Assignment Project Exam Help

Turing Machines

COMP1600 / COMP6260

https://powcoder.com

Victor Rivera Dirk Pattinson
Australian National University

Add WeChat powcoder

Semester 2, 2021

Alan Turing (1912–1954)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Turing's Scientific contributions

1936

- introduces *Turing machines* and the study of computability

Assignment Project Exam Help

1950.

- introduces the *Turing test* that turns AI into a concrete research problem

https://powcoder.com

- current today: Mitsuku, a computer programme has convinced judges (for the last 5 competitions – from 2013 to 2019) that it was a 18-year-old female from Leeds, England.

Add WeChat powcoder

1952.

- Pioneering work on computation in *nature*;

Also. Key figure in the invention of the earliest modern computers.

Turing at War

Germany

- Germans used *enigma machines* – most sophisticated crypto equipment

Assignment Project Exam Help

The U.K.

- code breaking effort assembled at Bletchley park
- Turing chief scientific genius

https://powcoder.com

Achievements

- cracked secret Enigma code
- estimated to have shortened world war II by *2 years*

Add WeChat powcoder

Fallout

- may ideas and technologies discovered in Bletchley park fed directly into modern computers

Death and Legacy

1952

- less than 10 years after heroic war efforts for the UK
- Turing prosecuted for 'crime' of homosexuality
- sentenced to chemical castration

1954

- death by suicide

Legacy Now

- widely recognised as the *father of computing*
- *Turing award* equivalent of Nobel prize
- UK government apologised for persecution in 2009
- 2012 officially named the Turing year
- Royal Pardon in 2013

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Turing Machines – Introduction

Assignment Project Exam Help

Computability – 1936 paper

On computable numbers, with an application to the Entscheidungsproblem

https://powcoder.com

- solved long standing open problem posed by Hilbert
- changed the world (of mathematics and computing)
- simple mathematical device that is able to simulate any computer
- this device instrumental in solving Hilbert's problem

Add WeChat powcoder

Modern Computers

- didn't exist in 1936?
- ENIAC in 1946 generally considered to be the first

Computing as a profession

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

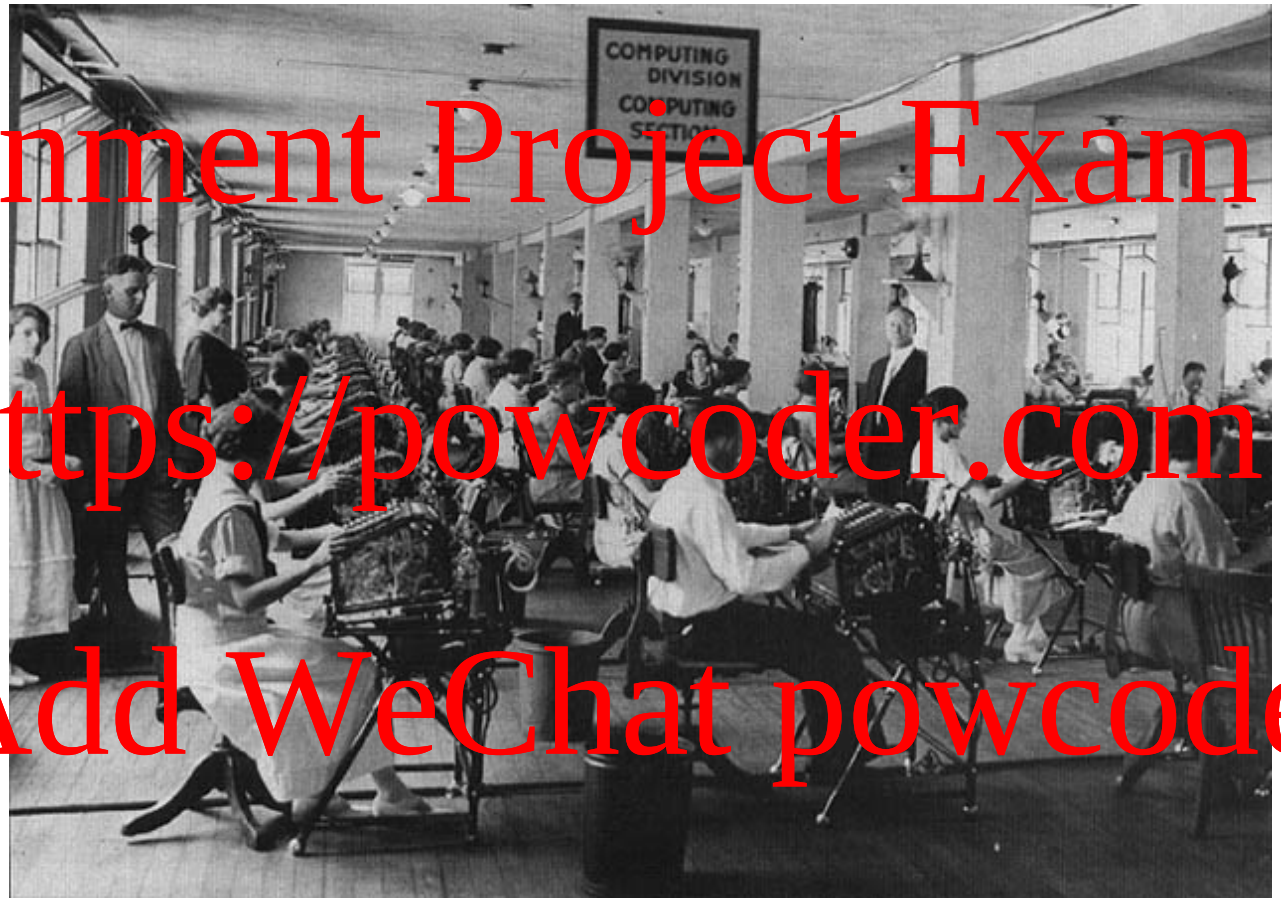


Photo c/o Early Office Museum Archives

A model for 'computers'

Computing in 1936

- computers not machines, it was a *profession*.
- Turing's contribution: mathematical *definition* of what computers can do.
- justification: references to 'state of mind' or similar
- see original paper, Section 9.I (quite readable!) `http://www.cs.virginia.edu/robins/Turing_Paper_1936.pdf`

