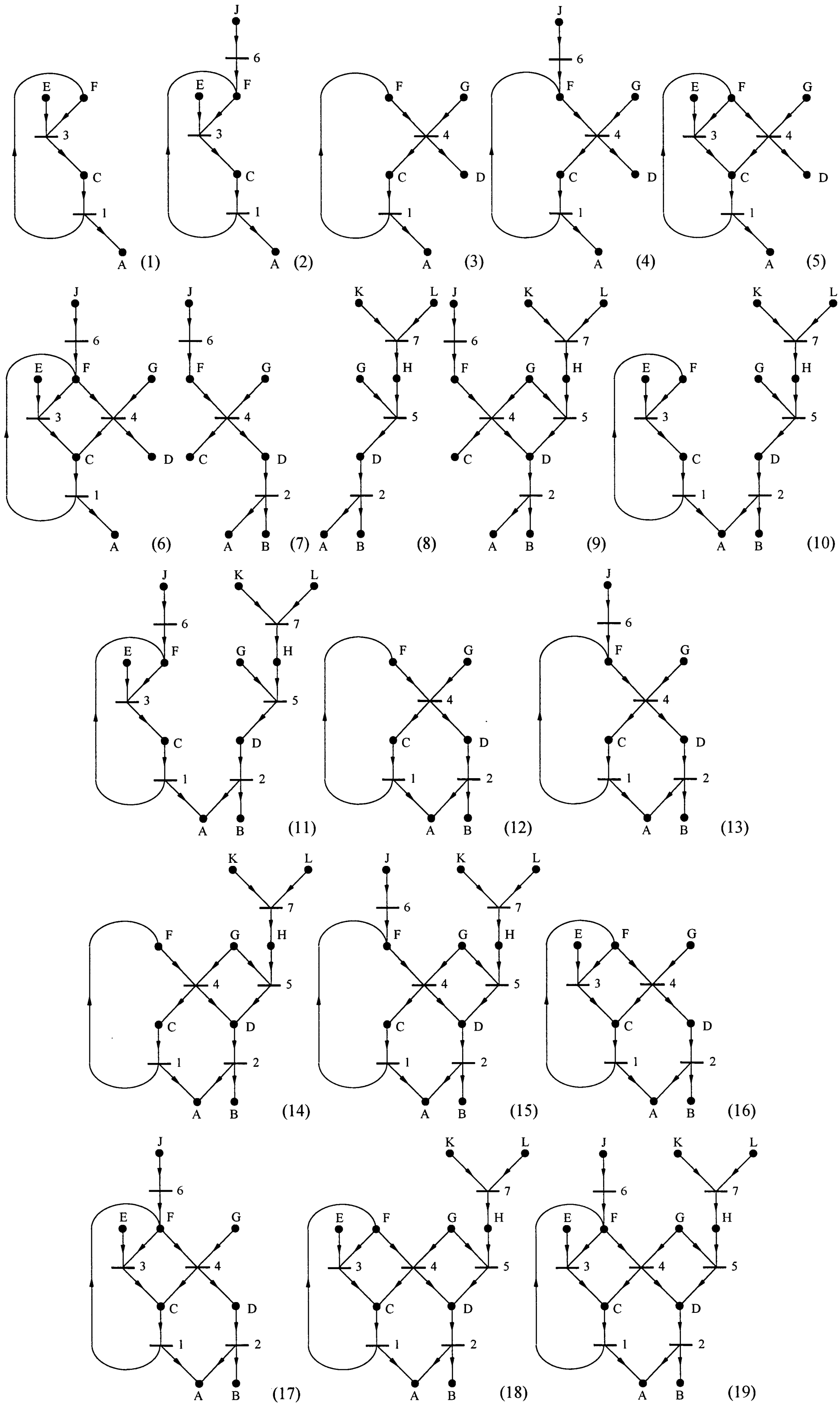

Fig. 3. Combinatorially feasible structures of the example (Friedler et al., 1995a).

5. Generation of the DNF by a combinatorial algorithm

The DNF can be generated from the CNF (Raman & Grossmann, 1991a); however, the algorithm for it is NP-hard, i.e. complex. Another way of generating the DNF is through the combinatorial algorithms of process synthesis, which directly obtain DNF without CNF, as described below.

