Intro to Theory Of Computing

510 FINAL PROJECT

Brain-Rot Language

Tuan Vu, Raj Kaura, Kobe Jordan

TABLE OF CONTENTS

Brain-Rot Language

Description of Formal Language

Our team has created a language representing the popular cultural adaptation of meme culture into everyday conversation, known as brain rot. This brain rot language is a formalization of modern internet slang, digital catchphrases, and cultural sayings that may seem nonsensical or even gibberish, but hold significant meaning within Generation Z. Our language captures the chaotic communication styles of the internet, showing how meaning can emerge from random strings.

The alphabet of our language consists of a set of widely recognized symbols: ∑ = {skibidi-toilet, fanum-tax, sigma, chat, quandale-dingle, glizzy, jit, karen, yns, opp, simp, aura, goat, session, baby-oil, low-taper-fade, rizzler, bro, rizzed, glazed, mogged, mewed, ratioed, cooked, yapped, faded, glazing, cooked, sus, the, ‘s, _}. In our alphabet we can define some of the symbols. For example, “Rizzler” refers to someone who is charismatic and charming. “Ratioed” refers to someone showing their superiority over someone and is often used to mock. “Glizzy” means hotdog, but derives its origins from a video of someone stuffing their face with hotdogs. “Fanum-tax,” a conjunction of two words, refers to a tax you must pay to your friend when ordering food. And “sus” refers to someone suspicious, etc.

The brain rot language follows specific patterns and rules to define valid strings. These strings encapsulate the structured nature of the internet slang, despite it seeming random. In our language, we have phrase patterns, and length constraints. Our phrase patterns follow “<subject><action><object> “ or <expression> format. An example of <subject><action><object> would be “jit_mogged_the_sigma,” which roughly translates to “the young person outshined or dominated his superior in terms of physical appearance.” An example of an expression would be “glazing,” which is used when someone is excessively or unnecessarily praising, defending, or overhyping someone. Lastly, strings that belong to our language must have 1, 5, or 9 symbols, any other number of symbols is invalid.

This language intends to see how languages evolve in online spaces and continue to be maintained through a shared understanding of an entire generation. This language is also used to educate people unfamiliar with brain rot to allow them to connect with Generation Z and even adapt some of the sayings to their personal lives. Specifically, this language was influenced by Professor Lohoefener's discussion in class about her own experience with her son using words that fall under this language. All in all, this brain rot language is a unique formal system that bridges the gap between internet culture, Generation Z, and other groups of people who are unfamiliar with brain rot.

Language

We've created subsections of the alphabet, defined below, that represent the <subject>, <action>, <object>, <expression> components, and connecting words.

$L = \{ vwxd^{j}wyw^{k}v^{k}w^{k}t^{k} + z \mid j, k \in \{0,1\}, d \in \{\text{'s}\}, w \in \{ _ \}, x, t \in \{$skibidi-toilet, fanum-tax, sigma, chat, quandale-dingle, glizzy, jit, karen, yns, opp, simp, aura, goat, session, baby-oil, low-taper-fade, rizzler, bro$\}, y \in \{$rizzed, glazed, mogged, mewed, ratioed, cooked, yapped, faded$\}, z \in \{$ratioed, simp, aura, sus, glazing, cooked, fanum-tax, skibidi-toilet$\}, v \in \{$the$\}\}$

Alphabet

$\sum = \{$skibidi-toilet, fanum-tax, sigma, chat, quandale-dingle, glizzy, jit, karen, yns, opp, simp, aura, goat, session, baby-oil, low-taper-fade, rizzler, bro, rizzed, glazed, mogged, mewed, ratioed, cooked, yapped, faded, glazing, cooked, sus, the, 's, _$\}$

Grammar

We chose not to simplify this to make it easier to interpret and understand the grammar

S -> VWXDWY | VWXDWYWVWX | Z

X -> skibidi-toilet | fanum-tax | sigma | chat | quandale-dingle | glizzy | jit | karen | yns | opp | simp | aura | goat | session | baby-oil | low-taper-fade | rizzler | bro

