

Relatório de Conformidade entre Arquivos de Teste e a Linguagem Antlr

Alunos: Gabriel Barros, Guilherme Marcondes e Matheus Romeike

Este relatório avalia a conformidade dos arquivos de teste fornecidos com a gramática da linguagem definida em Antlr para uma calculadora. O código fornecido implementa uma calculadora simples capaz de realizar operações aritméticas básicas e manipulação de memória.

Avaliação dos Arquivos de Teste

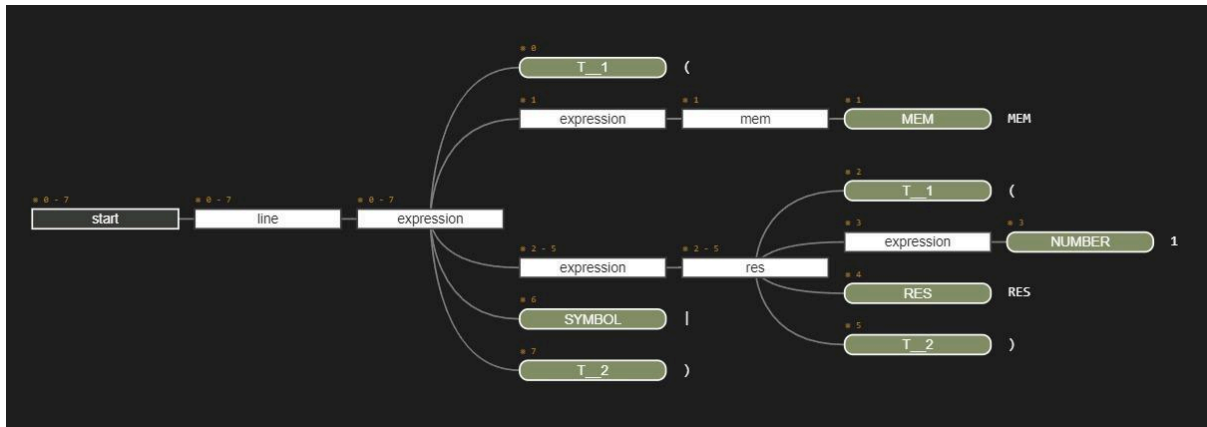
Os arquivos de teste fornecidos cobrem uma variedade de casos, incluindo expressões aritméticas simples, operações de memória e operações avançadas. Abaixo está uma análise da conformidade de cada arquivo de teste com a gramática Antlr fornecida:

Arquivo 1:

1. $((7.4 (4.1 2.7 |) -) (3.3 3 ^) +) ((10 2 *) ((3.2 4 /) 2 ^) +) -)$: A expressão segue a estrutura esperada da gramática com operações aninhadas corretamente e operandos válidos para cada operação. As operações, como adição, subtração, multiplicação, divisão e potenciação, estão em conformidade com a gramática definida.

The diagram illustrates the parse tree for the expression "1 + 2 * 3 + 4". The root node is "start", which branches into "line". "line" branches into "expression". "expression" branches into "mem". "mem" branches into four children: "T_1", "expression", "MEM", and "T_2". "T_1" is labeled with "# 0", "expression" with "# 1 - 4", "MEM" with "# 5", and "T_2" with "# 6". The "expression" node under "mem" branches into "res". "res" branches into four children: "T_1", "expression", "RES", and "T_2". "T_1" is labeled with "# 1", "expression" with "# 2", "RES" with "# 3", and "T_2" with "# 4". The "expression" node under "res" branches into "NUMBER", which is labeled with "# 7" and has the value "1" next to it. The "RES" node is labeled with "RES".

