Using Logic Rules for Concept Refinement Learning in First Order Logic

Zhenguo Shi [#1,2], Zongtian Liu [2], Jianping Chen [1]

[1] *School of Computer Science and Technology,Nantong University*
Nantong 226019,P.R.China
[1] chinaemail@sohu.com

[2] *School of Computer Engineering and Science,Shanghai University*
Shanghai 200072,P.R.China
[2] ztliu@mail.shu.edu.cn

***Abstract*—In this paper,it has been explored that the use of logic rules as key element in concept refinement Learning.A logic rule is a formal grammar in logic for expressing formation rules of a formal language.First order logic in Inductive Logic Programming(ILP) and programming language in Genetic Programming(GP) are formal languages,the logic rule is available to express syntax and semantics of them.Concept refinement learning including inductive concept learning by employing ILP and evolutionary concept learning by employing GP.A framework is presented that combining ILP and GP using logic rules for concept refinement learning in first order logic.The viability of our approach is illustrated by comparing the performance of our learner with that of other concept learners such as Progol,CfgGP,GGP on a variety of target concepts.We conclude with some observations about the merits of our approach and about possible extensions.**

***Keywords*—logic rule;refinement concept learning;inductive concept learning;evolutionary concept learning;inductive logic programming;genetic programming**

I. INTRODUCTION

A formal language(such as mathematical language in FCA[1],first order logic in ILP[2] and programming language in GP[3]) is often defined by means of a formal grammar also called its formation rules.The logic rule that will be described in this paper is a formal grammar for expressing formation rules in logic.A clear and powerful formalism for describing languages both natural and artificial follows from a method for expressing grammars in logic due to Colmerauer[4] and Kowalski[5].Logic rules similar to Definite Clause Grammars[6] and Logic Grammars[7] are the generalizations of Context Free Grammars[3].Their expressivenesses are much more powerful than that of Context Free Grammars,but equally amenable to efficient execution.Logic rules differ from Context Free Grammars in that their grammar symbols whether terminal or non-terminal may include arguments,and the arguments can be any term in the grammars.In this paper the notion of concept refinement learning is meant to include inductive concept learning[1][2] by employing FCA[11],ILP[12] and evolutionary concept learning[3] by employing GP[13] using logic rules.Concept refinement learning can be seen as a search process[14],starting from an initial hypothesis,what is done is searching the space of the possible hypotheses for one that fits the given set of training examples.In order to search hypothese space,when a problem can be represented by a fixed number of attributes,we introduce FCA with Galios concept lattice and trie searching in different granularity[8][9][10],for first order hypothese,we are interested in ILP with Horn clause lattice and refined searching[2].Due to the huge search space has to deal with,learning first order hypotheses is a hard task,so ILP systems try to overcome this problem by using specific search strategies,like the top-down[15] and the down-top[12].However,the greedy search strategies are adopted for reducing the computational effort,render techniques based on this approach often incapable of escaping from local optima.An alternative search approach is offered by GA[16] that have proved to be successful in solving comparatively hard optimization problems.GP[13] that extends traditional GA is a biologically inspired,domain independent method that automatically creates a computer program from a high-level statement of a problem's requirements.

So from views of formal grammars,the search space of ILP is determined by the syntax of Horn clauses and the background knowledge,and this space of GP is determined by the syntax of symbolic expressions and the sets of terminals and functions.those motivate the aim of this paper is using logic rules for concept refinement learning by combining ILP and GP.Our approach starts with an initial population of concepts generated randomly induced by other learner[1][2][12][15] and population evolution by GP[13] which individuals can be refined during their lifetime.

The paper is organized as follows.The next section presents details of logic rules for concept refinement learning including the notions about logic rules based derivation trees and Genetic Programming.Section three describes a concept refinement learning algorithm.In section fourth we delineate the experimentation and some evaluations of our approach.The last section is the conclusion.

II. LOGIC RULES

A. *Formal language*

A formal language is a set of words,i.e. finite strings of letters,symbols,or tokens.The set from which these letters are taken is called the alphabet over which the language is defined.

