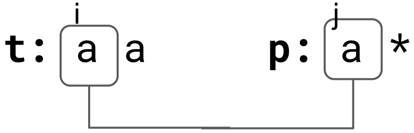
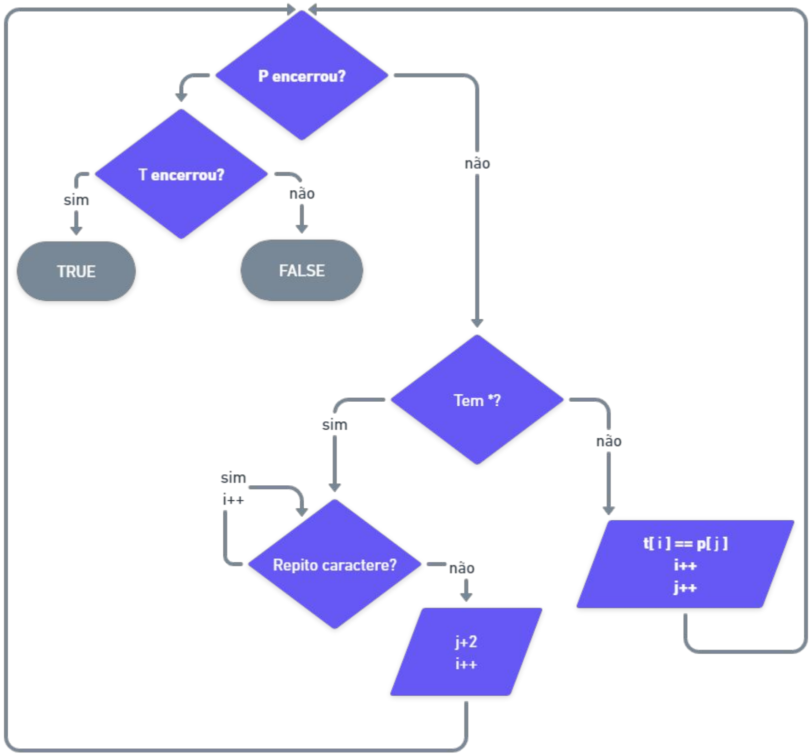
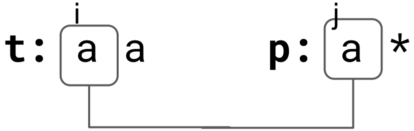
Simulação
Regular Expression Matching

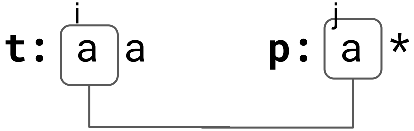
Teste 1


i
j
t: a a
p: a *

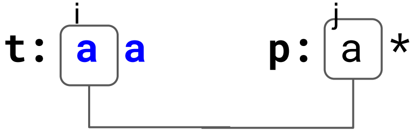

P encerrou?
T encerrou?
sim
não
TRUE
FALSE
não
Tem *?
sim
não
sim
i++
Repito caractere?
não
j+2
i++
t[i] == p[j]
i++
j++



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



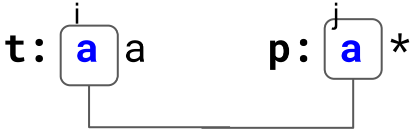
i

t: a a

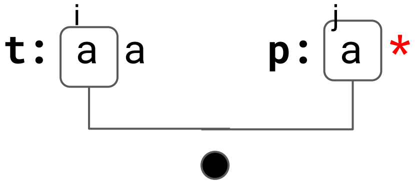
j

p: a *

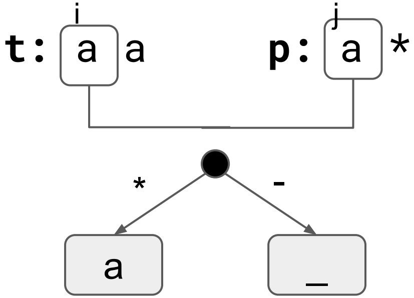
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



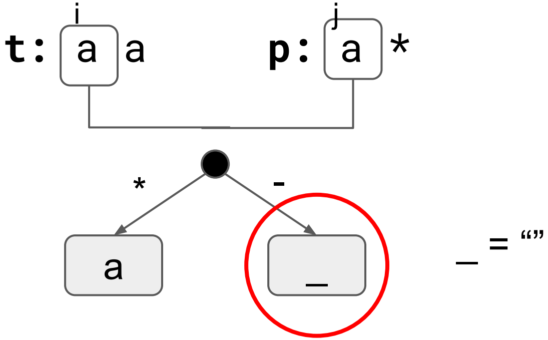
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



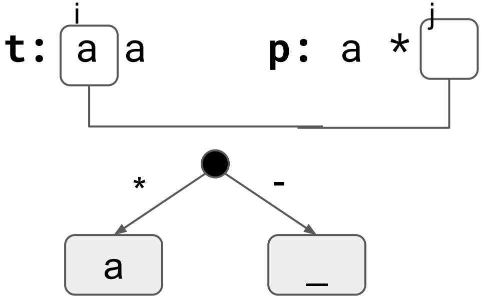
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
TRUE    first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



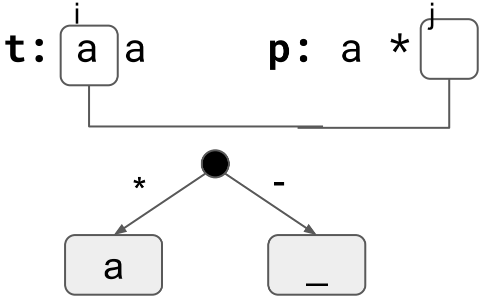
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
TRUE first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



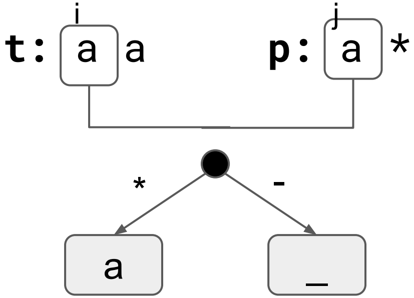
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
TRUE    first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

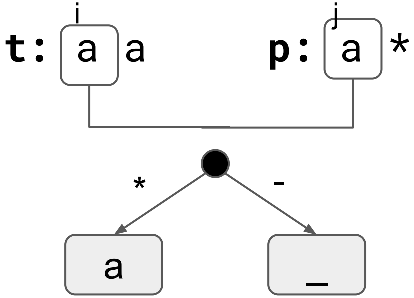


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

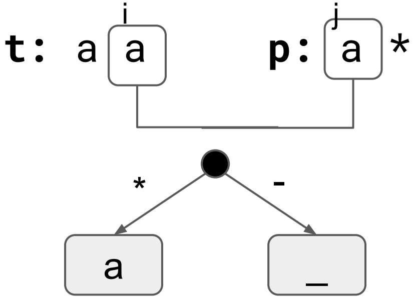


True

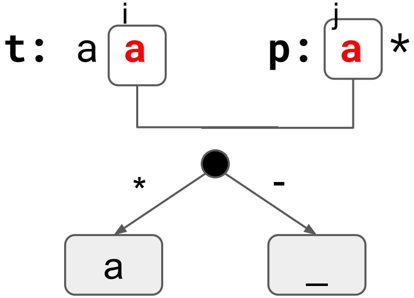
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
                     False              True
      else:
        return first_match and match(i+1, j+1)
```

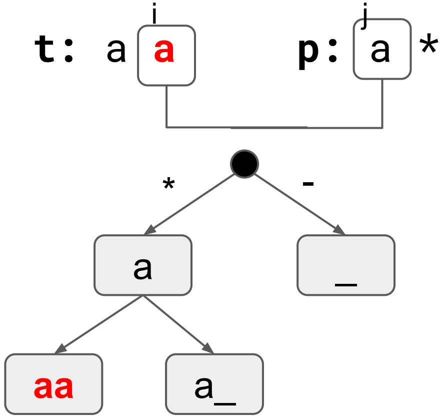


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



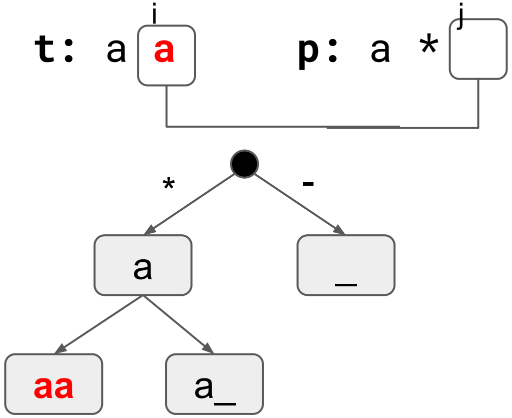
True

```python
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

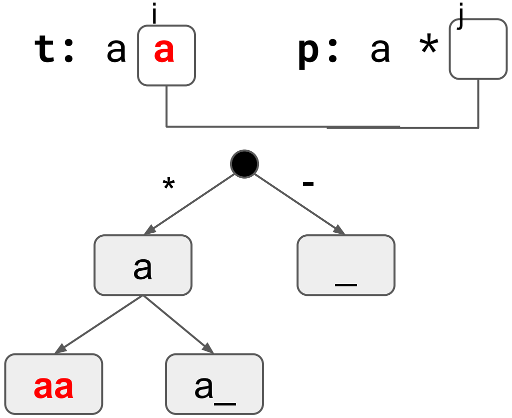


True

```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

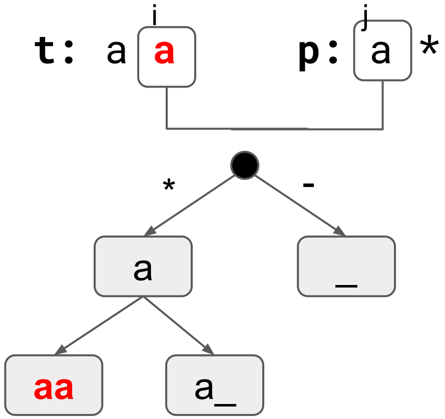


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
    return ( i == len(text) )
  else:
    first_match = i < len(text) and pattern[j] in {text[i], '.'}
    if (j+1 < len(pattern) and pattern[j+1] == '*'):
      return match(i, j+2) or (first_match and match(i+1, j))
    else:
      return first_match and match(i+1, j+1)
```

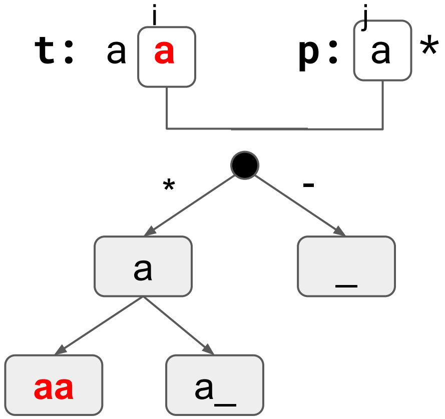
False



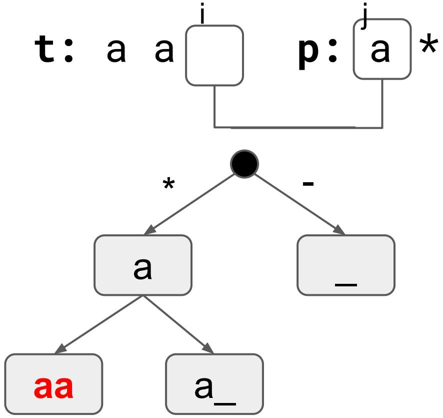
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True

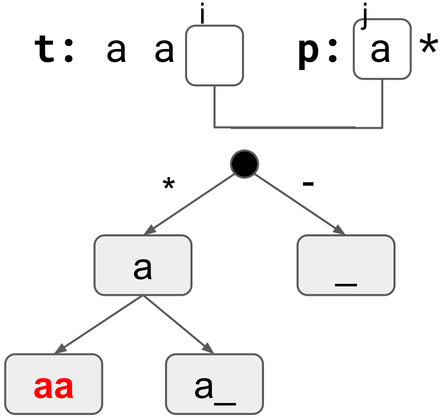
False



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
                       False                True
      else:
        return first_match and match(i+1, j+1)
```

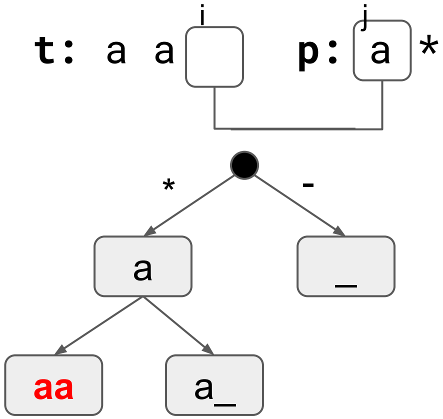


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

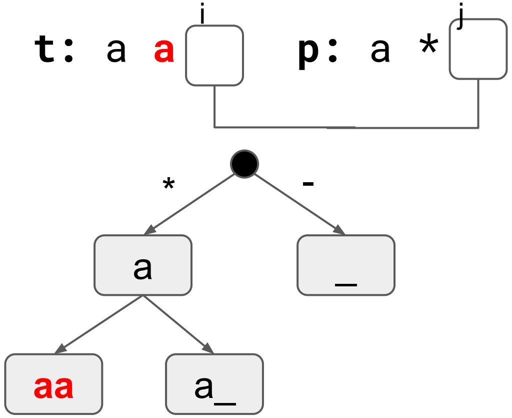


False

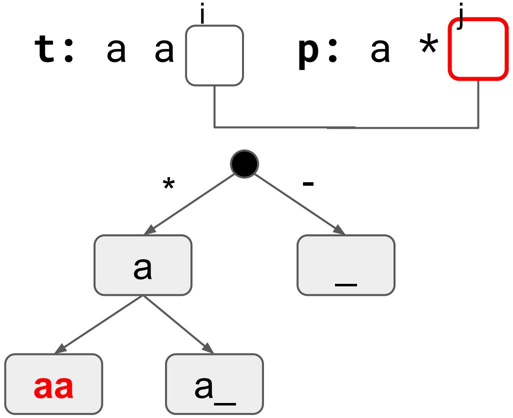
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



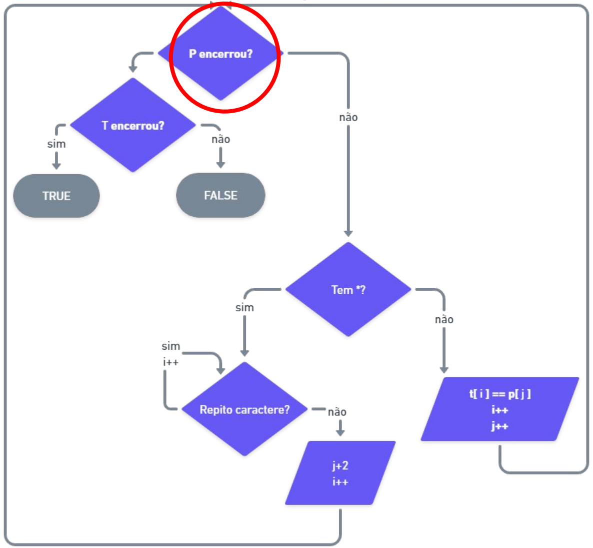
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

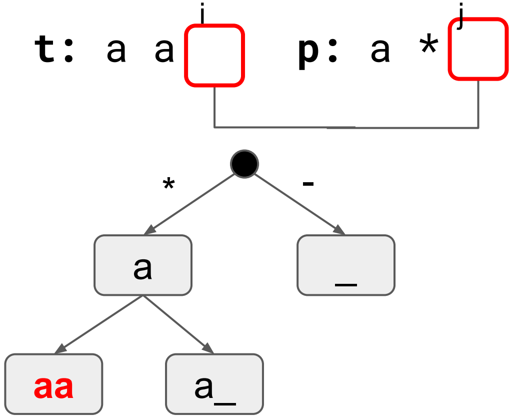