Assignment Project Exam Help
https://powcoder.com

Turing's model today

- no computer has been built that is *more* powerful than Turing's model
- Turing has discovered the *essence of computation*

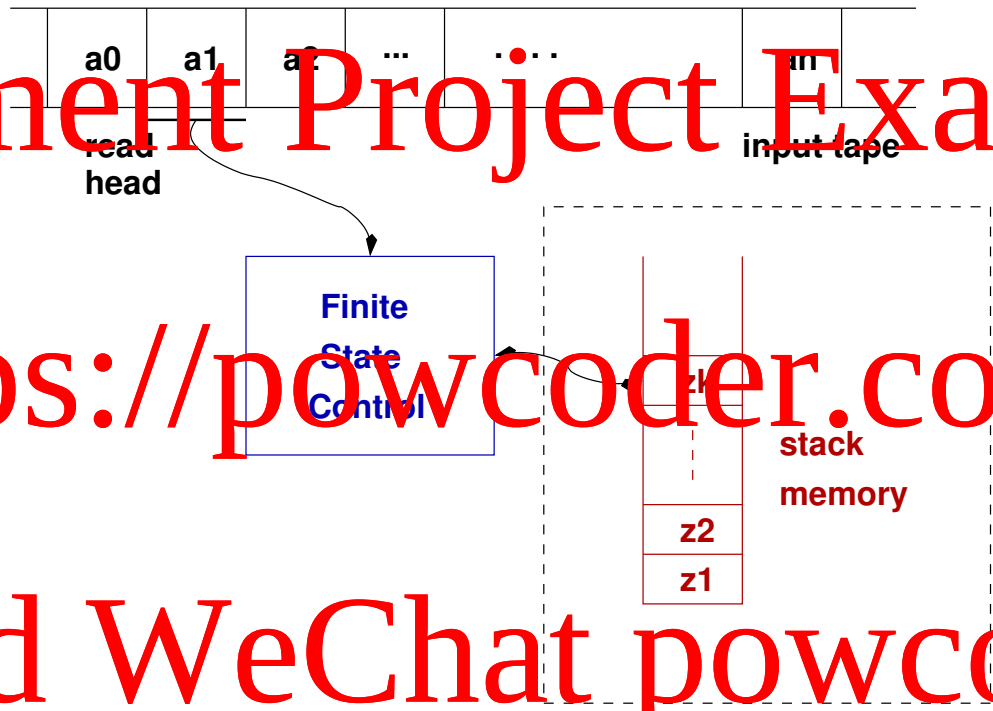
Add WeChat powcoder

Mathematical Models vs Reality

- no formal *proof* that the model is the "right" one
- but widely accepted
- new paradigms emerge, e.g. quantum computing

Push Down Automata, reloaded

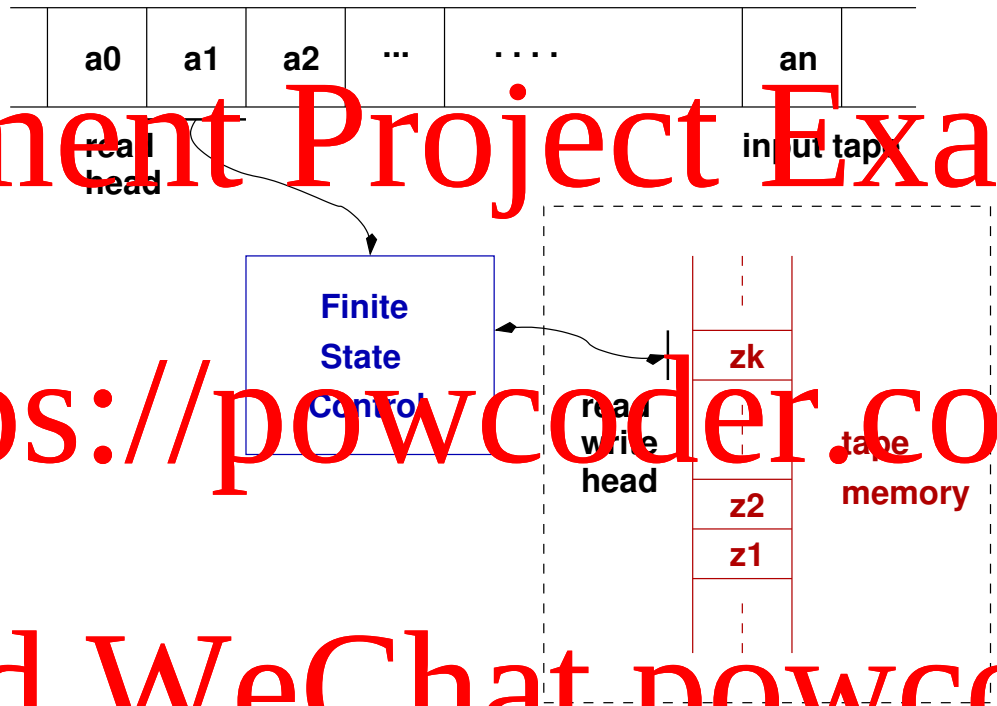
Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



- A PDA with its auxiliary store is *almost* a whole computer, except we can only directly access the symbol on the top of the stack.