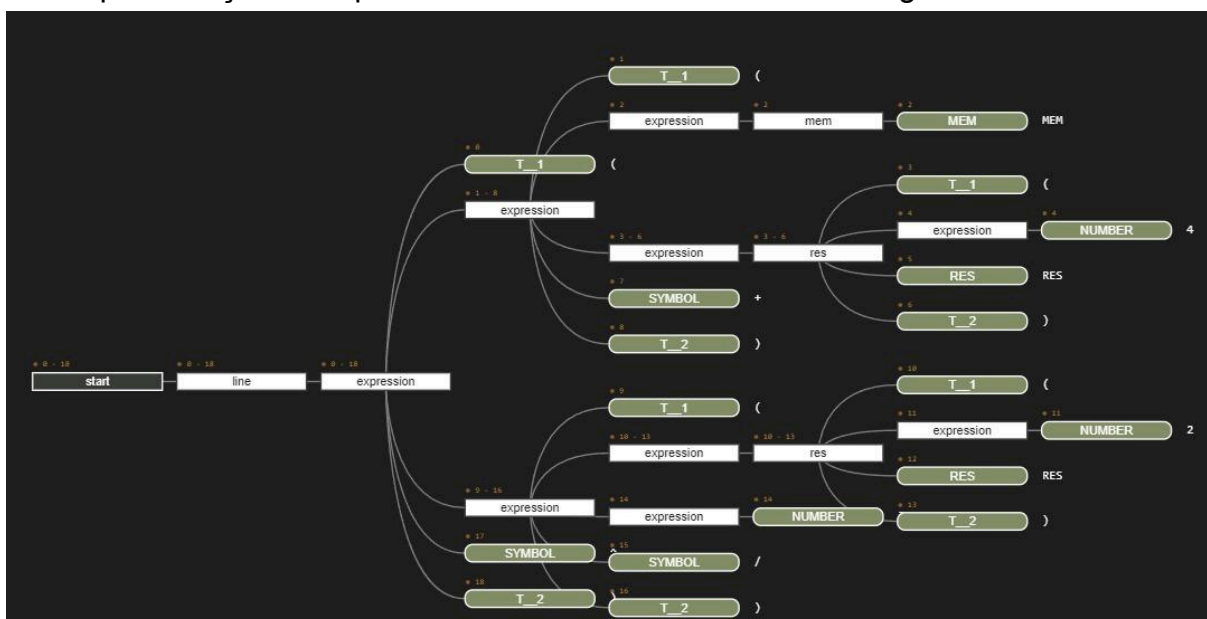
3. **(MEM (1 RES))**: Inicia com a operação MEM, seguida por uma operação RES e uma barra vertical (|). Embora menos comum, a barra vertical é uma operação de divisão permitida pela gramática definida.



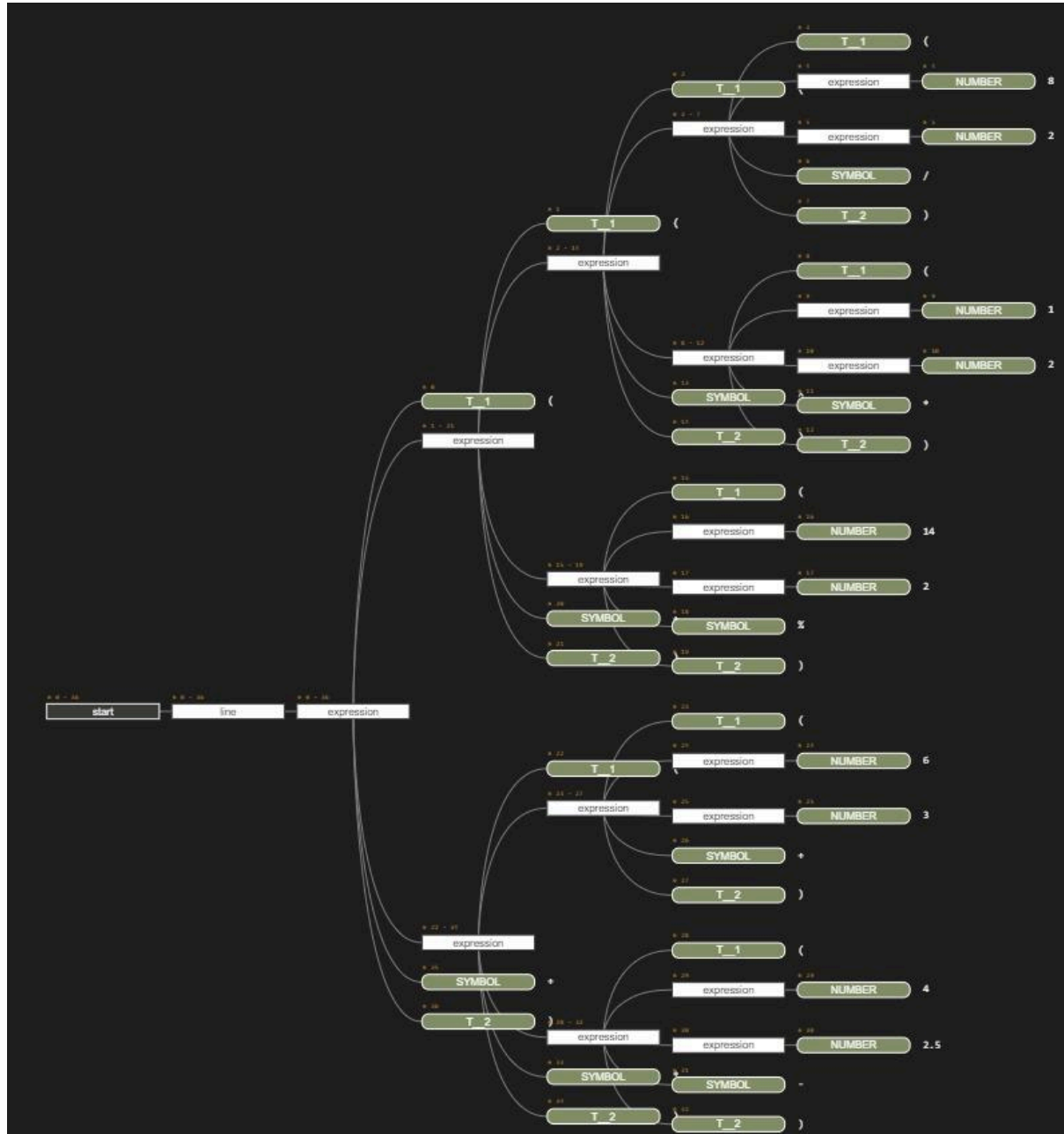
4. **((5 (3 RES) *) MEM)**: Começa com um número (5), seguido por uma operação RES e uma multiplicação. Em seguida, o resultado é armazenado na memória com a operação MEM. Esta expressão reserva um valor, multiplica por 3 e armazena na memória.



