



Report

Algebras and combinatorics

Author(s):

Emgeler, Erwin

Publication Date:

1979

Permanent Link:

<https://doi.org/10.3929/ethz-a-005363190> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

ETH

**Eidgenössische Technische Hochschule
Zürich**

Institut für Informatik

Erwin Engeler

ALGEBRAS AND COMBINATORS

June 1979

32

E T H

EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE
ZÜRICH

INSTITUT FÜR INFORMATIK

ERWIN ENGELER

ALGEBRAS AND COMBINATORS

Address of the author:
Institut für Informatik
ETH-Zentrum
CH-8092 Zürich

© 1979 Institut für Informatik der ETH Zürich

ALGEBRAS AND COMBINATORS

Erwin Engeler

§1 A general representation theorem

We shall prove our representation theorem for the case of algebras with one binary operation only; the generalization to arbitrary algebraic structures is sketched at the end of this section.

Let A be non-empty. Let B be a set of "formulas" defined as the smallest set $\ni A$ which contains the formula $(\alpha \rightarrow b)$ whenever α is a non-empty finite subset of B and $b \in B$.

Definition. For $M, N \subseteq B$ let $M \cdot N = \{b : \exists \alpha \subseteq N. \alpha \rightarrow b \in M\}$. A 2-algebra over A is a collection of subsets of B closed under \cdot .

THEOREM. Every algebra $\underline{A} = \langle A, \cdot \rangle$ with one binary operation is isomorphic to a 2-algebra over A .

Proof. Construct the set of formulas B as above, starting with the carrier set A of the given algebraic structure \underline{A} . Then define a map f of A into the power-set of B recursively by

$$f(a) = \bigcup_i f_i(a),$$

where

$$f_0(a) = \{a\},$$

$$f_{i+1}(a) = f_i(a) \cup \{\alpha \rightarrow y : \exists b \in A. b \in \alpha \subseteq f_i(b) \\ \wedge y \in f_i(a \cdot b) \wedge \alpha \text{ finite}\}.$$

Note that $f(a) \cap A = \{a\}$. Hence

$$(1) \quad \text{if } f(a) = f(b) \quad \text{then } a = b,$$

because then $\{a\} = f(a) \cap A = f(b) \cap A = \{b\}$. Thus, it remains to prove

$$(2) \quad f(a \cdot b) = f(a) \cdot f(b).$$

For this we compute as follows:

$$\begin{aligned} f(a) \cdot f(b) &= \{y : \exists \alpha \subseteq f(b). \alpha \rightarrow y \in f(a)\} \\ &= \{y : \exists \alpha \subseteq f(b) \exists \text{minimal } i. \alpha \rightarrow y \in f_{i+1}(a)\} \\ &= \{y : \exists \alpha \subseteq f(b) \exists i \exists u, v \in A. au = v \\ &\quad \wedge u \in \alpha \subseteq f_i(u) \wedge y \in f_i(v)\}. \end{aligned}$$

Because $u \in \alpha \subseteq f(b) \cap f_i(u)$ and $u \in A$, we have $u = b$ and $v = a \cdot b$, using $f(a) \cap A = \{a\}$ again. Hence

$$\begin{aligned} f(a) \cdot f(b) &= \{y : \exists \alpha \subseteq f(b) \exists i. b \in \alpha \subseteq f_i(b) \wedge y \in f_i(a \cdot b)\} \\ &= \{y : \exists i. y \in f_i(a \cdot b)\} = \bigcup_i f_i(a \cdot b) = f(a \cdot b). \end{aligned}$$

Thus (2) holds, and f is an isomorphic embedding as claimed. \square

If the structure to be represented has other operations, e.g. a ternary operation o , we augment the definition of B

accordingly: $A \subseteq B$ and if α, β are finite subsets of B and $c \in B$ then $(\alpha, \beta \overset{\circ}{+} c) \in B$ as well as $(\alpha \overset{\circ}{+} c) \in B$.

Definition. For $M, N, L \subseteq B$ let $o(M, N, L) = \{c : \exists \alpha \subseteq N \exists \beta \subseteq L. (\alpha, \beta \overset{\circ}{+} c) \in B\}$. A 2-3-algebra over A is a class of subsets of B closed under \cdot and o .

