International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 4, Number 1 (2014), pp. 79-84

http://www. irphouse.com /ijict.htm

Implementation of Recursively Enumerable Languages using Universal Turing Machine in JFLAP

Ankur Singh[1] and Jainendra Singh[2]

[1]*Department of Computer Science, IEC College of Engineering, Greater Noida, INDIA.*
[2] *Department of Computer Science, Maharaja Surajmal Institute, C-4, Janakpuri, New Delhi, INDIA.*

Abstract

This paper presents the implementation of recursively enumerable language using Universal Turing machine for JFLAP platform. Automata play a major role in compiler design and parsing. The class of formal languages that work for the most complex problems belongs to the set of Recursively Enumerable Language (REL).RELs are accepted by the type of automata as Turing Machine. Turing Machines are the most powerful computational machines and are the theoretical basis for modern computers. Turing Machine works for all classes of languages including regular language, Context Free Languages as well as Recursive Enumerable Languages. Still it is a tedious task to create and maintain Turing Machines for all problems. The Universal Turing Machine (UTM) is a solution to this problem. A UTM simulates any other TM, thus providing a single model and solution for all the computational problems. Universal Turing Machine is used to implementation of RELs for JFLAP platform. JFLAP is most successful and widely used tool for visualizing and simulating all types of automata.

Keywords: Automata; Compiler; Turing Machine; TM; JFLAP; PDA.

1. Introduction

An automaton is a mathematical model for a finite state machine (FSM). A FSM is a machine that has a set of input symbols and transitions and jumps through a series of states according to a transition function. Automata play a major role in compiler design and parsing. Turing machines are the most powerful computational machines. They possess an infinite memory in the form of a tape, and a head which can read and change the tape, and move in either direction along the tape or remain stationary. Turing machines are equivalent to algorithms, and are the theoretical basis for modern computers. A Turing machine that is able to simulate any other Turing machine is called a Universal Turing machine (UTM, or simply a universal machine). A UTM is the abstract model for all computational models. A UTM TU is an automaton that, given as input the description of any Turing Machine TM and a string w, can simulate the computation of M on w. JFLAP is software for experimenting with formal languages topics including nondeterministic finite automata, nondeterministic pushdown automata, and multi-tape Turing machines, several types of grammars, parsing, and L-systems. JFLAP is extremely useful in constructing Turing Machine with multiple inputs. Complex Universal Turing Machines can also be built by using other Turing Machines as components or building blocks for the same.

It is possible to introduce various ICT Applications for each category. Farmers can use B2B portals, Information and knowledge managements systems, information systems and mobile technologies to obtain information and direct market to reach the vision. Peelers must be updated with the technologies and market requirements, market trends and standards. Since this is the most critical category in the production process and difficult category for the technology adoption, special attention must required. Collectors are the category who collect cinnamon from farmers, sorting and grading cinnamon. There for they should be updated with market requirements, market trends and standards. They can use knowledge management systems, information systems and web sites to get information to optimize product qualities. Exporters must use B2B portals, interactive webs, informational webs and latest ICT tools for marketing, promotions and communication. Certification authorities are using international standard web sites and ICT tools. But they should be reported to the government authority. Buyers have no much control over the other parties and government. They can access latest prices, details and can directly communicate with exporters, certification authorities, and other categories on demand via the ICT channels.

2. Universal Turing Machine in JFLAP

2.1 JFLAP

JFLAP (Java Formal Languages and Automata Package) is software for experimenting with formal languages topics including nondeterministic finite automata, nondeterministic pushdown automata, and multi-tape Turing machines, several types of grammars, parsing, and L-systems. JFLAP is extremely useful in constructing Turing Machine with multiple inputs. Complex Turing Machines can also be built by using other Turing Machines as components or building blocks for the same. The implementation of a Turing Machine and Universal Turing Machine for the JFLAP platform has been described. JFLAP is most successful and widely used tool for visualizing and simulating automata such as finite state machines, pushdown automata, and Turing Machines. By executing our Universal Turing Machine in JFLAP, everyone get a direct and interactive experience of how this Turing Machine is capable of emulating other Turing Machines.

2.2 Turing Machine

A Turing machine is an automation whose temporary storage is tape. This tape is divided into cells, each of which is capable of holding one symbol. Associated with the tape is read-write head that can travel right or left on the tape and that can read and write a single symbol on each move. Turing Machines are the most powerful computational machines. The Turing Machine (TM) is the solution for the halting problem and all other problems that exist in the domain of computer science. Turing Machines provide an abstract model to all problems. It can work with Recursively Enumerable Language.