Turing Machines

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

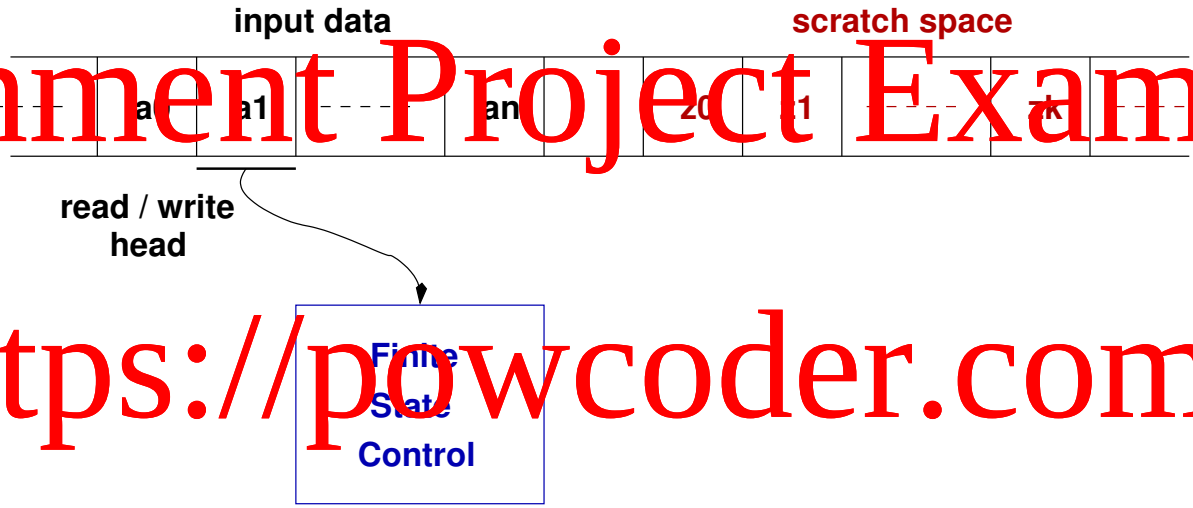


Generalisation from PDA to Turing machine

- replace *stack memory* by *tape memory*
- can access *arbitrary* symbols on tape by moving tape head

Single tape Turing Machines

Assignment Project Exam Help



https://powcoder.com

Simplification

Add WeChat powcoder

- have *the same* tape both for input and for storage
- tape is assumed to be *infinite* in both directions
- analogy: "get more paper if you run out"

Turing Machines as language recognisers

Assignment Project Exam Help

Deterministic Machines for the time being

- at every time of the computation *at most* one action is possible
- If there is *no* action that is legal, the TM *halts*
- If a TM halts in a *final* state, it has accepted the original input. The set of strings accepted by a TM is its *language*
- If it halts in a *non-final* state, this is an 'error', i.e. the input is rejected.
- A TM may also loop forever...

https://powcoder.com

Add WeChat powcoder

Definition. If a language is recognised by a Turing machine, then it is called *recursively enumerable*.

Output

Assignment Project Exam Help

Turing Machines as Computing Devices

- TMs can calculate *any* computable function.
- input: a string written onto the tape before the machine starts
- output: whatever is left on the tape when the machine halts

https://powcoder.com

Infinite Tapes aren't a problem

- computation halts after *finitely many steps*
- only finitely many tape cells will be written to

Add WeChat powcoder

Turing Machine – formal definition

A Turing Machine has the form $(S, s_0, F, \Gamma, \Sigma, \Lambda, \delta)$, where

- S is the set of states, $s_0 \in S$ is the initial state and $F \subseteq S$ are the final states;
- Γ is the set of tape symbols, $\Sigma \subseteq \Gamma$ is the set of input symbols, and $\Lambda \in \Gamma - \Sigma$ is the blank symbol;
- δ is a (partial) transition function

$$\delta : S \times \Gamma \nrightarrow S \times \Gamma \times \{L, R, S\}$$

$$(\text{state}, \text{tape symbol}) \mapsto (\text{new state}, \text{new tape symbol}, \text{direction})$$

The direction tells the read/write head which way to go next: Left, Right, or Stay.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Running a TM

Initialisation.

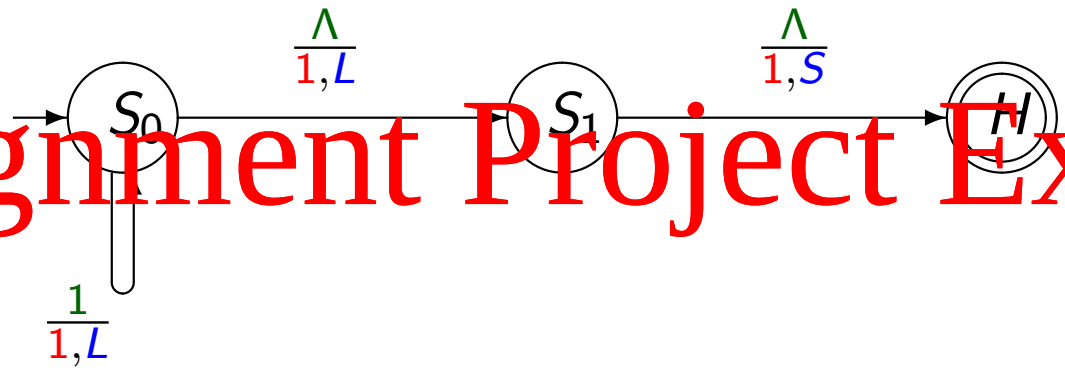
- some input (a finite string over Σ) is written on the tape;
- every other tape cell is a blank – Λ;
- the read/write head sits over the some cell of the input (or over any Λ is the input is ϵ);
- the state is the start state s_0.

Running.

- in a cycle: read symbol and execute action (state dependent): move/write/change state
- until a final state is reached (or the machine gets stuck)

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Graphical Representation of the Transition Function



