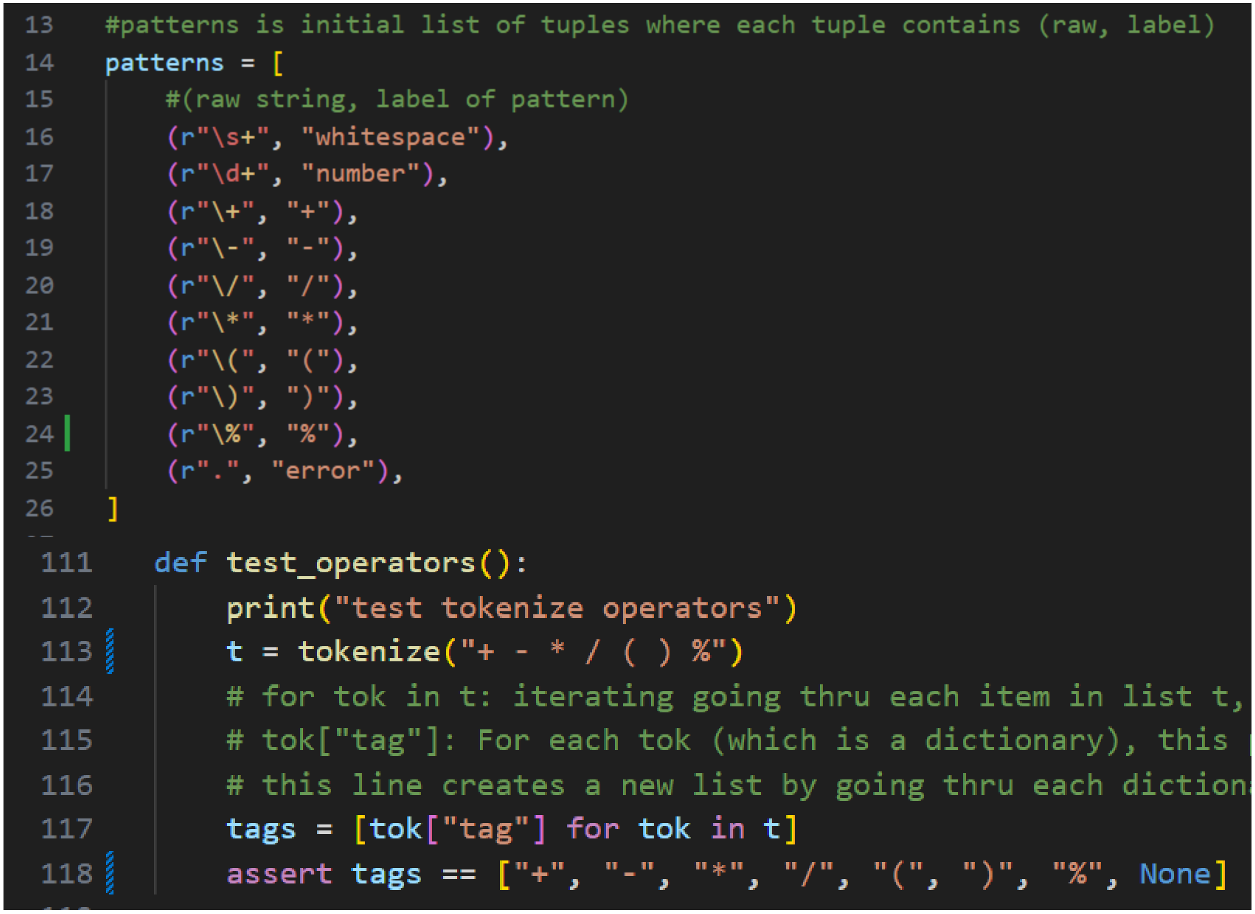
CS33101 – Structure of Programming Languages