978-1-4244-6439-5/10/$26.00 ©2010 IEEE

Definition1:A formal language L over an alphabet A is just a subset of A^* (the asterisk represents the Kleene star operation) that is a set of strings(including empty string) over that alphabet.

The alphabet of first order logic as a formal language example is defined as follows.

Definition2:The terms and formulas of first order logic are strings of symbols which together form the alphabet of the language.It is common to divide into logical symbols which always have the same meaning,and non-logical symbols whose meaning varies by interpretation.The logical symbols usually include:the quantifier symbols $S_q=\{\exists,\forall\}$,the connective symbols $Scon=\{\wedge,\vee,\neg,\rightarrow,\leftrightarrow\}$ for conjunction,disjunction,negation,implication,biconditional ,an infinite set of variables by $S_v=\{V_1,V_2,V_3,...\}$,brackets and other punctuation symbols $S_{pun}=\{\{,\},(,),...\}$ .The non-logical symbols represent predicates,functions and constants on the universe,there is a collection of n-ary predicate symbols $Sp=\{p_1,p_2,p_3,...\}$,there are infinitely many n-ary function symbols $Sf=\{f_1,f_2,f_3,...\}$,an infinite set of constant denoted by $S_c=\{c_1,c_2,c_3,...\}$ .So the alphabet of first order logic is often denoted by $A=S_q \cup S_{con} \cup S_v \cup S_{pun} \cup S_p \cup S_f \cup S_c$.

B. Logic rules

A formal language is often defined by means of a formal grammar also called its formation rules.A logic rule is a formal grammar for expressing formation rules of a formal language in logic, defined as follows.

Definition3:A logic rule R_l is defined by a quintuplet $R_l=(S_U,S_T,S_N,S_S,S_P)$. S_U is the set of terms that often called a universe.Any variable $V_i \in S_v$ and $V_i \in S_U$,any expression $f_i(t_1,...,t_n) \in S_U$ where each argument $t_i \in S_U$ and $f_i(t_1,...,t_n) \in S_f$ is a function symbol of valence n ,any constant $c_i \in S_c$ and $c_i \in S_U$ that is simply a 0-arity function. S_T is a set of terminal symbols with $S_T \subset S_U$, S_N is a set of non-terminal symbols with $S_N \subset S_U$ and $S_T \cap S_N=\varnothing$. S_S is a start symbol with $S_S \in S_N$. S_P is the set of productions.

To describe how grammars for Definition3 can be expressed in logic,we begin by considering context-free grammars example.

For example,a context-free grammar[3] G_C is a four-tuple: $G_C=(S_N,S_T,S_S,S_P)$, S_N is a finite set of non-terminal characters or variables,they represent different types of phrase or clause in the sentence,they sometimes called syntactic categories,each variable represents a language. S_T is a finite set of terminals,disjoint from S_N ,which make up the actual content of the sentence,the set of terminals is the alphabet of the language defined by the grammar. S_S is the start symbol used to represent the whole sentence or program,it must be an element of S_N . S_P is a set of productions or rewrite rules. $G_C=(S_N=\{S,M\},S_T=\{+,-,*,/,\Re\},S_S=S,$ $S_P=\{S\rightarrow M,M\rightarrow +MM|-MM|*MM|/MM\}|\Re\})$, G_C defines a language for generating all possible mathematical expressions by syntactic variable S,M using the operators $+,-,*,/$ and a set of random real numbers $\Re$.

According to Definition3,logic rules are a natural extension of context-free grammars.A logic rule differs from a CFG in that the grammar symbols whether terminal or nonterminal may include arguments.The arguments can be any term in the grammar that be used to enforce context dependency and to represent the semantics of the concept.A term as data objects of the language is either a variable,a function or a constant.A variable is represented by a capital letter V and digits.A function is a grammar symbol followed by a bracketed n-tuple of terms,a constant is simply a 0-arity function.A compound term comprises a functor and a sequence of one or more terms.The terminal symbols which are enclosed in square brackets,correspond to the set of words of the language specified by list,list either is the atom which has two arguments which are respectively the head and tail of the list representing the empty list or compound term with functor.The non-terminal symbols are expressions similar to functions in Lisp and literals in Prolog.The special non-terminal start symbol corresponds to all concepts of the language.The left-hand side of a logic rule is non-terminal symbols allowed to be compound terms,the right-hand side of a logic rule is a sequence of one or more items separated by commas,each item is either a non-terminal symbol or a sequence of terminal symbols,may contain logic goals and grammar symbols.A goal is a special kind of term,terminals and goals in the right-hand side of a logic rule will be referred to as procedure call.Commas denote concatenation and each rule ends with a full stop.