Assignment Project Exam Help
https://powcoder.com

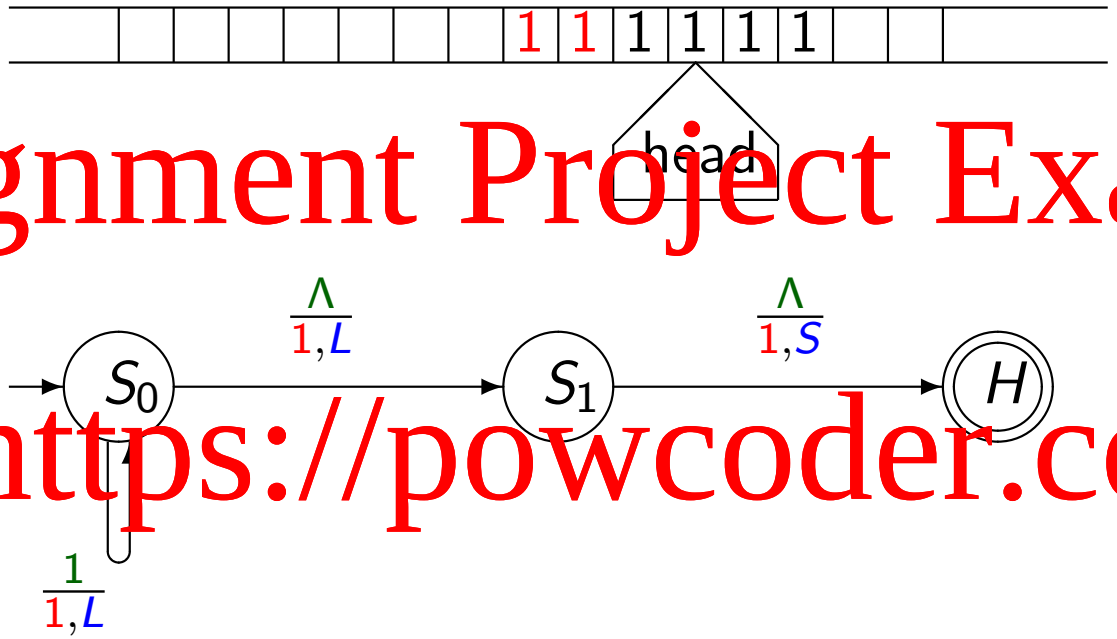
(Like FSA, annotate transition edges with commands for accessing tape.)

Convention

Add WeChat powcoder

- Numerator: *symbol read from tape*.
  - Λ means the tape is blank at that position.
- Denominator: *symbol written* / *direction of head movement*.
  - direction one of L, R, S for Left, Right, Stay.

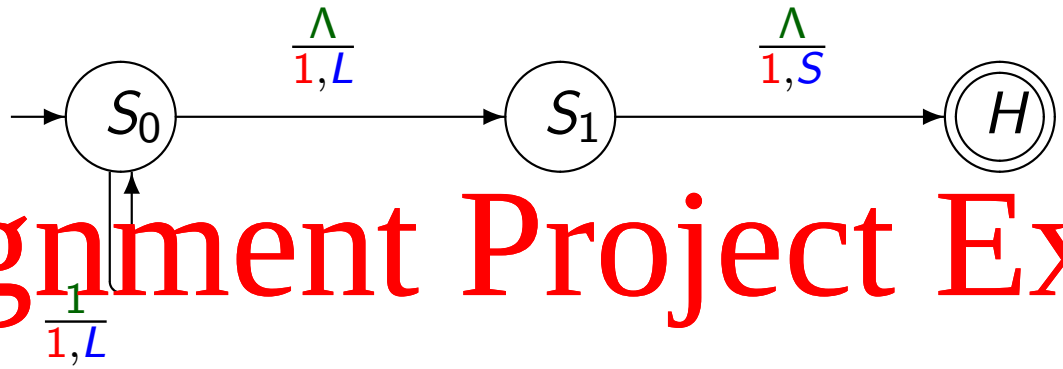
What does it do?



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- Adds two to a unary number!
- Assume the head starts over the input data.
- First phase scans left.
- Second phase writes two extra *1*s.

The Convention for Errors



Assignment Project Exam Help

TMs getting Stuck

https://powcoder.com

- suppose that the input contains a token other than 1.
- TM would halt in state S_0, as there is no arc telling us what to do if we meet such a token (this is the job of the rightwards scan).
- this is an error - the input is *rejected*.
- S_0 *not* an accepting state!

Add WeChat powcoder

Language

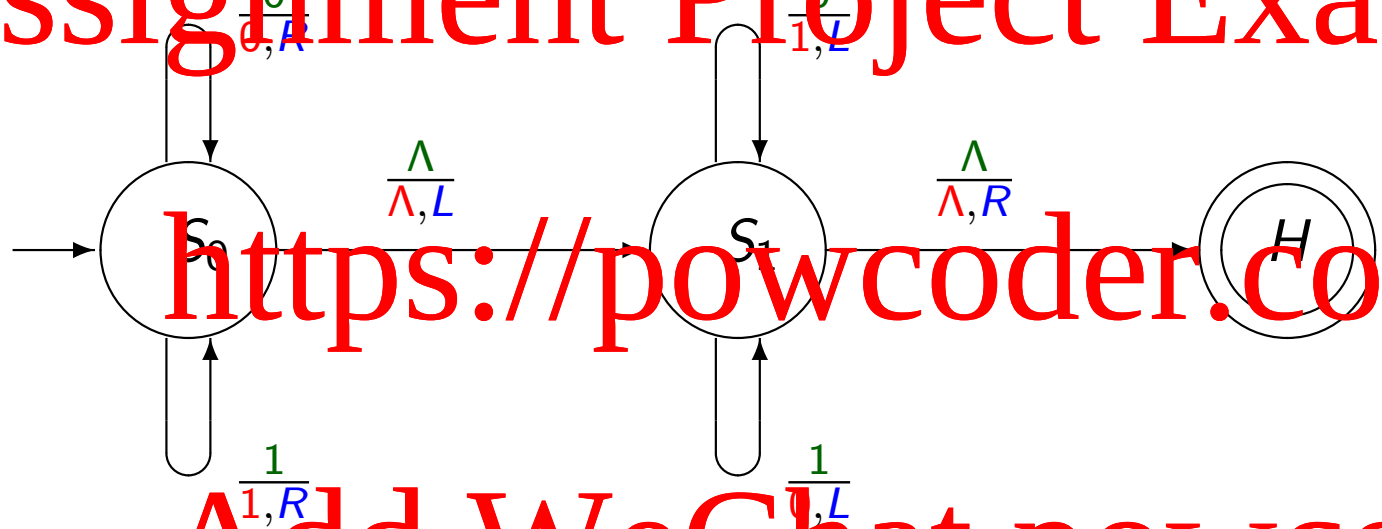
- TM accepts precisely $\{1^n \mid n \in \mathbb{N}\}$.

Alternative Formulation (not used here)

- could add an error state that the machine transitions to
- error state *not* accepting

What does this one do?