```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



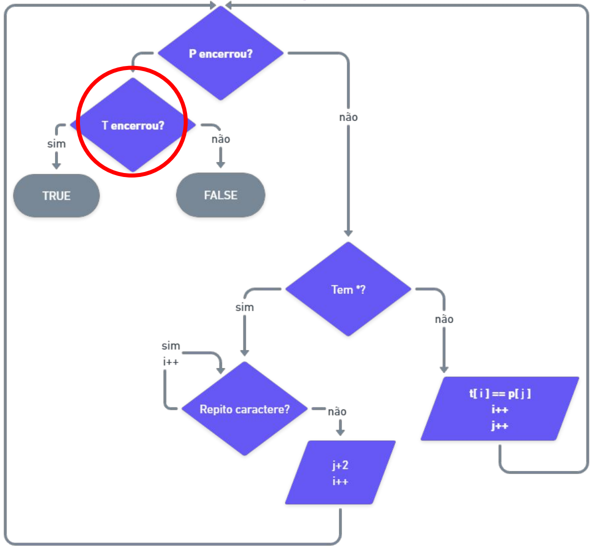
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

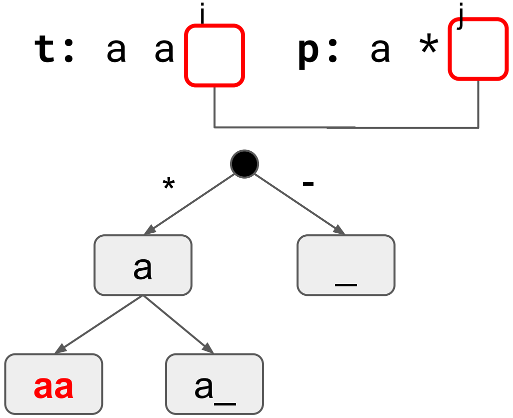


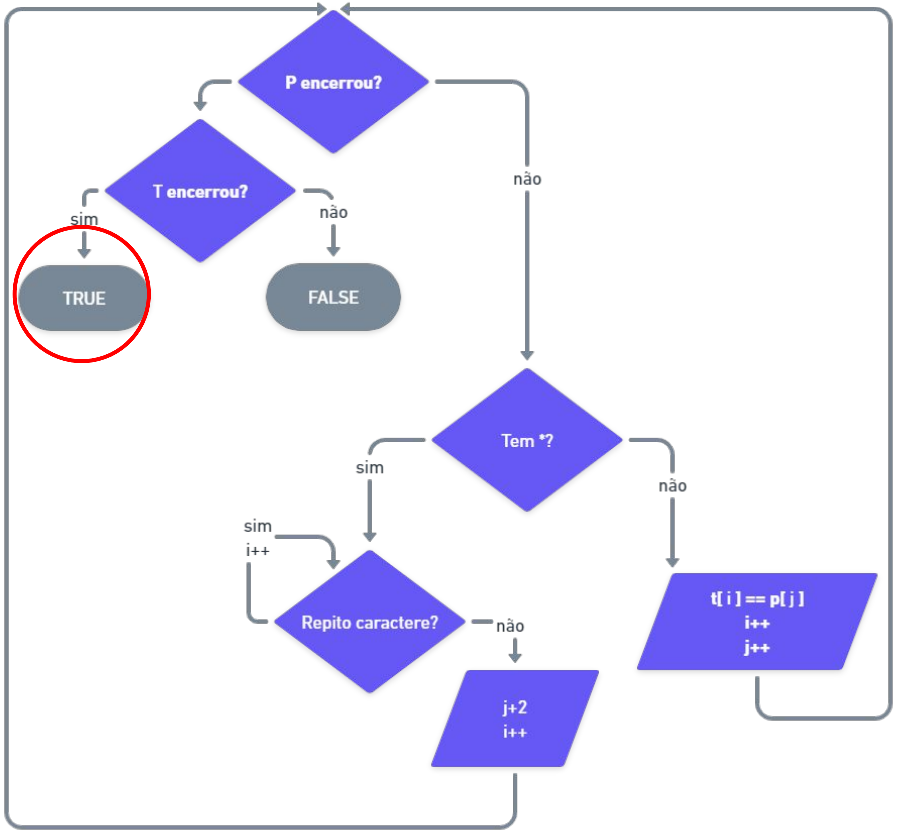


```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

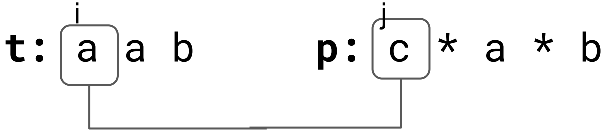
True




i
j
t: a a
p: a *
*
-
a
_
aa
a_


P encerrou?
T encerrou?
sim
não
TRUE
FALSE
não
Tem *?
sim
não
sim
i++
Repito caractere?
não
j+2
i++
t[i] == p[j]
i++
j++

Teste 2

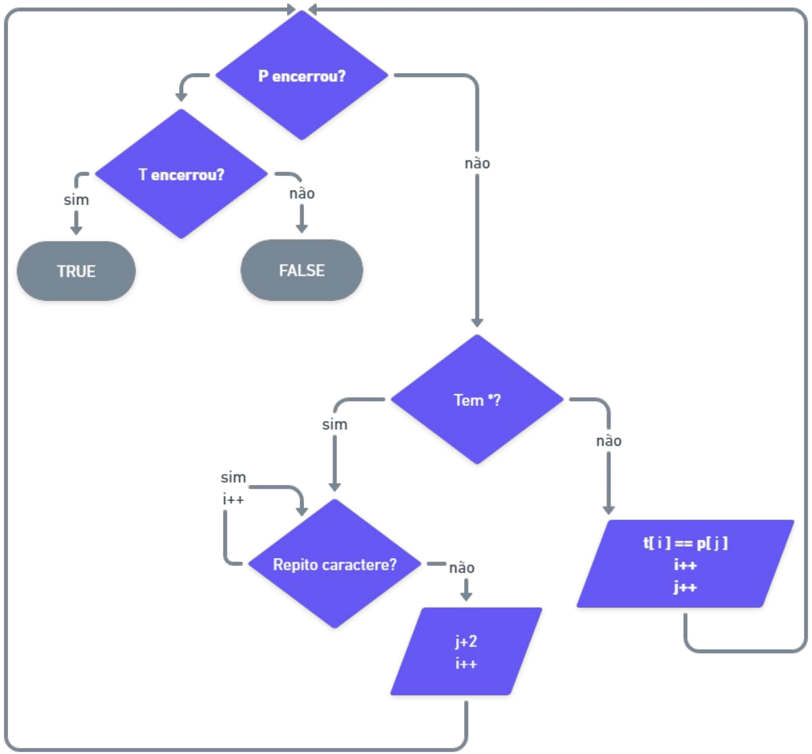


i

t: a a b

j

p: c * a * b



P encerrou?

T encerrou?

sim

TRUE

não

FALSE

não

Tem *?

sim

Repito caractere?

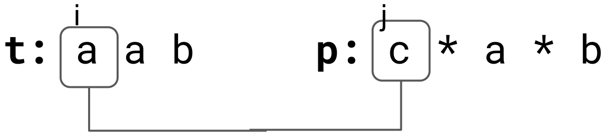
sim
i++

não

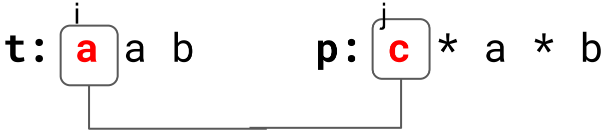
j+2
i++

não

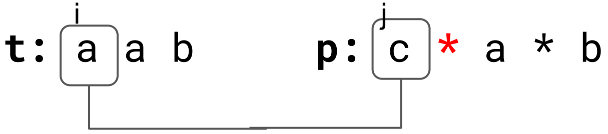
t[i] == p[j]
i++
j++



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
False   first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



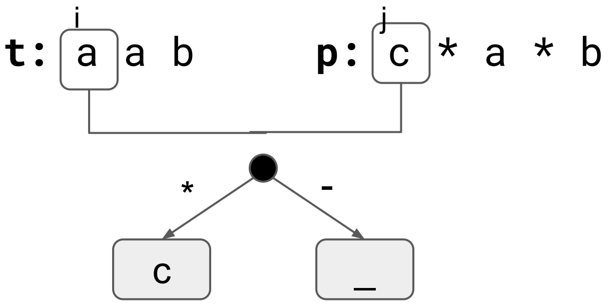
i

t: a a b

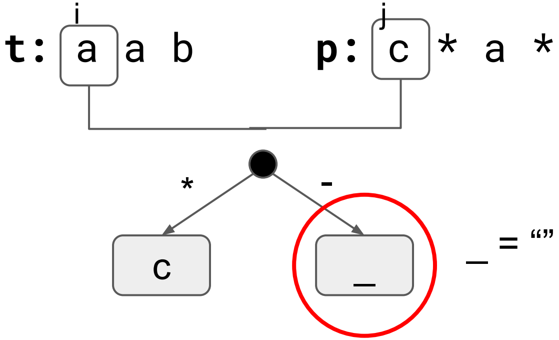
j

p: c * a * b

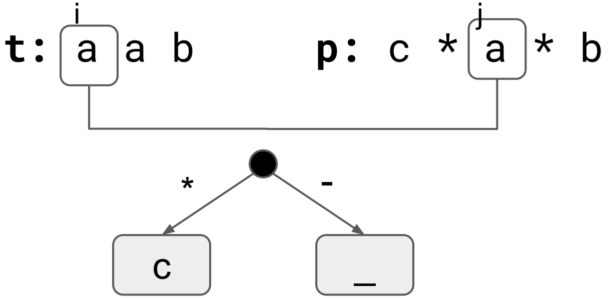
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
False   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



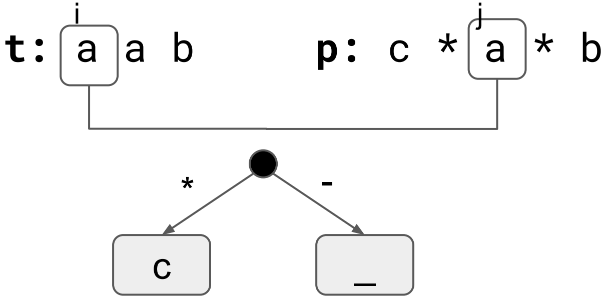
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False   first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



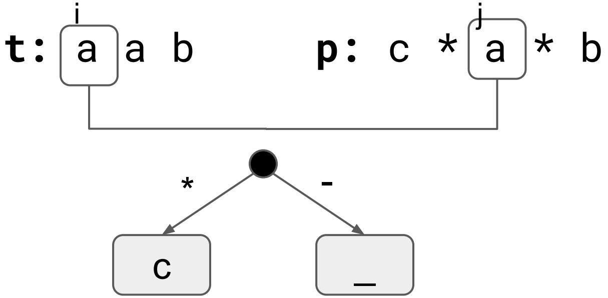
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



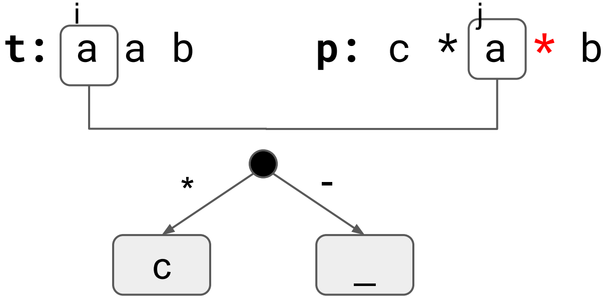
```python
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