For example, a formal grammar with arguments and goals can be represented by logic rules as follows[6].

$$G_R=(S_U=\{V_1,V_2,noun,rootform,is\_noun,Scp\},S_N=\{Ss,noun(V_1),$$
$$\{rootform(V_2,V_1)\},is\_noun(V_1)\},S_T=\{V_1,V_2\},S_S=Scp,S_P=\{noun(V_1)\rightarrow[V_2],$$
$$\{rootform(V_2,V_1)\},is\_noun(V_1)\})$$

As a formal language example,key notions of ILP[2][12][15] are defined as follows.

Definition4:ILP framework is a logic system.**Given:**1)training example set E ,consisting of true $E+$ and false $E-$ ground facts of an unknown predicate p ;2)description language L ,specifying syntactic restrictions on the definition of predicate p ;3)background knowledge B , defining predicate q_i (other than p),which may be used in the definition of p and provides additional information about the arguments in predicate p ;**Find:**A definition of H for p , expressed in L ,so that H is complete and consistent with respect to example E and background knowledge B .

For example,the Quinlan's network reachability problem[15] ,the problem involves a directional network depicted as follows,we extend it to a fuzzy network[1][9] in Figure1.

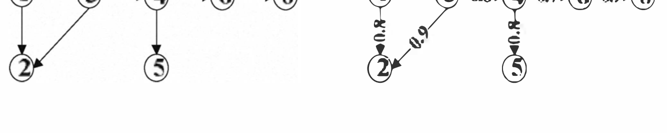

Fig. 1. Exmple for Quinlan's network(the left) and fuzzy network(the right).

Consider the fuzzy network in Figure1,this network represents the fuzzy relation $linkTo(V_1,V_2)$ that denotes node V_1 is directly linked to node V_2 with truth value t , where $t\in[0,1]$. In this network,the edges represent the instances of the $linkTo(V_1,V_2)$ relation and the number on an edge is the truth value of the corresponding instance.For example,the truth value of the instance $linkTo(3,4)$ is 0.8.A fuzzy relation is associated with a n-ary fuzzy predicate and can be described extensionally as a set of ordered pairs.Thus the relation $linkTo(V_1,V_2)$ can be represented as:

$linkTo(V_1,V_2) = \{(<0,1>,0.9),(<0,3>,0.8),(<1,2>,0.8),$
$(<3,2>,0.9),(<3,4>,0.8),(<4,5>,0.8),(<4,6>,0.7),$
$(<6,8>,0.9),(<7,6>,0.9),(<7,8>,0.9)\}$

The first element of an ordered pair is a n-tuple of constants that satisfies the associated fuzzy predicate.The second element of the ordered pair is the corresponding truth value.Other fuzzy relations can be obtained from the fuzzy network.One of them is $canReach(V_1,V_2)$ which is represented explicitly as:

$canReach(V_1,V_2) = \{(<0,1>,0.9),(<0,2>,0.7),(<0,3>,0.8),$
$(<0,4>,0.7),(<0,5>,0.6),(<0,6>,0.5),(<0,8>,0.5),(<1,2>,0.8),$
$(<3,2>,0.9),(<3,4>,0.8),(<3,5>,0.7),(<3,6>,0.6),(<3,8>,0.5),$
$(<4,5>,0.8),(<4,6>,0.7),(<4,8>,0.6),(<6,8>,0.9),(<7,6>,0.9),$
$(<7,8>,0.9)\}$

The negative examples of the relation $canReach(V_1,V_2)$ can be found using the close world assumption[15].

According to Definition3 and Definition4,productions of the logic rule for the fuzzy network are defined as follows.

