KWIC (Key Word in Context)

5G [520] VLSI and System Architecture—the Development of System
5th Generation Computer [87] aimed PROLOG—as a Preliminary Kernel Language for
8 Queens Problem [216] Simulating Coroutining for the Abstract Model to Efficient Compilation of Patterns [166] From
Abstract PROLOG Instruction Set [672] An
Abstract PROLOG Machine—a Specification [383]
Abstraction in Logic Programming [198] Modularization and
Abstraction in Logic Programming [199] Modularization and
Act 1 [304] age Based Upon Unification Which Unifies (Much Of) LISP, PROLOG, and
Act 1 [305] age Based Upon Unification Which Unifies (Much Of) LISP, PROLOG, and
Act 1 [307] age Based Upon Unification Which Unifies (Much Of) LISP, PROLOG, and
Algorithmic Program Debugging [555]
Algorithms and Proving Properties by Computing Terms
[454]Computing
Algorithms from a Specification in Logic [700] DDeriving
Different Unification
Algorithms in Concurrent PROLOG: the MAXFLOW
Experience [266] Implementing Parallel
Algorithms in Concurrent PROLOG: the MAXFLOW
Experience [267] Implementing Parallel
Algorithms, Architecture, and Technology [10]
All Solutions [480]
Alternation and the Computational Complexity of Logic
Programs [548]
Alternative Method of Adapting Resolution for Logic
Programming [18] niclipse—an
Alternative to PROLOG [512] LOGLISP: an
Alternative to Structure Sharing in the Implementation
of a PROLOG Interpreter [390]
Alternative to Structure-sharing in the Implementation
of a PROLOG Interpreter [394]
Alternative to Structure-sharing in the Implementation
of a PROLOG Interpreter [395]
Amalgamating Language and Metalanguage in Logic
Programming [51]
Amalgamating Language and Metalanguage in Logic
Programming [52]
Analysis [698] Semantic Code
Analysis—a Survey of the Formalism and a Comparison
with Augmented Transition [463]
Analysis in CAD [155] A Knowledge-based Expert
System for Automatic
Analysis of Dependencies to Improve the Behavior of
Logic Programs [64]
Analysis of Imprecise Data [410] Information System: a
Data-driven System for the
Analysis [145] A Logic-based Expert System for Model
Building in Regression
Analysis [459] Logic for Natural Language
Analysis [615] Fifth Generation Computer Architecture
Analysis [617] Fifth Generation Computer Architecture
Analysis [697] Semantic Code
And Parallel PROLOG with Divided Assertion Set [433]
And Parallelism in Logic Programs [131]
And/or Process Model for Parallel Interpretation of
Logic Programs [129] The
And/or Trees [471] Backtracking Intelligently in
Annotated Logic Programs [234] Towards the Compilation of
Another Qlog [79] (Re) Implementing PROLOG in LISP
or YAP—Yet
Answers [139] A Short Cut to More Informative
Answers [666] PROLOG/ex1, an Inference Engine
Which Explains Both Yes and No
APES: a User Manual [243]
APL, LISP, MODULA-2; Smalltalk, PROLOG Com-
puter Languages of the Future [12] TH.
Appendix to PROLOG: a Language for Implementing
Expert Systems [245]
Application of Hash to Data Base Machine and its
Architecture [328]
Application of Meta-language Programming to Fault
Finding in Logic Circuits [174]
Application of Multiple Rewrite Rule Sets in Algebraic
Manipulation [72]
Application of PROLOG in Designing Many-storied
Dwelling Houses [373]
Application of PROLOG in Symbolic Computing
[36] ing Contours of Integration: an
Application of PROLOG to the Development of QA
and DBM Systems [206] The
Application to Partial Evaluation [336] on of an Abstract
PROLOG Machine and its
Applications: User and Social Acceptability of FGCS
[534] New
Applications in Hungary [521] PROLOG
Applications in Hungary [522] PROLOG
Applications in Hungary [588] PROLOG
Applications in Hungary [589] PROLOG
Applications of Theorem Proving Based Machine
Intelligence in Qsar: Automation [146]
Applications [492] Prograph as an Environment for
PROLOG DB
Applicable Communicating Processes in First Order
Logic [34]
Applied Logic—its Use and Implementation as a Pro-
gramming Tool [673]
Applied to the PROLOG and Itself Which in Turn is
Intended to be Given to [300]
Applying an Inductionless Technique to Prove Properties
of Restricted PROLOG [25]
Approach to Self-descriptive Deductive Database
[450] A Logic Programming
Approach [160] Logic Programming: a Parallel
Programming
Architecture—Research and Development Plan for
—Computer Architecture [620]
Architecture—the Development of System 5G
[520] VLSI and System
Architecture Analysis [615] Fifth Generation Computer
Architecture Analysis [617] Fifth Generation Computer
Architecture and Hardware System [629] nce Machine
(SIM-P or PSI) Outline of its
Architecture and Vision [157] Parallel
Architecture in the Fifth Generation Computer Project
[620] Plan for—Computer
Architecture of Computer Vision [387] A Cognitive
Architecture of Fifth Generation Computer Systems
[600] nce Machine as the Basic
Architecture [236] Developments in Dataflow
Architecture [328] Application of Hash to Data Base
Machine and its
Architecture [445] of Inference Machine (Psi): its Design
Philosophy and Machine
Architecture [446] The Personal Sequential Machine
(Psi): its Design and Machine
Architecture [619] Functional Programming and Com-
puter
Architecture [624] Toward a New Generation Computer
Architecture, and Technology [10] Algorithms,
Architectures for Inference Mechanisms [626] New
Architectures [35] On Compiling PROLOG Programs on
Demand Driven
Architectures [616] VLSI Processor
Artificial Intelligence Commercial Products Begin to Emerge from Decades of Research [370].
Artificial Intelligence Practical Systems Use Natural Languages and Store Data [175].
Artificial Intelligence and Computers [604].
Artificial Intelligence and Japan's Computer Challenge to the World [181].
Artificial Intelligence LISP and PROLOG Machines are Proliferating [369].
Artificial Intelligence Methods [209] A Discrete Simulation System Based on ASPLE Using Predicate Logic [419].
Using PROLOG to Assimilation Method for Logic Databases [402].
Assimilation Method for Logic Databases [403]. Knowledge Acquisition with Knowledge Assistant [668]. KBOI: a Knowledge Based Garden Store.
Survey of the Formalism and a Comparison with Automated Deduction [380].
Intelligent Backtracking for Automatic Analysis in CAD [155].
A Knowledge-based Expert System for Automatic Calculation of Molecular Properties and Automatic Interpretation [146].
Automatic Generation of Mode Declaration for PROLOG Programs [392].
The Automatic Generation of Mode Declaration for PROLOG Programs [393].
The Automatic Generation of the Input-output Relation [566].
Agreement Transformation and Automatic Interpretation of Quantitative Structure-activity Relationships [146].
Automatic Synthesis of Expert Knowledge through Qualitative Modelling [426] in Average Size of Turner's Translation to Combinator Programs [269].
Axiomatic Data Base Theory [608].
Back-up Parallel Interpreter of PROLOG Program [204].
Backtrack in Horn Clause Programming [356]. Controlling Backtracking, and Sidetracking in Horn Clause Programs—Implementation [484].
Backtracking and Sidetracking in Horn Clause Programs—Implementation [487].
Backtracking and Sidetracking in Horn Clause Programs—the Theory [488].
Backtracking at Work [483].
Selective Backtracking for an Interpreter of Horn Clause Logic Programs [66].
Intelligent Backtracking for Automated Deduction [380].
Intelligent Backtracking for Logic Programs [485].
Selective Backtracking for Logic Programs [486].
Selective Backtracking in Plan-based Deduction [381].
Intelligent Backtracking Intelligently in AND/or Trees [471].
Selective Backtracking [481].
Selective Backtracking [482].
Backtracking [482].
Solving Combinatorial Search Problems by Intelligent Backtracking [483].
Solving Combinatorial Search Problems by Intelligent Backtracking [487].
Revision of Top-down Logical Reasoning through Intelligent Backtracking [68].
Revision of Top-down Logical Reasoning through Intelligent Backtracking [68].
Bagel: a Systolic Concurrent PROLOG Machine [540].
Lecture Notes on Base de Dados Face [473].
Manual de Utilizacao Da Base Machine and its Architecture [528].
Application of Hash to Data Base Machine as the Basic Architecture of Fifth Generation Computer Systems [600].
Base Machine [429]. A Relational Database Machine: First Step to Knowledge Base Machine [430].
A Relational Database Machine: First Step to Knowledge Base Machine [430].
A Relational Database Machine: First Step to Knowledge Base Machine [430].
A Relational Database Machine: First Step to Knowledge Base Machine [430].
Agnement, Knowledge Base Management System [441].
Towards a Co-operative Data Base Management, Knowledge Base Management and Expert System Development [455].
New Architecture for Knowledge Base Mechanisms [581].
Knowledge Base Query Languages [494].
High Level Data Base Support System for PROLOG [88].
Data Base System Based on Logic [399].
An Experimental Relational Data Base Theory [608].
An Axiomatic Data Base [20].
An Inference based on Fuzzy Logic Knowledge Base: Theory vs. Interpretation [442].
Data Base Expert Systems [404].
PROLOG Based Garden Store Assistant [668].
KBOI: a Knowledge Base System.
A Knowledge Description and its Verification with PROLOG [195].
Temporal Logic Based Machine Intelligence in Qsar: Automatic Calculation of Molecular Pr [146].
Based on a Natural Deduction System [256].
Logical Programming Based on a Natural Deduction System [606].
A Programming Language Based on a Natural Deduction System [607].
A Programming Language Based on Artificial Intelligence Methods [209].
A Discrete Simulation System Based on ATN and PROLOG [605].
Sova—an Integrated Question-answering System Based on ATN and PROLOG [605].
Sova—an Integrated Question-answering System Based on ATN and PROLOG [605].
Sova—an Integrated Question-answering System Based on ATN and PROLOG [605].
Sova—an Integrated Question-answering System Based on ATN and PROLOG [605].
Sova—an Integrated Question-answering System Based on ATN and PROLOG [605].
Complexity of Unification [705]On the Parallel Computational
Computation and Deductive Information Retrieval [648]
Computation and the Proof in Logic [452]The Relation Between
Computation Interpreter [465]A PROLOG Demand Driven
Computation Mechanisms [709]PROLOG and Data-flow
Computation Principle—an Alternative Method of Adapting Resolution for L [18]
Computation [295] The Scope of Symbolic
Computational and Deductive Information Retrieval [651]
Computational Complexity of Logic Programs [360] A Note on
Computational Complexity of Logic Programs [548] Alternation and the
Computational Formalism [97] Predicate Logic as a
Computational Models [286] Order: its Implications to Programming Languages and
Computer—Social Needs and its Impact [319] is Required of the Fifth Generation
Computer Architecture—Research and Development Plan for—Computer Arch [620]
Computer Architecture Analysis [613] Fifth Generation
Computer Architecture Analysis [617] Fifth Generation
Computer Breed Uses Trampeters for Parallel Processing [568] New
Computer Challenge to the World [181] ration Artificial Intelligence and Japan’s
Computer Language in Schools [169] Teaching Logic as a
Computer Language [183] PROLOG a Step Toward the Ultimate
Computer Language [345] Logic as a
Computer Language [348] Logic as a
Computer Languages of the Future [12] TH, APL, LISP, MODULA-2, Smalltalk PROLOG
Computer Languages [164] Superfast Unit Exploits Fifth-generation
Computer Modelling of Mathematical Reasoning [69] The
Computer Project—a Trip Report [547] Japan’s Fifth Generation
Computer System Project [422] Overview to the Fifth Generation
Computer System Project [614] Japan’s Fifth Generation
Computer System [325] Culture Creation and a New Generation
Computer Systems [180] Innovation in Symbol Manipulation in the Fifth Generation
Computer Systems [282] and Development Themes and Plans of the Fifth Generation
Computer Systems [283] nec of Research and Development Plans for Fifth Generation
Computer Systems [353] PROLOG and Relational Databases for Fifth Generation
Computer Systems [424] Fifth Generation
Computer Systems [600] use Machine as the Basic Architecture of Fifth Generation
Computer Systems [618] Japan’s Fifth-generation
Computer Utilisation of Knowledge [643] Chess Endgame Advice: a Case Study in
Computer [710] Logic Programming and a Dedicated High performance Personal
Computer [87] ained PROLOG—as a Preliminary Kernal Language for 5th Generation
Computer-based Synthesis of Logic Programs [171] Computers [604] Artificial Intelligence and
Computers [86] ESP as a Preliminary Kernal Language of Fifth Generation
Computing Algorithms and Proving Properties by Computing Terms [454]
Computing [291] Japan Moves to Dominate ‘Fifth-generation’
Computing [36] ing Contours of Integration: an Application of PROLOG in Symbolic
Conceptual Basis and Evaluation Strategy for Integrating Functional and L [579]
Conceptual Structures Information Processing in Mind and Machine [569]
Concerns [179] Worldwide Fifth Generation Computing: Developments, Issues and
Concurrency [150] A First Order Semantics of a Connective Suitable to Express
Concurrency [406] A Horn Clause-like Logic for Specifying
Concurrency Algorithm [275] Logic Representation of a Concurrent Execution of Logic [44]
Concurrency Logic Programming [273]
Concurrency Programming in Logic [408] A New Proposal for
Concurrency Programming Language [447] Description of the PROLOG Interpreter by a
Concurrency Programming [201] PROLOG Interpreter Based on
Concurrency Programming [202] PROLOG Interpreter Based on
Concurrency Programming [203] PROLOG Interpreter Based on
Concurrency PROLOG and its Interpreter [541] A Subset of
Concurrency PROLOG and its Interpreter [546] A Subset of
Concurrency PROLOG Machine [540] Lecture Notes on
Bagel: a Systolic Concurrent PROLOG [227] A Note on Systems Programming in
Control of Activities in the Or-parallel Token Machine

