CSC 240

DECIDABILITY

TURING LANGUAGES

A Turing machine accepts a string if it enters an accept state while processing that string.

A Turing machine rejects a string if it enters a reject state while processing that string.

A Turing machine loops forever on a string if it never enters an accept or reject state while processing that string.

A Turing machine halts on a string if it accepts or rejects that string.

The set of strings accepted by a Turing Machine are the language of that machine.

$$\mathscr{L}(M) = A$$

The $\mathscr{L}$anguage of Turing machine M

A is the set of all strings accepted by M.

"A is the language of M"

"A is the *recognized by* M"

TURING LANGUAGES

A language is “Turing Recognizable” if a Turing machine recognizes it.

A language is “Turing Decidable” if a Turing machine decides it, that is if the strings of that language cause the Turing machine to end up in an accept or reject state.

These machines are sometimes called “deciders”

TURING MACHINE COMPUTER EQUIVALENCE

Turing Machines ≅ Computers

If we can think of a way to do something with a computer, we can do it with a Turing Machine.

ENCODING NOTATION

O - Some object

<O> - Some object encoded as a string

<svg width="580" height="400"><g><rect fill="#fff" id="canvas_background" height="402" width="582" y="-1" x=“-1”/>...

01010101011111100001010...

Everything you have ever seen or heard on a computer can be represented as a string.

ACCEPTANCE PROBLEMS

D - Some DFA

<D, w> - Some DFA and its string input encoded as a string

$$A_{DFA} = \{ \langle D, w \rangle \mid D \text{ is a DFA that accepts string } w \}$$

Can we build a Turing Machine that *decides* A_{DFA}?

In other words, can we build a Turing Machine that tells us, for an arbitrary DFA, D and an arbitrary input string w, if D will accept or reject w?

ACCEPTANCE PROBLEMS

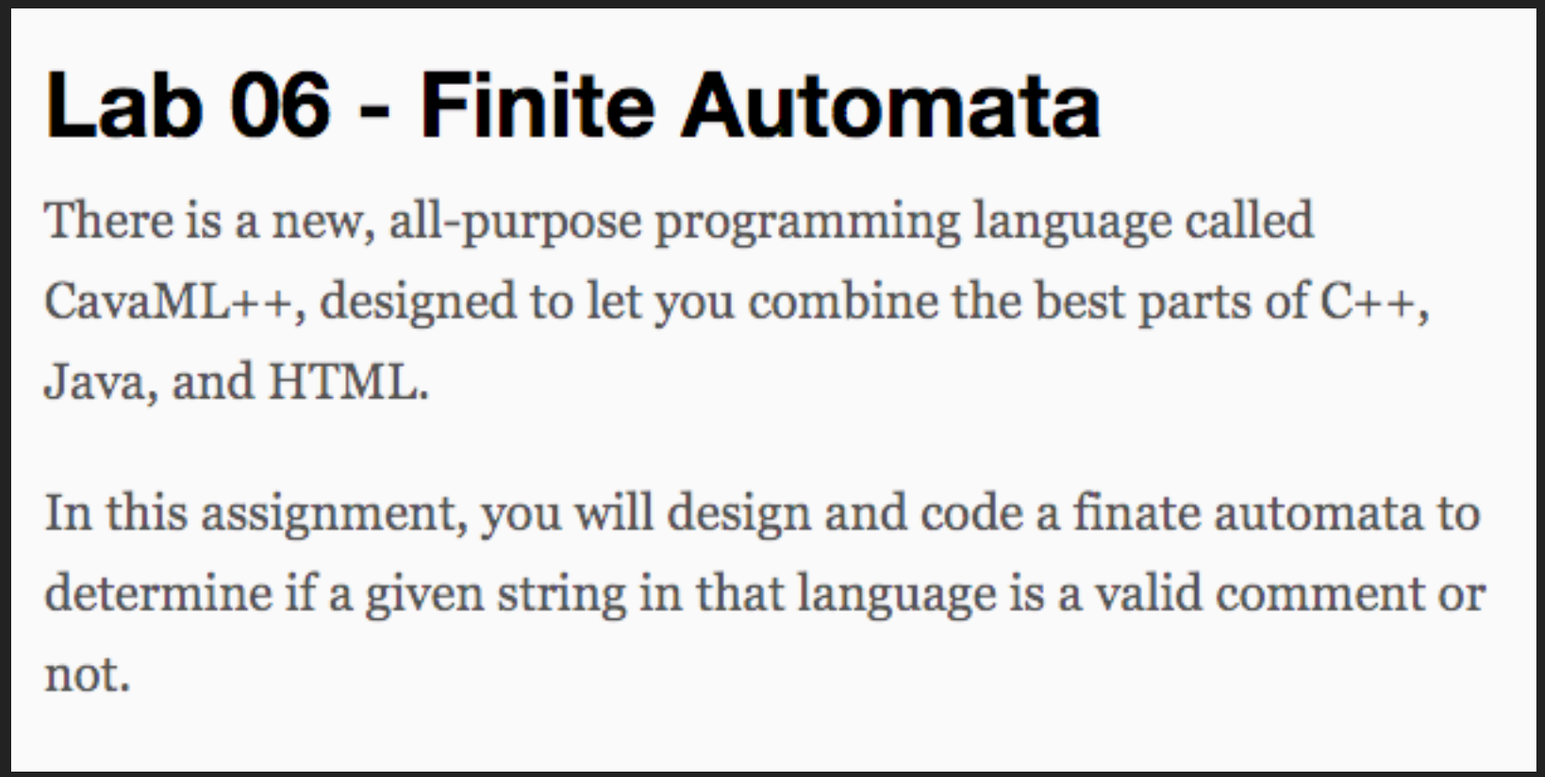
A_{DFA} = { <D, w> | D is a DFA that accepts string w }

