Tichafa Rinomhota tar351

Recursive Descent Parser for a subset of English Sentence

1. In lab 2 I created a Recursive Descent Parser for a subset of English sentence. For my development process I implemented and tested my grammar rules step by step. This is how I went about it; in total I wrote seven functions and this is their purpose.

firstOf\_sentence() = this checks to see if the next token is the first set of sentence

noun\_phrase() = this function is for the noun phrase production

firstOf\_noun\_phrase() = this checks if the next token belongs to the set of noun phrase

adjective\_phrase() – handles the adjective phrase production

firstOf\_adjective\_phrase() – checks if the next token belongs to the first set of adjective

verb\_phrase() – this is to handle the verb phrase production

firstOf\_verb\_phrase() – this checks to see if the next token belongs to the first set of verb phrase

sentence() – this is to handle sentence production

1. The sentence I shall use for number two shall be “the poor church mouse ate the money”

A paper with writing on it

AI-generated content may be incorrect.

Here is my parser output

enter <sentence>

| enter <noun phrase> 1

| | enter <adjective phrase> 1

| | | found ARTICLE: |the|

| | | found ADJECTIVE: |poor|

| | exit <adjective phrase> 1

| | found NOUN: |church|

| exit <noun phrase> 1

| enter <verb phrase> 1

| | found VERB: |mouse|

| exit <verb phrase> 1

| enter <noun phrase> 2

| | enter <adjective phrase> 2

| | | found ARTICLE: |ate|

| | | found ADJECTIVE: |the|

| | exit <adjective phrase> 2

| | found NOUN: |money|

| exit <noun phrase> 2

exit <sentence>

1. The role of the program call stack in the parsing process is very crucial. Each time a function calls another function the current functions’ state is pushed on to the stack which allows the program to return to that state after the function called is complete. The deepest call stack of the input was input 5 which was “his furious nose awkwardly sort of barely incompetently invented my timey-wimey homework.” This is because this sentence contains lots of adverbs. What happens when an adverb is called is due to the function verb\_phrase being recursive it creates another level in the call stack. So this makes the fifth input have the deepest call stack among the rest of the inputs.

Sources I used for this exercise were

* lecture notes
* <https://www.geeksforgeeks.org>
* https://www.reddit.com/r/grammar/comments/1c3ky5u/parse\_tree\_of\_a\_english\_sentence/?rdt=60874