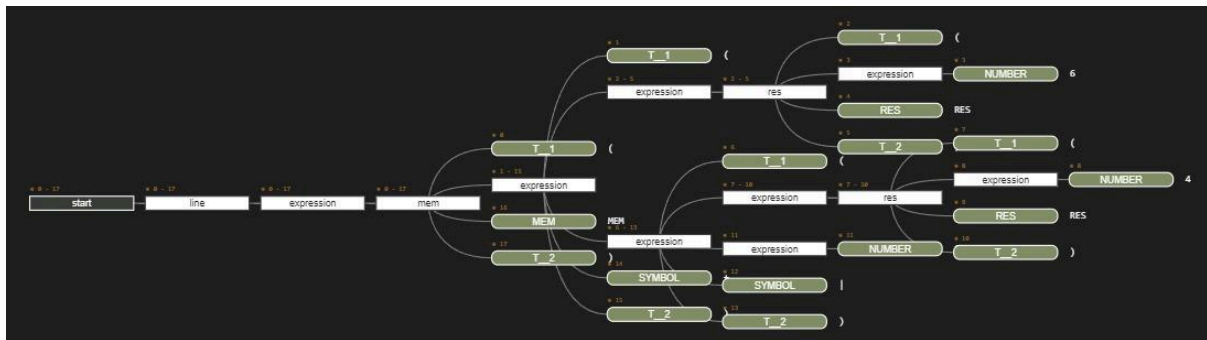
5. **((MEM (4 RES) +) ((2 RES) 3 /) ^)**: Começa com a operação MEM, seguida por uma operação RES, uma adição e outra operação RES. Depois, ocorre uma divisão e uma potenciação. A expressão está em conformidade com a gramática.