A truth assignment of the complete DNF satisfies exactly one of its clauses; consequently, each clause of the DNF of the synthesis problem corresponds to one combinatorially feasible network. The complete DNF of the synthesis problem, therefore, can be readily obtained if the set of feasible networks is available. Algorithm SSG based on the combinatorial theory is capable of generating such feasible networks (Friedler et al., 1992b); consequently, the results can be obtained either as a set of P-graphs or as the complete DNF depending on the type of output desired.

6. Example

Let a process network synthesis problem be given by triplet (P, R, O), where P = {A} is the set of products; R = {E, G, J, K, L}, the set of raw materials; and O = {Y1, Y2, Y3, Y4, Y5, Y6, Y7}, the set of operating units. The operating units in set O are: Y1 = ({C},{A, F}); Y2 = ({D},{A, B}); Y3 = ({E, F},{C}); Y4 = ({F, G},{C, D}); Y5 = ({G, H},{D}); Y6 = ({J},{F}); and Y7 = ({K, L},{H}). Note that the set of materials is M = {A, B, C, D, E, F, G, H, J, K, L}. This example is the same as Example 9 of Friedler, Varga and Fan (1995a); the corresponding P-graph is given in Fig. 2. Specifically, the CNF is generated simply as (O1) ∧ (O2) ∧ (O3) ∧ (O4), where

(O1) $Y1 \vee Y2$

(O2) true

(O3)

$$(\neg Y1 \vee Y3 \vee Y4) \wedge (\neg Y2 \vee Y4 \vee Y5)$$
$$\wedge (\neg Y3 \vee Y1 \vee Y6) \wedge (\neg Y4 \vee Y1 \vee Y6) \wedge (\neg Y5 \vee Y7)$$

(O4)

$$(\neg Y3 \vee Y1) \wedge (\neg Y4 \vee Y1 \vee Y2) \wedge (\neg Y5 \vee Y2)$$
$$\wedge (\neg Y6 \vee Y3 \vee Y4) \wedge (\neg Y7 \vee Y5)$$

The DNF of the problem is determined by algorithm SSG (Friedler et al., 1992b); each structure generated by this algorithm is represented as one clause in the DNF given below and as a P-graph given in Fig. 3 (Friedler et al., 1995a).

$$(Y1 \wedge \neg Y2 \wedge Y3 \wedge \neg Y4 \wedge \neg Y5 \wedge \neg Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge \neg Y2 \wedge Y3 \wedge \neg Y4 \wedge \neg Y5 \wedge Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge \neg Y2 \wedge \neg Y3 \wedge Y4 \wedge \neg Y5 \wedge \neg Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge \neg Y2 \wedge \neg Y3 \wedge Y4 \wedge \neg Y5 \wedge Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge \neg Y2 \wedge Y3 \wedge Y4 \wedge \neg Y5 \wedge \neg Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge \neg Y2 \wedge Y3 \wedge Y4 \wedge \neg Y5 \wedge Y6 \wedge \neg Y7) \vee$$
$$(\neg Y1 \wedge Y2 \wedge \neg Y3 \wedge Y4 \wedge \neg Y5 \wedge Y6 \wedge \neg Y7) \vee$$
$$(\neg Y1 \wedge Y2 \wedge \neg Y3 \wedge \neg Y4 \wedge Y5 \wedge \neg Y6 \wedge Y7) \vee$$
$$(\neg Y1 \wedge Y2 \wedge \neg Y3 \wedge Y4 \wedge Y5 \wedge Y6 \wedge Y7) \vee$$
$$(Y1 \wedge Y2 \wedge Y3 \wedge \neg Y4 \wedge Y5 \wedge \neg Y6 \wedge Y7) \vee$$
$$(Y1 \wedge Y2 \wedge Y3 \wedge \neg Y4 \wedge Y5 \wedge Y6 \wedge Y7) \vee$$
$$(Y1 \wedge Y2 \wedge \neg Y3 \wedge Y4 \wedge \neg Y5 \wedge \neg Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge Y2 \wedge \neg Y3 \wedge Y4 \wedge \neg Y5 \wedge Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge Y2 \wedge \neg Y3 \wedge Y4 \wedge Y5 \wedge \neg Y6 \wedge Y7) \vee$$
$$(Y1 \wedge Y2 \wedge \neg Y3 \wedge Y4 \wedge Y5 \wedge Y6 \wedge Y7) \vee$$
$$(Y1 \wedge Y2 \wedge Y3 \wedge Y4 \wedge \neg Y5 \wedge \neg Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge Y2 \wedge Y3 \wedge Y4 \wedge \neg Y5 \wedge Y6 \wedge \neg Y7) \vee$$
$$(Y1 \wedge Y2 \wedge Y3 \wedge Y4 \wedge Y5 \wedge \neg Y6 \wedge Y7) \vee$$
$$(Y1 \wedge Y2 \wedge Y3 \wedge Y4 \wedge Y5 \wedge Y6 \wedge Y7)$$