Definition5:Productions of the logic rule for the fuzzy network S_P are defined in Figure2 as follows.

Fig. 2. Productions of the logic rule for the fuzzy network.

C. Derivation Tree for fuzzy network

According to Definition3 and Figure2 ,the derivation tree for the fuzzy network is a natural extension of a traditional derivation tree[3] and similar to a derivation tree[7].A derivation step represents the application of a production in a logic rule,so derivation trees can express the construction of concepts in fuzzy network.For example,the concept definition

scp→clauseSet.
clauseSet→clauseSet,clauseSet.
clauseSet→clause.
clause→{random(0,1,V1)},consequent,[:-(V1)],antecedentSet,[.].
consequent→[canReach(B,C)].
antecedentSet→antecedentSet,[,],antecedentSet.
antecedentSet→antecedent.
antecedent→{member(V1,[A,B,C,D])},{member(V2,[A,B,C,D])}, literal(V1,V2).
literal(V1,V2)→[linkedTo(V1,V2)].
literal(V1,V2)→[canReach(V1,V2)].

$canReach(B,C){:-}(0.6)linkTo(B,C).$,
$canReach(B,C){:-}(0.5)linkTo(B,C),canReach(D,A).$

has the derivation tree are shown in Figure3,scp is start symbol corresponds to all concepts.

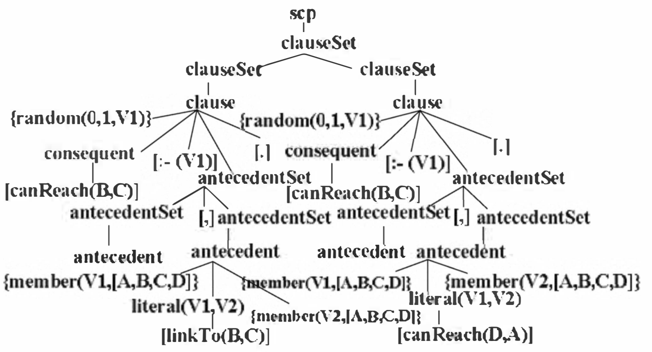

Fig. 3. The primary derivation tree of concepts.

D. Logic rules based Genetic Programming

Concepts are designated as the primary parent and the other one as the secondary parent.Their derivation trees are called the primary and secondary derivation trees respectively.The crossover is a sexual operation[13][16] that starts with two parental concepts and the corresponding derivation trees.The mutation operation[13][16] in concept refinement learning algorithm introduces random modifications to concepts in the population.Concepts in the population are selected as the parental concept.The selection[13][16] is based on various methods such as fitness proportionate and tournament selections.The crossover and mutation steps used to produce an offspring concept are presented in concept refinement learning algorithm in Figure4.

For example,the primary parent is
$canReach(B,C){:-}(0.6)linkTo(B,C).$,
$canReach(B,C){:-}(0.5)linkTo(B,C),canReach(D,A).$,
and the secondary parent is
$canReach(B,C){:-}(0.7)linkTo(B,D).$,
$canReach(B,C){:-}(0.6)linkTo(B,D),canReach(A,C).$,
corresponding derivation trees are depicted in Figure3 and Figure5 respectively,the valid offspring concept
$canReach(B,C){:-}(0.9)linkTo(B,C).$,
$canReach(B,C){:-}(0.8)linkTo(B,C),canReach(A,D).$
is obtained by crossover and its derivation tree is shown in Figure6.

For example,assume that the primary parent being mutated is
$canReach(B,C){:-}(0.6)linkTo(B,C).$,
$canReach(B,C){:-}(0.5)linkTo(B,C),canReach(D,A).$
corresponding the derivation tree is depicted in Figure3,the derivation tree of the offspring concept produced by mutation
$canReach(B,C){:-}(0.8)linkTo(B,C).$,
$canReach(B,C){:-}(0.7)linkTo(B,C),canReach(D,B).$
can be found in Figure7.