Assignment Project Exam Help



https://powcoder.com

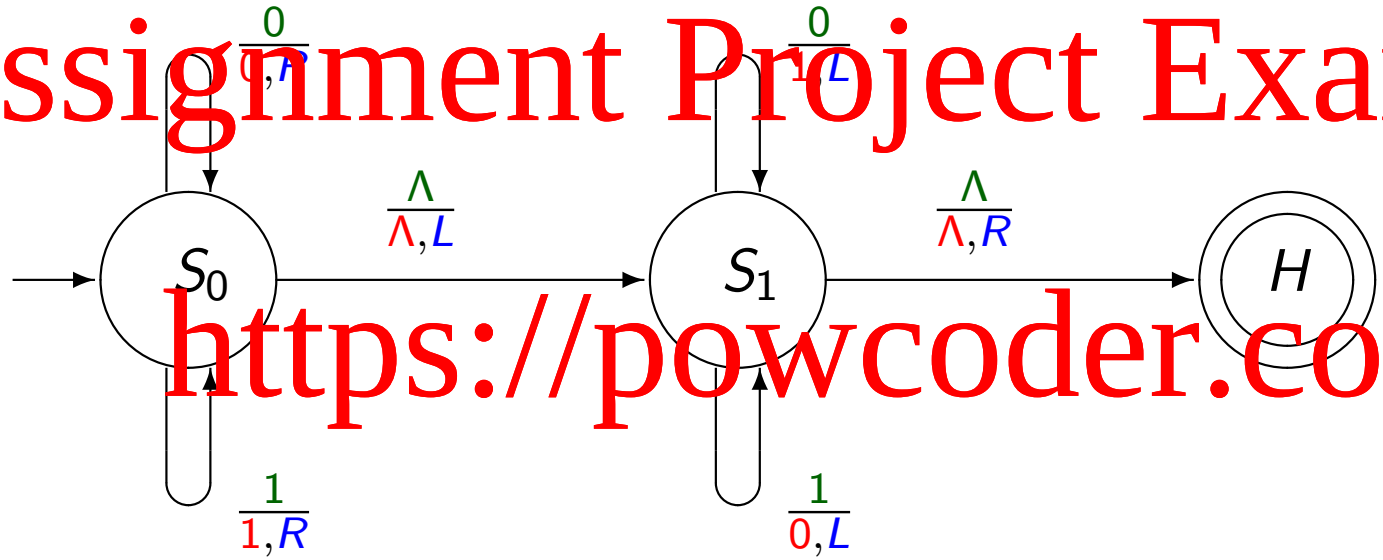
Add WeChat powcoder

Questions.

- Do you see two phases?
- What does each phase accomplish?

What does this one do?

Assignment Project Exam Help



https://powcoder.com

Add WeChat powcoder

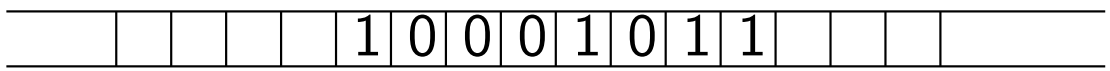
Answer.

- Phase 1: initialisation.
- Phase 2: computation, in this case, complement a binary number.

Harder Problems?

Assignment Project Exam Help

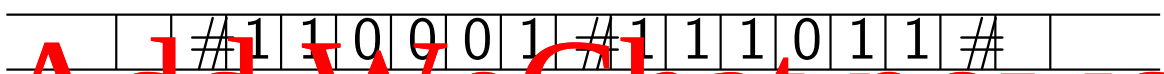
- Incrementing a binary number



You should try this!

https://powcoder.com

- Adding numbers - need terminators



Add WeChat powcoder

Convenient to write the result before the data.

- Multiplication - and so on!

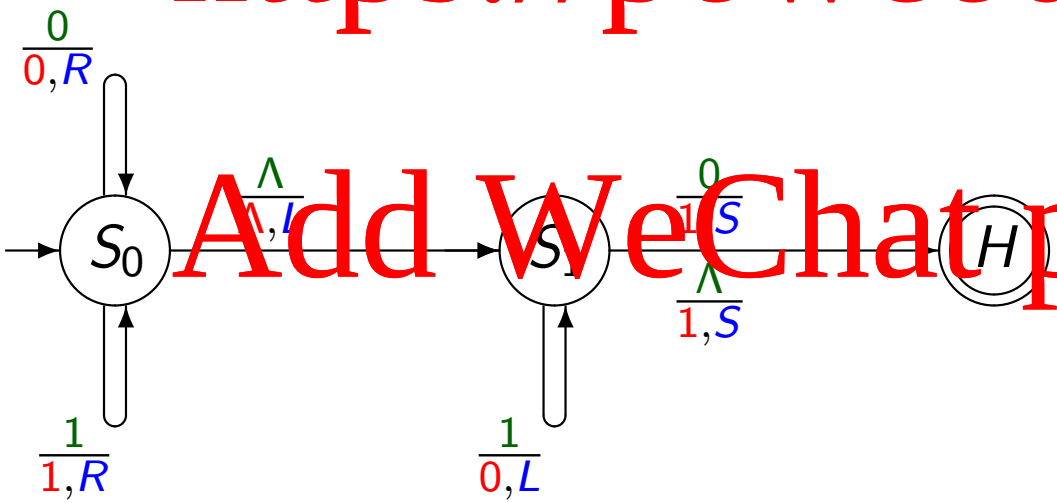
Incrementing a binary number

Assignment Project Exam Help

Example. Number increment

Solution:

https://powcoder.com

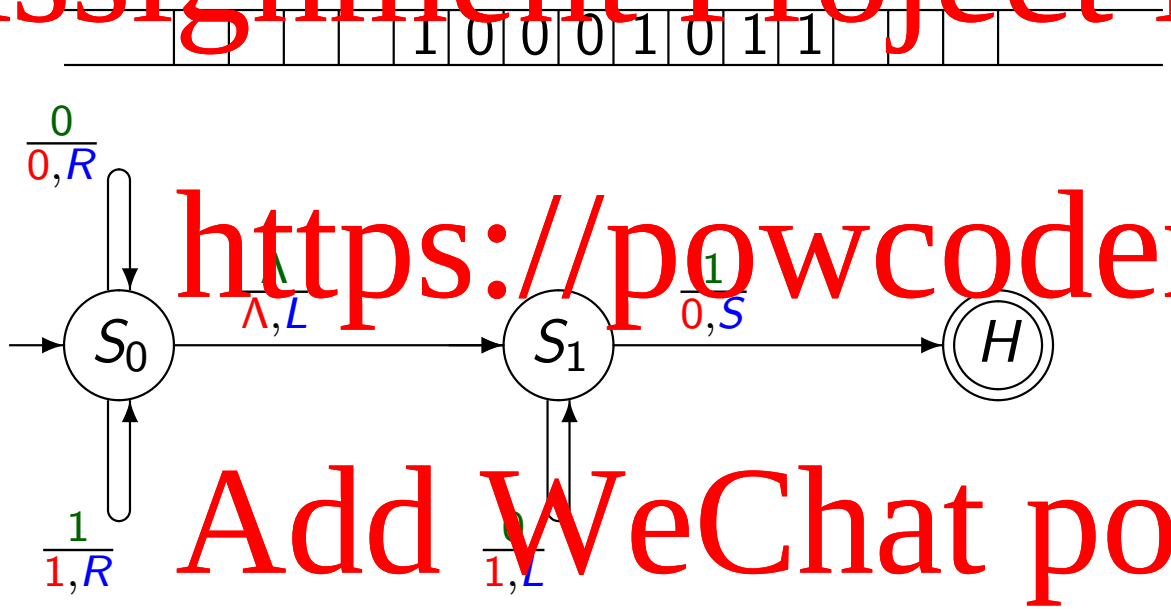


Add WeChat powcoder

Decrement

Example. Number decrement (similar)

Assignment Project Exam Help



https://powcoder.com

Add WeChat powcoder

Failure (or non-acceptance): If given number is 0 it fails (at state S_1),

How to add two numbers?

Input. Binary numbers separated by #, say

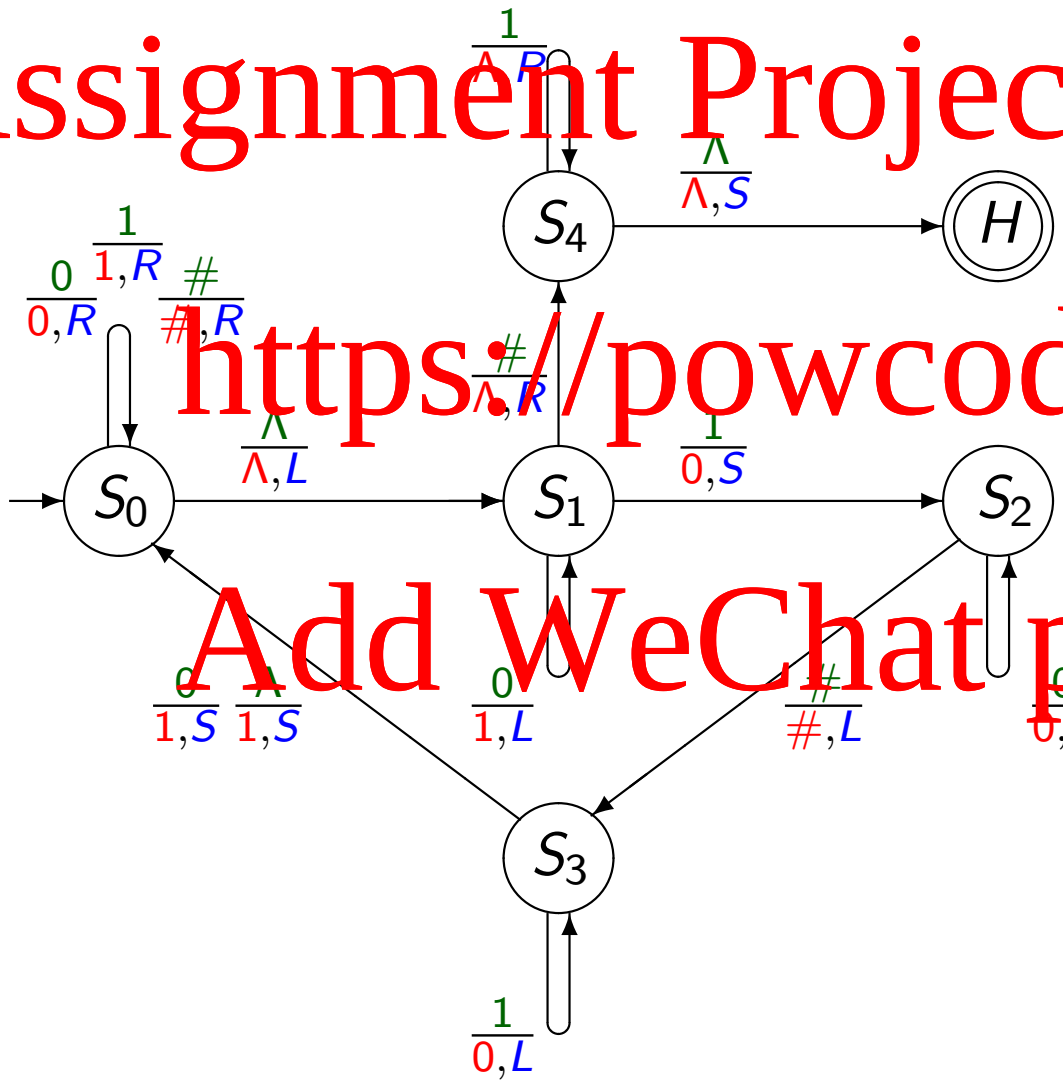
$$\overbrace{10100101}^{n}\#\overbrace{100101010}^{m}$$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Operation.

- Go back and forth between m and n, decrementing one (until this fails) and incrementing the other.
- decrement m, and increment n because n will expand leftwards.
- m gets changed to $00\ldots0$, n is replaced by the sum.
- Finally, delete the $\#00\cdots0$ on the right.

How to add two numbers? ctd.



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

How to multiply two numbers?

Input. as for addition

$$\underbrace{\quad}_{n}\#\underbrace{10100101}_{n}\#\underbrace{100101010}_{m}$$

Assignment Project Exam Help

Operation.