Control for Logic Programs [222] Metalevel

Control Flow of PROLOG Programs [228] Architecture for Fifth Generation Computer [600]

Control Flow of Design Errors in Logic-based CAD Programs [318]

Control of Logic Programs Using Integrity Constraints [156]"Logical": Algorithmic

Control Using Integrity Constraints [332] Intelligent

Control with Logic [466] Logic

Control with Logic [468] Logic

Control [349] Algorithm = Logic +

Controlling Backtrack in Horn Clause Programming [356]

Controlling Inference in the Semantic Interpretations of Mechanics Probie [391]

Coroutines in PROLOG [329] Remarks on

Corouting for the 8 Queens Problem [216] Simulating

Correctness of Digital Hardware Designs [28] Proving

the

Country [505] PROLOG Simulation of Migration Decision-making in a Less Developed Course at Brandeis University [637]


Creation and a New Generation Computer System [325] Culture

Cube Production System [196] Rubik’s

Culture Creation and a New Generation Computer System [325]

Current Trends in Logic Grammars [137]

Customizable Microprogram Assembler [280]

Cuttable Formulas for Logic Programming [228]

D’exemple [316] PROLOG II Manuel

D’utilisation [517] PROLOG Manuel de Reference et D’utilisation [636] PROLOG II Manuel

D-lists [253] Program Transformation by a Function That Maps Simple Lists Onto

DADM [321] Logic Programming in

Dado Machine: a Parallel System for High-speed Logic Programming [610] OG on the


Data Base Machine as the Basic Architecture of Fifth Generation Computer [600]

Data Base Machine: First Step to Knowledge Base Machine [430] A Relational

Data Base Management System [441] Towards a Cooperative

Data Base Management, Knowledge Base Management and Expert System Develop [455]

Data Base Query Languages [494] High Level

Data Base Support System for PROLOG [88] A

Data Base System Based on Logic [399] An Experimental Relational

Data Base Theory [608] An Axiomatic

Data Base: Theory vs. Interpretation [442]

Data Bases "à La Carte" [189] Relational
Epilog a Language for Extended Programming in Logic [497]
Epilog [193] Predicate Logic Programming a Proposal of Equality for PROLOG [339]
Flowcharts, and “Lucid” Style Programming in Logic
Focalizers, the Scoping Problem, and Semantic Interpretation Rules in Logic [386]
Foollog—a Small and Efficient PROLOG Interpreter [444]
Form and Function [511]Logic:
Form [411]Problems in Logical
Formal Definition of ASPLE Using Predicate Logic [419]
Formal Model for Or-parallel Execution of Logic Programs [91]
Formal Specifications—Dijkstra’s Three Maxes [194]
ion of Logic Programs from Formal System for Representing Grammatical
Relations—In PROLOG—Toward a
Formalism and a Comparison with Augmented Transition Networks [463]
Formalism and its Implementation Technique [631]
_intermediate Logic: A Survey of the
Formalism [97]Predicate Logic as a Computational
Format [628]Sequential Inference Machine Object Data Formulas for Logic Programming [228]
Cuttable FORTH, APL, LISP, MODULA-2, Smalltalk, PROLOG Computer Languages of the Future [12]
Forward Chaining Problem Solver [412]A Foundations of Logic Programming [361]
Foundations [262]Fifth Generation
Founding of this Journal [421]On the
Four Obstacles to End User Access [261]The
Friendly Interface [519]Modelling Human-computer Interactions in a
Fujitsu Research Activities in Knowledge Information Processing System [525]
Fun [170]PROLOG Can Link Diverse Subjects with Logic and
Function That Maps Simple Lists Onto D-lists [254]Program Transformation by a
Function [511]Logic: Form and
Functional and Logic Programming [579]
Formal and Logic Programming Implementation of Lexical
Functional Interpretation of a Logical Sentence—a Preliminary Note—[453]
Functional Languages [263]Efficient Storage Management for
Functional Plus Predicate Logic Programming Language [33]
Functional Programming and Computer Architecture [619]
Functional Programming [24]On the Integration of Logic Programming and
Functional Programming [78]On Implementing PROLOG in
Functional Programs [50]Transformation of Logic Programs into
Functional Style and its Algebra of Programs [19] from the Von Neumann Style? a
Functions Defined in First Order Logic [7]Evaluating Functions [16]me Aspects of the Static Semantics of Logic Programs with Monadic
Functions [32]The Call by Name Semantics of a Clause Language with
Functions [53]Horn Clause Programs for Recursive Functions [53]Horn Clause Programs Suggested by Recursive
Further Investigations of Deduction in Relational Data Bases [84]Deduce 2:
Game of Mastermind [64]Relational Programming Illustrated by a Program for the
Game Program [59] Using PROLOG for a
Garbage-collection in PROLOG Interpreters [59]A Note on
Garden Store Assistant [66]KBO1 : a Knowledge Based
Generating Contours of Integration: an Application of PROLOG in Symbolic C [36]
Generation Computer—Social Needs and its Impact [319]
_is Required of the Fifth
Generation Computer Architecture—Research and Development Plan for—Co [620]
Generation Computer System Project [422] Overview to the Fifth
Generation Computer System Project [614] Japan’s Fifth Generation Computer System [525] Culture Creation and
a New
Generation Computer Systems [180] Innovation in Symbol Manipulation in the Fifth
Generation Computer Systems [282] and Development Themes and Plans of the Fifth
Generation Computer Systems [283] Use of Research and Development Plans for Fifth
Generation Computer Systems [353] PROLOG and Relational Databases for Fifth
Generation Computer Systems [424] Fifth
Generation Computer Systems [600] as a Machine as the Basic Architecture of Fifth
Generation Computer [87] ained PROLOG—as a Preliminary Kernel Language for 5th
Generation Computers [86] ESP as a Preliminary Kernel Language of Fifth
Generation Foundations [262] Fifth
INDEX