III. CONCEPT REFINEMENT LEARNING ALGORITHMS

According to logic rules and derivation trees,a concept refinement learning algorithm combined ILP[2] and GP[13] is presented in Figure4.

```
Induced algorithm for initial population of concepts
    Input:Clause template.
        Generate derivation trees using the logic rule for the fuzzy network
    Output:Derivation trees of concepts.
Evolute algorithm for derivation trees of concepts
    Input:The primary derivation tree P,The secondary derivation tree S.
    Output:A new derivation tree.
        Step1:Execute the algorithm in the current population of concepts
            according to the fitness function;
        Step2:If the termination criterion is satisfied,terminate the
            algorithm,return the best derivation tree of concepts;
        Step3:Create a new population of concepts from the current population
            by applying the reproduction,crossover,and mutation
            operations.Proceed to the next generation and go to the step1.
    Function crossover(P,S)
    {
        C1:Find all subtrees of P,S and store them into a global variable
            PsubTree,SsubTree;
        C2:If PsubTree is not empty,select randomly a subtree,otherwise,without
            generating any new derivation tree;
        C3:Designate the subtree selected as the SelPsubTree and the root of it as the
            primary crossoverPoint.Remove the SelPsubTree from PsubTree;
        C4:Copy SsubTree to TsubTree.If TsubTree is not empty,select randomly a
            subtree,otherwise,go to C2;
        C5:Designate the subtree selected in C4 as the SelSsubTree.Remove it from
            TsubTree.If the new derivation tree produced by performing crossover
            between the SelPsubTree and the SelSsubTree is invalid go to C4;
        C6:Copy P to the new derivation tree,remove SelsubTree from it and
            impregnatinga copy of SelSsubTree at primary crossoverPoint;
        C7:Return the new derivation tree.
    }
    Function mutation(P)
    {
        M1:Find all subtrees of P and store them into a global variable
            subTree,excluding goals and terminal symbols;
        M2:If subTree is not empty,select randomly a subtree from
            subTree,otherwise,without generating any new derivation tree;
        M3:Designate the subtree selected as MutatesubTree.The root of
            MutatesubTree is called the mutatePoint.Remove MutatesubTree from
            subTree.Designate nonterminal symbol as NonTerminal that may have a list
            of arguments called Args;
        M4:For each argument in the Args,if contains variables,determine whether they
            are instantiated,otherwise,it is unbounded,store the modified bindings to
            NewBindings;
        M5:Create a new nonterminal symbol NewNonTerminal from NonTerminal
            and the bindings in NewBindings,try to generate a new derivation tree
            NewsubTree from NewNonTerminal;
        M6:If NewsubTree can be successfully created,the new derivation tree is
            obtained by deleting MutatedsubTree from a copy of P and impregnating the
            NewsubTree at the mutatePoint,otherwise, go to M3.
    }
```