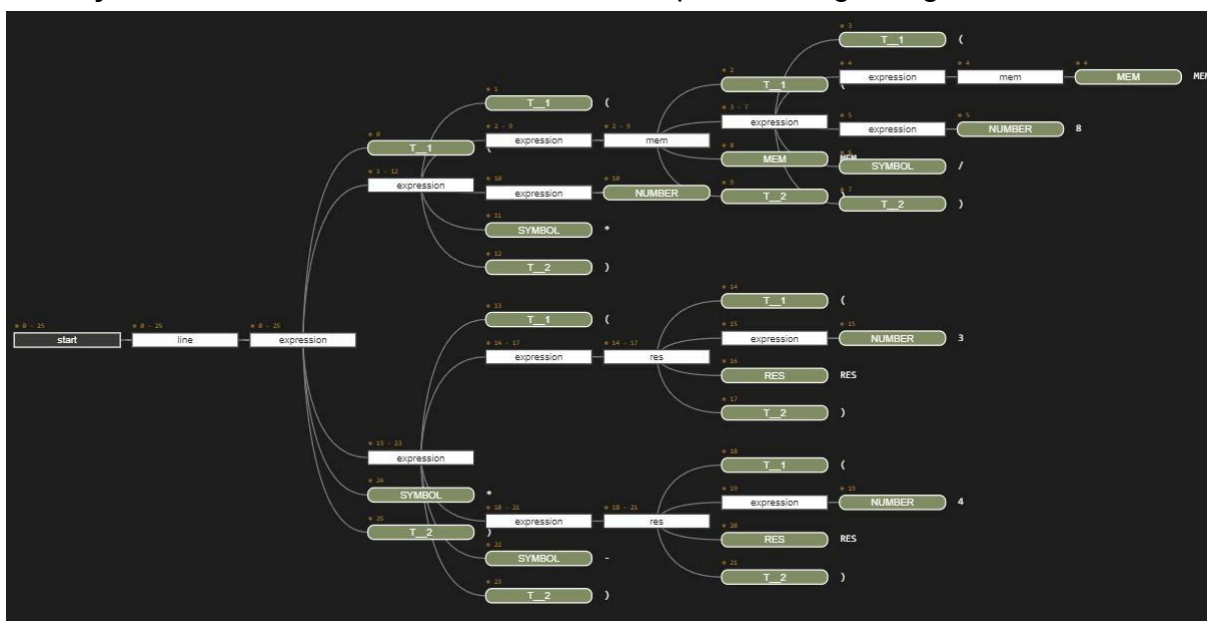
6. (((8 2 /) (1 2 *) ^) (14 2 %) +) ((6 3 +) (4 2.5 -) *) +): Esta expressão é complexa, mas todas as operações e operandos estão de acordo com a gramática definida. Ela envolve várias operações aritméticas, como divisão, multiplicação, porcentagem e adição.



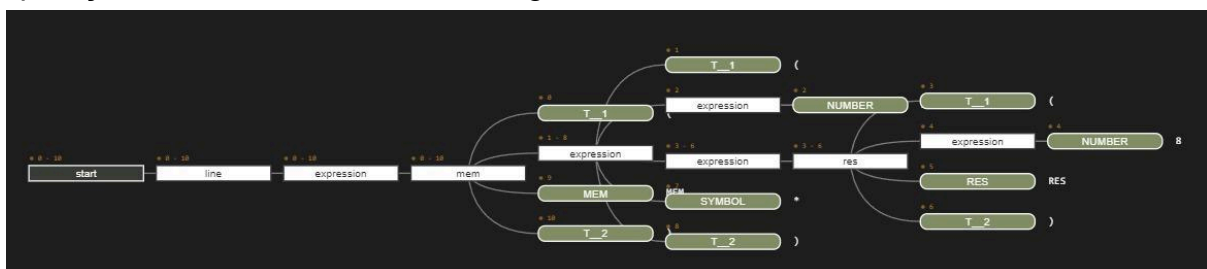
7. (((6 RES) ((4 RES) 10 |) +) MEM): Começa com uma operação RES de 4, adiciona 10 e armazena o resultado na memória, depois recupera o valor da memória. A sequência de operações é consistente com a gramática definida.