Generation Kernal Language Version 0.0 [714]
A Draft Proposal of Fifth Generation Kernal Language Version 0.1 [713]
A Draft Proposal of Fifth Generation of Computer Architecture [613]
The New Generation of Mode Declaration for PROLOG Programs [392]
The Automatic Generation of Mode Declaration for PROLOG Programs [393]
The Automatic Generation of the Input-output Relation [566]
Transformation and Automatic Generation Project in Japan [343]
The Fifth Generation Research in Japan—a Trip Report [674]
Fifth Generation VLSI Design with AI [575]
Fifth Generation [128]
Japan's Fifth Generation [290]
Japan Builds a Pillar of the Fifth Generation [344]
Logic Programming for the Fifth Generation [358]
The Role of Systems and Software Technology in the Fifth Generation [365]
In the Maze of the Fifth Generation [500]
Networks for the Fifth Generation [711]
SYSP: a New Programming Language to the Next Generation [238]
PROLOG Geometry Theorem Prover [114]
Geometry Theorem Prover [114]
Geometric: a PROLOG Compiler [300]
Given to Itself Together with the PROLOG to Produce a PROLOG Compiler [300]
Gp-pro Graphic Display Control Library Written in PROLOG [281]
Grammar in PROLOG [272]
Restriction
Grammar [504]
A PROLOG Implementation of Lexical Functional Grammar [699]
Beyond PROLOG: Software Specification by Grammars for Language Analysis—a Survey of the Formalism and a Compass [463]
Grammars in Logic [460]
Some Techniques for Writing Grammars in PROLOG [518]
Contextual Grammars [125]
Metamorphism
Grammars [137]
Current Trends in Logic
Grammars [1]
Definite Clause Translation
Grammars [367]
Comparison of the Logic Programming Language PROLOG with Two-level Grammars [386]
the Scoping Problem, and Semantic Interpretation: Rules in Logic
Grammars [461]
Extraposition
Grammars [462]
Extraposition
Grammars, and Programs [650]
Relational Equations, Grammatical Relations—[704]
In PROLOG—Toward a Formal System for Representing Grammatical Unification [368]
Grammer [564]
Toward a PROLOG Text
Graphic Display Control Library Written in PROLOG [281]
Gp-pro
Graphics with Infinite Terms [159]
Logic Programming
Graphs as Data in PROLOG Programs [530]
Graphs [352]
A Proof Procedure Using Connection
Guide to BS2000 T-PROLOG Simulation System [212]
User's
Guide to Decsystem-10 PROLOG [476]
User's
Guide to Version J of DEC-10 PROLOG [77]
A Guide to PROLOG II on VAX Installation and Utilisation
Handling and Drug Design [144]
Logic Programming in Chemical Information
Handling Negative Knowledge [6]
A PROLOG Extension for Hardware Description and its Verification with PROLOG [195]
Temporal Logic Based Hardware Designs [28]
Proving the Correctness of Digital Hardware System [629]
Nice Machine (SIM-P or PSI)
Outline of its Architecture and Hardware Takes Shape [115]
Fifth-generation Harmonic Mean of LISP, PROLOG, and Smalltalk [595]
Tao—a Hash to Data Base Machine and its Architecture [328]
Application of Hasl [2]
A Prological Definition of
HC Manual: Virginia Tech PROLOG [506]
The HCPRVR: an Interpreter for Logic Programs [85]
Heart for a Medical Expert System [425]
A Qualitative Model of the High Language on a Very Low Cost Computer [317]
Implementing a Very High Level Data Base Query Languages [494]
High Level Simulation System [210]
T-PROLOG Very High Level Simulation System [211]
T-PROLOG: a Very High Level Simulation System [213]
T-PROLOG Very High-performance Personal Computer [710]
Logic Programming and a Dedicated High-speed Logic Programming [610]
OG on the Dado Machine: a Parallel System for Higher Order Logic Programming Language [558]
Snepslog a
Higher Order: its Implications to Programming Languages and Computational [286]
Higher-order Extensions to PROLOG—are they Needed? [680]
Higher-order Extensions to PROLOG: are they Needed? [677]
Highly Parallel Machine [82]
Logic Programming on ZMOB: a Hoare's Program Find Revisited [561]
Horn Clause Logic Programs [66]
Intelligent Backtracking for an Interpreter of Horn Clause Programming [356]
Controlling Backtrack in Horn Clause Programs—Implementation [484]
ent Backtracking and Sidetracking in Horn Clause Programs—Implementation [487]
ent Backtracking and Sidetracking in Horn Clause Programs—the Theory [488]
ligicut Backtracking and Sidetracking in Horn Clause Programs for Recursive Functions [531]
Horn Clause Programs Suggested by Recursive Functions [332]
Horn Clause-like Logic for Specifying Currancy [406]
A Horn Clauses as a Production System [357]
The Denotational Semantics of
Horn Clauses with Infinite Terms [176]
On the Fixed-point Semantics of
Home Logic Programming System [190]
An Overview of the Houses [373]
Application of PROLOG in Designing Many-storied Dwelling
Human Cognition [238]
Deductive Modeling of Human Cognition [239]
Deductive Modeling of
Human Expertise [175]_ligence Practical Systems Use
Natural Languages and Store
Human-computer Interactions in a Friendly Interface
[519]Modelling
Hungary [521]PROLOG Applications in
Hungary [522]PROLOG Applications in
Hungary [585]PROLOG Applications in
Hungary [589]PROLOG Applications in

IC-PROLOG—Language Features [103]
IC-PROLOG Language Features [106]
IC-PROLOG Language Features [107]
IC-PROLOG [105]The Control Facilities of
IC-PROLOG [385]Getting Started with
IC-PROLOG: Aspects of its Implementation [104]
Illustrated by a Program for the Game of Mastermind
[647]Relational Programming
Impact [319]_is Required of the Fifth Generation
Computer—Social Needs and its
Impacts of Logic on Data Bases [220]
Imperative Complement to PROLOG [653]A Proposal
for an
Implement PROLOG [21]A Virtual Machine to
Implement PROLOG [458]A LISP-machine to
Implementation as a Programming Tool [673]Applied
Logic—its Use and
Implementation Compared to LISP [691]PROLOG—the
Language and its
Implementation in Concurrent PROLOG [543]_erge
Operators: Their Specification and
Implementation in Concurrent PROLOG [554]_erge
Operators: Their Specification and
System and its
Implementation Issues of PROLOG [58]Some Reflections
on
Implementation of a Data-independent Expert System
with Quasi-natural Lan. [574]
Implementation of a Knowledge Acquisition System
[326]A Methodology for
Implementation of a Knowledge Acquisition System
[327]A Methodology for
Implementation of a Large System on a Small Machine
[477]A PROLOG
Implementation of a Personal Sequential Inference
Machine: PSI [715]. Design and
Implementation of a PROLOG Interpreter [390]_native
to Structure Sharing in the
Implementation of a PROLOG Interpreter [394]_native
to Structure-sharing in the
Implementation of a PROLOG Interpreter [395]_native
to Structure-sharing in the
Implementation of Control in Logic Programming Lan-
guages [188]On the
Implementation of Lexical Functional Grammar [504]A
PROLOG
Implementation of PIL (PROLOG in LISP) [669]An
Easy
Implementation of PROLOG [507]An
Implementation of the Knuth-bendix Reduction System
[378] A PROLOG
Implementation of Uniform a Knowledge-
representation/programming Language_ [301]
Intelligent Backtracking [67]Revision of Top-down Logical Reasoning through

Intelligent Backtracking [68]Revision of Top-down Logical Reasoning through

Intelligent Control Using Integrity Constraints [332]

Intelligent Logic [563]PROLOG

Intelligent Man-machine Interface [601]

Intelligent UNIX Shell Project [229]

Intelligently in And/or Trees [471]Backtracking

Intended to be Applied to the PROLOG and Itself Which in Turn is Intended [300]

Intended to be Given to Itself Together with the PROLOG to Produce a PROL _[300]

Interaction Language [639]Logic as

Interactions in a Friendly Interface [519]Modelling Human-computer

Interactions [147]Logic-based Program System for Predicting Drug

Interactive Program Verifier for PROLOG Programs [223]On an

Interactive Relational Database Queries Expressed in Logic [678] t Processing of

Interactive Relational Database Queries Expressed in Logic [679] t Processing of

Interesting Subset of Natural Language [122]An

Interface [183]A Kernel for a General Natural Language

Interface [519]Modelling Human-computer Interactions in a Friendly

Interface [601]Intelligent Man-machine

Interfacing Predicate-logic Languages and Relational Data-bases [83]

Intermission—Actors in PROLOG [303]

Intermission—Actors in PROLOG [306]

Intermission—Actors in PROLOG [308]

Interpretation of a Logical Sentence—a Preliminary Note—[453]Functional

Interpretation of Logic Programs [129]The And/or Process Model for Parallel

Interpretation of Logic Programs [132]Parallel

Interpretation of Quantitative Structure-activity Relationships [146] Automatic

Interpretation of the "Japanese Knowledge" Bomb [396]

Interpretation Rules in Logic Grammars [386], the Scoping Problem, and Semantic

Interpretation [442]Data Base: Theory vs.