Fig. 4. Concept refinement learning algorithm for the fuzzy network

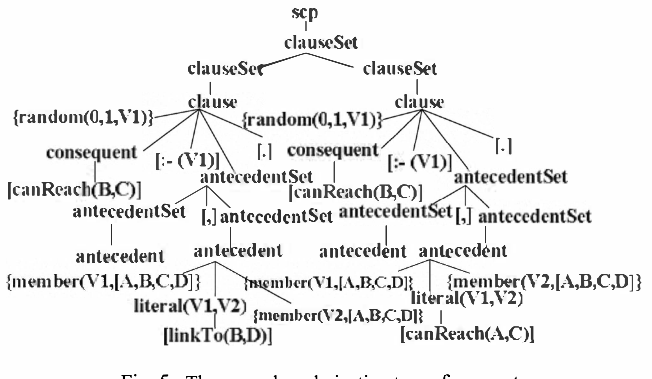

Fig. 5. The secondary derivation tree of concepts

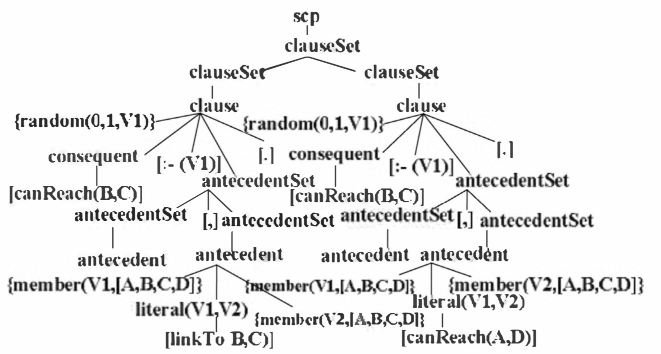

Fig. 6. The derivation tree of the offspring concept produced by crossover

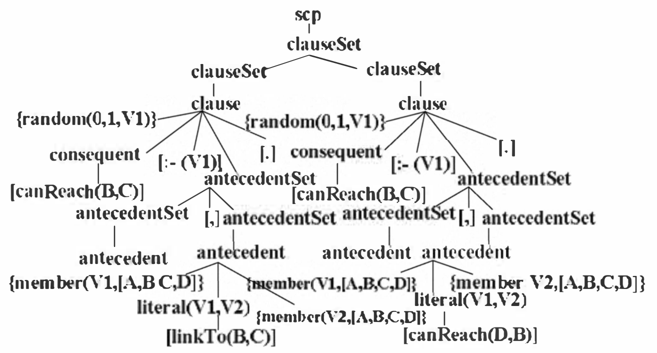

Fig. 7. The derivation tree of the offspring concept produced by mutation.

IV. Experiment Results and Evalution

This section compares the performance of our learner[2] with that of CfgGP[3],GGP[7],Progol[16].Our learner using the concept refinement learning algorithm employs selection,crossover,and mutation to generate potentially better functions.Two of various fitness functions[13] are test,they are used for top-down learner[3][7] and down-top learner[16].Learning curves are used to estimate the performances of various learner.The example space is divided randomly into disjoint training and testing sets.The example space contains 60 positive and 40 negative examples.The training sets contain 30 to 100 examples.The testing set consists of 55 positive and 15 negative examples.

The positive and negative examples are used as the fitness cases.The fitness function finds the sum,taken over all fitness cases of the absolute values of the difference between the desired truth value and the truth value returned by the generated concept definitions.A fitness case is said to be covered by a concept definition if the truth value returned is within 0.03 of the desired value.Our learner terminates if the maximum number of generations of 30,is reached or a fuzzy network that covers all fitness cases is found.The logic rule for the fuzzy network problem is shown in Figure2.The background knowledges are represented by the fuzzy relation $linkTo(V1,V2)$ and the predicate $random(0,1,V)$, $member(V,[A,B,C,D])$ that are goals as procedure call.

This process of dividing,training,and testing is repeated for 20 trials and the results are averaged to generate a learning curve.Experiment results for induced accuracy vs tranning size are depicted in Figure8.

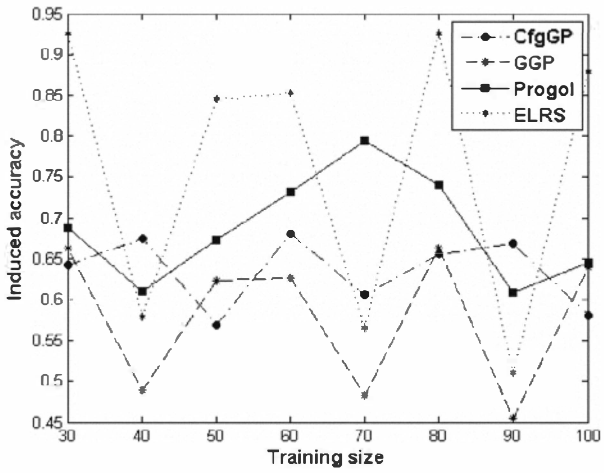

Fig. 8. Learning curves for estimating the performances of various learner

V. CONCLUSION

In this paper it has been explored that the use of logic rules as key element in concept refinement learning in first order logic.Concept refinement learning including lnductive concept learning by employing ILP and evolutionary concept learning by employing GP.The viability of our approach is illustrated by comparing the performance of our learner with that of other concept learners such as Progol,CfgGP,GGP on a variety of target concepts in first order logic.A framework is presented that combining ILP and GP using the logic rules for concept refinement learning and implement our learner ELRS[2].The system can learn concept definitions in various formal languages such as mathematical language in FCA,first order logic in ILP and programming language in GP.The logic rules represent context-sensitive information and domain-dependent knowledge.