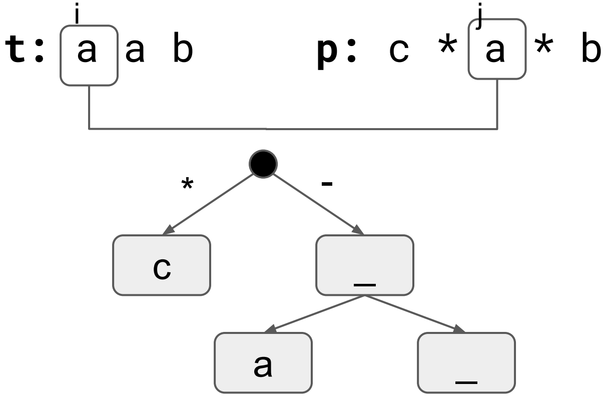
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
True first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

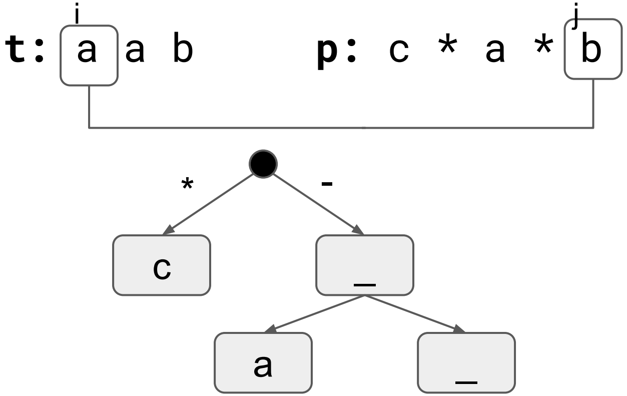


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

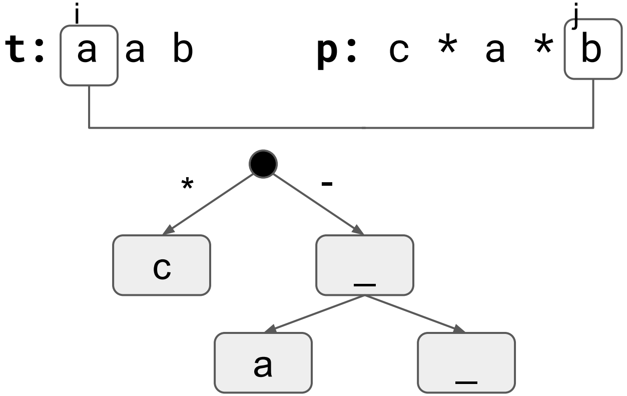


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

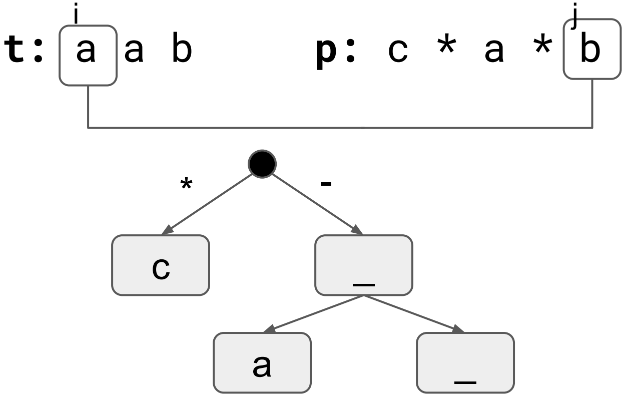
True



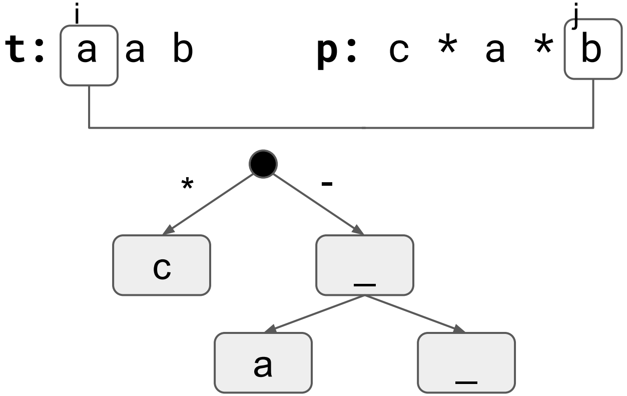
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



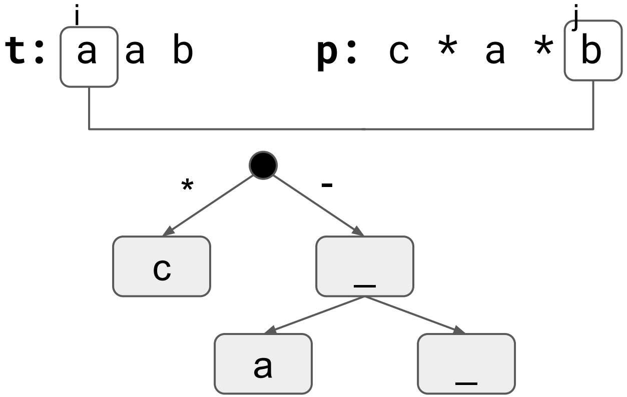
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



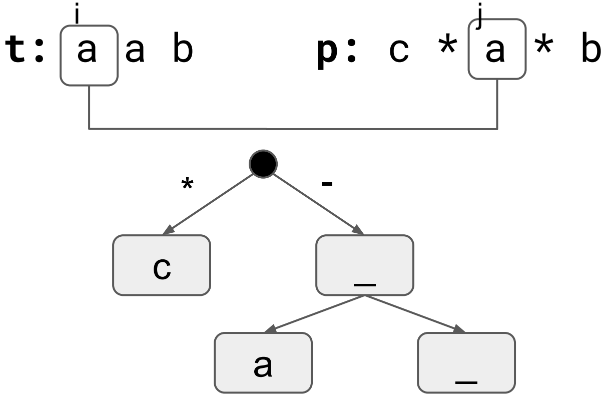
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
False   first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



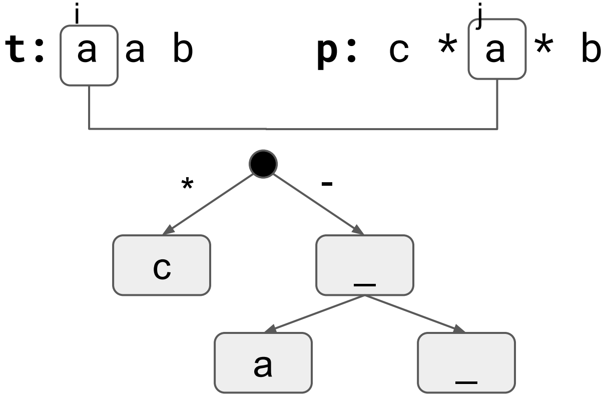
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:              False
        return first_match and match(i+1, j+1)
```



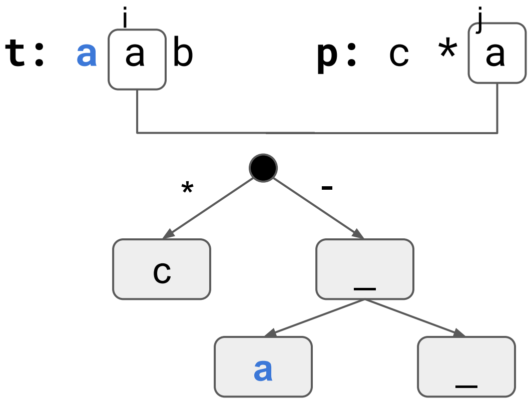
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True

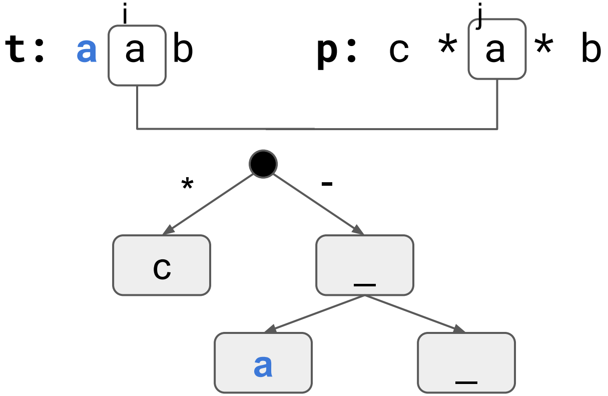
False



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
True    first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
                          False             True
        else:
            return first_match and match(i+1, j+1)
```



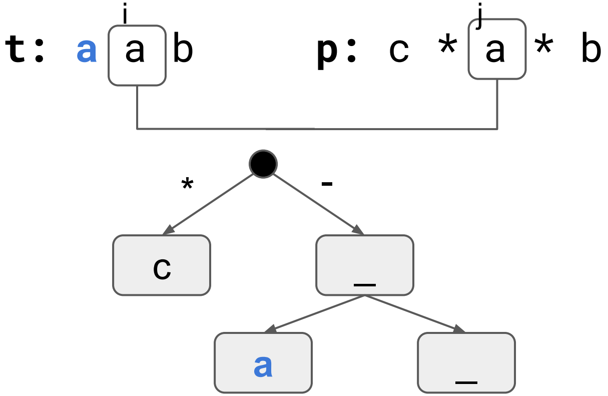
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
        first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



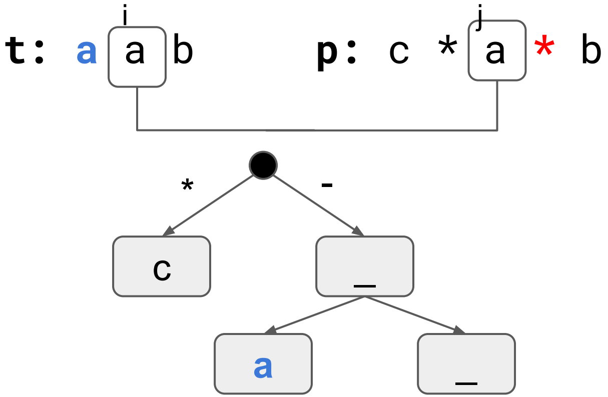
t: a a b

p: c * a * b

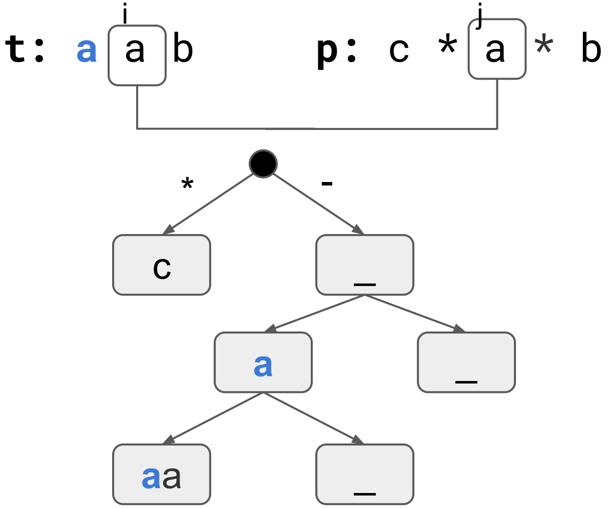
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
        first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



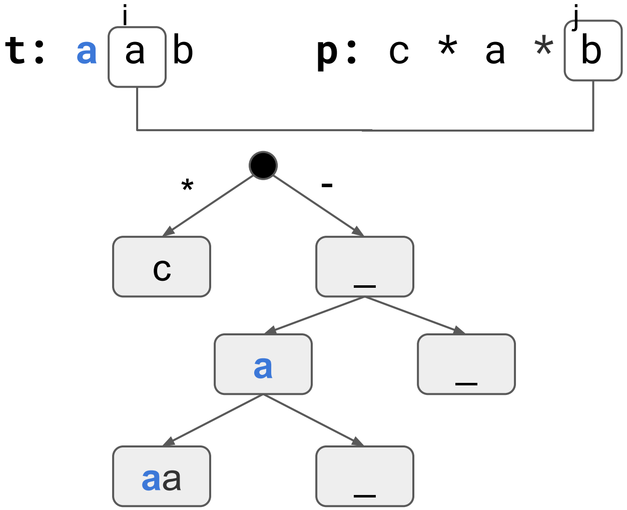
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



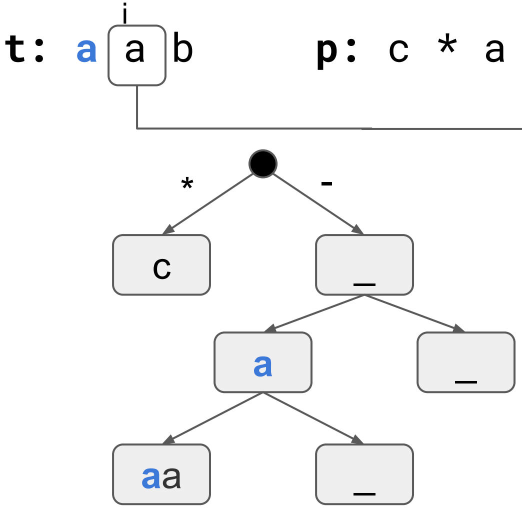
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



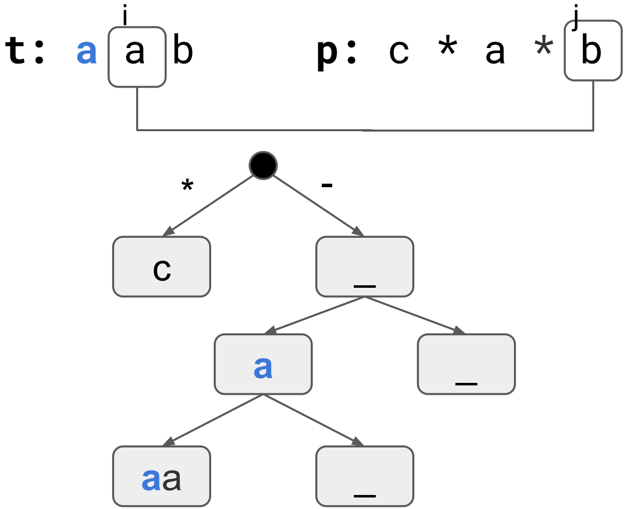
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
True    first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