Interpretations of Mechanics Problems [391]ontrolling Inference in the Semantic

Interpreter Based on Concurrent Programming [201]PROLOG

Interpreter Based on Concurrent Programming [202]PROLOG

Interpreter Based on Concurrent Programming [203]PROLOG

Interpreter by a Concurrent Programming Language [447]Description of the PROLOG

Interpreter de PROLOG [187]Un

Interpreter for Distributed Logic [407]A Small Interpreter for Logic Programs [415]A Dataflow


Interpreter [390]_native to Structure Sharing in the Implementation of a PROLOG

Interpreter [394]_native to Structure-sharing in the Implementation of a PROLOG

Interpreter [395]_native to Structure-sharing in the Implementation of a PROLOG

Interpreter [444]Foolog—a Small and Efficient PROLOG

Interpreter [465]A PROLOG Demand Driven Computation

Interpreter [541]A Subset of Concurrent PROLOG and its

Interpreter [546]A Subset of Concurrent PROLOG and its

Interpreters [59]A Note on Garbage-collection in PROLOG


Interprocess Communication in Concurrent PROLOG [593]

Interrogacao Em Portugues Das Priorida Des Em Ciencia E Technologia [478] a Para

Introduction of a Complexity Measure for Control of Design Errors in Logi [318]

Introduction to Logic and Data Bases [225]An Overview and

Introduction to Logic Programming [94]An

Introduction to Logic Programming [95]An

Introduction to Mu-PROLOG [432]An

Introduction to PROLOG, a 'Fifth-generation' Language [112]


Invertibility of Logic Programs [560]

Investigations of Deduction in Relational Data Bases [84]Deduce 2: Further Involving Data Structure as a Knowledge Representation Language [449] cate Logic

Iota on Decsystem 20—KWIC Example [240]

Issues and Concerns [179]Worldwide Fifth Generation Computing: Developments,

Issues in Developing Expert Systems [397]

Issues of PROLOG [58]Some Reflexions on Implementation itself Together with the PROLOG to Produce a PROLOG Compiler [300] o be Given to itself Which in Turn is Intended to be Given to Itself Together with the [300]

Japan—a Trip Report [674]Fifth Generation Research in

Japan Builds a Pillar of the Fifth Generation [290]

Japan Moves to Dominate 'Fifth-generation' Computing [291]

Japan [343]The Fifth Generation Project in
INDEX

Japan's Computer Challenge to the World [181]_ration
Artificial Intelligence and
Japan's Developments Observed [366]
Japan's Fifth Generation Computer Project—a Trip
Report [547]
Japan's Fifth Generation Computer System Project [614]
Japan's Fifth Generation [128]
Japan's Fifth-generation Computer Systems [618]
Japanese FGCS Project [707]_Perspective of the
Japanese Knowledge Bomb [396]_Interpretation of the

