RECOGNITION OF TAMIL IN UNIVERSAL TURING MACHINE

A. Maheshwari¹, M. A. DoraiRangaswamy²

¹Research Scholar, Sathyabama University, Chennai
²Professor, Department of CSE, AVIT, Chennai

78mahee@gmail.com
drdorairs@yahoo.co.in

Abstract- Turing Machines are the most powerful computational machines. Turing machines are equivalent to algorithms, and are the theoretical basis for modern computers. Still it is a tedious task to create and maintain Turing Machines for all the 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. A universal Turing machine is one which can accept the description of another Turing machine as input and simulate the operation of that Turing machine. The creation of UTM is very tedious because we need to devise encoding scheme for Tamil language to serve all TM's. Also many of the existing tools do not support the creation of UTM which makes the task very difficult to accomplish. Hence a Universal Turing Machine is developed for the JFLAP platform to recognize Tamil language. JFLAP is most successful and widely used tool for visualizing and simulating all types of automata.

Keywords- Tamil words; Delta rule; FSA; JFLAP; Transitions; UTM

I. TURING MACHINE

Turing Machines are the most powerful Finite State Machines. They can simulate exactly what a digital computer can do. Informally, TM consists of a finite set of states and a controller that can read or write symbols on an infinite length tape. Unlike DFA, NFA, and ε-NFA, a TM controller can move in both directions on the tape.

The machine starts with an initial state, a finite number of input symbols written on the tape (all other infinite number of tape cells are blank), and the controller is set to the first input symbol from the left. According to the current state and the current scanned symbol on the input tape, the controller takes the next move. It can overwrite the current scanned symbol or leave it untouched, change the current state, then move to either the left or the right, and so on. If no more moves the left or the right, and so on. If no more moves are possible, then the machine halts. In some cases, the machine may run forever and never halt.

II. TURING MACHINE MODEL

The Turing machine is a simple and powerful mathematical model of computation that can be used as language acceptor, language translator, and function evaluator. The languages that the Turing machine can compute are recursively enumerable.

The graphical model of the Universal Turing machine is as follows:

The model consists of an input output relation that the machine computes. The input is given in binary form on the machine's tape, and the output consists of the contents of the tape when the machine halts.

At every step, the current state and the character read on the tape determine the next state the FSM will be in, the character that the machine will output on the tape (possibly the one read, leaving the contents unchanged), and which direction the head moves in, left or right.

The problem with Turing Machines is that a different one must be constructed for every new computation to be performed, for every input output relation.

This is why we introduce the notion of a universal turing machine (UTM), which along with the input on the tape, takes in the description of a machine M. The UTM can go on then to simulate M on the rest of the contents of the input tape. A universal turing machine can thus simulate any other machine.

A. RELATIONSHIP BETWEEN AUTOMATA AND LANGUAGE CLASSES
III. VISUAL EXAMPLES

A. Rice Cooker

Rice cooker operations can also be modeled by an FSM. Examples of the operations include cooking, reheating, and keeping warm. A timer is also considered in this model. The student can operate the rice cooker simulator by pressing the operations, and then the corresponding state of the underlying automaton is highlighted. In the rice cooker automaton model, every state represents an operation, for example, the state labeled q0 represents the waiting (initial) state, q1 represents the keep warm (final state) operation, and q2 represents the reheating operation.

The input alphabet is represented by the symbols A, B, C, and D, where

1. “A” corresponds to heating and reheating operations,
2. “B” corresponds to the keep warm and cancel operations,
3. “C” corresponds to the timer, and
4. “D” corresponds to the finish operation.

In a real rice cooker, after we set the initial conditions, it completes the task and finishes automatically after a certain amount of time. The applet simulates such behavior; first, the user can set the initial conditions by pressing the buttons A, B, and C; then, the finish button D will take place automatically by the automaton after a certain amount of time.

IV. UNIVERSAL TURING MACHINE IN JFLAP

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. Still it is a tedious task to create and maintain TMs for all the 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.