- Repeatedly decrement m (until this fails) and add n to p (p is initially blank)
- Must modify the addition routine to not erase the number n being added.

https://powcoder.com

Modification of addition routine

- Two new tape symbols, $0'$ and $1'$.
- Before each addition stage, change all the 1s in n to $1'$.
- When decrementing n, swap 0s and 1s as usual, but keep the primes.
- When finished adding n to p, go back and use the primes to restore n.

Add WeChat powcoder

Observation.

- this is *very tedious* – but the model is simple and easy to analyse
- tricks that you see here are typical

Programming Issues – Data

Data Types and Gadgets

- not present in the model
- but can be simulated ...

Numbers.

- Usually use unary or binary for integers.

Vocabulary.

- Can be arbitrary, and $\{0, 1\}$ suffice. Characters are represented as strings of bits.

Variables, Arrays, Files

- Use markers on the tape to separate values.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Programming Issues – Control

Common Idioms.

Assignment Project Exam Help

- Scan to blank or to find, insert, delete a symbol
- Use control states to remember information
  In particular, we often need to "remember" a symbol, to write it elsewhere: this typically requires a set of states, one per symbol

https://powcoder.com

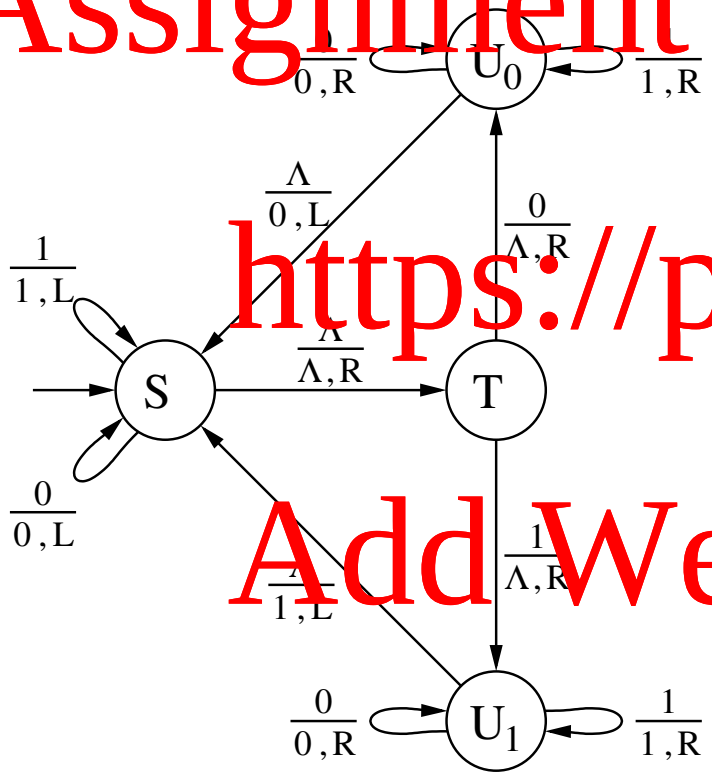
- Composition
  If you have a TM to multiply by 3 and one to multiply by 5, put them together to multiply by 15.

Add WeChat powcoder

- Decisions (conditional computation)
  As we have seen, we can branch on 0 or 1 (or T or F).
- Loops — of course.

Using States to Remember a Tape Symbol

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



Given a string of 0 or 1 surrounded by blanks, this machine repeatedly forever erases the leftmost bit, and writes it on the right hand end. (Not so useful, but illustrates the point)

We use the choice of states U_0 or U_1 to remember which symbol has been erased and is to be written

Universal Turing Machines and Turing Completeness

- We can construct a TM to simulate any conventional computer. (Cost effectiveness is not brilliant.)
- From this point, just think of a TM as like a computer program (with a machine that will run it)
- We can construct a TM that first reads a description of some other TM and then simulates it. This is a *universal* TM. (Think of a Haskell program whose purpose is to read any other Haskell program and run that)
- Turing machines can simulate themselves.
- Turing machines can *compute properties of themselves*.
- Any computing device which can simulate a universal Turing Machine is also called *universal* or *Turing Complete*.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Church-Turing Thesis

Church-Turing Thesis.

> If a function is computable, then it can be calculated by a Turing machine.

Assignment Project Exam Help

Equivalent Formulation.

- if a problem can be solved by an algorithm, then it can be solved by a Turing machine.

https://powcoder.com

This is a Thesis.

- more a definition than a theorem
- can never be *proved*: what does algorithmic mean?
- however, there's lots of *evidence*

Add WeChat powcoder

Evidence.

- all *other* definitions of the term computable give the same class of computable functions
- there are many: λ-calculus, register machines, while programs, etc.

Church and λ-calculus

Assignment Project Exam Help

Example. The (untyped) λ-calculus

- proposed by Alonzo Church (the C in the Church-Turing thesis)
- in 1932, even before Turing's paper
- Rosser showed that both notions are equivalent

https://powcoder.com

Equivalence. (Rosser, 1939)

- if a function is computable by a Turing machine, then it is computable in the λ-calculus

Add WeChat powcoder

- ... and vice versa
- simulation of the respective formalism in the other approach.

Can real computers simulate Turing Machines?

Differences between computers and Turing machines

- A Turing Machine has an *infinite tape*.
- "real" computers have finite memory (sometimes a lot, but nowhere near infinite)
- physical devices *necessarily* have finite memory. They're more like finite automata.
- . . . but the number of states can be very, very large. How big is $2^{4294967296}$?
- physical computers are an *approximation* of TMs.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Intuitive Understanding

- Turing machine model *feels* closer to what we can program
- e.g. we *can* recognise the language $\{a^n b^n \mid n \geq 0\}$
- if the real machines runs out of memory . . . we can always buy more
- this is about computation *in principle*

Church-Turing Thesis: Argument 1

Turing Machines emulate humans.

- comparison how humans perform tasks
- Turing's 'computers' only perform TM operations: writing, reading, refocussing (state change)

TMs are finite like humans

- can't do infinite operations, e.g. work with infinitely many symbols
- can't read the whole infinite tape

Taken together

- convincing (in the 1930s) that Turing's model is correct
- but no mathematical evidence (yet)

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Church-Turing Thesis: Argument 2

Stability of the TM Model.