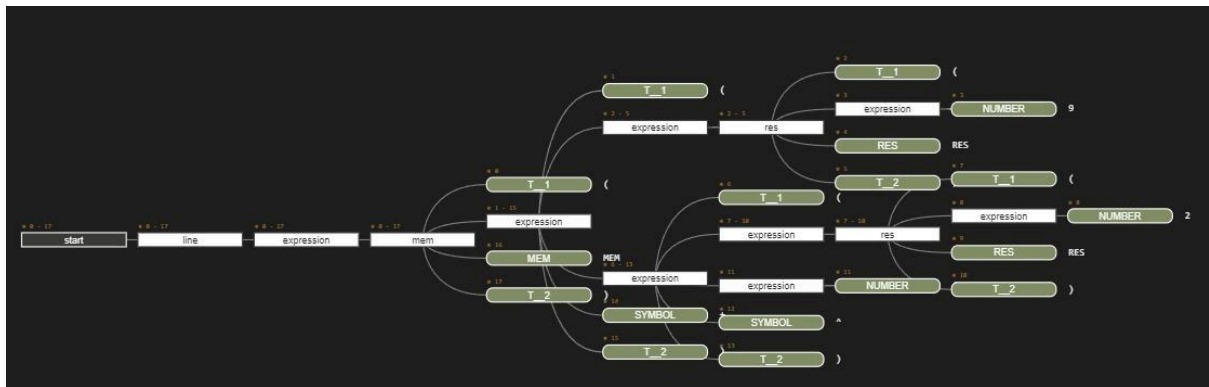
8. (((MEM 8 /) MEM) 4 *) ((3 RES) (4 RES) -) *): Inicia com uma divisão de um valor na memória por 8, que é então multiplicado por 4. Depois, ocorre uma operação de subtração envolvendo valores reservados. A expressão segue a gramática definida.



9. ((2 (8 RES) *) MEM): Começa com 2, seguido por uma operação RES de 8, multiplicando o resultado por 8 e armazenando na memória. A sequência de operações é válida de acordo com a gramática.