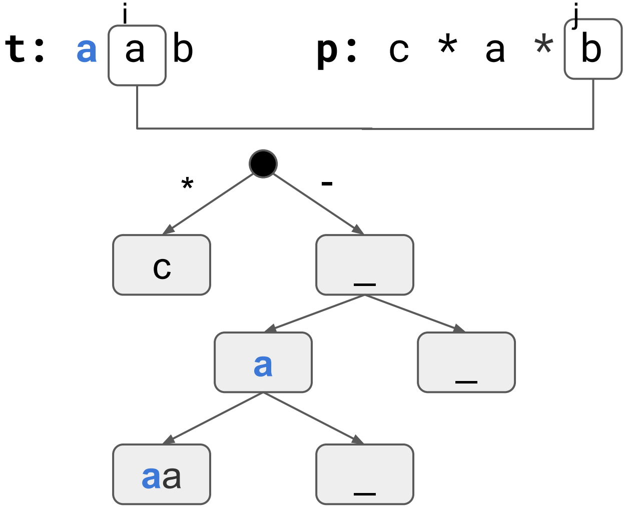


```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

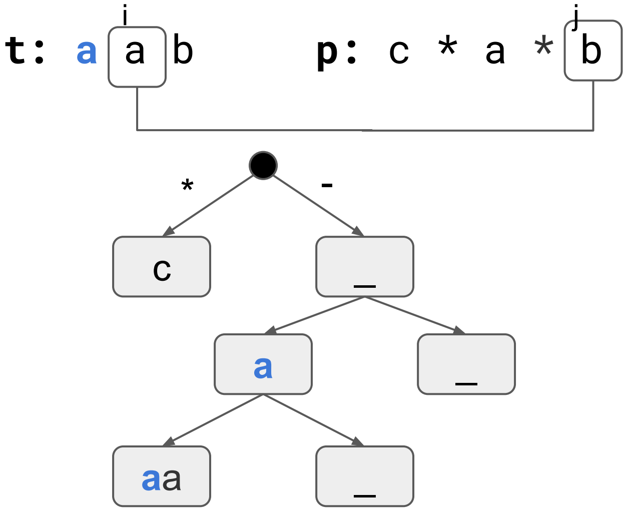


False

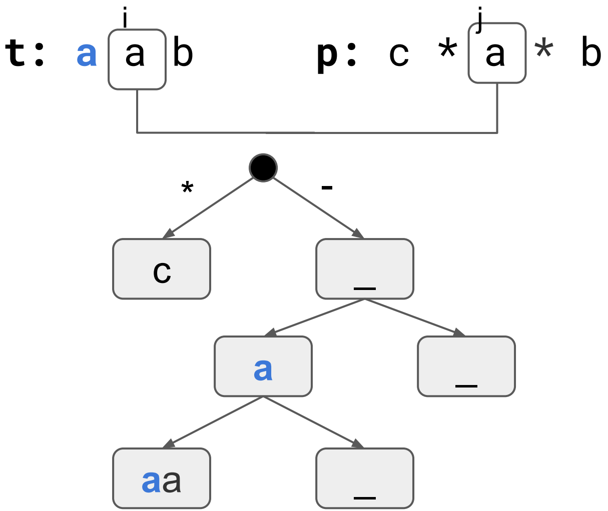
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
False   first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



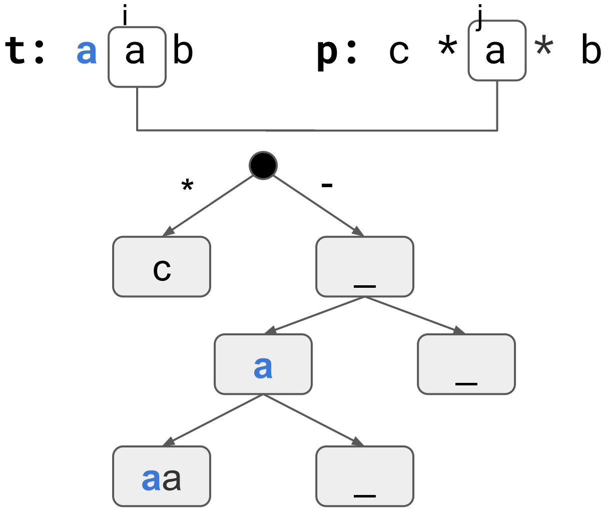
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:           False
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True

False

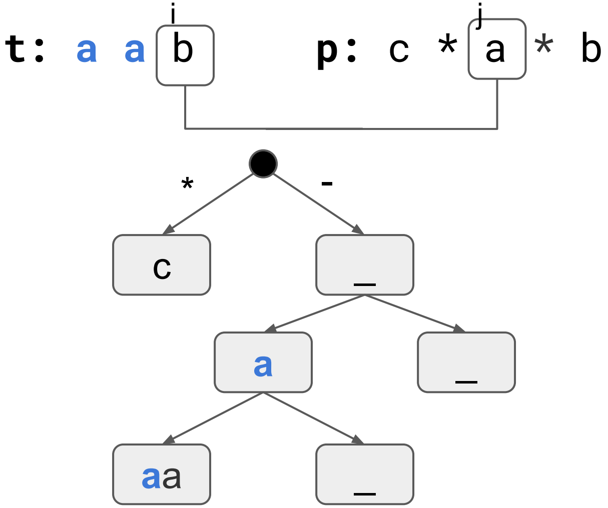


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

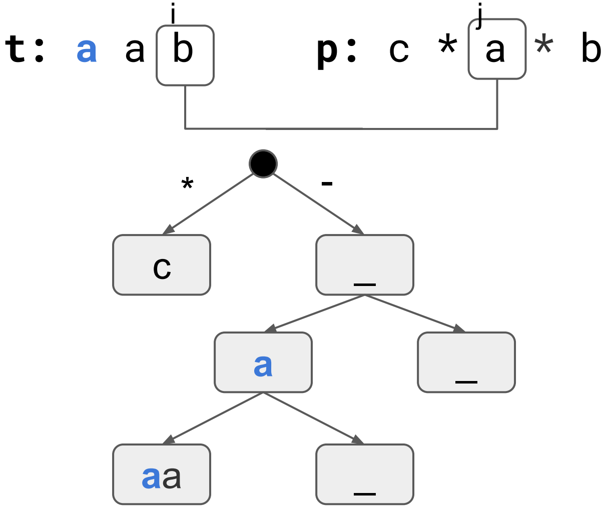
True

False

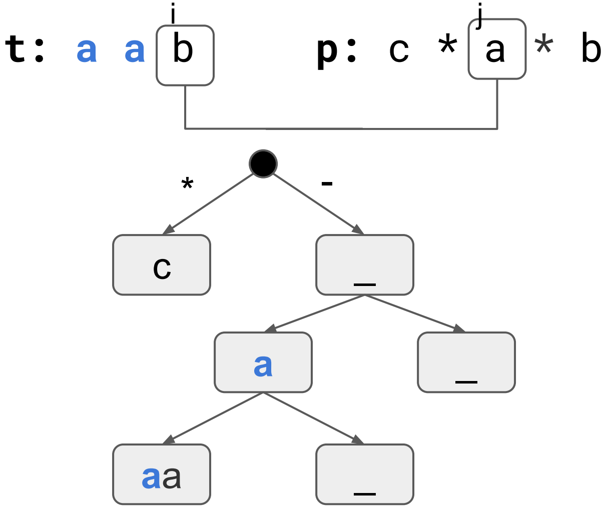
True



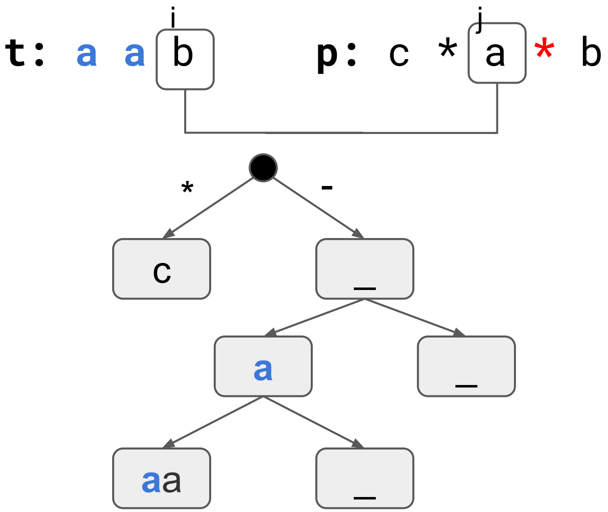
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



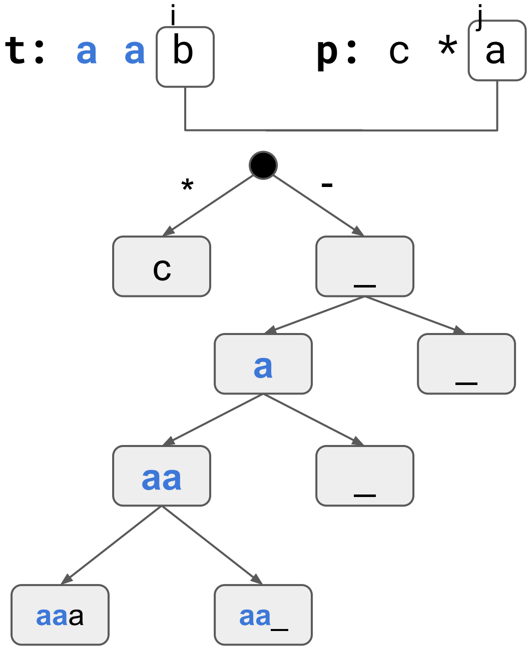
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



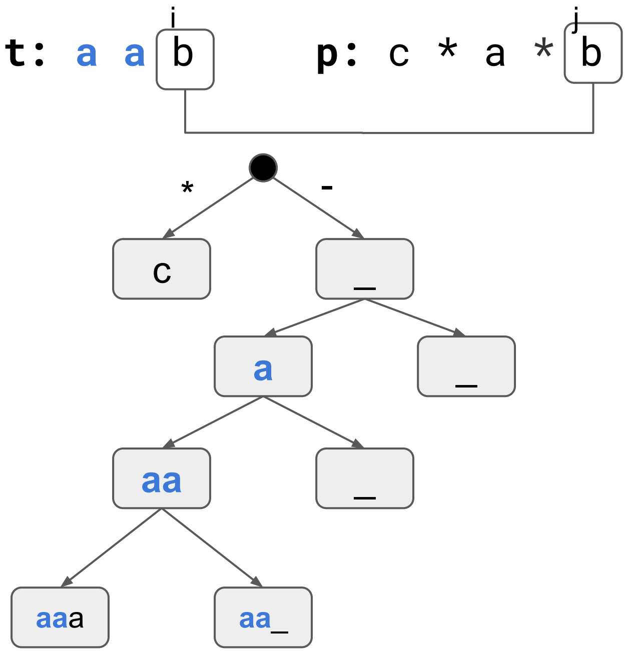
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



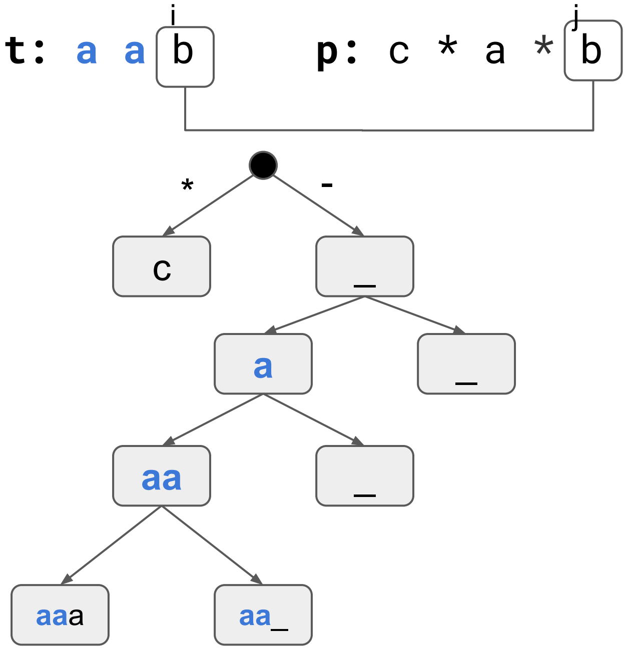
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
False  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



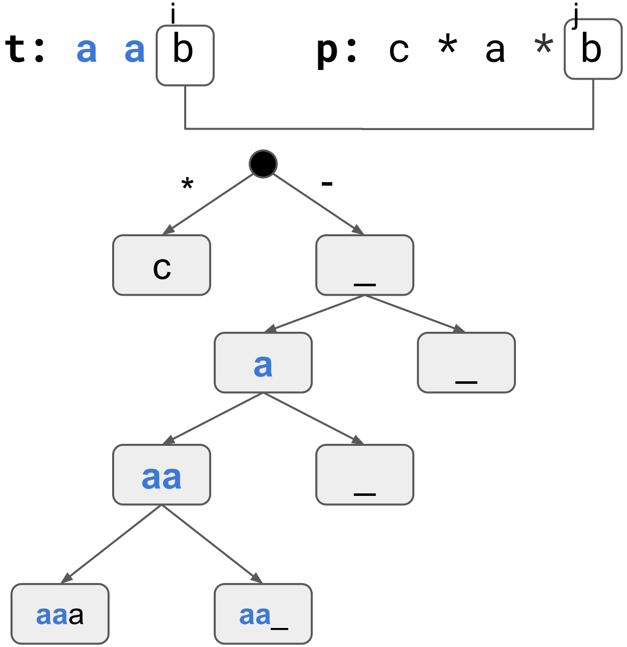
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
False   first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



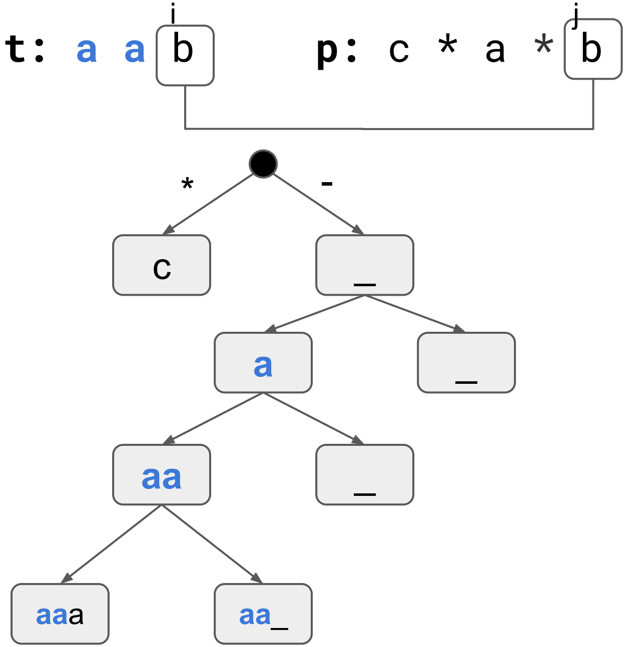
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



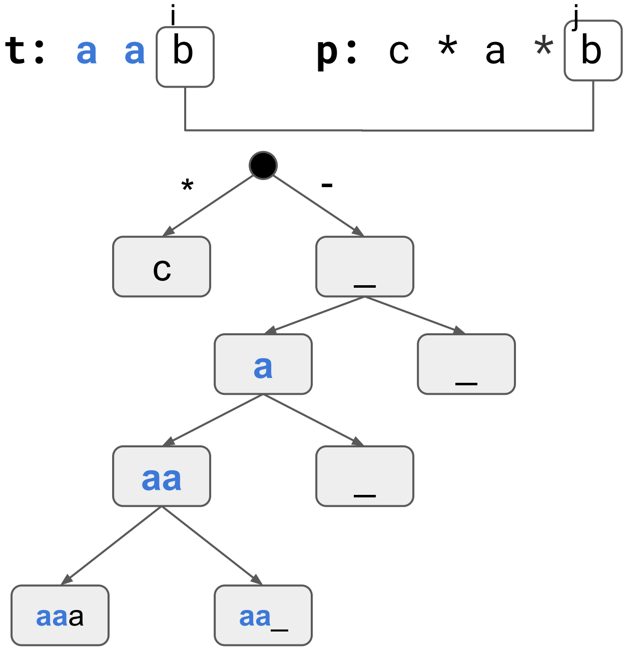
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



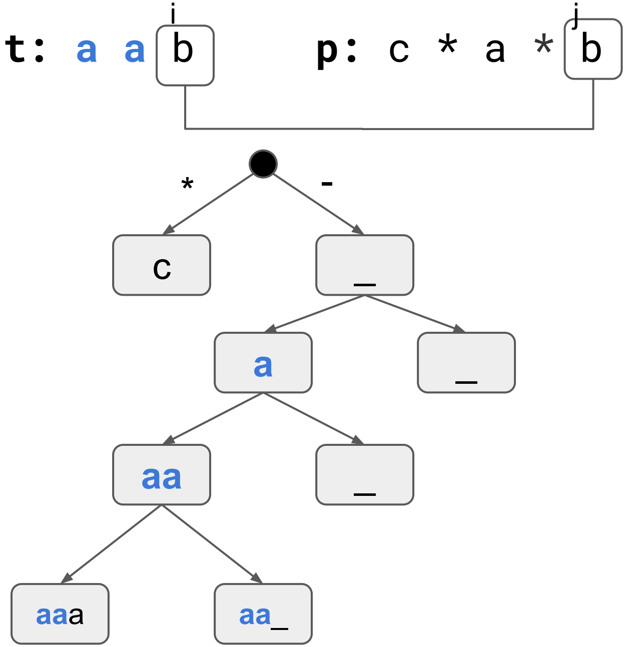
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
True    first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



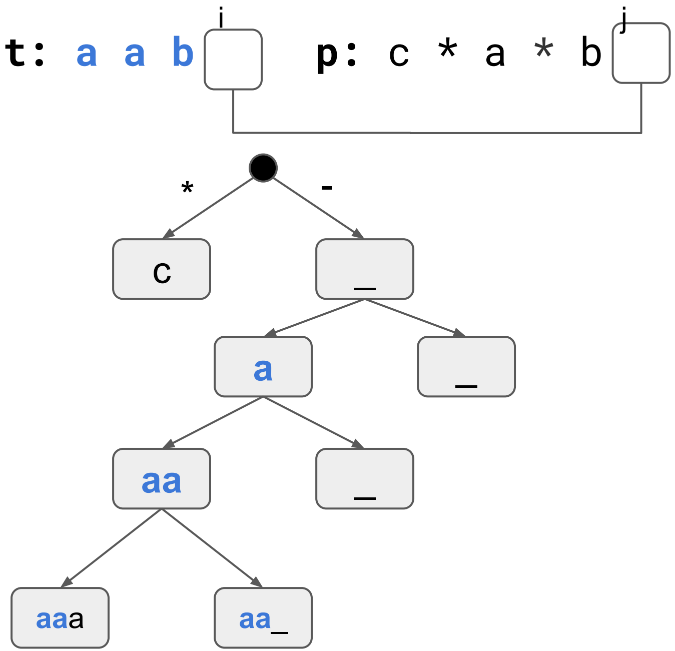
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
True    first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:        True
            return first_match and match(i+1, j+1)
```