THEOREM. Every algebraic structure $\underline{A} = \langle \underline{A}, \cdot, o \rangle$ is isomorphic to a 2-3-algebra.

Proof. Same as above with the map f redefined by setting

$$\begin{aligned}
 f_{i+1}(a) = & f_i(a) \cup \{ \alpha \overset{\circ}{+} \gamma : \exists b \in A. b \in \alpha \subseteq f_i(b) \\
 & \wedge \gamma \in f_i(a \cdot b) \wedge \alpha \text{ finite} \} \\
 \cup & \{ \alpha, \beta \overset{\circ}{+} \gamma : \exists b, c \in A. b \in \alpha \subseteq f_i(b) \\
 & \wedge c \in \gamma \subseteq f_i(c) \wedge \gamma \in f_i(o(a, b, c)) \\
 & \wedge \alpha, \beta \text{ finite} \}. \quad \square
 \end{aligned}$$

It is easy to extend the representation theorem to relational structures.

§2 Combinatory algebras

A combinatory algebra is an algebraic structure $\underline{A} = \langle A, \cdot \rangle$ which is "combinatorially complete", i.e.

For every expression $\phi(x_1, \dots, x_n)$ built up from constants (denoting elements of A) and variables x_1, \dots, x_n by means of the operation symbol " \cdot " there exists an element f in A such that for all $a_1, \dots, a_n \in A$

$$(\dots((f \cdot a_1) \cdot a_2) \dots \cdot a_n) = \phi(a_1, \dots, a_n).$$

The existence of non-trivial combinatory algebras follows either from a Church-Rosser theorem as an algebra of equivalence-classes of terms or by constructions such as Scott's D_∞ or Plotkin-Scott's P_ω . Our general representation theorem suggests that combinatory algebras be constructed as 2-algebras. Indeed, all combinatory algebras are isomorphic to 2-algebras.

Let $A \neq \emptyset$ and B be constructed as in the first part of section 1. Then the 2-algebra of all subsets of B already forms a combinatory algebra. Following Curry's remark that combinatorial completeness follows from two of its instances, it suffices to isolate two different subsets K and S of B such that for all $M, N, L \subseteq B$ we have

- (1) $KMN = M,$
- (2) $SMNL = ML(NL).$

The following definitions accomplish this.

Definition.

$$K := \{\sigma \rightarrow (\rho \rightarrow s) : \sigma, \rho \subseteq B, s \in \sigma\}$$

$$S := \{ \{\tau \rightarrow (\{r_1, \dots, r_n\} \rightarrow s)\} \rightarrow (\{\sigma_1 \rightarrow r_1, \dots, \sigma_n \rightarrow r_n\} \rightarrow (\sigma \rightarrow s)) : \\ n \geq 1, r_1, \dots, r_n \in B, \tau \cup \bigcup \sigma_i = \sigma \subseteq B \}.$$

THEOREM. The 2-algebra of subsets of B is a combinatory algebra.

Proof. Clearly $K \neq S$, since $(\{a\} \rightarrow (\{a\} \rightarrow a)) \in K$, $(\{a\} \rightarrow (\{a\} \rightarrow a)) \notin S$. The combinatorial laws follow by straightforward verification:

$$\begin{aligned} KMN &= \{s : \exists \alpha \subseteq N \exists \beta \subseteq M. \beta \rightarrow (\alpha \rightarrow s) \in K\} \\ &= \{s : \exists \beta \subseteq M. s \in \beta\} = M. \end{aligned}$$