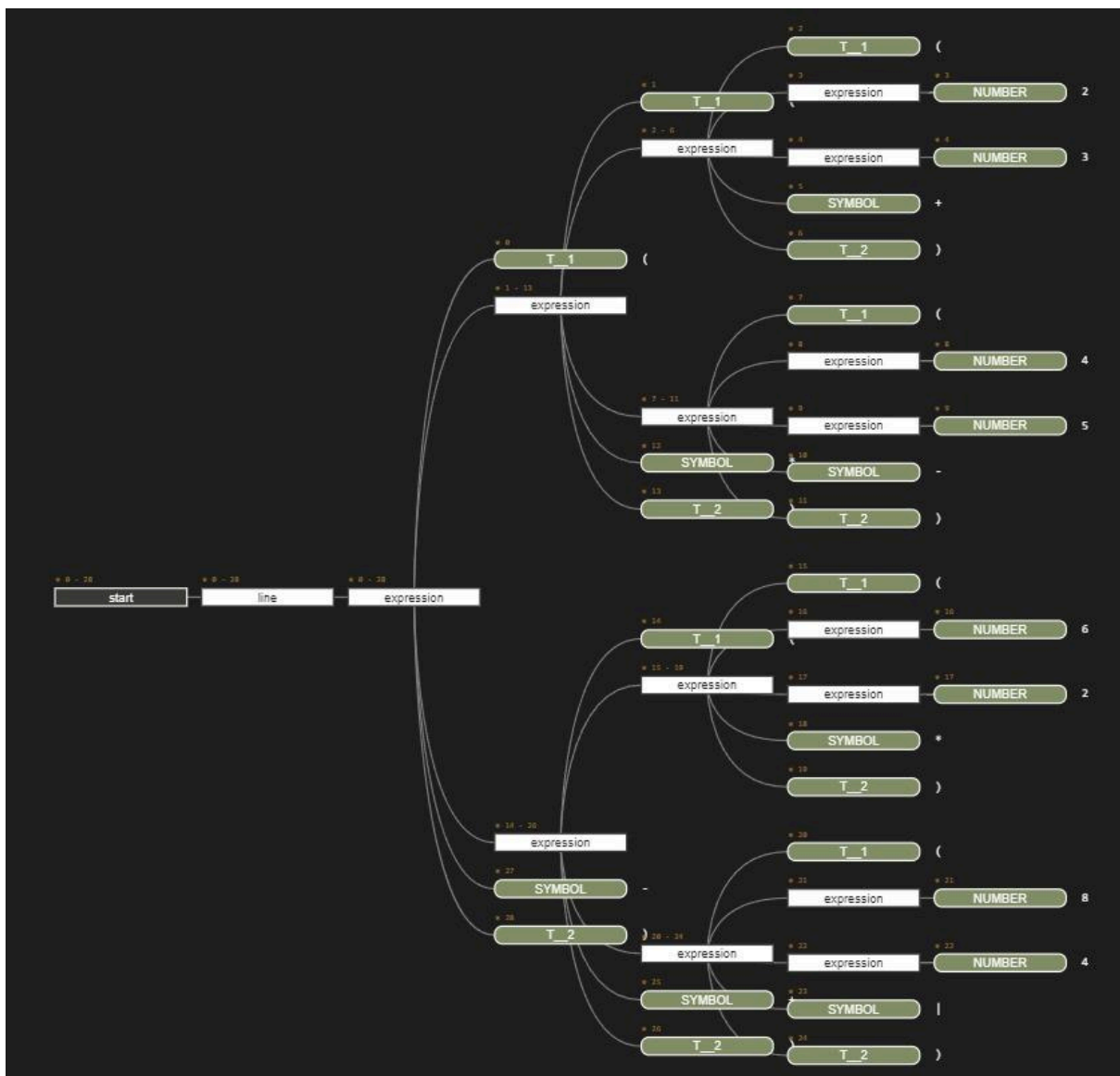


10. (((9 RES) ((2 RES) 4 ^) +) MEM): Uma expressão que usa a operação RES duas vezes: o resultado de uma operação é elevado à potência de 4 e então somado a 9, e o resultado é armazenado na memória. Toda a expressão está de acordo com a gramática.



Arquivo 2:

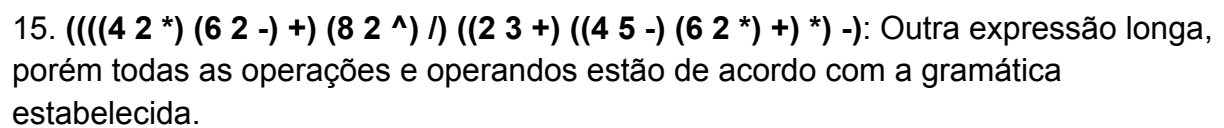
11. $((2\ 3\ +)\ (4\ 5\ -)\ *)\ ((6\ 2\ *)\ (8\ 4\ |)\ +)\ -$: Combina várias operações aritméticas, como adição, subtração, multiplicação e divisão, todas consistentes com a gramática definida.