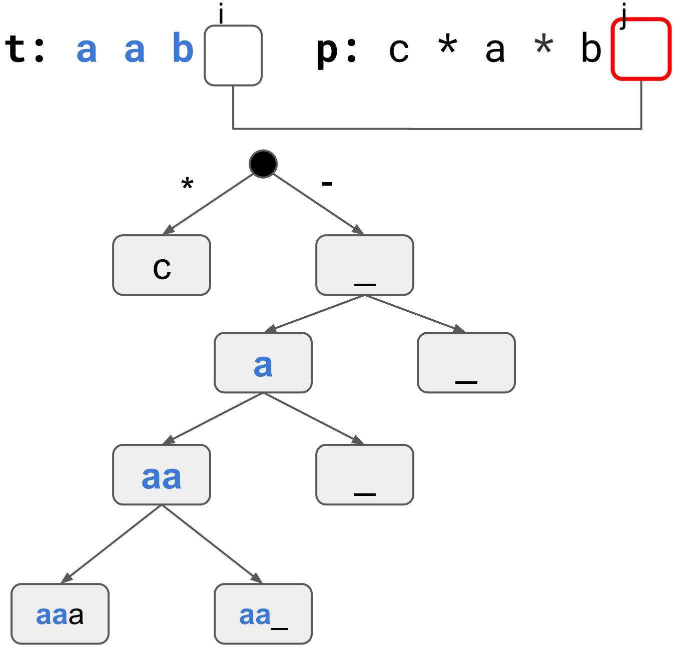
```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True

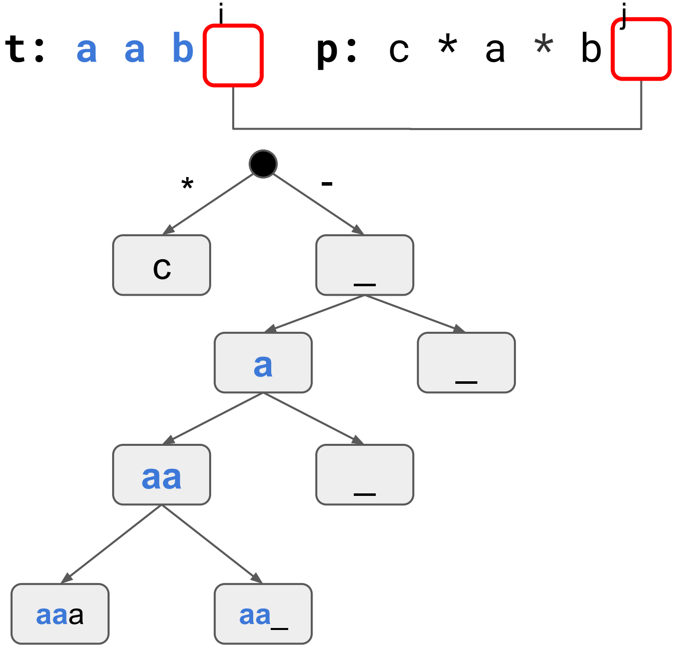
True



```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

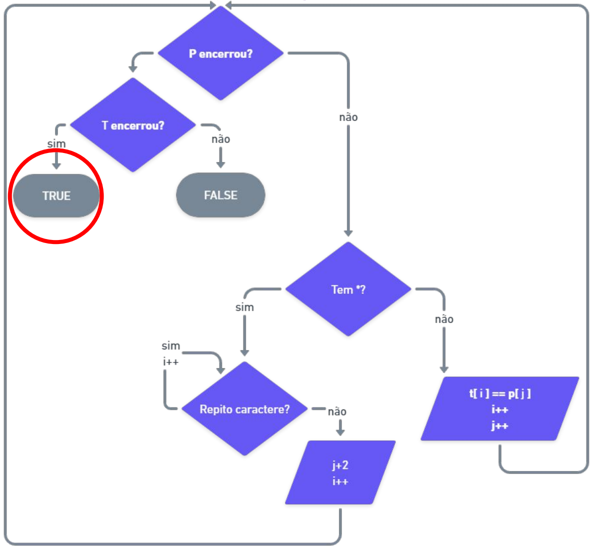


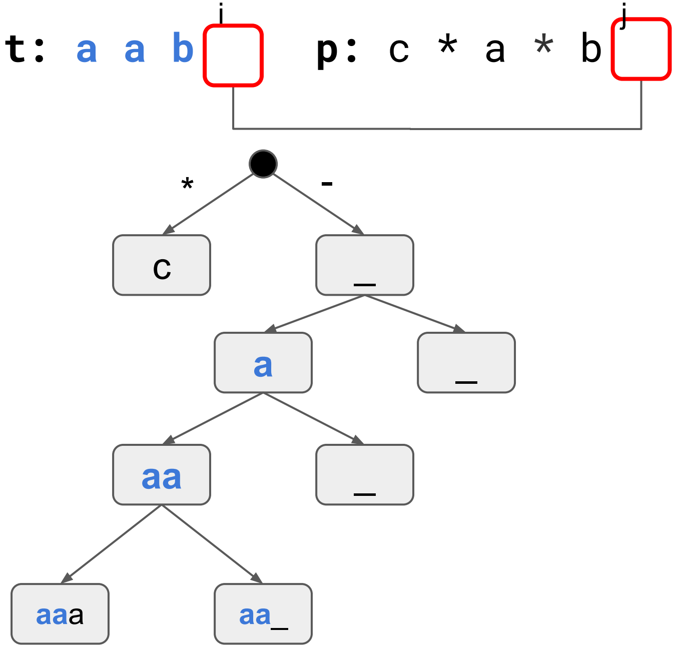
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
        first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
        first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```

True





i

j

t: a a b

p: c * a * b

*

-

c

_

a

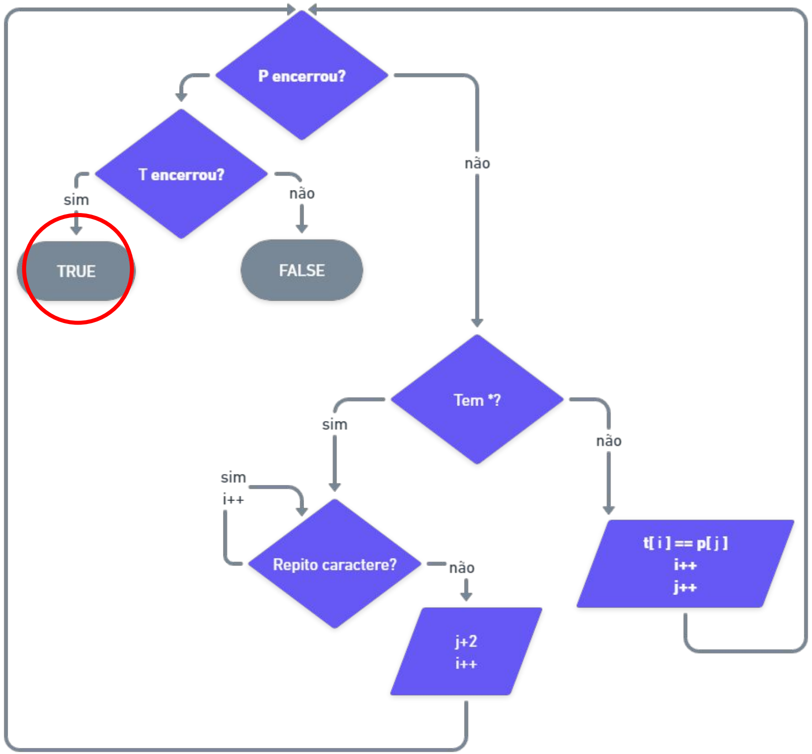
_

aa

_

aaa

aa_



P encerrou?

T encerrou?

sim

não

TRUE

FALSE

não

Tem *?

sim

não

sim
i++

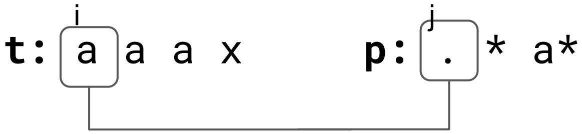
Repito caractere?