Can we build a Turing Machine that *decides* A_{DFA}?

1. Can we encode D as a string?

    Defined by a 5-tuple: $(Q, \Sigma, \delta, q_0, F)$

2. Can we simulate D in a Turing Machine?

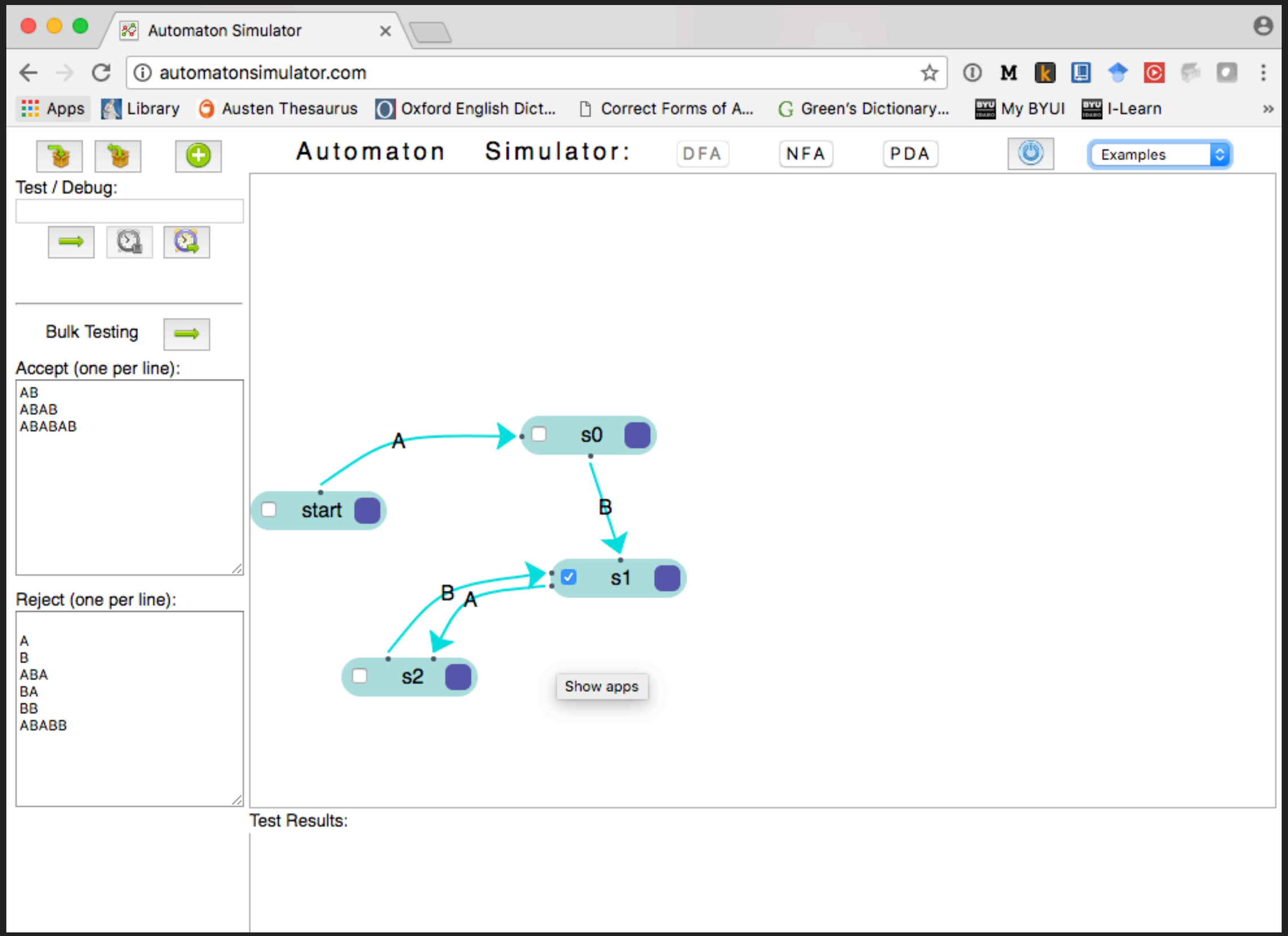


Lab 06 - Finite Automata

There is a new, all-purpose programming language called CavaML++, designed to let you combine the best parts of C++, Java, and HTML.

In this assignment, you will design and code a finate automata to determine if a given string in that language is a valid comment or not.

ACCEPTANCE PROBLEMS



ACCEPTANCE PROBLEMS

N - Some NFA

<N, w> - Some NFA and its string input encoded as a string

$$A_{NFA} = \{ \langle N, w \rangle \mid N \text{ is an NFA that accepts string } w \}$$

Can we build a Turing Machine that *decides* A_{NFA}?