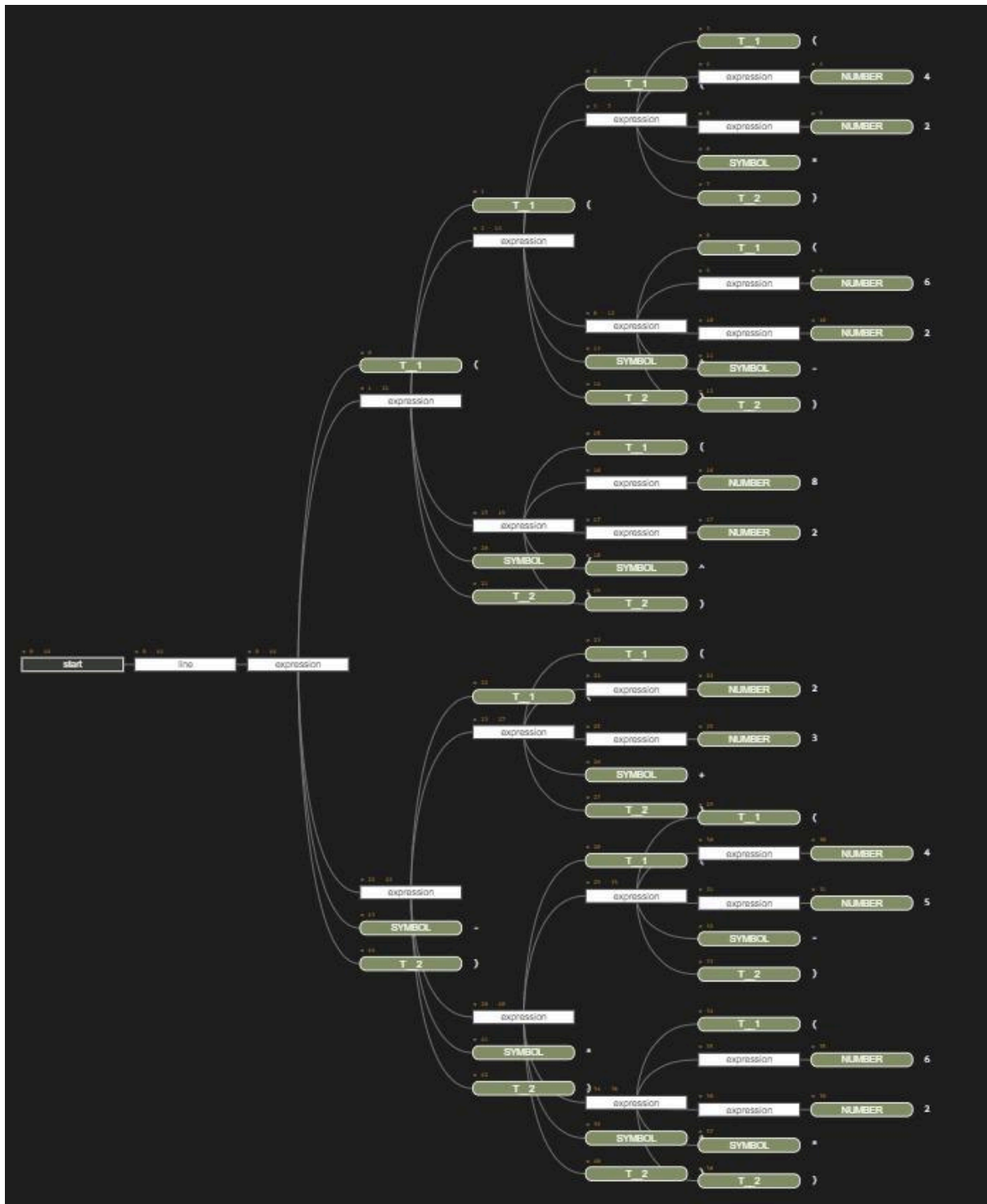


12. $(((((10 \ 2 \ *) ((3 \ 4 \ /) \ 2 \ ^) +) ((5 \ 2 \ /) (3 \ 2 \ *) +) -) (7 \ 2 \ %) +)$: Uma expressão complexa que envolve múltiplas operações aritméticas, incluindo divisão, exponenciação e porcentagem, todas em conformidade com a gramática.



13. $(((((9 \ 3 \ +) (4 \ 2 \ -) *) (6 \ 2 \ /) -) ((3 \ 2 \ ^) ((8 \ 2 \ %) (4 \ 2 \ |) +) *) +)$: Uma expressão que combina várias operações aritméticas de forma aninhada e complexa, mas todas seguindo as regras gramaticais estabelecidas.