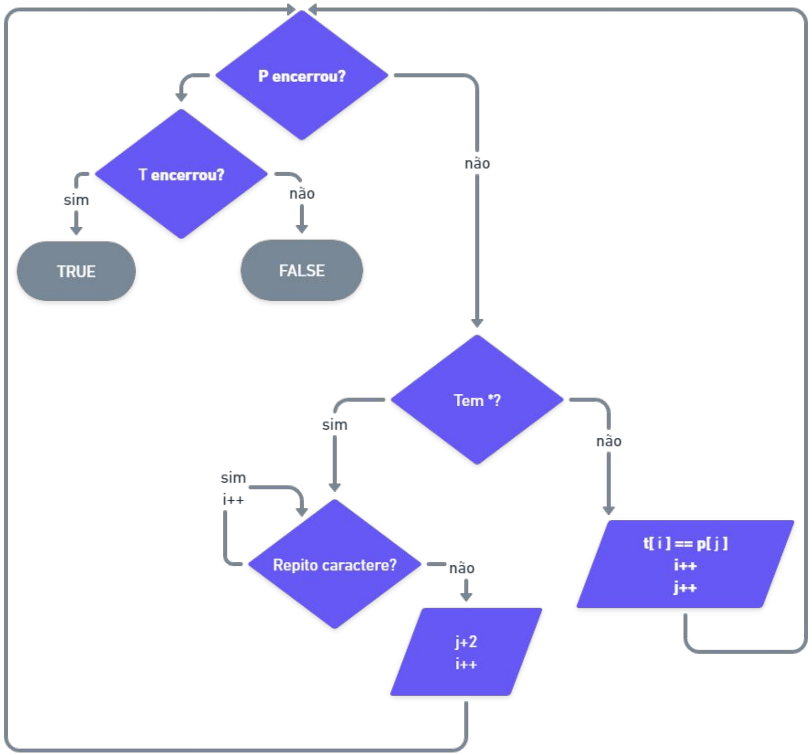
não

j+2
i++

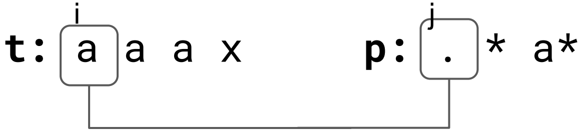
t[i] == p[j]
i++
j++

Teste 3

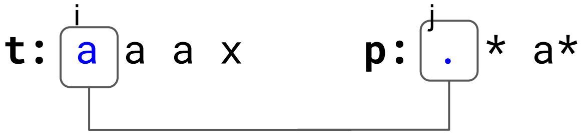

i
t: a a a x
j
p: . * a*


P encerrou?
T encerrou?
sim
não
TRUE
FALSE
não
Tem *?
sim
não
sim
i++
Repito caractere?
não
j+2
i++
t[i] == p[j]
i++
j++

Ta errado!   Ç_Ç)

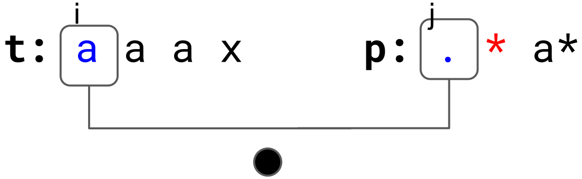


```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
    return ( i == len(text) )
  else:
    first_match = i < len(text) and pattern[j] in {text[i], '.'}
    if (j+1 < len(pattern) and pattern[j+1] == '*'):
      return match(i, j+2) or (first_match and match(i+1, j))
    else:
      return first_match and match(i+1, j+1)
```



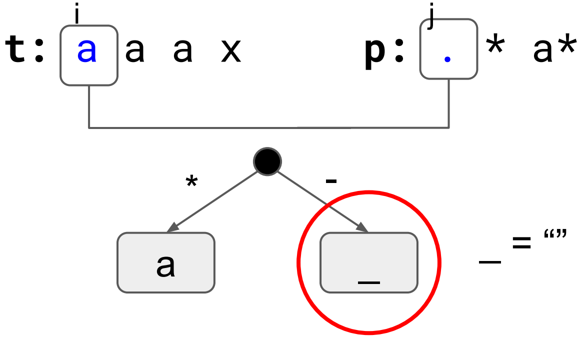
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True



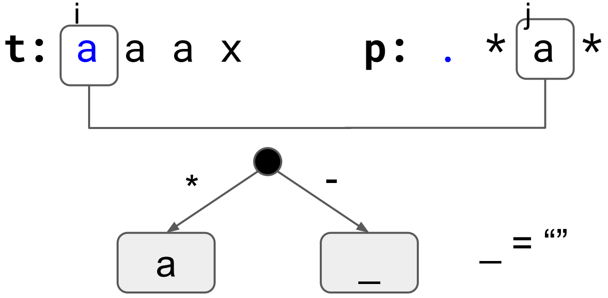
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

True

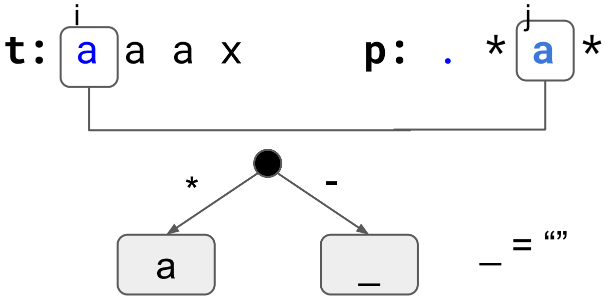


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

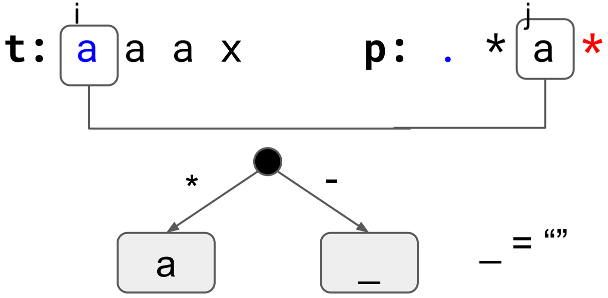
True



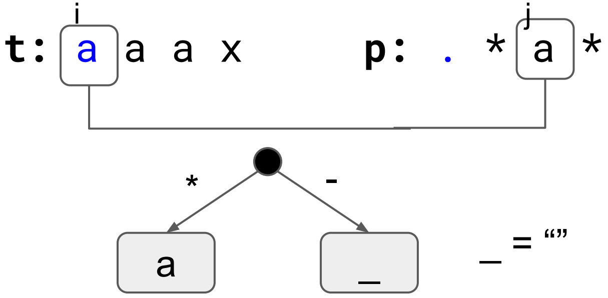
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



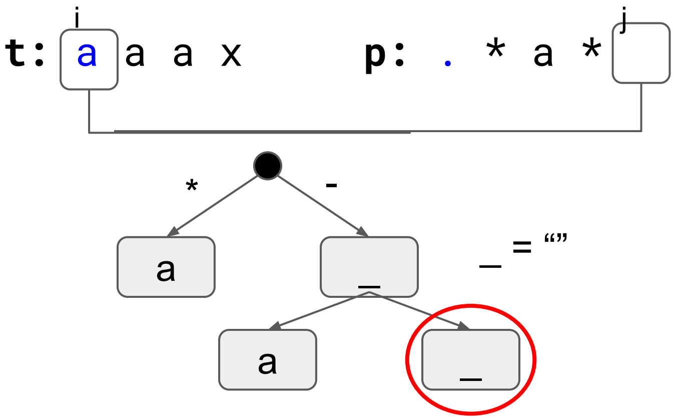
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
TRUE first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



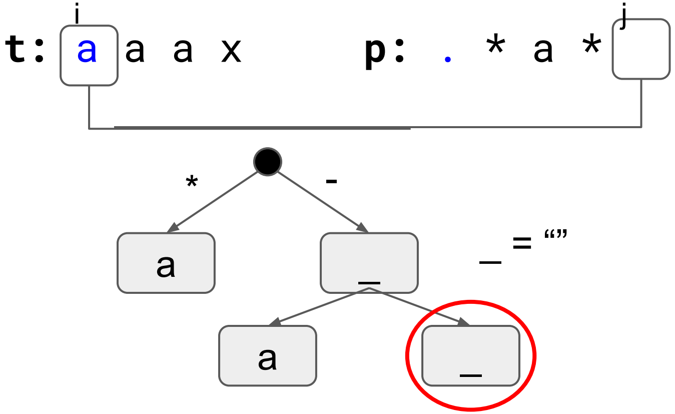
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
TRUE first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```



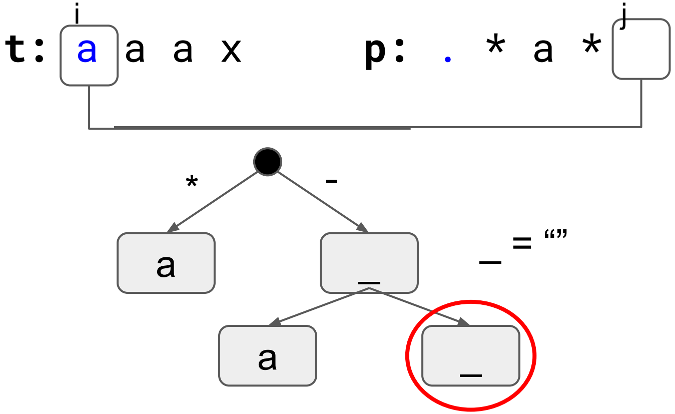
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
TRUE  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

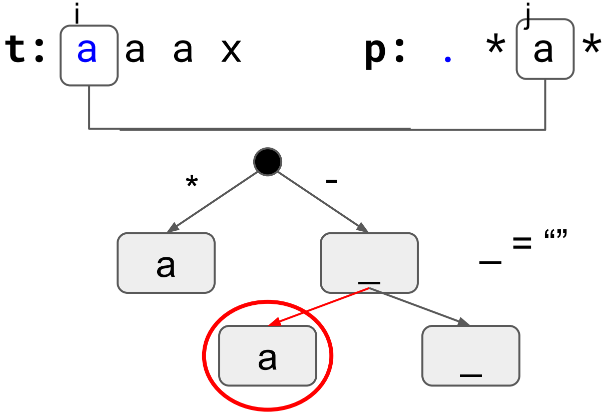


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

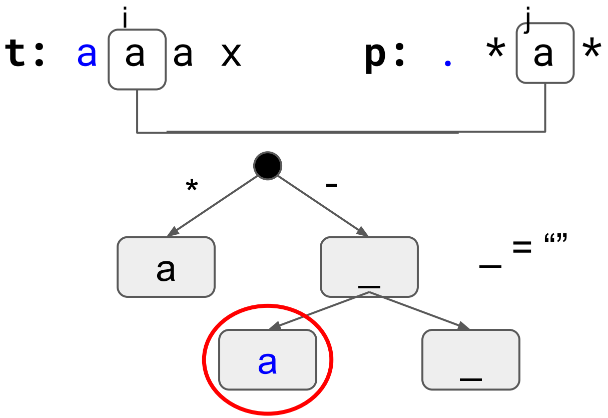
False



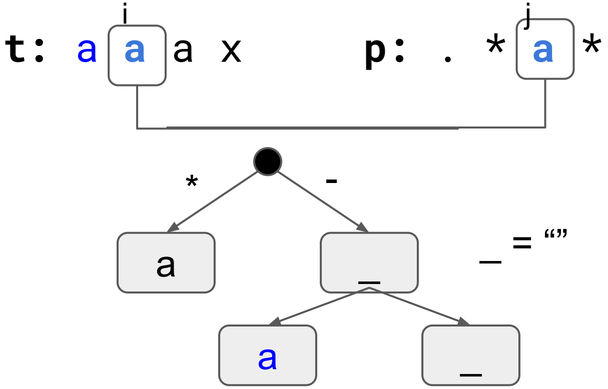
```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

True

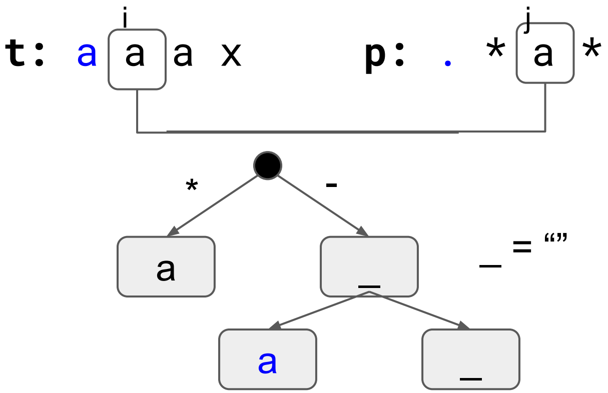
False True



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

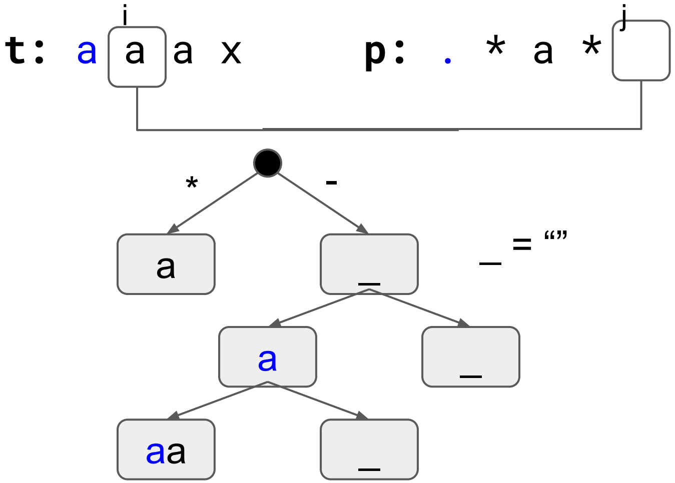


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

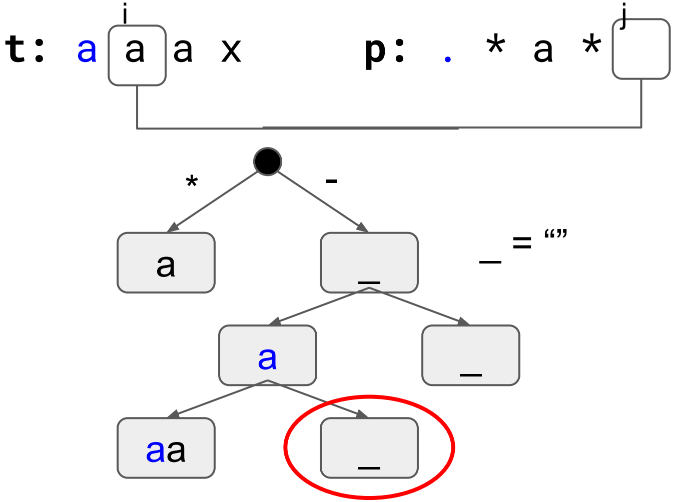


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True

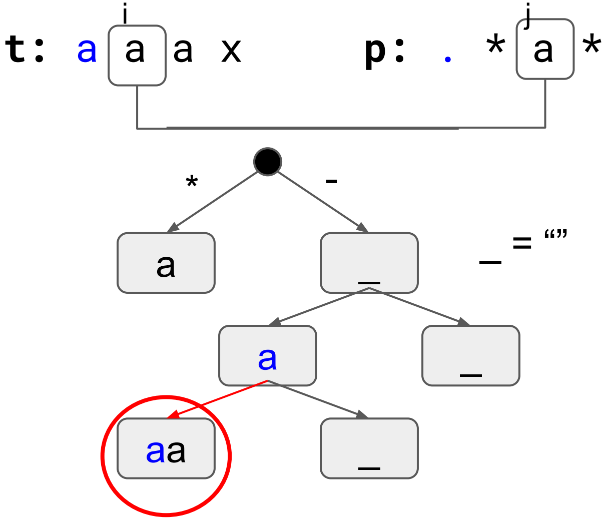