Y -> rizzed | glazed | mogged | mewed | ratioed | cooked | yapped | faded

Z -> ratioed | simp | aura | glazing | cooked | fanum-tax | skibidi-toilet | sus

D -> ‘s | λ

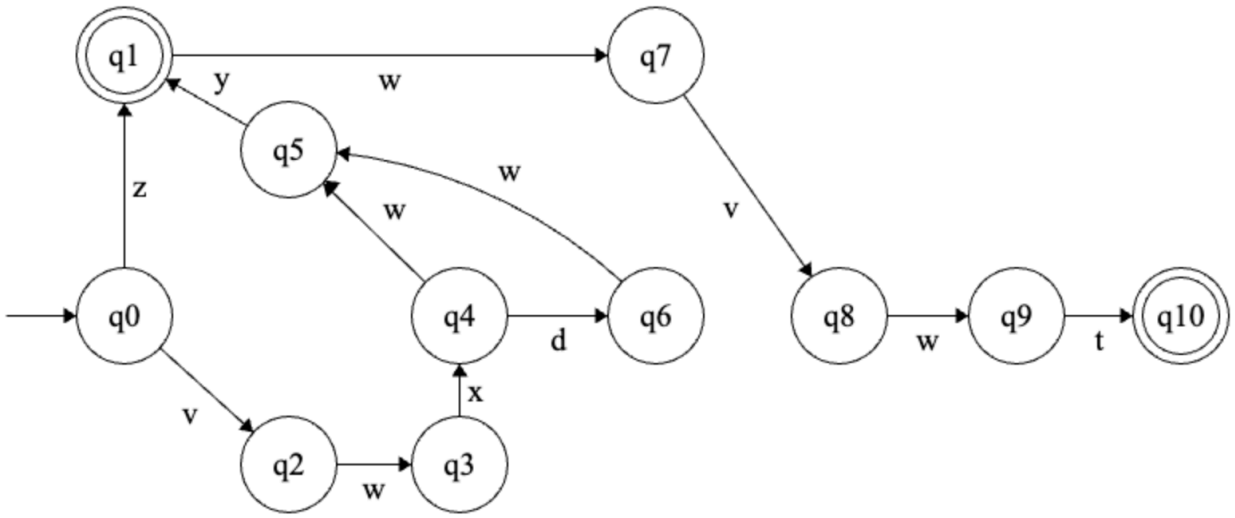
V -> the

W -> _

We’ve broken sections of our language into their respective <subject>

L = { $vwxd^jwyw^kv^kw^kt^k$ + z | j, k ∈ {0,1}, d ∈ {‘s}, w ∈ { _ }, x, t ∈ {skibidi-toilet, fanum-tax, sigma, chat, quandale-dingle, glizzy, jit, karen, yns, opp, simp, aura, goat, session, baby-oil, low-taper-fade, rizzler, bro}, y ∈ {rizzed, glazed, mogged, mewed, ratioed, cooked, yapped, faded}, z ∈ {ratioed, simp, aura, sus, glazing, cooked, fanum-tax, skibidi-toilet}, v ∈ {the}}

Automaton



x = skibidi-toilet | fanum-tax | sigma | chat | quandale-dingle | glizzy | jit | karen | yns | opp | simp | aura | goat | session | baby-oil | low-taper-fade | rizzler | bro

y = rizzed | glazed | mogged | mewed | ratioed | cooked | yapped | faded

z = ratioed | simp | aura | glazing | cooked | fanum-tax | skibidi-toilet | sus

v = the

w = _

d = ‘s

t = x

Data Structure

Our data structure, an NFA, represents the automaton in memory. A list of states, the alphabet, the initial state, and the final states define our NFA class.

Our NFA string representation returns a string that tells us the current NFA's initial state, final states, and the transitions of each state.

We had five functions for our NFA class: add_state(state), set_initial_state(state), add_final_state(state), run(input_string), and parse_input_to_nfa(input_string).

The "add_state" function takes an object with the class "State" and adds it to the list of states for the NFA.