K/R Language Feature [436]_PROLOG
KB01 : a Knowledge Based Garden Store Assistant [668]
Kernal Language for 5th Generation Computer [87]_ained
PROLOG—a Preliminary
Kernal Language of Fifth Generation Computers [86]_ESP
as a Preliminary
Kernal Language Version 0.0 [14]_A Draft Proposal of
Fifth Generation
Kernal Language Version 0.1 [713]_A Draft Proposal of
Fifth Generation
Kernel for a General Natural Language Interface [185]_A
Knowledge Bomb [396]_Interpretation of the “Japanese
Knowledge Acquisition in PROLOG [235]
Knowledge Acquisition System [326]_A Methodology for
Implementation of a
Knowledge Acquisition System [327]_A Methodology for
Implementation of a
Knowledge Acquisition [491]_On the Use of Self-
description for
Knowledge Assimilation Method for Logic Databases [402]_A
Knowledge Assimilation Method for Logic Databases [403]_A
Knowledge Base Machine [429]_A Relational Database
Machine: First Step to
Knowledge Base Machine [430]_A Relational Data Base
Machine: First Step to
Knowledge Base Management and Expert System
Development in PROLOG [455]_agement,
Knowledge Base Mechanisms [581]
Knowledge Base [20]_An Inferential Fuzzy Logic
Knowledge Based Garden Store Assistant [668]_KB01 : a
Knowledge Information Processing Language : Shapeup [338]
Knowledge Information Processing System [525]_Fujitsu
Research Activities in
Knowledge Information Processing Systems [192]_Aiming
for
Knowledge Information Processing Systems [423]_Challenge for
Knowledge Programming Language on Concurrent
PROLOG [278]_Mandala a
Knowledge Representation in an Efficient Deductive
Inference System [573]
Knowledge Representation in PROLOG/KR [434]
Knowledge Representation Language [449]_cate Logic
Involving Data Structure as a
Knowledge Representation of Design in Many-sorted
Logic [375]
Knowledge Representation System in PROLOG [580]_A
Knowledge Representation Systems? [161]_How
Complete are
Knowledge through Qualitative Modelling [426]_t in
Automatic Synthesis of Expert
Knowledge Utilization and Realization [355]_A Logic
Programming Language for
Knowledge [136]_Logic Programming as a Representa-
tion of
Knowledge [643]_Chess End-game Advice: a Case Study
in Computer Utilisation of
Knowledge [6]_A PROLOG Extension for Handling
Negative
Knowledge-based Expert System for Automatic Analysis
in CAD [155]_A
Knowledge-based Problem-solving in AL3 [54]
Knowledge-representation/programming Language
Based Upon Equivalence of D_ [301]
Knuth-bendix Reduction System [378]_A PROLOG
Implementation of the
KS as a Query Language [153]_The Predicate Calculus-
language
KWIC Example [240]_Iota on Decsystem 20—
Language : Shapeup [338]_Knowledge Information
Processing
Language Access to PROLOG Database Systems [529]_Natural
Language Analysis—a Survey of the Formalism and a
Comparison with Augme_ [463]
Language Analysis [459]_Logic for Natural
Language and its Implementation Compared to LISP [691]_PROLOG— the
Language and Metalanguage in Logic Programming [50]_Amalgamating
Language and Metalanguage in Logic Programming [51]_Amalgamating
Language and Metalanguage in Logic Programming [52]_Amalgamating
Language and Understanding [708]
Language Based on a Natural Deduction System [606]_A
Programming
Language Based on a Natural Deduction System [607]_A
Programming
Language Based Upon Equivalence of Descriptions [301]_representation/programming
Language Based Upon Unification Which Unifies (Much
Of) LISP, PROLOG, and_ [304]
Language Based Upon Unification Which Unifies (Much
Of) LISP, PROLOG, and_ [305]
Language Based Upon Unification Which Unifies (Much
Of) LISP, PROLOG, and_ [307]
Language Consultable Data Bases [143]_Logical Design of
Deductive Natural
Language Feature [436]_PROLOG K/R
Language Features (a Brief Survey) [437]_PROLOG/KR : the
Language Features of LPL, a Logic Programming Lan-
guage [248]
Language Features [103]_IC-PROLOG—
Language Features [106]_IC-PROLOG
Language Features [107]_IC-PROLOG
Language for 5th Generation Computer [87]_ained
PROLOG—as a Preliminary Kernal
Language for Extended Programming in Logic [497]_Epilog a
Language for Implementing Expert Systems
[101]_PROLOG: a
Liberated from the Von Neumann Style? a Functional Style and its Algebra o...[19]
Libraries in PROLOG [184]Building
Library Written in PROLOG [281]Gp-pro Graphic Display Control
Linear Resolution Theorem Proving Program in PROLOG [428]An Ordered
Link Diverse Subjects with Logic and Fun [170]PROLOG Can
LISP and PROLOG Machines are Proliferating [369]Artificial Intelligence
LISP in Pure PROLOG [479]Pure
LISP Intended to be Applied to the PROLOG and Itself Which in Turn is Int...[300]
LISP Written in PROLOG Written in LISP Intended to be Applied to the PROLOG. [300]
LISP, PROLOG, and Act 1 [304] age Based Upon Unification Which Unifies (Much Of)
LISP, PROLOG, and Act 1 [305] age Based Upon Unification Which Unifies (Much Of)
LISP, PROLOG, and Act 1 [307] age Based Upon Unification Which Unifies (Much Of)
LISP, PROLOG, and Smalltalk [595] Tao --- a Harmonic Mean of
LISP-machine to Implement PROLOG [458]A LISP [448]PROLOG Compared with Listing of a PROLOG Program Describing Entitlement to Supplementary Benef...[242]
Logic Based Hardware Description and its Verification with PROLOG [195]Temporal Logic Based Software Development Method [587]LDM—a
Logic Bases of the System Sphinx [294]Medical Decision Aid:
Logic Programming as a Representation of Knowledge [136] Logic Programming Based on a Natural Deduction System [256] Logic Programming for Expert Systems [244]
Logic Programs [548]| Alternation and the Computational Complexity of
Logic Programs [560]| Invertibility of
Logic Programs [633]| A Parallel Execution Model of
Logic Programs [634]| Parallel Execution of
Logic Programs [64]| Analysis of Dependencies to Improve the Behavior of
Logic Programs [66]| Intelligent Backtracking for an Interpreter of Horn Clause
Logic Programs [686]| Implementing PROLOG—Compiling Predicates
Logic Programs [687]| Implementing PROLOG—Compiling Predicates
Logic Programs [85]| HCPRVR: an Interpreter for
Logic Programs [91]| A Formal Model for Or-parallel Execution of
Logic Programs [93]| The Synthesis and Verification of
Logic Programs: a Research Proposal [133]| Efficient
Logic Programs: a Research Proposal [134]| Summary of Efficient
Logic Representation of a Concurrent Algorithm [275]
Logic Simulation in PROLOG [276]| Digital
Logic Simulation in PROLOG [625]
Logic T-PROLOG [215]| A Modelling Tool Based on Mathematical
Logic through Logic [140]| Translating Spanish into
Logic [138]| On Database Systems Development through
Logic [140]| Translating Spanish into Logic through
Logic [277]| Programming Law in
Logic [34]| Applicative Communicating Processes in First Order
Logic [374]| Using Programming Language
PROLOG PROLOG Based Upon Mathematical
Logic [375]| Knowledge Representation of Design in Many-sorted
Logic [399]| An Experimental Relational Data Base System Based on
Logic [405]| A Proposal for Distributed Programming in
Logic [407]| A Small Interpreter for Distributed
Logic [408]| A New Proposal for Concurrent Programming in
Logic [419]| A Formal Definition of ASPLE Using Predicate
Logic [44]| Concurrent Execution of
Logic [452]| The Relation Between Computation and the
Logic [45]| Programming with Full First-order
Logic [460]| Some Techniques for Writing Grammars in
Logic [466]| Logic Control with
Logic [468]| Logic Control with
Logic [46]| Programming with Full First Order
Logic [47]| Programming with Full First Order
Logic [497]| Epilog: a Language for Extended Programming in
Logic [563]| PROLOG Intelligent
Logic [565]| A Narrative Schema in Procedural
Logic [577]| A Decision Method for Process
Logic [57]| Dataflow, Flowcharts, and "Lucid"—Style Programming in
Logic [621]| Inference Machine: from Sequential to Parallel
Logic [652]| Programming Machine: from Sequential to Parallel
Logic [657]| Programming with Resolution
Logic [678]| t Processing of Interactive Relational Database Queries Expressed in
Logic [679]| t Processing of Interactive Relational Database Queries Expressed in
Logic [700]| Deriving Different Unification Algorithms from a Specification in
Logic [7]| Evaluating Functions Defined in First Order
Logic-based CAD Programs [318]| Complexity Measure for Control of Design Errors in
Logic-based Expert System for Model Building in Regression Analysis [145]| A
Logic-based Program System for Predicting Drug Interactions [147]
Logic—its Use and Implementation as a Programming Tool [673]| Applied
Logic-programming Era is Dawning [217]
Logic: a Calculus for Deriving Programs [92]| Predicate
Logic: a Survey of the Formalism and its Implementation Technique [631]| Predicate
Logic: Form and Function [511]
Logica Para Procesos Distribuidos [409]| Uma Logica Action Systems [496]
Logical Basis of Programming by Assertion and Query [510]| The
Logical Data Bases vs Deductive Data Bases [218]
Logical Design of Deductive Natural Language Consultable Data Bases [143]
Logical Form [411]| Problems in
Logical Program Synthesis [39]
Logical Programs [382]| The Meaning of
Logical Reasoning through Intelligent Backtracking [67]| Revision of Top-down
Logical Reasoning through Intelligent Backtracking [68]| Revision of Top-down
Logical Reconstruction of PROLOG II [662]| A
Log i c a l S e n t e n c e — a P r e l i m i n a r y Note—[453]| Functional Interpretation of a Logically [544]| Playing Mastermind
LOGLISP: an Alternative to PROLOG [512]
LOGLISP: Motivation, Design and Implementation [513]
LOGLISP: Motivation, Design, and Implementation [514]
Long-range Regional Planning Problem [214]| Using T-PROLOG for a
Low Cost Computer [317]| Implementing a Very High Language on a Very
LPG Bulletin [660]
LPL, a Logic Programming Language [248]| Language Features of
Lucid"—Style Programming in Logic [57]| Dataflow, Flowcharts, and "
Machine "Delta" (Translated from Ipsj) [311]| A Relational Database
Machine "Delta" [310]| A Relational Database
Machine "Delta" [559]| A Relational Database
Machine (SIM-P or PSI) Outline of its Architecture and Hardware System [629]_nce
Machine (SIM-P or PSI) [630]_The Personal Sequential Inference
Machine—a Specification [383]_Abstract PROLOG
Machine and Data Base Machine as the Basic Architecture of Fifth Generation [600]
Machine and its Application to Partial Evaluation [336]_on of an Abstract PROLOG
Machine and its Architecture [328]_Application of Hash to Data Base
Machine Architecture [445]_al Inference Machine (Psi): its Design Philosophy and
Machine Architecture [446]_The Personal Sequential Machine (Psi): its Design and
Machine as the Basic Architecture of Fifth Generation Computer Systems [600]_ase
Machine Based on the Data Flow Mechanism [288]_PROLOG
Machine Intelligence in Qsar: Automatic Calculation of Molecular Property [146]
Machine Micromegas [313]_PROLOG II La
Machine Object Data Format [623]_Sequential Inference
Machine PROLOG [297]_Unique Features of LISP
Machine PROLOG [302]_Unique Features of LISP
Machine Prototype [249]_Some Aspects on a Logic
Machine to Implement PROLOG [21]_A Virtual
Machine Translation System Using the Active Dictionary [602]_An English-Japanese
Machine Translation [42]_A Future for
Machine [205]_A Preliminary Study for Designing PROLOG
Machine [258]_An Or-parallel Token
Machine [334]_An Abstract PROLOG
Machine [429]_A Relational Database Machine: First Step to Knowledge Base
Machine [430]_A Relational Data Base Machine: First Step to Knowledge Base
Machine [477]_A PROLOG Implementation of a Large System on a Small
Machine [540]_Lecture Notes on Bagel: a Systolic Concurrent PROLOG
Machine [567]_Logic Programming on an Ffp
Machine [569]_Conceptual Structures Information Processing in Mind and
Machine [599]_Implementing Parallel PROLOG on a Multiprocessor
Machine [670]_New PROLOG Runs on LISP
Machine [87]_Logic Programming on ZMOB: a Highly Parallel
Machine [89]_Control of Activities in the Or-parallel Token
Machine [90]_Control of Activities in the Or-parallel Token
Machine: a Parallel System for High-speed Logic Programming [610]_OG on the Dado
Machine: First Step to Knowledge Base Machine [429]_A Relational Database
Machine: First Step to Knowledge Base Machine [430]_A Relational Data Base
Machine: from Sequential to Parallel Logic [621]_Inference
Machine: from Sequential to Parallel [622]_Inference
Machine: PSI [627]_Outline of the Personal Sequential Inference
Machine: PSI [628]_Outline of the Personal Sequential Inference
Machine: PSI [715]_Design and Implementation of a Personal Sequential Inference
Machines are Proliferating [369]_Artificial Intelligence LISP and PROLOG
Magic [118]_PROLOG Without Man-machine Interface [601]_Intelligent
Management and Expert System Development in PROLOG [455]_agement, Knowledge Base
Management for Flexible Control Strategies [692]_Efficient PROLOG Memory
Management for Functional Languages [763]_Efficient Storage
Management of PROLOG Implementations [61]_The Memory
Management of PROLOG Implementations [63]_The Memory
Management System [230]_The Dlog Database
Management System [441]_Towards a Co-operative Data Base
Management, Knowledge Base Management and Expert System Development in PR [455]
Manipulation in the Fifth Generation Computer Systems [180]_Innovation in Symbol
Manipulation [72]_lectic Application of Multiple Rewrite Rule Sets in Algebraic
Manual CP/M Version [384]_Micro-PROLOG 3.0 Programmer's Reference
Manual de Utilizacao Da Base de Dados Facc [473]
Manual de Utilizacao Do Programa Para Interrogacao Em Portugues Das Pri_
Manual of Examples [312]_PROLOG II
Manual: Virginia Tech PROLOG [506]_The HC
Manuel D'exemples [316]_PROLOG II
Manuel D'utilisation [636]_PROLOG II
Manuel de Reference et D'utilisation [517]_PROLOG
Manuel de Reference et Modele Theorique [120]_PROLOG II
Many-sorted Logic [375]_Knowledge Representation of Design in
Many-storied Dwelling Houses [373]_Application of PROLOG in Designing
Mapping [250]_Program Transformation by Data Structure
Mapping [251]_Program Transformation by Data Structure
Mapping [252]_Program Transformation by Data Structure
Maps Simple Lists Onto D-lists [253]
Program Transformation by a Function That Mastermind Logically [544]
Playing Mastermind [647]
Relational Programming Illustrated by a Program for the Game of Mathematical Logic T-PROLOG [215]
A Modelling Tool Based on Mathematical Logic [374]
Using Programming Language PROLOG—PROLOG Based Upon
Mathematical Reasoning [69]
The Computer Modelling of
Mazes [194]
ion of Logic Programs from Formal Specifications—Dijkstra’s Three
MAXFLOW Experience [266]
menting Parallel Algorithms in Concurrent PROLOG: the
MAXFLOW Experience [267]
menting Parallel Algorithms in Concurrent PROLOG: the
Maze of the Fifth Generation [365]
In the McDermott on PROLOG: a Rejoinder [646]
Mean of LISP, PROLOG, and Smalltalk [595]
Ta—a Harmonic
Meaning of Logical Programs [382]
The Measure for Control of Design Errors in Logic-based CAD Programs [318]
Complexity
Performance
Mechanics Problems [391]
ontrolling Inference in the Semantic Interpretations of
Mechanism for Generating Relational Algebra Queries [712]
An Enhanced Inference
Mechanism [288]
PROLOG Machine Based on the Data Flow
Mechanism [701]
G + Data Flow: Arguments for Combining PROLOG with a Data Driven
New Architecture for Knowledge Base
Mechanisms [200]
Problem Solving and Inference
Mechanisms [581]
Knowledge Base
Mechanisms [626]
New Architectures for Inference
Mechanisms [709]
PROLOG and Data-flow Computation
Mechanization of an Oracle in a Debugging System [158]
Medical Decision Aid: Logic Bases of the System Sphinx [294]
Medical Expert System [425]
A Qualitative Model of the Heart for a
Memory Management for Flexible Control Strategies [692]
Efficient PROLOG
Memory Management of PROLOG Implementations [61]
The Memory Management of PROLOG Implementations [63]
The Memory Multiprocessors [41]
Parallel PROLOG Using Stack Segments on Shared
Merge Operators: Their Specification and Implementation in Concurrent PROL [543]
Merge Operators: Their Specification and Implementation in Concurrent PROL [554]
Meta-language Programming to Fault Finding in Logic Circuits [174]
Application of Meta-level Inference for Selective Application of Multiple Rewrite Rule Se [72]
Meta-level Inference in Algebra [71]
Meta-shifting [598]
Program Transformation through Metacntrol of Process Synchronisation in T-PROLOG [208]
Metalanguage for Logic Programming [223]
A Control
Metalanguage in Logic Programming [50]
Amalagating Language and
Metalanguage in Logic Programming [51]
Amalagating Language and
Metalanguage in Logic Programming [52]
Amalagating Language and
Metalevel Control for Logic Programs [221]
Metalevel Control for Logic Programs [222]
Metalog Problem-solving System an Informal Presentation [154]
The Metamorphosis Grammars [125]
Method for Logic Databases [402]
A Knowledge Assimilation
Method for Logic Databases [403]
A Knowledge Assimilation
Method for Process Logic [577]
A Decision
Method for Producing Efficient PROLOG Programs [586]
ed Language Programming—a Method of Adapting Resolution for Logic Programming [18]
ciple—an Alternative(Method [587]
LDM—a Logic Based Software Development
Method) [330]
PROLOG for Programmers (an Outline of a Teaching
Methodology for Implementation of a Knowledge Acquisition System [326]
A Methodology for Implementation of a Knowledge Acquisition System [327]
A Methodology in PROLOG Programming [376]
A Design
Methodology of Logic Programming [545]
Methodology [163]
Logic Programming
Methods Based One Extended Dempster & Shafer’s Theory for Problems of Unc [287]
Methods [209]
A Discrete Simulation System Based on Artificial Intelligence
Micro-PROLOG 3 [364]
CP/M Version [384]
Micro-PROLOG Primer [99]
Micro-PROLOG [167]
Beginning
Microcomputers [671]
PROLOG Compiler Due for Micromegas [313]
PROLOG II La Machine
Microprogram Assembler [280]
Customizable
Migration Decision-making in a Less Developed Country [505]
PROLOG Simulation of Mind and Machine [569]
Conceptual Structures Information Processing in
Mister P and Mister S Problem in Pure PROLOG [121]
A Solution to the Mister S Problem in Pure PROLOG [121]
A Solution to the Mister P and
Mixed Language Programming—a Method for Producing Efficient PROLOG Prog [586]
Modal Extension of PROLOG Language [231]
Dural: a Modal Extension of PROLOG [232]
Dural: a Modal Inference System [552]
The Mode Declaration for PROLOG Programs [392]
The Automatic Generation of Mode Declaration for PROLOG Programs [393]
The Automatic Generation of
Model Building in Regression Analysis [145] A Logic-based Expert System for
Model for Or-parallel Execution of Logic Programs
[91] A Formal
Model for Parallel Interpretation of Logic Programs
[129] The And/or Process
Model of Logic Programs [633] A Parallel Execution
Model of the Heart for a Medical Expert System [425] A
Qualitative
Model to Efficient Compilation of Patterns [166]
From Abstract
Model [119] PROLOG II Reference Manual and Theoretical
Model [702] A Parallel PROLOG: the Construction of a
Data Driven
Modele Theorique [120] PROLOG II Manuel de Référence et
Modeling of Human Cognition [238] Deductive
Modeling of Human-computer Interactions in a Friendly
Interface [519]
Modelling of Mathematical Reasoning [69] The
Computer
Modelling Tool Based on Mathematical Logic T-
PROLOG [215] A
Modelling [426] in Automatic Synthesis of Expert
Knowledge through Qualitative
Models [127] Nondeterministic Languages Used for the
Definition of Data
Models [286] Order: its Implications to Programming
Languages and Computational
MODULA-2, Smalltalk, PROLOG Computer Lan-
guages of the Future [12] TH, APL, LISP,
Modularization and Abstraction in Logic Programming
[198]
Modularization and Abstraction in Logic Programming
[199]
Module Concepts for PROLOG [585]
Module Development Based on Program Transformation
and Automatic Generali [566]
Molecular Properties and Automatic Interpretation of
Quantitative Stuctru [146]
Monadic Functions [16] me Aspects of the Static
Semantics of Logic Programs with
Motivation, Design and Implementation [513] LOGLISP:
Motivation, Design, and Implementation [514] LOGLISP:
MPROLOG Environment: Today and Tomorrow
[340] The
MPROLOG System [37] The
MPROLOG System [38] The
MPROLOG User’s Manual -Draft- [584]
Mu-PROLOG [432] An Introduction to
Multiple Rewrite Rule Sets in Algebraic Manipulation
[72] lective Application of
Multiprocessor Machine [599] Implementing Parallel
PROLOG on a
Multiprocessors [41] Parallel PROLOG Using Stack
Segments on Shared Memory
My Experiences with PROLOG [70]
Mycin: the Expert System and its Implementation in
LOGLISP [440]
Name Semantics of a Clause Language with Functions
[32] The Call by
Narrative Schema in Procedural Logic [565] A
Nated, a Derivation Editor [172]
Natural Deduction (Corrected Copy) [260] Evaluation of
Logic Programs Based on
Natural Deduction System [256] Logic Programming
Based on a
Natural Deduction System [606] A Programming Lan-
guage Based on a
Natural Deduction System [607] A Programming Lan-
guage Based on a
Natural Deduction [259] Evaluation of Logic Programs
Based on
Natural Language Access to PROLOG Database
Systems [529]
Natural Language Analysis [459] Logic for
Natural Language Consultable Data Bases [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Natural Language Sentences [493] On a Semantic Repre-
sentation of
Natural Language [122] An Interesting Subset of
Natural Language [456] A Dialogue in
Natural Language [489] An Interesting Subset of
Natural Language [143] Logical
Design of Deductive
Natural Language Interface [185] A Kernel for a General
Natural Language Processing [603] PROLOG and
Natural Language Queries [690] Efficient Easily
Adaptable System for Interpreting
Natural Language Semantics: a Logic Programming Ap-
proach [499]
Predicate Logic Programming Language [398]A Set—Oriented
Predicate Logic Programs [686]Implementing PROLOG —Compiling
Predicate Logic Programs [687]Implementing PROLOG —Compiling
Predicate Logic [419]A Formal Definition of ASPLE Using
Predicate Logic: a Calculus for Deriving Programs [92]
Predicate Logic: a Survey of the Formalism and its Implementation Techniq [631]
Predicate [596]Semantics of a Logic Programming Language with a Reducibility
Predicate-logic Languages and Relational Data-bases [83]Interfacing
Predicting Drug Interactions [147]Logic-based Program System for
Preliminary Kernal Language for 5th Generation Computer [81]aimed PROLOG—as a
Preliminary Kernal Language of Fifth Generation Computers [86]ESP as a
Preliminary Note— [453]Functional Interpretation of a Logical Sentence -a
Preliminary Research on Data Flow Machine and Data Base Machine as the Ba [600]
Preliminary Study for Designing PROLOG Machine [205]A
Prescriptive to Descriptive Programming a Way Ahead for Caad [583]
Primentive for the Control of Logic Programs [298]A Primer [99]A Micro PROLOG
Primitive for the Control of Logic Programs [296]A Principle—an Alternative Method of Adapting Resolution for Logic Program [18]
Priorida Des Em Ciencia E Technologia [478] a Para t Interrogacao Em Portugues Das
Prism—a Parallel Inference System for Problem Solving [320]
Program Describing Entitlement to Supplementary Benefit [242] listing of a PROLOG
Program Development [341] Programming Support Environment for PROLOG
Program Find Revisited [561] Hoare’s
Program for the ‘S-P Problem’ [498] A PROLOG
Program for the Game of Mastermind [647] Relational Programming Illustrated by a
Program in PROLOG [428] An Ordered Linear Resolution Theorem Proving
Program Specification of Numerical Integration [108] Logic
Program System for Predicting Drug Interactions [147] Logical-
Program Transformation and Automatic Generation of the Input-output Relation [566]
Program Transformation by a Function That Maps Simple Lists Onto D-lists [253]
Program Transformation by Data Structure Mapping [250]
Program Transformation by Data Structure Mapping [251]
Program Transformation by Data Structure Mapping [252]
Program Transformation through Meta-shifting [598]
INDEX