Homework 1: Add an operator

2/15/2026

Yusairah Haque

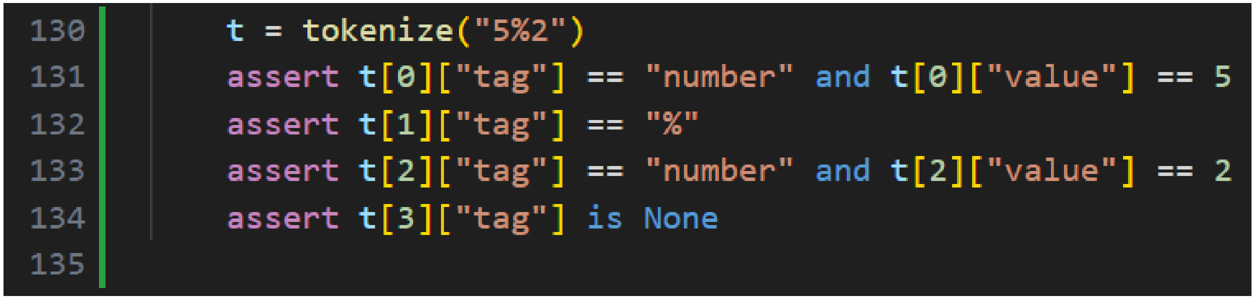
1. Changes and additions made to tokenizer.py



```
#patterns is initial list of tuples where each tuple contains (raw, label)
patterns = [
    #(raw string, label of pattern)
    (r"\s+", "whitespace"),
    (r"\d+", "number"),
    (r"\+", "+"),
    (r"\-", "-"),
    (r"\/", "/"),
    (r"\*", "*"),
    (r"\(", "("),
    (r"\)", ")"),
    (r"\%", "%"),
    (r".", "error"),
]
```

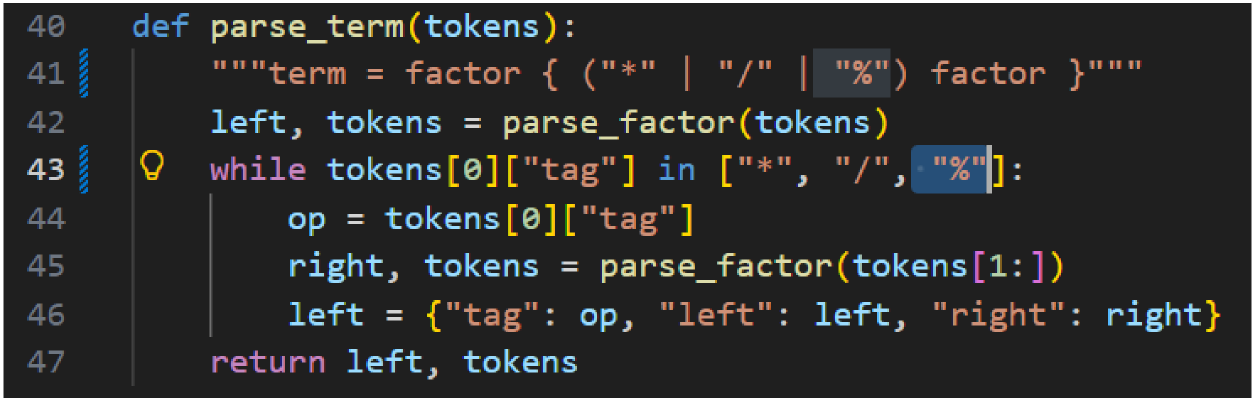
```
def test_operators():
    print("test tokenize operators")
    t = tokenize("+ - * / ( ) %")
    # for tok in t: iterating going thru each item in list t,
    # tok["tag"]: For each tok (which is a dictionary), this
    # this line creates a new list by going thru each diction
    tags = [tok["tag"] for tok in t]
    assert tags == ["+", "-", "*", "/", "(", ")", "%", None]
```

Added test cases in def test_expression()