A. Universal Turing Machine

A UTM simulates any other TM, thus providing a single model and solution for all the computational problems. A UTM T_U is an automaton that, given as input the description of any Turing Machine T_M 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

δ: Q × Γ → Q × Γ × {L, R} \tag{1}

The UTM is represented as

T_U = (Q, Σ, Γ, δ, q_0, B, F) \tag{2}

Where

Q is the set of all internal states,
Σ is the input alphabet,
Γ is a finite set of symbols called the tape alphabet,
δ is the transition function,
qu_0 is the initial state
B ∈ Γ is a special symbol called the blank,
F ∈ Q is the set of all final states.

1) Working of the UTM

The UTM has an infinite tape extendable in both directions to hold the input and perform the computation. It also has a read-write head to position the input symbol. The UTM has infinite memory. There are two other tapes also that are used for the processing. The first tape holds the description of the original Turing Machine T_M and the other tape to hold the internal state of T_M.

The input to the UTM T_U is given in the
form of \( \langle T_M, w \rangle \) where \( T_M \) is the Turing Machine that has to be manipulated and \( w \) is an input string for \( T_M \). The execution of the Turing Machine is specified by transition rules or delta rules. Each transition is of the form:

\[
\delta (q_i, a) = (q_j, b, R)
\]

(3)

where

- \( q_i \) is the current state,
- \( a \) is the current read symbol,
- \( q_j \) is the next state or destination state,
- \( b \) is the write symbol and
- \( R \) is the direction to which the tape head has to move.

2) Manipulation of Turing Machine in the UTM

\( T_U \) repeatedly carries out the classic von Neumann fetch-and-execute process. The fetch process determines which delta-rule of \( T_M \) to be emulated next. \( T_U \) begins from the initial configuration section of its tape. It begins by reading the current state-id digit and the alphabet character that lays one cell to its right in its copy of the tape of \( T_M \), which is present in the last section of the tape. \( T_U \) will then locate the correct transition rule of \( T_M \) from the list-of-rules in the portion of its tape enclosed by the two colon symbols, which is the middle section of the tape. If no matching transition rule is found, \( T_U \) will crash similar to what \( T_M \) would have done.

V. TEST RESULTS

For a deterministic turing machine with \( m \) symbols in the alphabet such that \( | \Sigma | = m \) and total number of states \( n \), \( m \times n \) transitions are possible.

A UTM with \( n \) states, \( | \Sigma | = m \) and \( p \) possible directions branches into \( m \times n \times p \) states for execution. A problem that can be solved with a multi tape Turing machine with \( m \) tapes in \( O(n^m) \) moves can be done with a UTM in \( O(n^m) \) moves.

RECOGNIZATION OF ENGLISH TO TAMIL

The actions that should be carried out are:

1. Copies the input from input tape.
2. Positions head 1 at the beginning of the string in input tape.
3. At a time only one character is recognized.
4. All Tamil words are stored in the memory tape.
5. Convert the English words to the corresponding Tamil words.
VI. EXISTING SYSTEM
A. Turing Machines are the most powerful computational machines.
B. Turing machines are equivalent to algorithms, and are the theoretical basis for modern computers.

VII. PROPOSED SYSTEM
A. Developing a Universal Turing Machine for South Indian Languages.
B. Implement the concept of universality by including more symbols in the input alphabet as well as the tape alphabet.

VIII. CONCLUSION AND FUTURE WORK
Turing machines are the most powerful computational machines. A Universal Turing Machine simulates any other Turing Machine, thus providing a single model and solution for all the computational problems. The Turing Machines provide an abstract model to all the problems. This paper describes the working of a Turing Machine as well as a Universal Turing Machine for Context Free Languages.

The Turing Machines differ from all other automata as it can work with recursively Enumerable Languages. The language $a^m b^n a^m b^n$ is a recursively enumerable language which cannot be implemented using a PDA but can be done using a T.M. This requires more storage than for Context Free Languages and hence the Turing Machines with the infinite tapes, extendable in both directions are used for this.

The future work includes
i. Developing a Universal Turing Machine for South Indian Languages.
ii. Implement the concept of universality by including more symbols in the input alphabet as well as the tape alphabet.

IX. REFERENCES