$$\begin{aligned}
 \text{ML(NL)} &= \{s : \exists \rho \subseteq \text{NL}. \rho \rightarrow s \in \text{ML}\} \\
 &= \{s : \exists n \geq 1 \exists r_1, \dots, r_n \in B \exists \sigma_1, \dots, \sigma_n \subseteq L. \\
 &\quad \{r_1, \dots, r_n\} \rightarrow s \in \text{ML} \wedge \sigma_1 \rightarrow r_1, \dots, \sigma_n \rightarrow r_n \in N\} \\
 &= \{s : \exists n \geq 1 \exists r_1, \dots, r_n \in B \exists \sigma_1, \dots, \sigma_n \subseteq L \exists \tau \subseteq L. \\
 &\quad \tau \rightarrow (\{r_1, \dots, r_n\} \rightarrow s) \in M \wedge \sigma_1 \rightarrow r_1, \dots, \sigma_n \rightarrow r_n \in N\} \\
 \\
 \text{SMNL} &= \{s : \exists \sigma \subseteq L \exists \eta \in N \exists \varepsilon \in M. (\varepsilon \rightarrow (\eta \rightarrow (\sigma \rightarrow s))) \in S\} \\
 &= \{s : \exists \sigma \subseteq L \exists n \geq 1 \exists r_1, \dots, r_n \in B \exists \tau, \sigma_1, \dots, \sigma_n \subseteq B. \\
 &\quad \tau \rightarrow (\{r_1, \dots, r_n\} \rightarrow s) \in M \wedge \sigma_1 \rightarrow r_1, \dots, \sigma_n \rightarrow r_n \in N \\
 &\quad \wedge \sigma = \tau \cup \bigcup \sigma_i\} \\
 &= \{s : \exists n \geq 1 \exists r_1, \dots, r_n \in B \exists \tau, \sigma_1, \dots, \sigma_n \subseteq L. \\
 &\quad \tau \rightarrow (\{r_1, \dots, r_n\} \rightarrow s) \in M \wedge \sigma_1 \rightarrow r_1, \dots, \sigma_n \rightarrow r_n \in N\} \\
 &= \text{ML(NL)}. \quad \square
 \end{aligned}$$

§3 Lambda calculi

Lambda calculi are based on binary algebraic structures $\underline{A} = \langle A, \cdot \rangle$; they enforce combinatorial completeness by providing a name

$$\lambda X.M,$$

for each expression M , to denote the element $f \in A$ for which

$$f \cdot N = M_X^N$$

where M_X^N stands for the expression obtained from M by replacing the variable X everywhere by N .

The language of a lambda calculus consists of constant symbols and variables X, Y, Z, \dots and is provided with the mechanisms of application (if M and N are λ -terms then so is MN) and abstraction (if M is a λ -term and X is a variable, then $\lambda X.M$ is a λ -term).

We now present an interpretation of λ -terms in the 2-algebra of all subsets of B which will make use of the latter a model of the $\lambda\beta$ -calculus. To each variable X, Y, \dots of the lambda calculus we associate an infinite set of new symbols $\{x_1, x_2, \dots\}$, resp. $\{y_1, y_2, \dots\}, \dots$. Let C be the smallest set $\supseteq B$ such that both $(\alpha + b)$ and $(\alpha ; b)$ are in C whenever α is a finite subset of C and $b \in C$. Let $C(X), C(X, Y), \dots$ be defined the same way, taking $B \cup \{x_1, x_2, \dots\}$, resp. $B \cup \{x_1, x_2, \dots\} \cup \{y_1, y_2, \dots\}$ to start. Elements of $C, C(X), C(X, Y), \dots$ can be reduced by replacing parts of expressions of the form $\alpha ; (\beta + c)$ by c if $\beta \subseteq \alpha$.

LEMMA 1. For every w in C (resp. $C(X), C(X, Y), \dots$) there is a unique irreducible element w^* of C (resp. $C(X), C(X, Y)$) which can be obtained from w by repeated applications of the reduction rule.

Proof. If $w = (\{a_1, \dots, a_n\} + b)$ then the unique w^* is clearly $(\{a_1^*, \dots, a_n^*\} + b^*)$. If $w = (\{a_1, \dots, a_n\}; b)$ and b can be reduced to $(\{c_1, \dots, c_m\} + d)$, where each c_i is obtained from an a_j by (repeated) reductions, then w^* equals $\{c_1^*, \dots, c_m^*\} + d^*$ by the previous case. Thus w reduces uniquely to d^* . If b cannot be so reduced, then the unique w^* is $(\{a_1^*, \dots, a_n^*\}; b^*)$. \square

To indicate the occurrence of a symbol x_i or a set of symbols $\xi \subseteq \{x_1, x_2, \dots\}$ in an element of $C(X)$, we write it $a(x_i)$, respectively $a(\xi)$. If b , respectively β is substituted for x_i , respectively ξ , we write the result as $a(b)$, respectively $a(\beta)$.