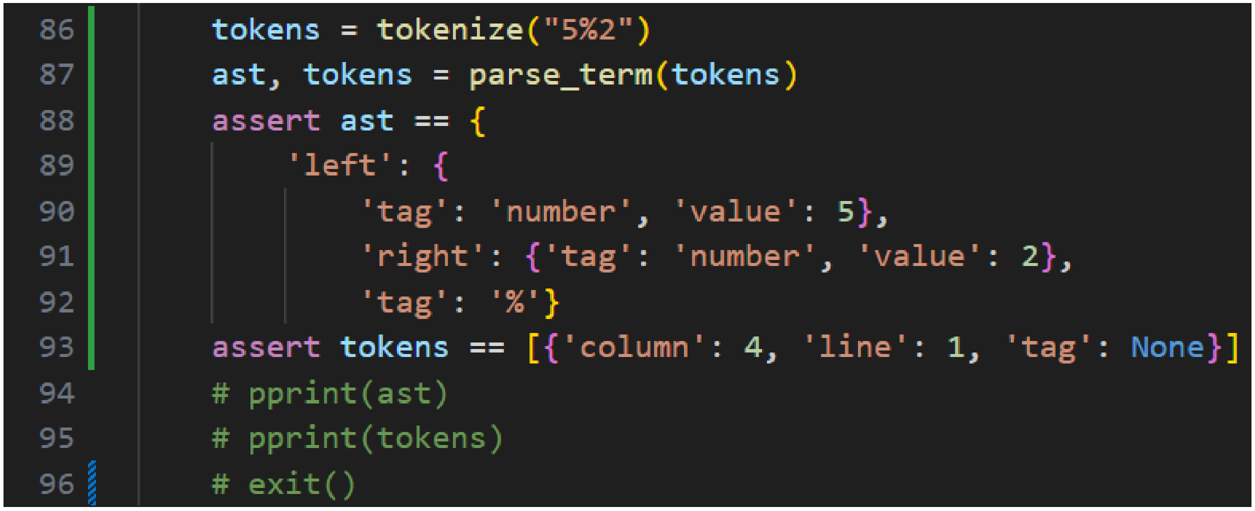
```
    t = tokenize("5%2")
    assert t[0]["tag"] == "number" and t[0]["value"] == 5
    assert t[1]["tag"] == "%"
    assert t[2]["tag"] == "number" and t[2]["value"] == 2
    assert t[3]["tag"] is None
```

2. Changes and additions made to parser.py



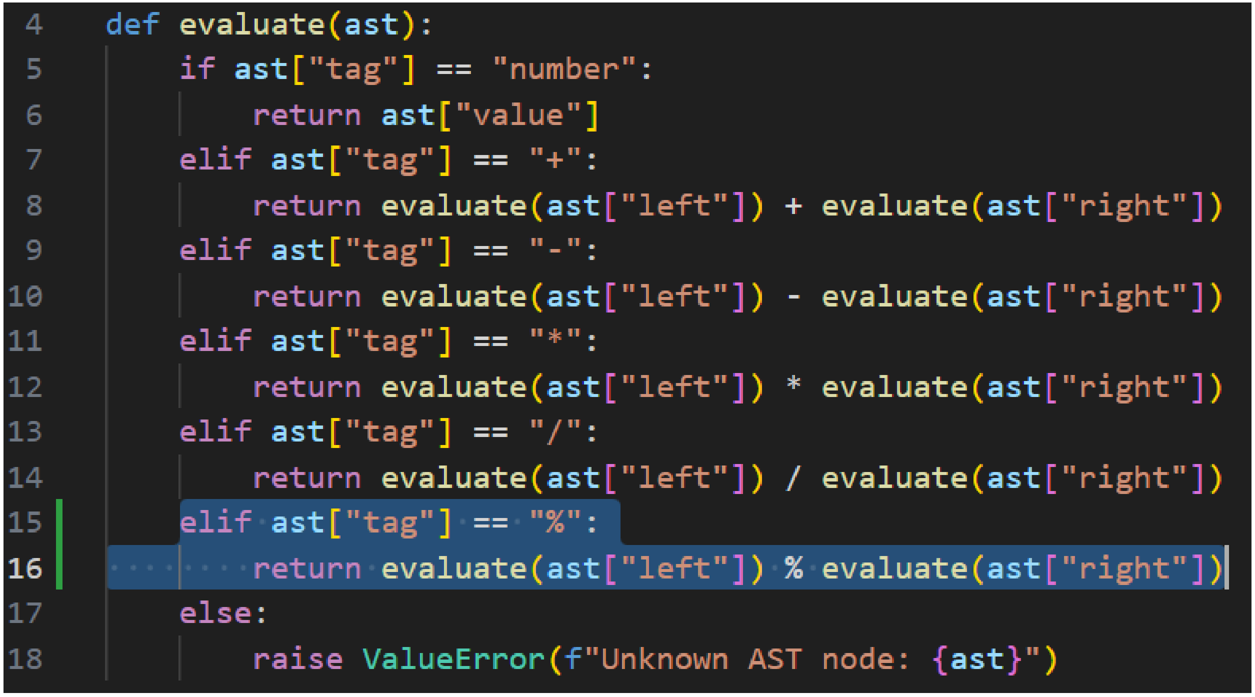
```python
def parse_term(tokens):
    """term = factor { ("*" | "/" | "%") factor }"""
    left, tokens = parse_factor(tokens)
    while tokens[0]["tag"] in ["*", "/", "%"]:
        op = tokens[0]["tag"]
        right, tokens = parse_factor(tokens[1:])
        left = {"tag": op, "left": left, "right": right}
    return left, tokens
```

