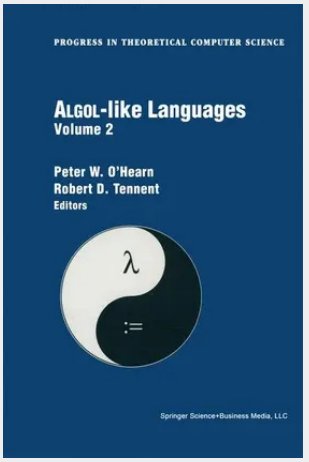
O'Hearn / Tennent

Algol-like Languages

To construct a compiler for a modern higher-level programming languagel one needs to structure the translation to a machine-like intermediate language in a way that reflects the semantics of the language. little is said about such struc turing in compiler texts that are intended to cover a wide variety of program ming languages. More is said in the literature on semantics-directed compiler construction [1] but here too the viewpoint is very general (though limited to 1 languages with a finite number of syntactic types). On the other handl there is a considerable body of work using the continuation-passing transformation to structure compilers for the specific case of call-by-value languages such as SCHEME and ML [21 3]. In this paperl we will describe a method of structuring the translation of ALGOL-like languages that is based on the functor-category semantics devel oped by Reynolds [4] and Oles [51 6]. An alternative approach using category theory to structure compilers is the early work of F. L. Morris [7]1 which anticipates our treatment of boolean expressionsl but does not deal with procedures. 2 Types and Syntax An ALGOL-like language is a typed lambda calculus with an unusual repertoire of primitive types. Throughout most of this paper we assume that the primi tive types are comm(and) int(eger)exp(ression) int(eger)acc(eptor) int(eger)var(iable) I and that the set 8 of types is the least set containing these primitive types and closed under the binary operation -.

To construct a compiler for a modern higher-level programming languagel one needs to structure the translation to a machine-like intermediate language in a way that reflects the semantics of the language. little is said about such struc turing in compiler texts that are intended to cover a wide variety of program ming languages. More is said in the literature on semantics-directed compiler construction [1] but here too the viewpoint is very general (though limited to 1 languages with a finite number of syntactic types). On the other handl there is a considerable body of work using the continuation-passing transformation to structure compilers for the specific case of call-by-value languages such as SCHEME and ML [21 3]. In this paperl we will describe a method of structuring the translation of ALGOL-like languages that is based on the functor-category semantics devel oped by Reynolds [4] and Oles [51 6]. An alternative approach using category theory to structure compilers is the early work of F. L. Morris [7]1 which anticipates our treatment of boolean expressionsl but does not deal with procedures. 2 Types and Syntax An ALGOL-like language is a typed lambda calculus with an unusual repertoire of primitive types. Throughout most of this paper we assume that the primi tive types are comm(and) int(eger)exp(ression) int(eger)acc(eptor) int(eger)var(iable) I and that the set 8 of types is the least set containing these primitive types and closed under the binary operation -.



106,99 €
99,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9780817639372
Medium: Buch
ISBN: 978-0-8176-3937-2
Verlag: Birkhäuser
Erscheinungstermin: 01.12.1996
Sprache(n): Englisch
Auflage: 1997
Serie: Algol-like Languages
Produktform: Gebunden
Gewicht: 1520 g
Seiten: 349
Format (B x H): 160 x 241 mm