Let now $[.]$ be a map, which associates a subset of B to every variable of the lambda calculus. We also consider modified maps $[.]_X, [.]_{XY}, \dots$ defined as follows:
 $[X]_X = \{x_1, x_2, \dots\}, [Y]_X = [Y]$ for all variables $Y \neq X$,
 $[X]_{XY} = \{x_1, x_2, \dots\}, [Y]_{XY} = \{y_1, y_2, \dots\}, [Z]_{XY} = [Z]$ for all $Z \neq X, Y$. The maps $[.], [.]_X, \dots$ are extended to all lambda-terms by:

Definition.

$$\begin{aligned} [MN] &= \{(\alpha; b)^* : \alpha \subseteq [N], b \in [M]\}, \\ [MN]_X &= \{(\alpha; b)^* : \alpha \subseteq [N]_X, b \in [M]_X\}, \\ [\lambda X.M] &= \{(\tau \rightarrow a(\tau))^* : \tau \subseteq B, a(\xi) \in [M]_X\}, \\ [\lambda X.M]_X &= [\lambda X.M], \\ [\lambda Y.M]_X &= \{(\tau \rightarrow a(\xi, \tau))^* : \tau \subseteq B, a(\xi, \eta) \in [M]_{XY}\}, Y \neq X. \end{aligned}$$

LEMMA 2. Let L be a lambda-term in which the variable X does not occur free, and let M_X^L result from M by replacing X everywhere in M by L . Then

$$[M_X^L] = \{a(\lambda)^* : \lambda \subseteq [L], a(\xi) \in [M]_X\}.$$

If Y is also not free in L then

$$[M_X^L]_Y = \{a(\lambda, \eta)^* : \lambda \subseteq [L], a(\xi, \eta) \in [M]_{XY}\}.$$

Proof. It suffices to prove the first statement, because $[L]_Y = [L]$. The first statement is shown by induction on the structure of M .

$$\begin{aligned} \text{(a)} \quad [X_X^L] &= [L] = \{a(\lambda)^* : \lambda \subseteq [L], a(\xi) \in [X]_X = \{x_1, x_2, \dots\}\}, \\ [Y_X^L] &= [Y] = \{a(\lambda)^* : \lambda \subseteq [L], a(\xi) \in [Y]_X = [Y]\}. \end{aligned}$$

$$\begin{aligned}
 \text{(b)} \quad |(\text{MN})_X^L| &= |M_X^L N_X^L| = \{(\alpha; b)^* : \alpha \in [N_X^L], b \in [M_X^L]\} \\
 &= \{(\alpha(\lambda_1); b(\lambda_2))^* : \lambda_1, \lambda_2 \in [L], \alpha(\xi_1) \in [N]_X, \\
 &\quad b(\xi_2) \in [M]_X\} \\
 &= \{c(\lambda)^* : \lambda \in [L], c(\xi) \in [MN]_X\}.
 \end{aligned}$$

(c) Assume, without loss of generality, that Y is not free in L . Then

$$\begin{aligned}
 |(\lambda Y.M)_X^L| &= |\lambda Y.M_X^L| = \{(\tau \rightarrow a(\tau))^* : \tau \in B, a(\eta) \in [M_X^L]_Y\} \\
 &= \{\tau \rightarrow b(\lambda, \tau)^* : \tau \in B, \lambda \in [L], b(\xi, \eta) \in [M]_{XY}\} \\
 &= \{c(\lambda)^* : \lambda \in [L], c(\xi) \in [\lambda Y.M]_X\}.
 \end{aligned}$$

Definition. $|M| = [M] \cap B$.

THEOREM. If $M = N$ is provable in the lambda calculus, then for all maps $[\cdot]$ we have $|M| = |N|$.

Proof. For the only non-trivial axiom we have

$$\begin{aligned}
 |(\lambda X.M)N| &= [(\lambda X.M)N] \cap B = \{(\alpha; b)^* : \alpha \in [N], b \in [(\lambda X.M)]\} \cap B \\
 &= \{(\alpha; \tau \rightarrow b(\tau))^* : \alpha \in [N], \tau \in B, b(\xi) \in [M]_X\} \cap B \\
 &= \{b(\tau)^* : \tau \in [N], b(\xi) \in [M]_X\} \cap B, \text{ because for} \\
 &\quad \text{reduction must have } \tau \in \alpha \in [N], \\
 &= [M_X^N] \cap B = |M_X^N| \text{ by Lemma 2 and definition of } |\cdot|.
 \end{aligned}$$