Added test cases in def test_parse_term()



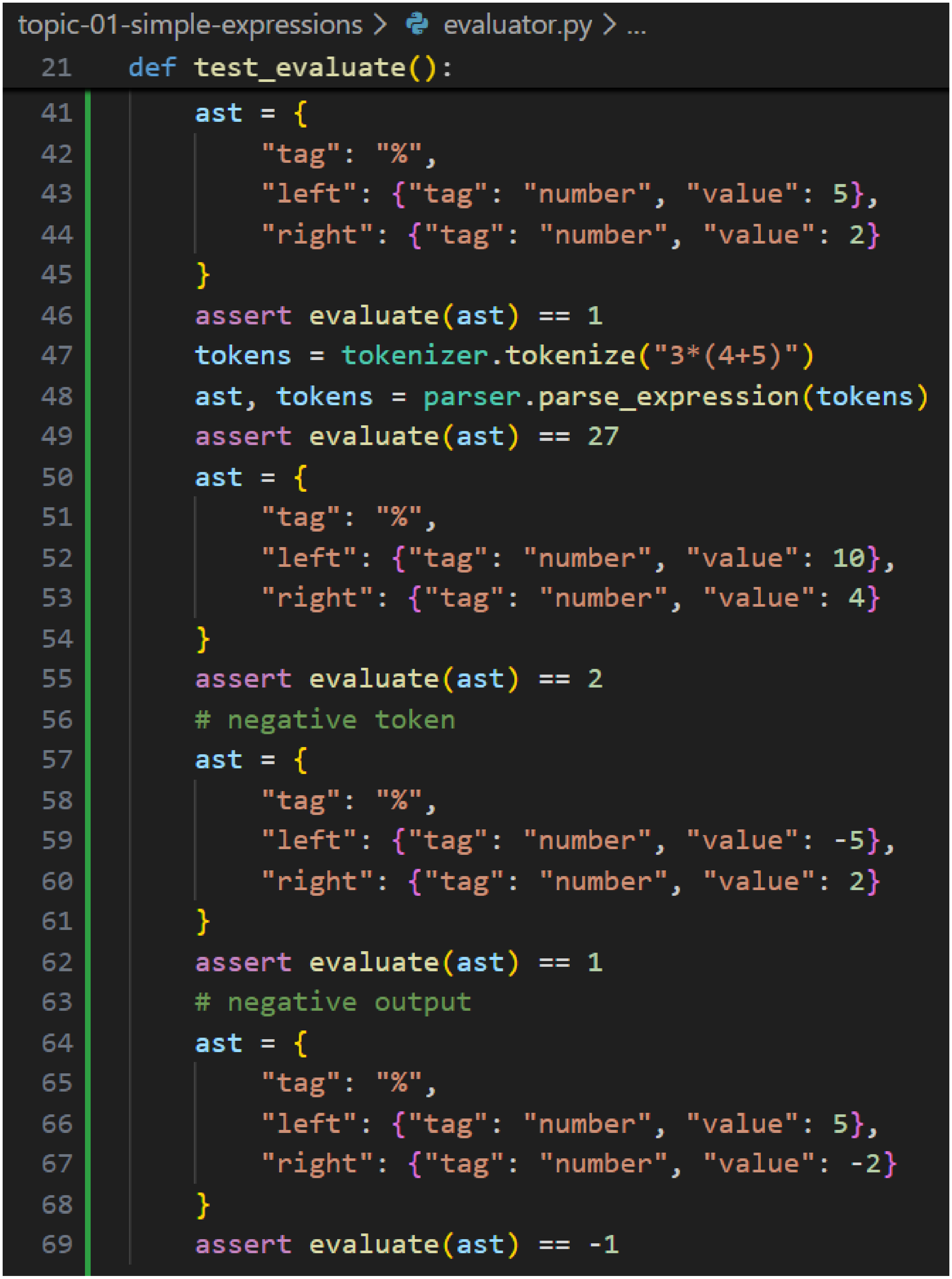
```python
        tokens = tokenize("5%2")
        ast, tokens = parse_term(tokens)
        assert ast == {
            'left': {
                'tag': 'number', 'value': 5},
                'right': {'tag': 'number', 'value': 2},
                'tag': '%'}
        assert tokens == [{'column': 4, 'line': 1, 'tag': None}]
        # pprint(ast)
        # pprint(tokens)
        # exit()
```