To induce and evolve target concepts,we have to determine the logic rules,the fitness function and other major parameters such as the population size,the maximum number of generations,and the probabilities of applying various genetic operations.For concept learning in first order logic,each individual clause in the population can be evaluated in terms of how well it covers positive examples and excludes negative examples.Thus,the fitness functions for concept learning problems calculate this measurement.Typically,each clause set is run over a number of training examples so that its fitness is measured as the total number of misclassified positive and negative examples.Sometimes,if the distribution of positive and negative examples is extremely uneven,this method of estimating fitness is not good enough to focus the search.

The Quinlan's network reachability problem is used as a demonstration,the problem involves a directional network ,we extend it to a fuzzy network in Figure1.The experiment illustrates the ability of our learner for inductive concept learning by employing ILP and evolutionary concept learning by empoying GP from fuzzy network.In the future,we plan to apply our approach to other problems of concept refinement learning such as functional program learning,recursive function learning,imperfect data learning and inductive concept learning in Formal Concept Analysis(FCA).

ACKNOWLEDGMENT

The work presented in this paper is supported by National Natural Science Foundation of China(No.60575035,60975033) and University Natural Science Foundation Projects in Jiangsu(No.07KJB520096).Thank for the anonymous reviewers valuable suggestions and reference literature authors.

REFERENCES

[1] Z.G.Shi,Z.T.Liu,Q.Wu,“Research on τ-parameter fuzzy concept lattice for grid resource”,in Proc.SKG'06,2006,paper 10.1109.76,p.16-22.
[2] Z.G.Shi,Z.T.Liu,J.P.Chen,“Research on a learning algorithm as inverse deduction”,Journal of Nantong University,vol.8,pp.10-16,Sep.2009.
[3] P.Whigham,“A Schema Theorem for context-free grammars”, in Proc.ICEC'95,1995,paper 10.1109.489140,p.178-181.
[4] A.Colmerauer,Ed.,Natural Language Communication with Computers:Metamorphosis Grammars,ser.Lecture Notes in Computer Science.Berlin,Heidelberg:Springer,1978,vol.63.
[5] R.Kowalski,“Algorithm=Logic+Control”,Communications of the ACM,vol.22,pp.424-436,Jul.1979.
[6] F.Pereira,D.Warren,“Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks”,Artificial Intelligence,vol.13,pp231–278,Mar.1980.
[7] M.L.Wong,“An adaptive knowledge acquisition system using generic genetic programming”,Expert Systems with Applications,vol.15,pp.47-58,Jul.1998.
[8] Z.G.Shi,Z.T.Liu,“Research on model of real concept lattice for grid resources and algorithms”,Computer Engineering and Applications ,vol.42,pp.4-8,Dec.2006.
[9] Z.G.Shi,Z.T.Liu,“Research on model of fuzzy concept lattice for grid resource and algorithms”,Application Research of Computers,vol.24,pp.70-74,Oct.2007.
[10] Q.Wu,Z.T.Liu,Z.G.Shi,“Constructing Galios Lattice in Hybrid Datasets Based on Rough Set Theory”,in Proc.ICICTA'08,2008,paper 10.1109.133,p.767-771.
[11] B.Ganter and R.Wille,Formal Concept Analysis:Mathematical Foundations. Berlin,Heidelberg:Springer-Verlag,1999.
[12] S.Muggleton,“Inductive logic programming”,New Generation Computing,vol.8,pp295-318,Apr.1991.
[13] J.Koza,Genetic Programming:on the Programming of Computers by Means of Natural Selection.Cambridge,MA:MIT Press,1992.
[14] T.M.Mitchell,“Generalization as search”,Artificial Intelligence,vol.18,pp.203-206,Mar.1982.
[15] J.R.Quinlan,“Learning logical definitions from relations”,Machine Learning,vol.5,pp239-266,Aug.1990.
[16] S.Muggleton,A.Tamaddoni-Nezhad,“QG/GA:A stochastic search for Progol”,Machine Learning,vol.70,pp.121-133,Mar.2008.