PROLOG [428] An Ordered Linear Resolution Theorem
PROLOG [431] A Polymorphic Type System for
PROLOG [455] Agent, Knowledge Base Management and Expert System Development in
PROLOG [457] A Solution of a CAD Problem in
PROLOG [458] A LISP-machine to Implement
PROLOG [476] User's Guide to Decsystem-10
PROLOG [479] Pure LISP in Pure
PROLOG [495] The Occur-check Problem in
PROLOG [506] An Implementation of
PROLOG [512] LOGLISP: an Alternative to
PROLOG [518] Contextual Grammars in
PROLOG [523] An Implementation of
PROLOG [531] Merge Operators: Their Specification and Implementation in Concurrent
PROLOG [538] Distributed Programming in Concurrent
PROLOG [539] Systems Programming in Concurrent
PROLOG [543] Merge Operators: Their Specification and Implementation in Concurrent
PROLOG [554] Merge Operators: Their Specification and Implementation in Concurrent
PROLOG [556] Object Oriented Programming in Concurrent
PROLOG [557] Object Oriented Programming in Concurrent
PROLOG [560] Deterministic and Bottom-up Parsing in
PROLOG [568] A Knowledge Representation System in
PROLOG [585] Module Concepts for
PROLOG [581] Some Reflections on Implementation Issues of
PROLOG [592] Interprocess Communication in Concurrent
PROLOG [593] Interprocess Communication in Concurrent
PROLOG [605] Sova—an Integrated Question-answering System Based on ATN and
PROLOG [625] Logic Simulation in
PROLOG [632] Logic Circuit Synthesis Using
PROLOG [642] AVL-tree Insertion: a Benchmark Program Biased Towards
PROLOG [653] A Proposal for an Imperative Complement to
PROLOG [667] Data Bases, Expert Systems, and
PROLOG [709] My Experiences with
PROLOG [716] Object-oriented Programming in
PROLOG [731] Utility Procedures in
PROLOG [888] A Data Base Support System for
PROLOG, a 'Fifth-generation' Language [112] Introduction to
PROLOG, and Act 1 [304] age Based Upon Unification Which Unifies (Much Of) LISP,
PROLOG, and Act 1 [305] age Based Upon Unification Which Unifies (Much Of) LISP,
PROLOG, and Act 1 [307] age Based Upon Unification Which Unifies (Much Of) LISP,
PROLOG, and Smalltalk [595] Tao—a Harmonic Mean of LISP,
PROLOG/ex1, an Inference Engine Which Explains Both Yes and No Answers [666]
PROLOG/KR: the Language Features (a Brief Survey) [437]