- adding 'features' doesn't make more functions computable

Multi-tape.

- have extra tapes to store data
- easier to program, but no extra "power"
- (single-tape can simulate multi-tape)

Multi-head.

- have more than one head, heads can move independently
- heads can access multiple symbols at once
- again, no extra power

Non-determinism. (as for nfa's)

- tm can make one of several possible next moves
- tm can "guess" the right next move
- *may* make it faster, but cannot compute more functions.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Church-Turing Thesis: Argument 3

Many Models of Computation

- plenty of *different* definitions of what computable may mean
- different purposes, different contexts

Assignment Project Exam Help

Main Insight.

- *any* (reasonable) model of computation can compute *precisely* the same functions as the TM model.
- "reasonable" in the sense of modelled on mechanical computation

https://powcoder.com

Examples.

- Lambda-Calculus (Church, 1932)
- Post-Systems (Post, 1939)
- Register Machines
- ...

Add WeChat powcoder

Doesn't include

- models based on physical phenomena
- ... or biology, or ...

Argument 3B: Grammars

Theorem. Any language that is generated by a grammar can be recognised by a Turing machine.

Assignment Project Exam Help

Proof (Sketch)

- write the *start symbol* S onto the tape (say, right of our input)
- search through all possible derivations from S
- each time we reach a *sentence*, check whether it matches the input

https://powcoder.com

Acceptance

Add WeChat powcoder

- if the grammar finds the input, we will find a derivation
- if the grammar does *not* generate the input, we may loop forever (and not accept)

Argument 3B – grammars ctd

Unrestricted Grammars. Have productions of the form

$$\alpha \to \beta$$

where β is arbitrary, and α contains at least one non-terminal.

Assignment Project Exam Help

Theorem. For any TM, there exists a grammar that generates *precisely* the words that the TM accepts.

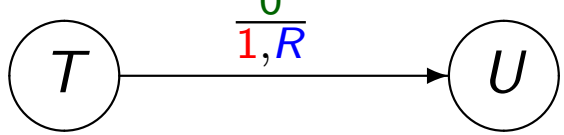
https://powcoder.com

Proof (Sketch)

- non-terminals (of the grammar) are states of the TM
- run TM "backwards" (we are interested in inputs, not outputs)

Add WeChat powcoder

TM Transition.



Grammar Production. $1U \to T0$.

(Detail missing, e.g. how to handle blanks)

Argument 3C – Computers & Programming Languages

Observation.

- no computer ever invented could do things that a TM can't do
- no programming language can do more than a TM
- back-and-forth translation TM $\leftrightarrow$ PL

Assignment Project Exam Help

https://powcoder.com

Common Terminology.

A programming language that can compute every function that can be computed by a Turing machine is called *Turing complete*.

Add WeChat powcoder

Examples

- The languages that you know: Haskell, Java, Python, ...
- even the simple while language that we used for Hoare logic
- implement TM simulator in your favourite programming language

Argument 3D – John Conway's Game of Life

Game of Life.

- infinite 2d grid
- finitely many cells marked *alive*, all others are *dead*

Rules of the Game. iterate through generations

- live cells with fewer than two live neighbours die (*under-population*)
- live cells with more than three live neighbours die *(over-population)*;
- dead cells with exactly three live neighbours come alive *(reproduction)*;
- all other cells stay as they are.

Emergent Behaviour.

- visualised by GoL, many interesting forms
- analogy of complex behaviour emerging from simple rules

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Argument 3D – John Conway's Game of Life ctd.

Assignment Project Exam Help

From TMs to Conway's Game

- can implement Game of Life on a Turing machine
- lots of coding, in particular 2d grid onto 1d tape

https://powcoder.com

From Conway's Game to TMs (Paul Rendell, 2011)

- showed that GoL can simulate Turing machines
- comes down to clever choice of initial configurations see `http://rendell-attic.org/gol/turing_js_r.gif`

Add WeChat powcoder

Metaprograms

Metaprograms are programs that have other programs as input or output.

Assignment Project Exam Help

Examples

- *Lexical analysers* and *parser generators SPARK Examiner*, which automatically proves correctness properties of programs written in SPARK Ada;

https://powcoder.com

- *Code generation* which automatically produces codes from e.g. a Z specification or a formal proof

Add WeChat powcoder

Next Goal. Scrutinise TMs that take TMs as input.

- main goal: *halting problem*

Coding a TM onto tape

Coding of a TM as binary strings

- can be written onto a tape
- just code the transition function

Assignment Project Exam Help

States are ordered $S_1, S_2, \ldots, S_k$, where S_1 is the start state and S_2 the unique final state

https://powcoder.com

Tape Symbols are ordered $X_1, X_2, X_3, \ldots, X_l$ where X_1 is 0, X_2 is 1, and X_3 is Λ.

Directions $\{L, R, S\}$ as D_1, D_2, D_3 respectively

Add WeChat powcoder

Transitions $\delta(S_i, X_j) = (S_k, X_l, D_m)$ mapped to

$$0^i\ 1\ 0^j\ 1\ 0^k\ 1\ 0^l\ 1\ 0^m$$

(the 0s carry information, the 1s act as separators.)

Coding a TM onto tape ctd.

Assignment Project Exam Help

Coding of all transitions. A TM with transitions $C_1, \ldots, C_n$ is codes as

$$C_1 \ 11 \ C_2 \ 11 \ \cdots \ 11 \ C_n$$

https://powcoder.com

(11 is used as a separator for transitions)

Additional Input.

Add WeChat powcoder

- code for a TM, and additional string
- use 111 to separate TM and string input

Coding a TM onto tape – example

Assignment Project Exam Help
https://powcoder.com

The transitions are

0	1	00	1	0	1	00	1	00
0	1	000	1	000	1	00	1	00
000	1	000	1	00	1	00	1	000

Add WeChat powcoder

Recall: States, Symbols, Directions.

So the TM as a whole is encoded by the binary string

0100101001 00 11 0100010001 00100 11 0001000100 1001000