There exists a one-to-one correspondence between the clauses of this DNF and the P-graphs of Fig. 3. For instance, the first clause of the DNF, $(Y1 \wedge \neg Y2 \wedge Y3 \wedge \neg Y4 \wedge \neg Y5 \wedge \neg Y6 \wedge \neg Y7)$, represents the structure including operating units 1 and 3 and excluding operating units 2 and 4 through 7, which is the structure given in Fig. 3 (1).

The solutions of the CNF and DNF are identical; the corresponding structures are combinatorially feasible because they satisfy axioms (S1) through (S5). Note that cutting the recirculation by eliminating the stream from operating unit Y1 to material F, as proposed in Raman and Grossmann (1993), excludes eight structures, e.g. the first one, represented by the above clauses in the DNF; nevertheless, any of these excluded structures can be potentially optimal.

7. Concluding remarks

The generation of the conjunctive and disjunctive normal forms to solve process synthesis problems has been mathematically established on the basis of the combinatorial theory of total flowsheet synthesis. It can be stated that in general, no advantage can be gained by switching to logical formulation once a solution of a synthesis problem is initiated by the combinatorial approach.

References

Friedler, F., Fan, L. T., & Imreh, B. (1998). Process network synthesis: problem definition. *Networks*, *28*, 119.

Friedler, F., Tarján, K., Huang, Y. W., & Fan, L. T. (1992a). Graph-theoretic approach to process synthesis: axioms and theorems. *Chemical Engineering Science*, *47*, 1973.

Friedler, F., Tarján, K., Huang, Y. W., & Fan, L. T. (1992b). Combinatorial algorithms for process synthesis. *Computers and Chemical Engineering*, *16*, S313.

Friedler, F., Tarján, K., Huang, Y. W., & Fan, L. T. (1993). Graph-theoretic approach to process synthesis: polynomial algorithm for maximal structure generation. *Computers and Chemical Engineering*, *17*, 929.

Friedler, F., Varga, J. B., & Fan, L. T. (1995a). Decision-mapping: a tool for consistent and complete decisions in process synthesis. *Chemical Engineering Science*, *50*, 1755.

Friedler, F., Varga, J. B., & Fan, L. T. (1995b). Algorithmic approach to the integration of total flowsheet synthesis and waste minimization. *AIChE Symposium Series*, *90*, 86.

Friedler, F., Varga, J. B., Fehér, E., & Fan, L. T. (1996). Combinatorially accelerated branch-and-bound method for solving the MILP model of process network synthesis. In C. A. Floudas, & P. M. Pardalos, *Nonconvex optimization and its applications* (pp. 609–626). Norwell, MA: Kluwer.

Hooker, J. N., Yan, H., Grossmann, I. E., & Raman, R. (1994). Logic cuts for process networks with fixed charges. *Computers and Operations Research*, *21*, 265.

Raman, R., & Grossmann, I. E. (1991a). Logical inference in branch and bound techniques for process synthesis, *Proceedings of PSE 1991*. Montebello, Canada.

Raman, R., & Grossmann, I. E. (1991b). Relation between MILP modeling and logical inference for chemical process synthesis. *Computers and Chemical Engineering*, *15*, 72.

Raman, R., & Grossmann, I. E. (1993). Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis. *Computers and Chemical Engineering*, *17*, 909.

Raman, R., & Grossmann, I. E. (1994). Modeling and computational techniques for logic based integer programming. *Computers and Chemical Engineerng*, *18*, 563.