In other words, can we build a Turing Machine that tells us, for an arbitrary NFA, N and an arbitrary input string w, if N will accept or reject w?

"Every nondeterministic finite automaton has an equivalent deterministic finite automaton. "

ACCEPTANCE PROBLEMS

R - Some Regular Expression

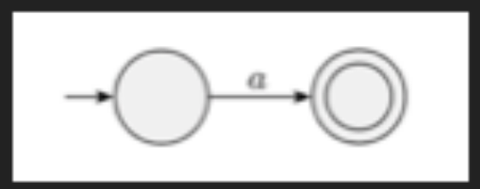
<R, w> - Some Regular Expression and its string input encoded as a string

A_{REX} = { <R, w> | R is a Regular Expression that matches string w }

Can we build a Turing Machine that *decides* A_{REX}?

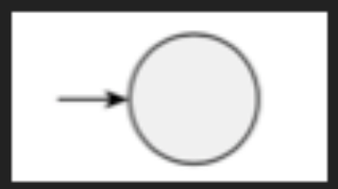
In other words, can we build a Turing Machine that tells us, for an arbitrary Regular Expression, R and an arbitrary string w, if R matches w?

L(R) = {a}



L(R) = {ε}

L(R) = ∅



ACCEPTANCE PROBLEMS

We can build a Turing Machine that tells us:

If a DFA will accept a string.