PROLOG/KR [434] Knowledge Representation in
PROLOG/KR [439] Data Abstraction in
PROLOG/LISP Type Language for Logic Programming [523] Quine: a
PROLOG/LISP Type Language for Logic Programming [524] Quine: a
PROLOG: a Language for Implementing Expert Systems [101]
PROLOG: are they Needed? [677] Higher-order Extensions to
PROLOG: Software Specification by Grammar [699] Beyond
PROLOG: the Construction of a Data Driven Model [702] A Parallel
PROLOG: the MAXFLOW Experience [266] Lementing Parallel Algorithms in Concurrent
PROLOG: the MAXFLOW Experience [267] Lementing Parallel Algorithms in Concurrent
Prover 11141 Geom: a PROLOG Geometry Theorem
Prover 11141 Geom: a PROLOG Technology Theorem
INDEX

Provers [55] The Relation Between Semantic Tableaux and Resolution Theorem
Provers [56] The Relation Between Semantic Tableaux and Resolution Theorem
Proving Based Machine Intelligence in Qsar: Automatic Calculation of Mole [146]
Proving Program in PROLOG [428] An Ordered Linear Resolution Theorem
PSI [627] Outline of the Personal Sequential Inference Machine:
PSI [628] Outline of the Personal Sequential Inference Machine:
PSI [715] Design and Implementation of a Personal Sequential Inference Machine:
PSI: Outline of its Architecture and Hardware System [629] The Personal Sequential Inference Machine (SIM-P or PSI):
PSI [630] The Personal Sequential Inference Machine (SIM-P or PSI):
PSI): its Design and Machine Architecture (446) The Personal Sequential Machine (PSI):
Qlog—the Programming Environment for PROLOG in LISP [335]
Qlog—the Software for PROLOG and Logic Programming [337]
Qlog [79] Re)implementing PROLOG in LISP or YAQ — Yet Another
Queries Expressed in Logic [679] t Processing of Interactive Relational Database
Queries [690] Efficient Easily Adaptable System for Interpreting Natural Language Queries [712] An Enhanced Inference Mechanism for Generating Relational Algebra
INDEX

Relational Algebra Queries [712] An Enhanced Inference Mechanism for Generating
Relational Data Base Machine: First Step to Knowledge Base Machine [430]
An Experimental
Relational Data Bases "à La Carte" [189]
Relational Data Bases "à La Carte" [472]
Relational Data Bases [322] Deductive Question-answering on
Relational Data Bases [654] Deductive Information Retrieval on Virtual
Relational Data Bases [84] Deduce 2: Further Investigations of Deduction in
Relational Database Machine "Delta" (Translated from [fps]) [311]
Relational Database Machine "Delta" [310]
Relational Database Machine "Delta" [559]
Relational Database Machine: First Step to Knowledge Base Machine [429]
Relational Database Queries Expessed in Logic [678] Processing of Interactive
Relational Database Queries Expessed in Logic [679] Processing of Interactive
Relational Databases for Fifth Generation Computer Systems [353] PROLOG and
Relational Databases [48] Logic Programming &
Relational Dataflow System [130]
Relational Equations, Grammars, and Programs [650]
Relational Production Systems and Logic Programs [414]
Relational Programming Illustrated by a Program for the Game of Mastermind [647]
Relational Programming Language for Parallel Programming [100]
Relations—[704] in PROLOG—Toward a Formal System for Representing Grammatical
Relationships [146] Automatic Interpretation of Quantitative Structure-activity
Relative Assertions [413] Transporting Values via
Release 1.0 Draft [309] L.M.-PROLOG User Manual Reliable and Readable PROLOG Programs [60]
Adding Redundancy to Obtain More
Remarks on Coroutines in PROLOG [329]
Report of FGCS Project’s Research Activities, 1982 [279]
Report [547] Japan’s Fifth Generation Computer Project—a Trip
Report [674] Fifth Generation Research in Japan—a Trip
Representation in an Efficient Deductive Inference System [373]
Representation Language [449] cate Logic Involving Data Structure as a Knowledge
Representation of a Concurrent Algorithm [275] Logic
Representation of Design in Many-sorted Logic [375] Knowledge
Representation of Knowledge [136] Logic Programming as a
Representation of Natural Language Sentences [493] On a Semantic
Representation of Real Numbers [241]
Unr: Universal
Representations for Programs [656] Verification Conditions as
Representing Grammatical Relations—[704] in PROLOG—Toward a Formal System for
Representing the Law as Logic Programs [537] Prospects
for
Research Activities in Knowledge Information Processing System [525] Fujitsu
Research and Development Plans for Fifth Generation Computer Systems [283]
Research and Development Themes and Plans of the Fifth Generation Computer [282]
Research [370] Intelligence Commercial Products Begin to Emerge from Decades of
Resolution for Logic Programming [18] Clople—an Alternative Method of Adapting
Resolution Logic [652] Programming with Resolution Logic [657] Programming with
Resolution Theorem Provers [55] The Relation Between Semantic Tableaux and
Resolution Theorem Provers [56] The Relation Between Semantic Tableaux and
Resolution Theorem Proving Program in PROLOG [428] An Ordered Linear
Resources Evaluation through Natural Language [489]
ert-system for Environmental
Resources Evaluation through Natural Language [490]
ert-system for Environmental
Restricted PROLOG Programs [25] n Inductionless Technique to Prove Properties of
Restriction Grammar in PROLOG [272]
Retrieval for Dynamic Files [363] Partial-match
Retrieval on Virtual Relational Data Bases [654] Deductive Information
Retrieval [648] Computation and Deductive Information
Retrieval [651] Computation and Deductive Information
Revision of Top-down Logical Reasoning through Intelligent Backtracking [67]
Revision of Top-down Logical Reasoning through Intelligent Backtracking [68]
Revisited [561] Hoare’s Program Find
Revisited [676] The “S-P Problem”
Revolution in Education [168]
Rewrite Rule Sets in Algebraic Manipulation [72] lective Application of Multiple
Risks in Data Processing Systems [81] Using PROLOG to Assess Security