A Turing Machine *M* is defined by

$$M = (Q, \Sigma, \Gamma, \delta, q_s, \square, F)$$

where

Q is the set of internal states {qi | i is a nonnegative integer}

Σ is the input alphabet

Γ is the finite set of symbols in the tape alphabet

δ is the transition function

S is Q * Γn → subset of Q * Γn * {L, S, R}n

□ is the blank symbol.

qs (is member of Q) is the initial state

F (is a subset of Q) is the set of final states

2.3 Universal Turing Machine

A UTM simulates any other TM, thus providing a single model and solution for all the computational problems. A UTM TU is an automaton that, given as input the description of any Turing Machine TM and a string w, can simulate the computation of M on w. It reduces the memory usage when compared to using multiple TMs. The transition function is the core part of a UTM. The UTM works on the basis of the rules defined in it. The transition function δ is defined as

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$$

The transition function δ is a partial function on Q X Γ and its interpretation gives the principle by which a Turing Machine operates. The arguments of δ are the current state of the control unit and the current tape symbol being scanned. The result is a new state of the control unit, a new tape symbol which replaces the old one and a move symbol L or R.

A UTM can accept regular languages, CFGs as well as RELs. A UTM can solve any problem that can be solved using a FSA, PDA or even a standard Turing Machine. The UTM designed in this paper supports a restricted alphabet of {a, b, c, x, y, z, }. It does not support non-determinism. Any standard TM with a maximum of ten states can be simulated using this UTM. It has over 1000 states to simulate a standard TM.

2.4 Recursively Enumerable Language

A language L is said to be recursively enumerable if there exists a Turing Machine that accepts it. It implies that there exists a Turing Machine M, such that, for every w € L

$$q_0w \vdash^*_M x_1q_fx_2$$

With q_f a final state. The definition says nothing about what happens for w not in L; it may be that machine halts in a no final state or that it never halts and goes into an infinite loop.

Regular languages form a proper subset of Context Free Languages. So PDA is more powerful than finite automata. But CFLs are limited in scope because many of the simple language like $a^nb^nc^n$ are not context free. So to incorporate the set of all languages that are not accepted by PDAs and hence that are not context free, more powerful language families has been formed. This creates the class of Recursively Enumerable Languages (REL).

3. Implementation

The Language L is said to be recursively enumerable if there exists a Turing Machine that accepts it. The working of Turing Machine for a recursively enumerable language can be explained with an example of $a^nb^nc^n$. The language $a^nb^nc^n$ is a recursively enumerable language which cannot be implemented using a FA as well as a PDA. The standard Turing machine T_M for the language $a^nb^nc^n$ is given in Fig. 1(a).

The various strings are applied to the Turing Machine with multiple run. A few results are shown in Fig. 1(b).

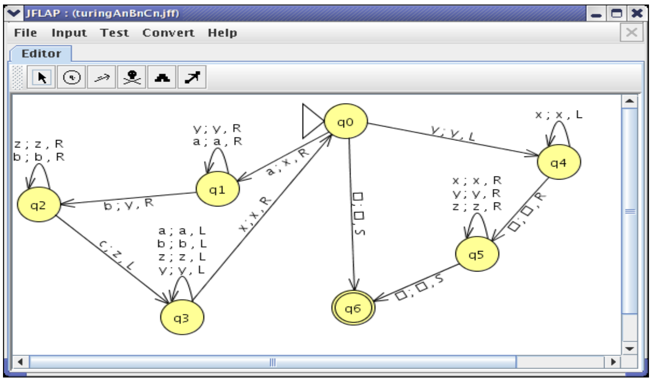

Fig. 1. (a) Turing Machine for anbncn

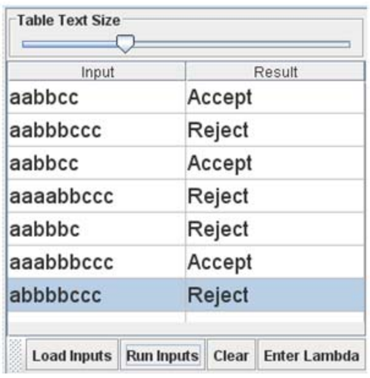

Table Text Size

Input	Result
aabbcc	Accept
aabbbccc	Reject
aabbcc	Accept
aaaabbccc	Reject
aabbbc	Reject
aaabbbccc	Accept
abbbbccc	Reject

Load Inputs Run Inputs Clear Enter Lambda

(b) Multiple Run by Turing Machine

The same problem can be solved with the UTM also. The difference lies in the way the UTM branches into states and transitions as a single move of TM corresponds to multiple moves of TU.

The same problem can be solved with the UTM also. The difference lies in the way the UTM branches into states and transitions as a single move of TM corresponds to multiple moves of TU.

For a deterministic Turing machine with m symbols in the alphabet such that | Σ | = m and total number of states n, m X n transitions are possible. A UTM with n states, | Σ | = m and p possible directions branches into m X n X p states for execution. A problem that can be solved with a multi tape Turing machine with m tapes in O (n) moves can be done with a UTM in O (nm) moves. For Fig. 2 shows the UTM for the recursively enumerable language $a^nb^nc^n$.