```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

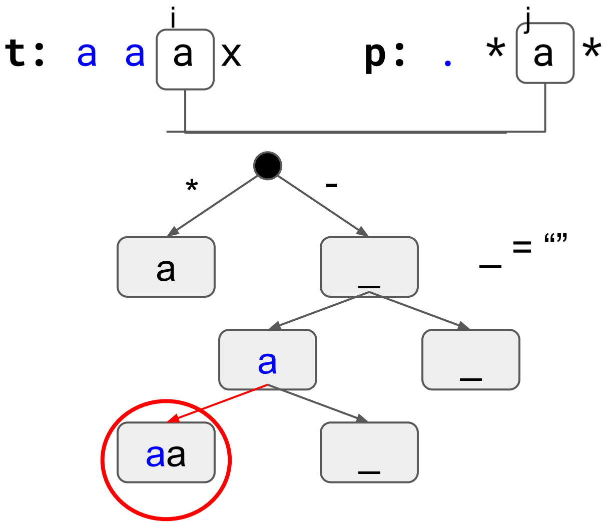


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

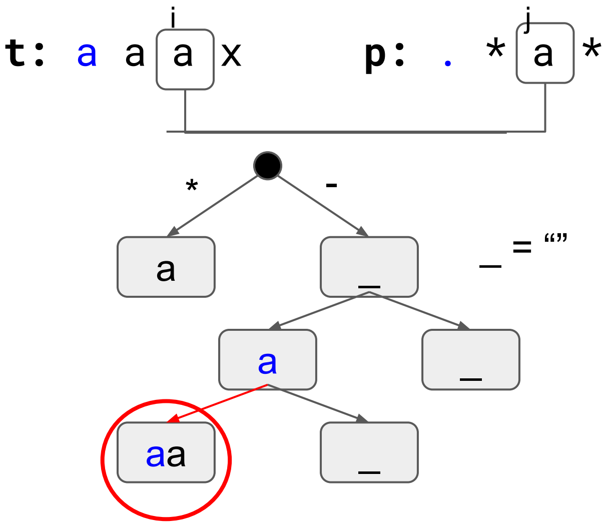
False



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
                      False               True
      else:
        return first_match and match(i+1, j+1)
```

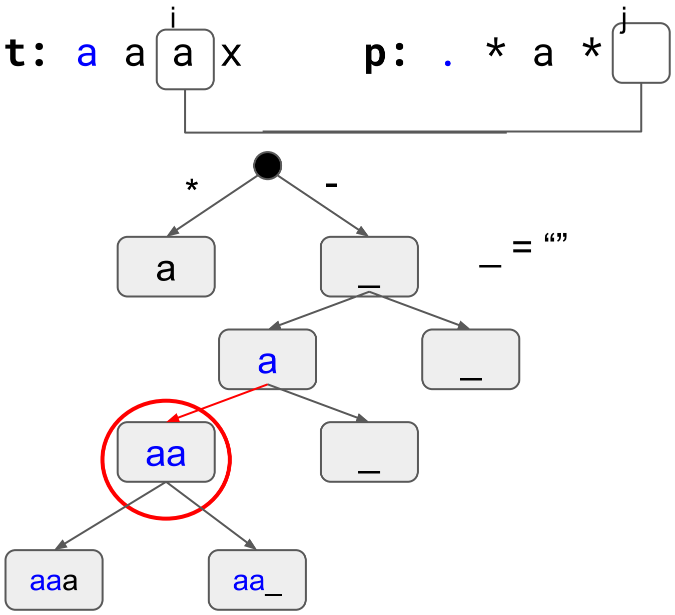


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

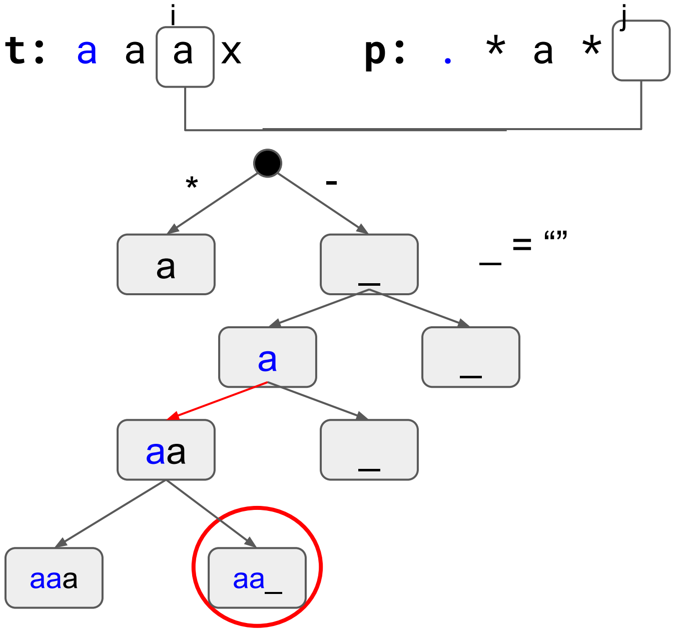


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True

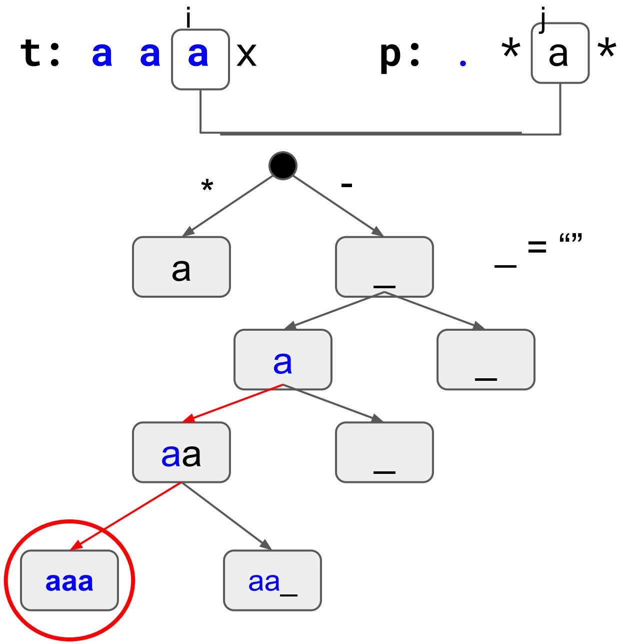


```python
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

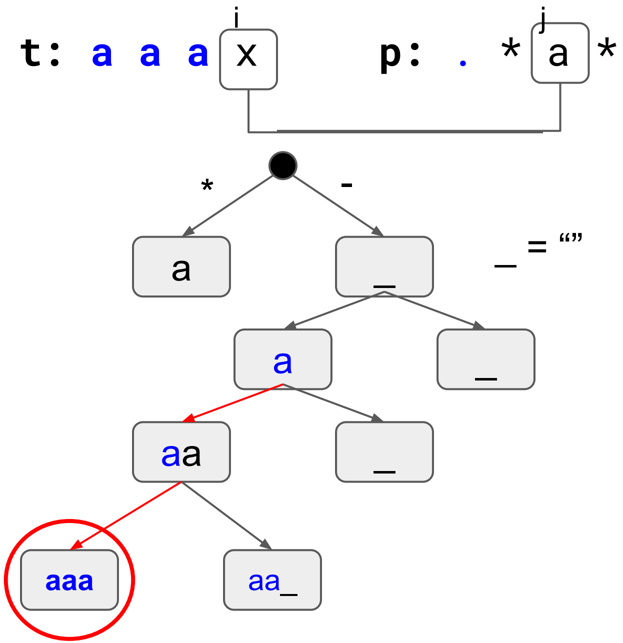


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

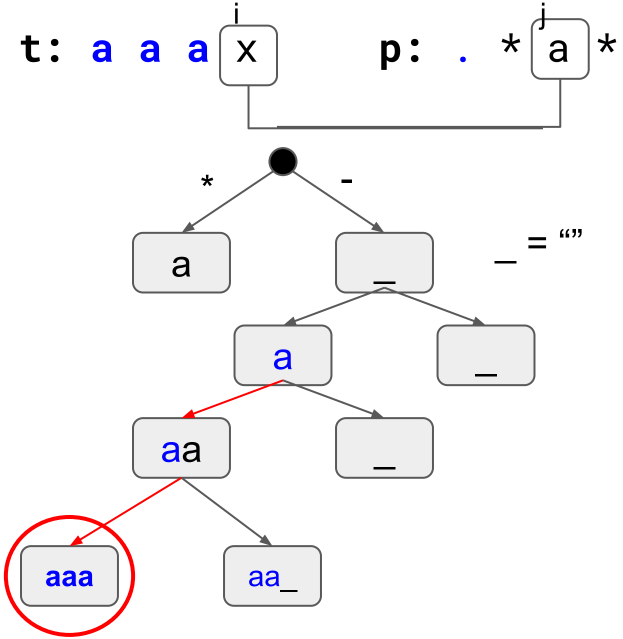
False



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
                       False              True
      else:
        return first_match and match(i+1, j+1)
```

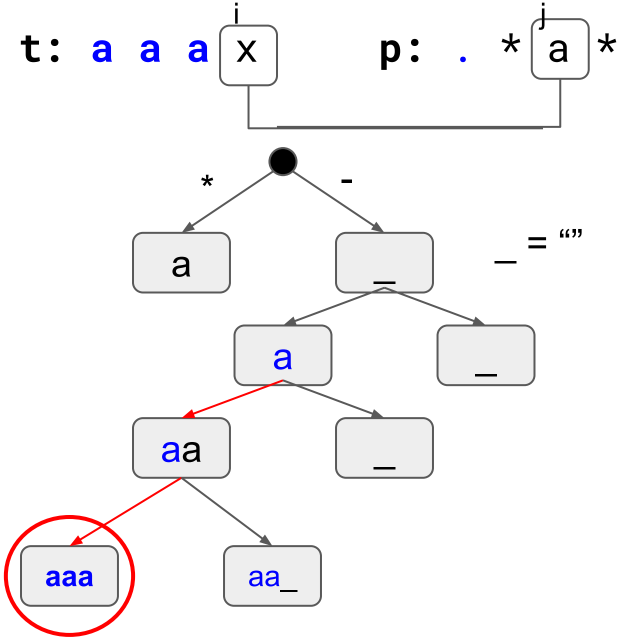


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



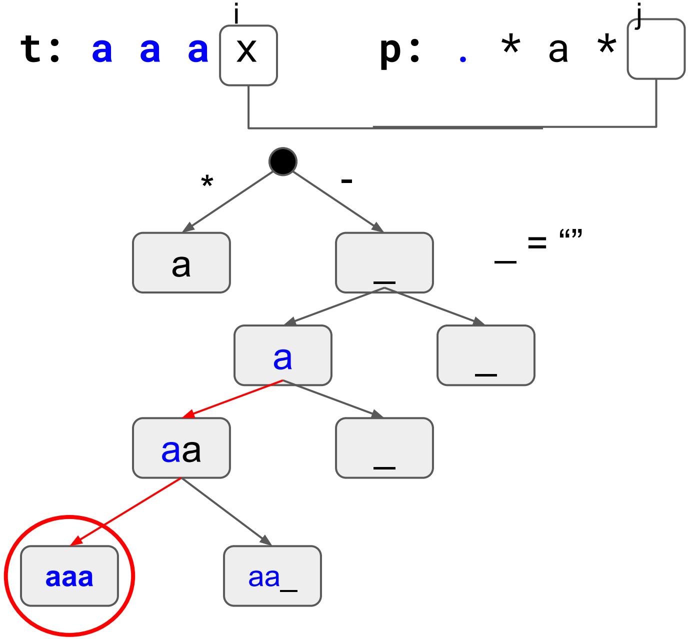
False