The "set_initial_state" function takes an object with the class "State" and sets it to the initial state for the NFA.

The "add_final_state" function takes an object with the class "State" and adds it to the list of final states for the NFA.

The "run" function takes a string and runs it through the current NFA and returns True if the automata ends in a final state and False if the automata doesn't end in a final state.

The "parse_input_to_nfa" function takes our defined automaton data structure that includes information about its states & alphabet, parses it to set up transitions, and returns a fully initialized NFA.

Our state class for the automaton represents a state in our automaton, encapsulating its name, transitions, and methods for managing the transitions.

We had two functions for our State class: add_transition(symbol, state) and next_state(symbol).

The "add_transition" function takes a symbol and state and adds a transition from the current state to a given state.

The "next_state" function takes a symbol and retrieves the next state for the given symbol.

Testing

We chose not to simplify this to make it easier to interpret and understand the grammar

S -> VWXDWY | VWXDWYWVWX | Z

X -> skibidi-toilet | fanum-tax | sigma | chat | quandale-dingle | glizzy | jit | karen | yns | opp |

simp | aura | bop | goat | session | baby-oil | low-taper-fade | rizzler | bro

Y -> rizzed | glazed | mogged | mewed | ratioed | cooked | yapped | faded

Z -> ratioed | simp | aura | glazing | cooked | fanum-tax | skibidi-toilet | sus

D -> ‘s | λ

V -> the

W -> _

Test case	String	Accepted / Rejected	Explanation
1	the_sigma_cooked_the_yns	Accepted	The superior being destroyed/embarrassed the group of young, rowdy individuals. Full syntax (VWXDWYWVWX): “the_<subject>_<action>_the_<object>” Transitions: q0 -> q2 -> q3 -> q4 -> q5 -> q1 -> q7 -> q8 -> q9 -> q10
2	the_opp_faded_the_rizzler	Accepted	The enemy beat up the charismatic young man; this follows an internet meme of a fight between two people.

			Full syntax (VWXDWYWVWX): “the_<subject>_<action>_the_<object>” Transitions: q0 -> q2 -> q3 -> q4 -> q5 -> q1 -> q7 -> q8 -> q9 -> q10
3	the_goat’s_cooked	Accepted	The greatest of all time is not doing good / is not in a good spot Half syntax (VWXDWY): “the_<subject>_<action>” Transitions: q0 -> q2 -> q3 -> q4 -> q6 -> q5 -> q1
4	the_glizzy_cooked	Accepted	The hotdog tasted really good. Half syntax (VWXDWY): “the_<subject>_<action>” Transitions: q0 -> q2 -> q3 -> q4 -> q5 -> q1
5	fanum-tax	Accepted	An expression used to take food from someone justified as a mandatory tax. Expression (Z): “<expresssion> Transitions: q0 -> q1
6	skibidi-toilet	Accepted	Random expression, the origin lies in a TV series made for children, but often used to express how much brain rot an individual has; used satirically. Expression (Z): “<expresssion> Transitions: q0 -> q1

7	skibidi-toilet_fanum-tax _baby-oil	Rejected	Doesn’t follow the grammar structure above
8	apple	Rejected	It doesn’t exist in the alphabet of the language
9	the	Rejected	“the” is not an expression, therefore it cannot be accepted as a single symbol string. Ends on q2, not accepting state
10	the_	Rejected	Ends on q3, not accepting state
11	the_goat	Rejected	Ends on q4, not accepting state
12	the_goat_	Rejected	Ends on q5, not accepting state
13	the_goat’s_	Rejected	Ends on q6, not accepting state
14	the_goat_cooked_	Rejected	Ends on q7, not accepting state
15	the_goat_cooked_the	Rejected	Ends on q8, not accepting state
16	the_goat_cooked_the_	Rejected	Ends on q9, not accepting state
17		Rejected	An empty string cannot be accepted as it must either be 1, 5, or 9 symbols. It ends on q0, not accepting the state
18	the_sigma_cooked_the _yns_aura	Rejected	More symbols than defined (1,5,9) and doesn’t follow grammar structure