```
class Parser:
    def __init__(self, input_str):
        self.input_str = input_str
        self.index = 0
        self.derivation = []

    def parse_E(self):
        start = self.index
        if self.parse_T():
            while self.index < len(self.input_str) and self.input_str[self.index] == '+':
                self.index += 1
                if self.parse_T():
                    self.derivation.append("E → E + T")
            return True
        self.index = start
        return False

    def parse_T(self):
        start = self.index
        if self.parse_F():
            while self.index < len(self.input_str) and self.input_str[self.index] == '*':
                self.index += 1
                if self.parse_F():
                    self.derivation.append("T → T * F")
                else:
                    return False  # Require at least one F expression after *
            return True
        self.index = start
        return False

    def parse_F(self):
        start = self.index
        if self.index < len(self.input_str) and self.input_str[self.index] == '(':
            self.index += 1
            if self.parse_E():
                if self.index < len(self.input_str) and self.input_str[self.index] == ')':
                    self.index += 1
                    self.derivation.append("F → (E)")
                    return True
        elif self.index < len(self.input_str) and self.input_str[self.index] == 'a':
            self.index += 1
            self.derivation.append("F → a")
            return True
        self.index = start
        return False

    def parse(self):
        if self.parse_E() and self.index == len(self.input_str):
            self.derivation.append("E")
            return True
        return False


input_str = "(a+a)*a"
parser = Parser(input_str)
result = parser.parse()
print("The input string is valid:", result)
if result:
  print("Derivation steps:")
  for step in reversed(parser.derivation):
    print(step)
else:
  print("Incorrect Structure")
```

```
The input string is valid: True
Derivation steps:
E
T → T * F
F → a
F → (E)
E → E + T
F → a
F → a
```

```
input_str = "a*a*"
parser = Parser(input_str)
result = parser.parse()
print("The input string is valid:", result)
if result:
  print("Derivation steps:")
  for step in reversed(parser.derivation):
    print(step)
else:
  print("Incorrect Structure")
```

```
The input string is valid: False
Incorrect Structure
```

✓ 0s completed at 10:30 PM