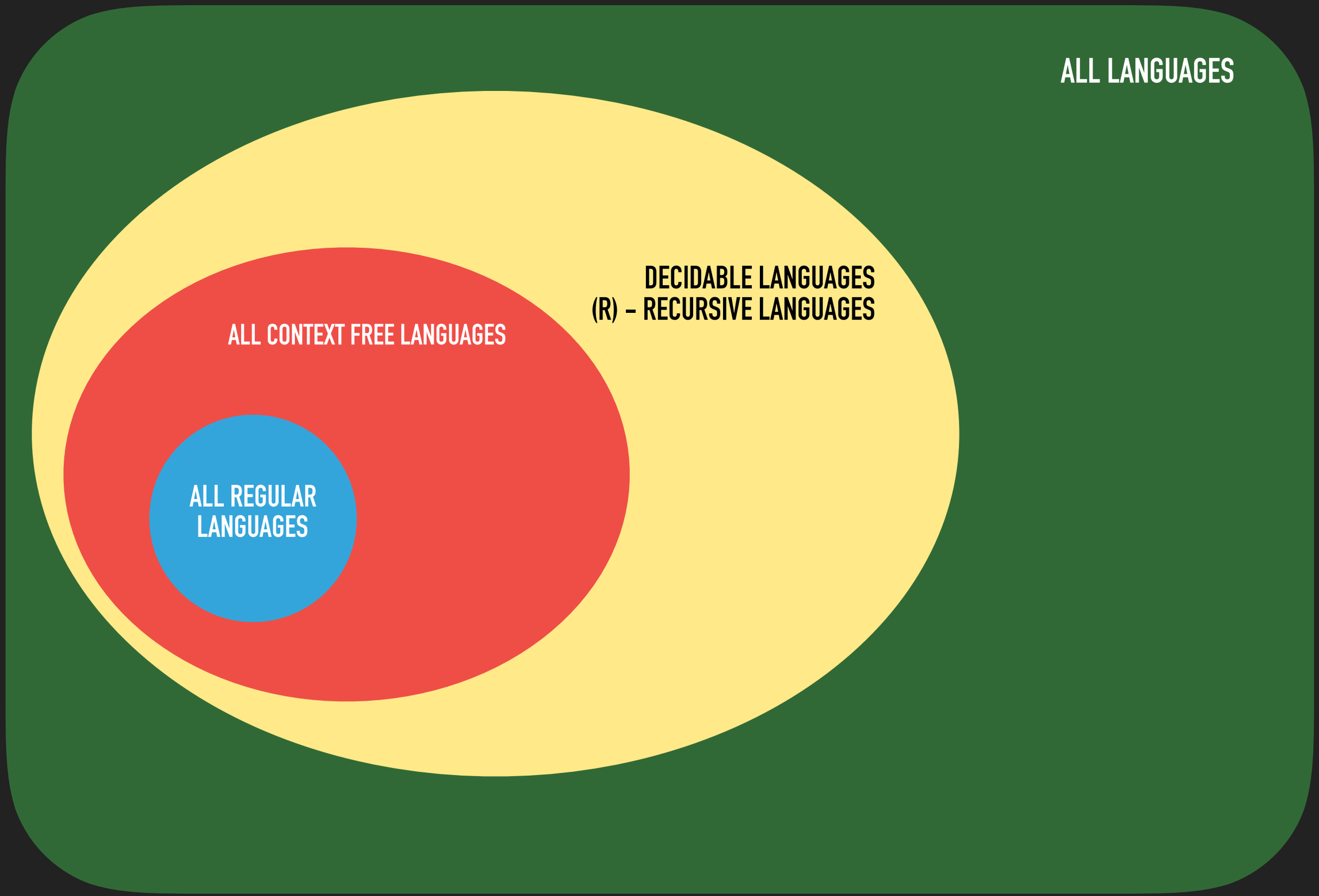
If an NFA will accept a string.

If a Regular Expression will accept a string.

If a CFG describes a string.

If a PDA accepts a string.

TURING DECIDABLE LANGUAGES



ACCEPTANCE PROBLEMS

M - Some Turing Machine

<M, w> - Some Turing Machine and its string input encoded as a string

$$A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM that accepts string } w \}$$

Can we build a Turing Machine that *decides* A_{TM}?

In other words, can we build a Turing Machine that tells us, for an arbitrary TM, M and an arbitrary input string w, if M will accept or reject w?

TURING LANGUAGES

A Turing machine accepts a string if it enters an accept state while processing that string.

A Turing machine rejects a string if it enters a reject state while processing that string.

A Turing machine loops forever on a string if it never enters an accept or reject state while processing that string.

A Turing machine halts on a string if it accepts or rejects that string.

TURING LANGUAGES

A language is "Turing Recognizable" if a Turing machine recognizes it.

A language is "Turing Decidable" if a Turing machine decides it, that is if the strings of that language cause the Turing machine to end up in an accept or reject state.

Can we build a Turing Machine that *recognizes* A_{TM}?

Can we build a Turing Machine that *decides* A_{TM}?