Shape [115]Fifth-generation Hardware Takes
Shapeup [338]Knowledge Information Processing Language:
Shared Memory Multiprocessors [41]Parallel PROLOG
Using Stack Segments on
Sharing in the Implementation of a PROLOG Interpreter
[390] native to Structure
Shell Project [229] Intelligent UNIX
Short Cut to More Informative Answers [139] A
 Sidetracking in Horn Clause Programs—Implementation [484]ent Backtracking and
 Sidetracking in Horn Clause Programs—Implementation [487]ent Backtracking and
Sidetracking in Horn Clause Programs—the Theory [488] lligent Backtracking and
SIM Operating System [264] Basic Constructs of the
SIM Operating System [265] Basic Constructs of the
SIM-P or PSI Outline of its Architecture and Hardware
System [629] nce Machine (SIM-P or PSI) [630] The Personal Sequential Inference Machine
Simple Lists Onto D-lists [253] Program Transformation by a Function That Maps
Simulating Coreouting for the 8 Queens Problem [216]
Simulation in PROLOG [276] Digital Logic
Simulation in PROLOG [625] Logic
Simulation of Migration Decision-making in a Less Developed Country [505] PROLOG
Simulation on PROLOG Basis [207] System
Simulation System Based on Artificial Intelligence Methods [209] A Discrete
Simulation System [210] T-PROLOG Very High Level
Simulation System [211] T-PROLOG: a Very High Level
Simulation System [213] T-PROLOG Very High Level
Size of Turner’s Translation to Combinator Programs [269] Average
Slash [640] Warren’s Doctrine on the
Small and Efficient PROLOG Interpreter
[444] Poo log—a Small Interpreter for Distributed Logic [407] A
Small Machine [477] A PROLOG Implementation of a Large System on a
Smalltalk [595] Tao—a Harmonic Mean of LISP, PROLOG, and
Smalltalk, PROLOG Computer Languages of the Future
[12] TH, APL, LISP, MODULA-2,
Snipslog a “Higher Order” Logic Programming Language [558]
Social Acceptability of FGCS [534] New Applications: User and
Social Needs and its Impact [319] is Required of the Fifth Generation Computer—
Software Development Method [587] LDM—a Logic Based
Software Engineering [354] Logic Programming—What Does It Bring to the
Software for Deductive Reasoning [255] Development of Software for PROLOG and Logic Programming
[337] Qlog—the Software Specification by Grammar [699] Beyond PROLOG:
Software Technology in the Fifth Generation [358] The Role of Systems and
Solution of a CAD Problem in PROLOG [457] A
Solution to the Mister P and Mister S Problem in Pure PROLOG [121] A
Solutions for the Negation Problem [141] Two Solutions [480] All
Solve It with PROLOG [113] How to Solve [412] A Forward Chaining Problem
Solving and Inference Mechanisms [200] Problem
Solving Combinatorial Search Problems by Intelligent Backtracking [62]—
Solving Combinatorial Search Problems by Intelligent Backtracking [65]
Solving on ZMOB [400] Parallel Problem
Solving with PROLOG [197] Problem
Solving [320] Prism—a Parallel Inference System for Problem
Solving [350] Logic for Problem
Sova—an Integrated Question-answering System Based on ATN and PROLOG [605]
Spanish into Logic through Logic [140] Translating
Specification and Implementation in Concurrent PROLOG [543] erge Operators: Their
Specification and Implementation in Concurrent PROLOG [554] erge Operators: Their
Specification as a Design Tool [148] Runnable
Specification as a Design Tool [149] Runnable
Specification in Logic [700] Deriving Different Unification Algorithms from a
Specification of an Abstract PROLOG Machine and its Application to Partia [336]
Specification of Numerical Integration [108] Logic Program
Specification Support System [177] LDM: a Program
Specification Support System [178] LDM—a Program
Specifications—Dijkstra’s Three Maxes [194] ion of Logic Programs from Formal
Specifying Concurrency [406] A Horn Clause-like Logic for
Sphinx [294] Medical Decision Aid: Logic Bases of the System
Stack Segments on Shared Memory Multiprocessors [41] Parallel PROLOG Using
Start of the FGCS Project*: Discussion [533] “at the
Started with IC-PROLOG [385] Getting
Static Semantics of Logic Programs with Monadic Functions [16] of Aspects of the
Stepwise Development of Operational and Denotational Semantics for PROLOG [292]
Storage Management for Functional Languages [263] Efficient
Stock Assistant [668] KB01: a Knowledge Based Garden
Store Human Expertise [175] lligence Practical Systems
Strategies [692] Efficient PROLOG Memory Management for Flexible Control
Strategy for Integrating Functional and Logic Programming [579] s and Evaluation
Stream-based Execution of Logic Programming [359]
Structure as a Knowledge Representation Language [449] cate Logic Involving Data
System Project [422] Overview to the Fifth Generation Computer
System Project [614] Japan’s Fifth Generation Computer
System Simulation on PROLOG Basis [207]
System Sphinx [294] Medical Decision Aid: Logic Bases of the
System with Quasi-natural Language Information Input [574] A ta-independent Expert
System [130] A Relational Dataflow
System [158] Mechanization of an Oracle in a Debugging System
System [178] LDM—a Program Specification Support
System [17] Xp’s: an Extended Or-parallel PROLOG
System [182] A PROLOG Interpreter for the UNIX Operating System
System [196] Rubik’s Cube Production
System [210] T-PROLOG Very High Level Simulation System
System [211] T-PROLOG: a Very High Level Simulation System
System [213] T-PROLOG Very High Level Simulation System
System [230] The Dlog Database Management System
System [256] Logic Programming Based on a Natural Deduction System [264] Basic Constructs of the SIM Operating System
System [265] Basic Constructs of the SIM Operating System
System [325] Culture Creation and a New Generation Computer System
System [326] A Methodology for Implementation of a Knowledge Acquisition System
System [327] A Methodology for Implementation of a Knowledge Acquisition System
System [525] Fujitsu Research Activities in Knowledge Information Processing System
System [606] A Programming Language Based on a Natural Deduction System
T-PROLOG: a Very High Level Simulation System [211]
Teaching Logic as a Computer Language in Schools [169]
Transformation by a Function That Maps Simple Lists Onto D-lists [253]
Transformation by Data Structure Mapping [250]Program
Transformation by Data Structure Mapping [251]Program
Transformation by Data Structure Mapping [252]Program
Transformation of Logic Programs into Functional Programs [501]
Transformation System for Logic Programs Which Preserves Equivalence [597]A
Transformation through Meta-shifting [598]Program
Transition Networks [463] Survey of the Formalism and a Comparison with Augmented
Translated from Ipsj [311] A Relational Database Machine “Delta” ( 
Translating Spanish into Logic Through Logic [140]
Translation Grammars [1] Definite Clause
Translation to Combinator Programs [269] Average Size of Turner’s
Translation [42] A Future for Machine
Transporting Values via Relative Assertions [413]
Trees [123] PROLOG and Infinite
Trees [124] PROLOG and Infinite
Trees [427] A Unification Algorithm for Infinite
Trees [471] Backtracking Intelligently in AND/OR
Trends in Logic Grammars [137] Current
Trip Report [547] Japan’s Fifth Generation Computer Project—a
Trip Report [674] Fifth Generation Research in Japan—a
Turing Institute a Prospectus [27] The
Turn is Intended to be Given to Itself Together with the
PROLOG to Produce [300]
Turner’s Translation to Combinator Programs [269] Average Size of
Tutorial [311] PROLOG
Two Complement Notions for the “Negation as Failure” Rule in Logic Programming [40]
Two Solutions for the Negation Problem [141]
Two-level Grammars [367] Comparison of the Logic Programming Language PROLOG with
Type Language for Logic Programming [523] Quete: a PROLOG/LISP
Type Language for Logic Programming [524] Quete: a PROLOG/LISP
Type System for PROLOG [431] A Polymorphic Types in PROLOG [40] Towards a Theory of
Ultimate Computer Language [183] PROLOG—a Step Toward the
Ultimate PROLOG [126] Last Steps Towards an Uma Logica Para Processos Distribuidos [409]
Uncertainties: a Tool for Implementing Rule-based Systems [542] ic Problems with
Uncertainty/fuzziness [287] Extended Dempster & Shafer’s Theory for Problems of
Undecidability of Two Completeness Notions for the “Negation as Failure” Rule [40]
Understanding the Control Flow of PROLOG Programs [74]