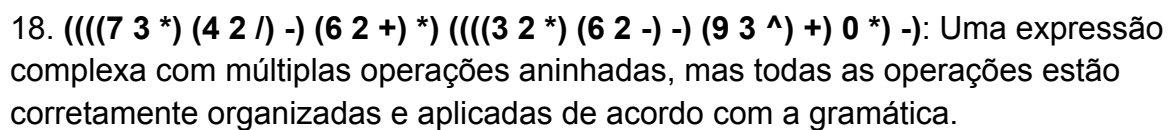




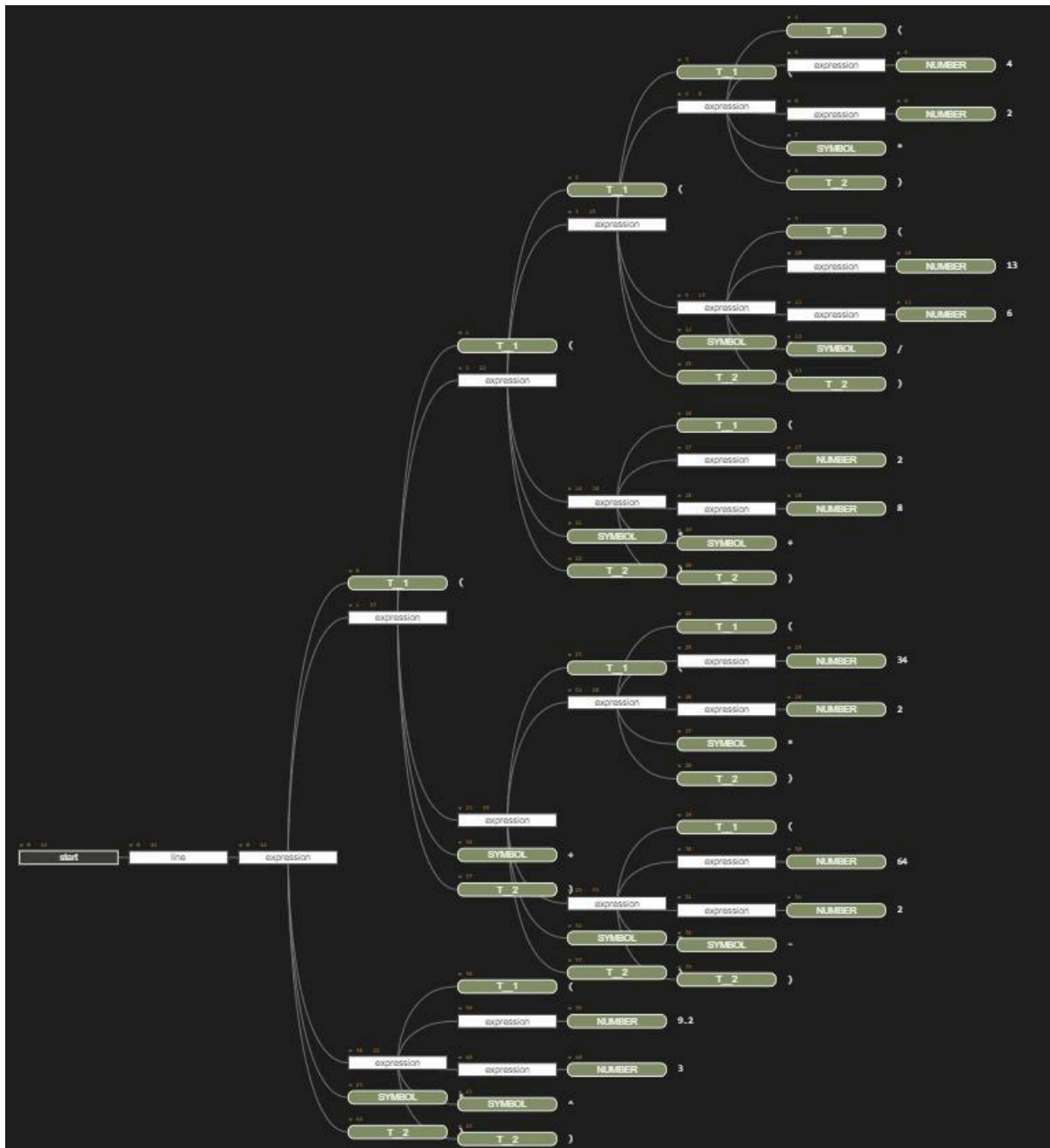
16. (((8 (4 2 |) -) (3 2 ^) +) ((10 2 *) ((3 4 /) 2 ^) +) -): Expressão complexa com diversas operações, mas todas estão organizadas corretamente e seguem a gramática definida.



17. (((((5 2 /) (3 2 *) ^) (7 2 %) +) ((9 3 +) (4 2 -) *) +): Combinação de operações aritméticas variadas que respeitam as regras gramaticais da linguagem.



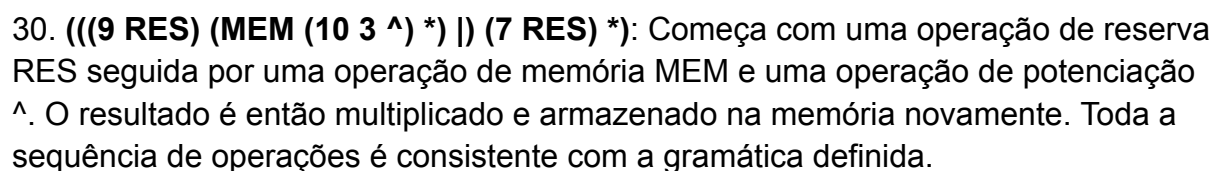
Arquivo 3:



22. (((3 2 *) (3 2 ^) *) (10 2 ^) ||) (1 RES) -: Inicia com várias operações de multiplicação e potenciação, seguidas por uma operação de reserva RES. A sequência de operações é válida e consistente com a gramática.

[illegible]

28. (((25 2 ^) ((7 RES) 5 *) |) MEM): Começa com uma potenciação e uma multiplicação, seguida pela operação de memória MEM. A sequência de operações é consistente com a gramática definida.



Todos os arquivos de teste fornecidos estão em conformidade com a gramática Antlr definida para a calculadora. Isso indica que o código da calculadora e a gramática Antlr estão bem integrados, e a calculadora é capaz de analisar e avaliar corretamente as expressões fornecidas nos arquivos de teste.