The verification of the rules of proof are all trivial, except

$$M = N \text{ implies } \lambda X.M = \lambda X.N .$$

Observe

$$|\lambda X.M| = \{\tau \rightarrow a(\tau)^* : a(\xi) \in [M]_X\} \cap B = \bigcup_{\tau \in B} \{\tau \rightarrow a_\tau : a_\tau \in |M|_{(\tau)}\}$$

where $[X]_{(\tau)} = \tau$, $[Y]_{(\tau)} = [Y]$ for all $Y \neq X$. Because $|M|_{(\tau)} = |N|_{(\tau)}$ for all τ by assumption, we have therefore $|\lambda X.M| = |\lambda X.N|$.

THEOREM. $|\lambda X.X| \neq |\lambda X.XX|$.

Proof. $|\lambda X.X| = \{\tau \rightarrow a(\tau)^* : \tau \in B, a(\xi) \in [X]_X = \{x_1, x_2, \dots\}\}$

$$\cap B = \{\{a\} \rightarrow a : a \in B\}.$$

$$|\lambda X.XX| = \{\tau \rightarrow a(\tau)^* : \tau \in B, a(\xi) \in [XX]_X = \{(\alpha; b)^* : \alpha \in [X]_X, b \in [X]_X\} \cap B\}$$

$$= \{\tau \rightarrow a(\tau)^* : \tau \in B, a(\xi) = (\{x_{i_1}, \dots, x_{i_n}\}; x_{i_{n+1}})\}$$

$$\text{for some } n \geq 1\} \cap B$$

$$= \{\{a_1, \dots, a_{n+1}\} \rightarrow (\{a_1, \dots, a_n\}; a_{n+1})^* : a_i \in B\} \cap B$$

$$= \{\{a_1, \dots, a_n, (\{a_{i_1}, \dots, a_{i_m}\} \rightarrow b)\} \rightarrow b : a_i, b \in B\}$$

$$\neq |\lambda X.X| . \quad \square$$

We thank H. Barendregt for pointing out some oversights in an earlier version of this model.

§4 Continuity

In previous constructions of models of combinatory algebras, continuity played an important rôle. - For an appropriate topology we can very easily prove here that the continuous maps from the powerset of B to itself are exactly the ones which are obtained by application.

Definition. Let $A \neq \emptyset$ and B be as before. The sets $\{M : \alpha \subseteq M \subseteq B\}$ for finite α form the base of our topology.

Observe that in this topology a map f from the powerset of B into itself is continuous iff

$$f(N) = \bigcup \{f(\alpha) : \alpha \subseteq N, \alpha \text{ finite}\}.$$

THEOREM. f is continuous iff $\exists M \subseteq B \forall N \subseteq B. f(N) = M \cdot N.$

Proof. (a) Suppose f continuous. Define

$$M = \{\alpha \rightarrow x : x \in f(\alpha), \alpha \subseteq B, \alpha \text{ finite}\}.$$

$$\text{Then } M \cdot N = \{x : \exists \alpha \subseteq N. \alpha \rightarrow x \in M\}$$

$$= \{x : \exists \alpha \subseteq N. x \in f(\alpha), \alpha \text{ finite}\}$$

$$= \bigcup \{f(\alpha) : \alpha \subseteq N, \alpha \text{ finite}\}$$

$$= f(N) \text{ by continuity.}$$

(b) Suppose f is given by $f(N) = M \cdot N.$ We have to show continuity, i.e. $f(N) = \bigcup \{f(\alpha) : \alpha \subseteq N, \alpha \text{ finite}\}.$ The latter set is, by definition equal to $\bigcup \{M \cdot \alpha : \alpha \subseteq N, \alpha \text{ finite}\}.$

Thus $x \in \bigcup \{M \cdot \alpha : \alpha \subseteq N, \alpha \text{ finite}\}$

iff $\exists \alpha \exists \beta \subseteq \alpha \subseteq N. \beta \rightarrow x \in M \wedge \alpha \text{ finite},$

iff $\exists \beta \subseteq N. \beta \rightarrow x \in M,$

iff $x \in M \cdot N,$

iff $x \in f(N).$



§5 Applications

The simplicity of the combinatory algebras above facilitates their use as models of computation. This will be elaborated in a future paper; one example should suffice here.