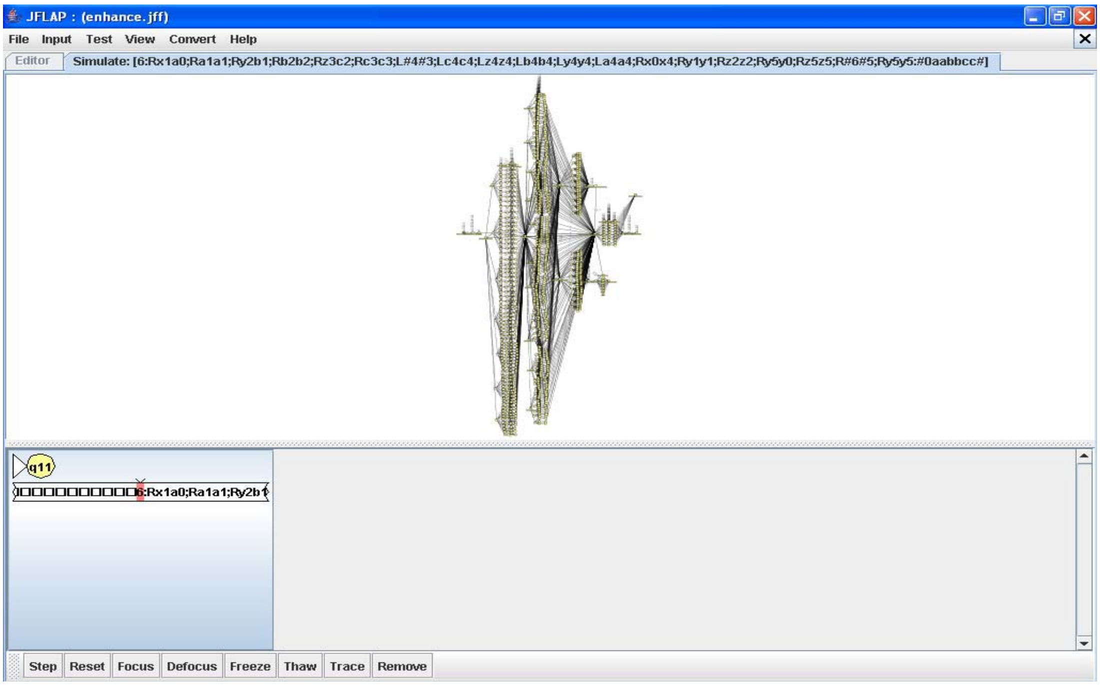

Fig. 2: Universal Turing Machine for anbncn.

4. Conclusions

Turing Machines are the most powerful computational machines. The Turing Machines provide an abstract model to all the problems. This paper describes the working of a Turing Machine and UTM for Recursively Enumerable Languages for JFLAP platform. A Universal Turing Machine simulates any other Turing Machine, thus providing a single model and solution for all the computational problems. The language $a^nb^nc^n$ is a recursively enumerable language which cannot be implemented using a Finite Automata or a PDA but can done using a Turing Machine. This requires more storage than for Context Free Languages and hence the Turing Machine with the infinite tapes, extendable in both directions is used for this. The UTM is designed to supports an alphabet of {a, b, c, x, y, z}.It simulates standard Turing machine with a maximum of ten states. It also does not support non-determinism.

The future work includes enhancing the concept of universality by including more symbols in the input alphabet as well as in the tape alphabet.

References

[1] Susan H.Rodger, Eric Wiebe, Kyung Min Lee, Chris Morgan, Kareem Omar and Jonathan Su, "Increasing engagment in automata theory with JFLAP", ACM Transactions,, 2009, 403-407.

[2] Eric Gramond and Susan H. Rodger, "Using JFLAP to interact with theorems in automata theory",ACM Poratl Proc. In SIGCSE,1999,336-340.

[3] Peter Linz, An Introduction to Formal Languages and Automata (3rd Edition, Narosa Publishing House,2003).

[4] J.Hopcroft, R.Motwani and J.Ullmann, Introduction to Automata Theory, Language and Computation (3rd Edition, Addison-Wesley, 2006).

[5] [Online]http://www.jfalp.org/tutorials.

[6] Susan H.Rodger and Thomas W.Finley, JFLAP: An Interactive Formal Languages and Automata Package, ISBN 0763738344, Jones & Bartlett Publishers, 2006.

[7] Sumitha C.H and Krupa Ophelia Geddam, "Implementation of Context Free Languages in Turing Machines", IEEE conf. on second International conference on Machine Learning and Computing, 2010.

[8] Jainendra Singh, Dr. S.K. Saxena, "Implementation of Recursively Enumerable Language in to the Unrestricted Grammar using Turing Machine", International Journal of Information And Computation Technology (IJICT), ISSN 0974-2239, Volume 3, Number 1 (2013), pages 23-26.

[9] Jainendra Singh, Dr. S.K. Saxena, " Implementation of Unrestricted Grammar in to the Recursively Enumerable Language using Turing Machine", The International Journal of Engineering And Science (IJES), ISSN 2319-1813 ISBN 2319-1805, Volume-2, Issue-3 (2013),pages 56-59.