3. Changes and additions made to evaluator.py



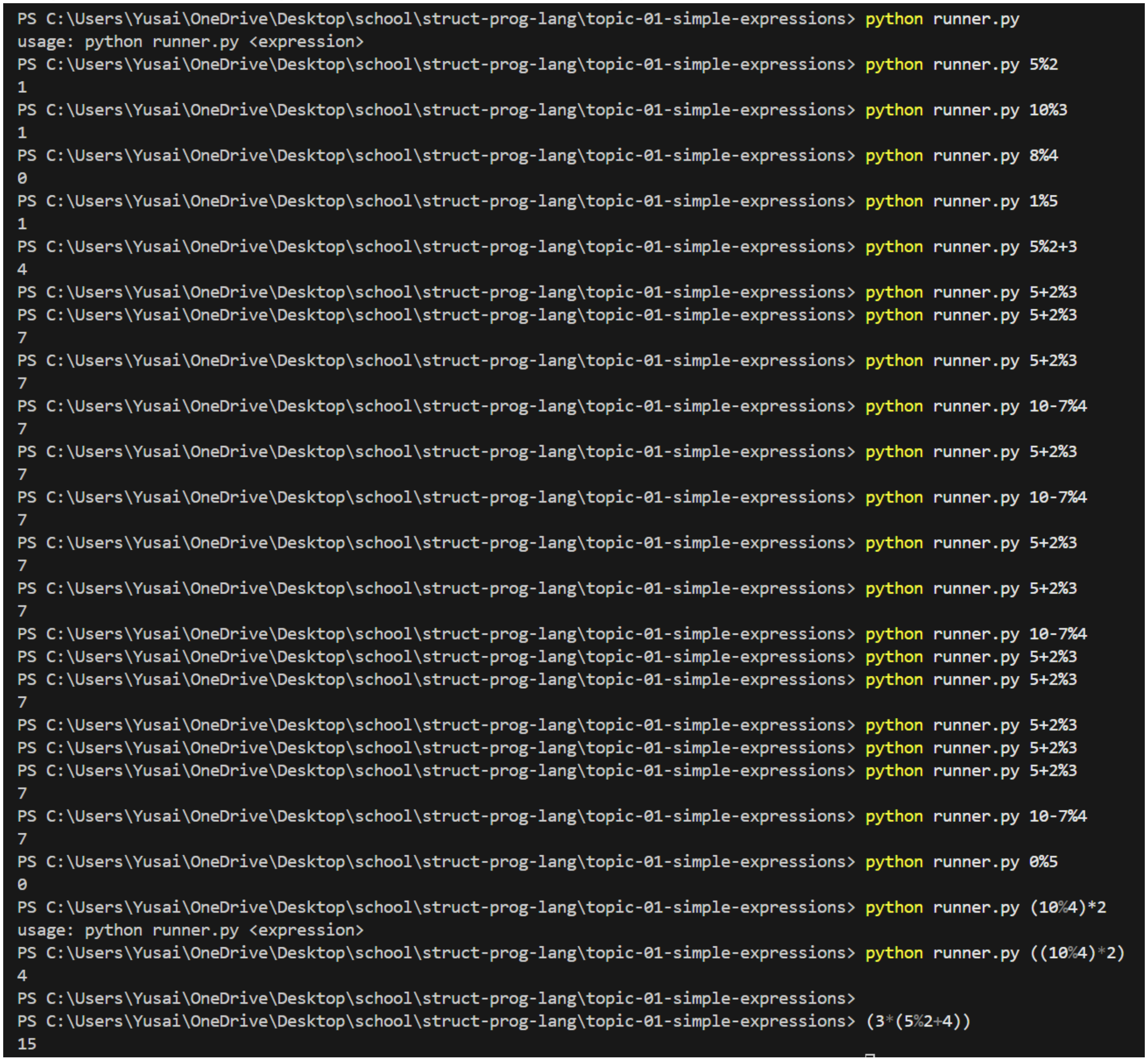
```python
def evaluate(ast):
    if ast["tag"] == "number":
        return ast["value"]
    elif ast["tag"] == "+":
        return evaluate(ast["left"]) + evaluate(ast["right"])
    elif ast["tag"] == "-":
        return evaluate(ast["left"]) - evaluate(ast["right"])
    elif ast["tag"] == "*":
        return evaluate(ast["left"]) * evaluate(ast["right"])
    elif ast["tag"] == "/":
        return evaluate(ast["left"]) / evaluate(ast["right"])
    elif ast["tag"] == "%":
        return evaluate(ast["left"]) % evaluate(ast["right"])
    else:
        raise ValueError(f"Unknown AST node: {ast}")
```

Added test cases in def test_evaluate()



```python
def test_evaluate():
    ast = {
        "tag": "%",
        "left": {"tag": "number", "value": 5},
        "right": {"tag": "number", "value": 2}
    }
    assert evaluate(ast) == 1
    tokens = tokenizer.tokenize("3*(4+5)")
    ast, tokens = parser.parse_expression(tokens)
    assert evaluate(ast) == 27
    ast = {
        "tag": "%",
        "left": {"tag": "number", "value": 10},
        "right": {"tag": "number", "value": 4}
    }
    assert evaluate(ast) == 2
    # negative token
    ast = {
        "tag": "%",
        "left": {"tag": "number", "value": -5},
        "right": {"tag": "number", "value": 2}
    }
    assert evaluate(ast) == 1
    # negative output
    ast = {
        "tag": "%",
        "left": {"tag": "number", "value": 5},
        "right": {"tag": "number", "value": -2}
    }
    assert evaluate(ast) == -1
```

4. Outputs of some % expressions with the runner.py



```
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py
usage: python runner.py <expression>
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5%2
1
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 10%3
1
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 8%4
0
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 1%5
1
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5%2+3
4
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 10-7%4
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 10-7%4
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 10-7%4
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 5+2%3
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 10-7%4
7
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py 0%5
0
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py (10%4)*2
usage: python runner.py <expression>
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> python runner.py ((10%4)*2)
4
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions>
PS C:\Users\Yusai\OneDrive\Desktop\school\struct-prog-lang\topic-01-simple-expressions> (3*(5%2+4))
15
```