Let Γ be a first-order theory with predicate symbol R (binary) and function symbol f (unary). The model of computation associated to Γ is a 2-algebra $\lambda\Gamma$ over the first-order language L of Γ , containing S and K (which makes it a combinatory algebra) and the following constants:

For each formula ϕ in L let

$$[\phi] := \{\psi \in L : \Gamma, \phi \vdash \psi\}.$$

For all variables x, y in L let

$$[y := f(x)] := \{\Delta \rightarrow \psi(x, y) : \Gamma, \Delta, y' = f(x), x' = x \vdash \psi(x', y')\}.$$

This model describes programmed computations on data and with operations that are incompletely known (only to the extent that Γ provides this knowledge) or whose description is infinite. It has been implemented at the ETH by Th. Fehlmann for the case where Γ is Peano arithmetic, and by P. Horak for exact computations with reals and power series. Descriptions of these implementations are forthcoming.

Berichte des Instituts für Informatik

- *Nr. 1 N.Wirth: The Programming Language PASCAL
- *Nr. 2 N.Wirth: Program development by step-wise refinement
- Nr. 3 P.Lüchli: Reduktion elektrischer Netzwerke und Gauss'sche Elimination
- Nr. 4 W.Gander, A.Mazzario: Numerische Prozeduren I
- *Nr. 5 N.Wirth: The Programming Language PASCAL (Revised Report)
- *Nr. 6 C.A.R.Hoare, N.Wirth: An Axiomatic Definition of the Language PASCAL
- Nr. 7 W.Gander, A.Mazzario: Numerische Prozeduren II
- Nr. 8 E.Engeler, E.Wiedmer, E.Zachos: Ein Einblick in die Theorie der Berechnungen
- *Nr. 9 H.P.Frei: Computer Aided Instruction: The Author Language and the System THALES
- *Nr.10 K.V.Nori, U.Ammann, K.Jensen, H.H.Nägeli, Ch.Jacobi: The PASCAL 'P' Compiler: Implementation Notes (Revised Edition)
- Nr.11 G.I.Ugron, F.R.Lüthi: Das Informations-System ELSBETH
- *Nr.12 N.Wirth: PASCAL-S: A subset and its Implementation
- *Nr.13 U.Ammann: Code Generation in a PASCAL Compiler
- Nr.14 K.Lieberherr: Toward Feasible Solutions of NP-Complete Problems
- *Nr.15 E.Engeler: Structural Relations between Programs and Problem
- Nr.16 W.Bucher: A contribution to solving large linear problems
- Nr.17 N.Wirth: Programming languages: what to demand and how to access them and Professor Cleverbyte's visit to heaven
- *Nr.18 N.Wirth: MODULA: A language for modular multiprogramming
- *Nr.19 N.Wirth: The use of MODULA and Design and Implementation of MODULA
- Nr.20 E.Wiedmer: Exaktes Rechnen mit reellen Zahlen

- *Nr.21 J.Nievergelt, XS-0, a Self-explanatory School Computer
H.P.Frei,
et al.:
- Nr.22 P.Läuchli: Ein Problem der ganzzahligen Approximation
- Nr.23 K.Bucher: Automatisches Zeichnen von Diagrammen
- Nr.24 E.Engeler: Generalized Galois Theory and its Application to
Complexity
- Nr.25 U.Ammann: Error Recovery in Recursive Descent Parsers and
Run-time Storage Organization
- Nr.26 E.Zachos: Kombinatorische Logik und S-Terme
- Nr.27 N.Wirth: MODULA-2
- Nr.28 J.Nievergelt, Sites, Modes and Trails: Telling the User of an
J.Weydert: Interactive System where he is, what he can do,
and how to get to Places.
- Nr.29 A.C.Shaw: On the Specification of Graphic Command Languages
and their Processors
- Nr.30 B.Thurnherr, Global Data Base Aspects, Consequences for the
C.A.Zehnder: Relational Model and a Conceptual Schema Language
- Nr.31 A.C.Shaw: Software Specification Languages based on regular
Expressions
- Nr.32 E. Engeler: Algebras and Combinators

* out of print