```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

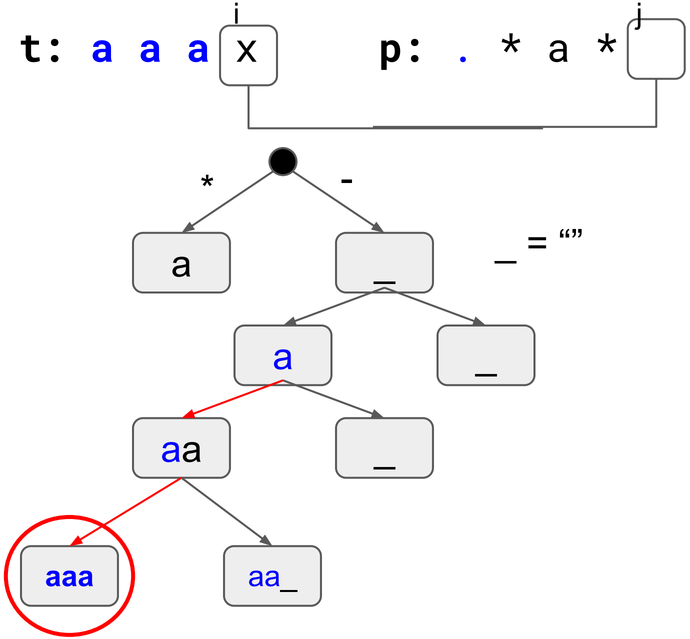


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i],
'.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1,
j))
      else:
        return first_match and match(i+1, j+1)
```

False

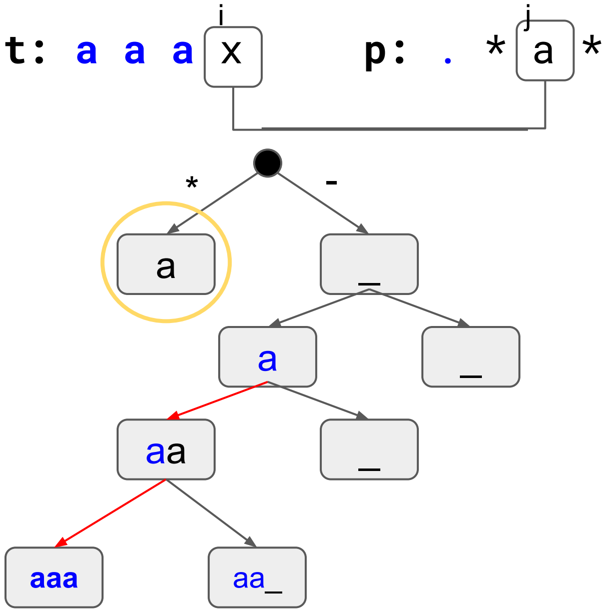


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

False

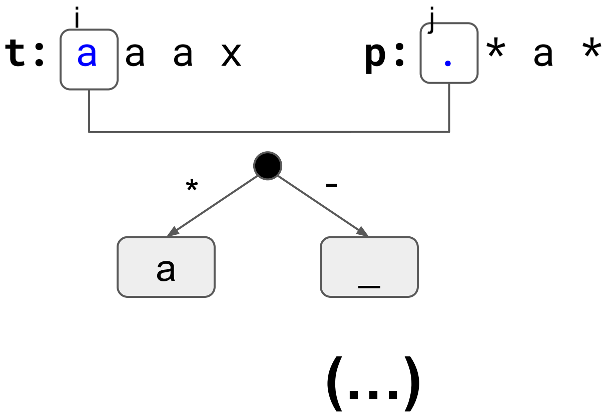


```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

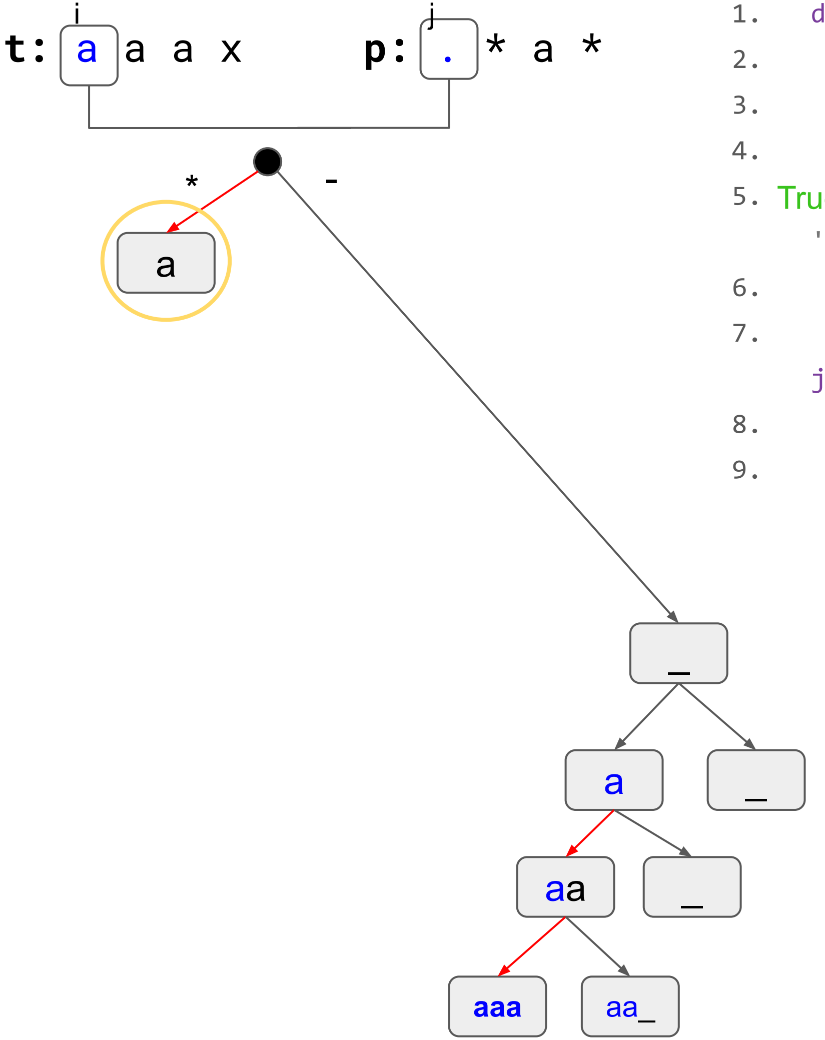
False

False

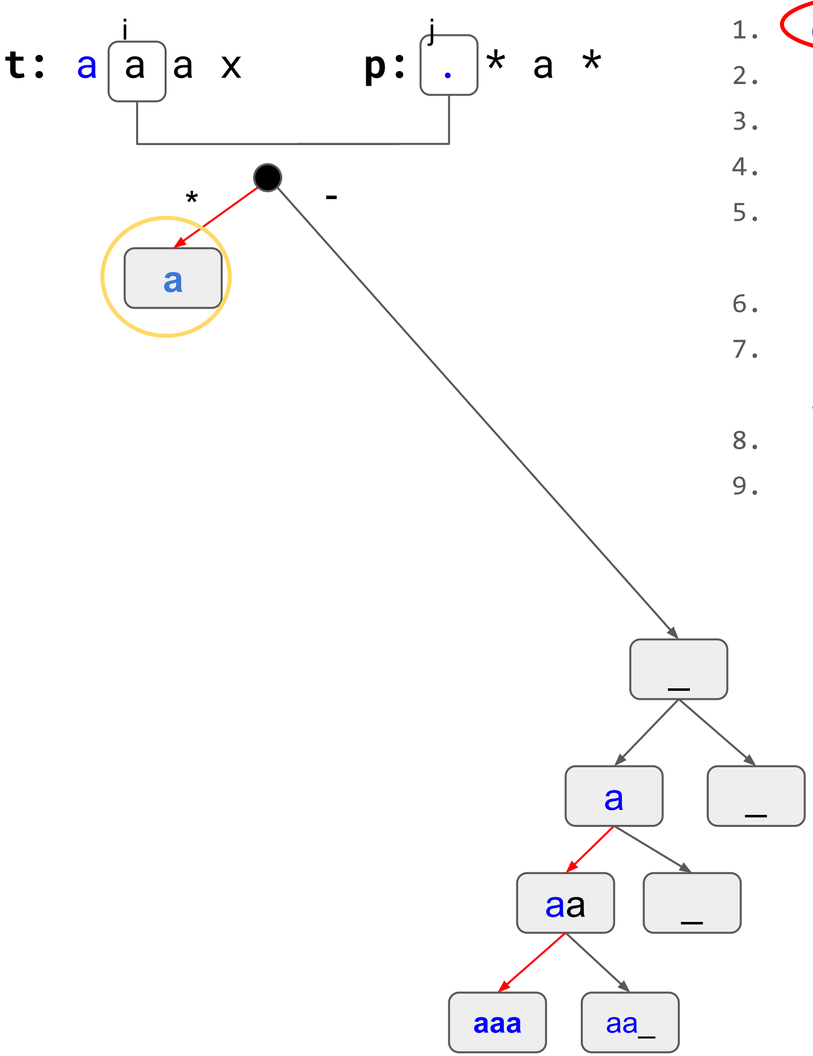
False



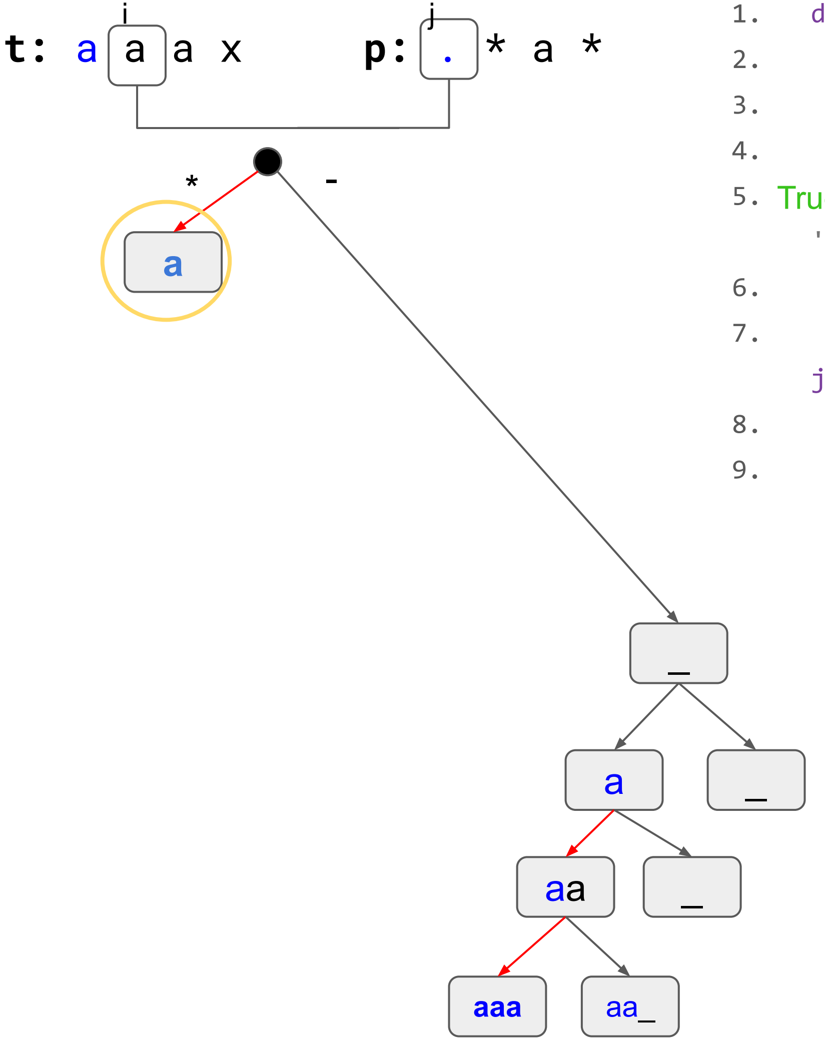
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
                     False                True
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
                   False                      True
      else:
        return first_match and match(i+1, j+1)
```

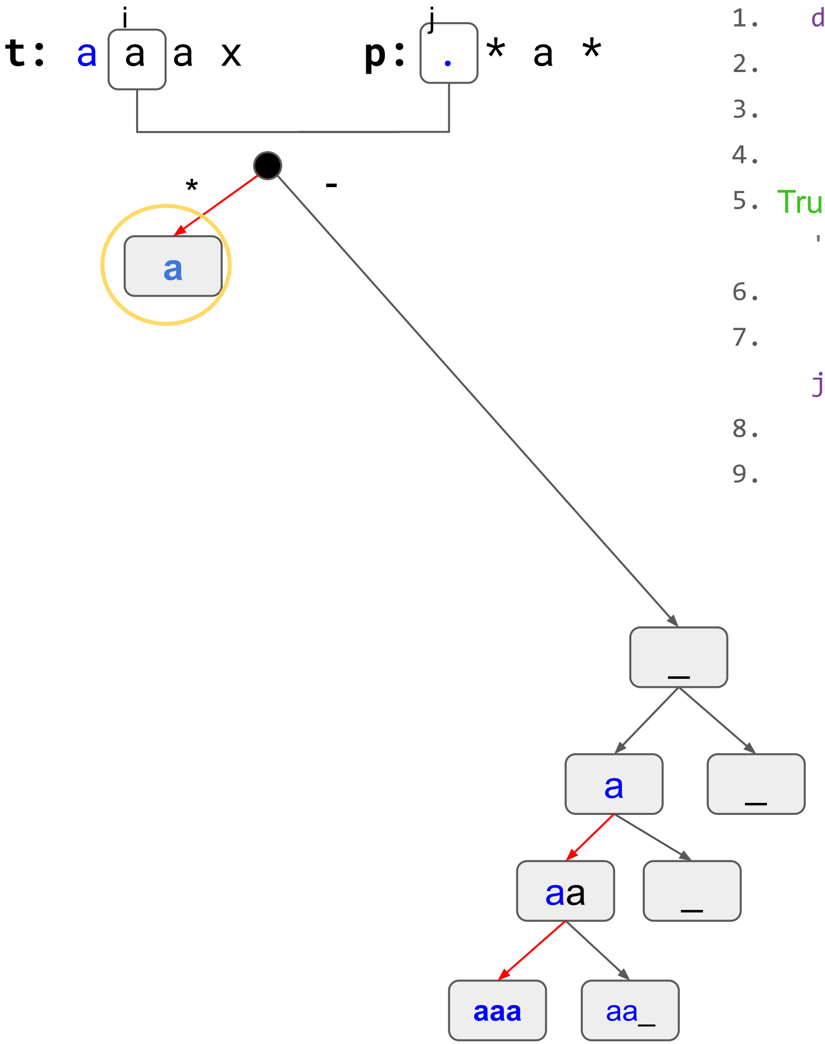


```python
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
   first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

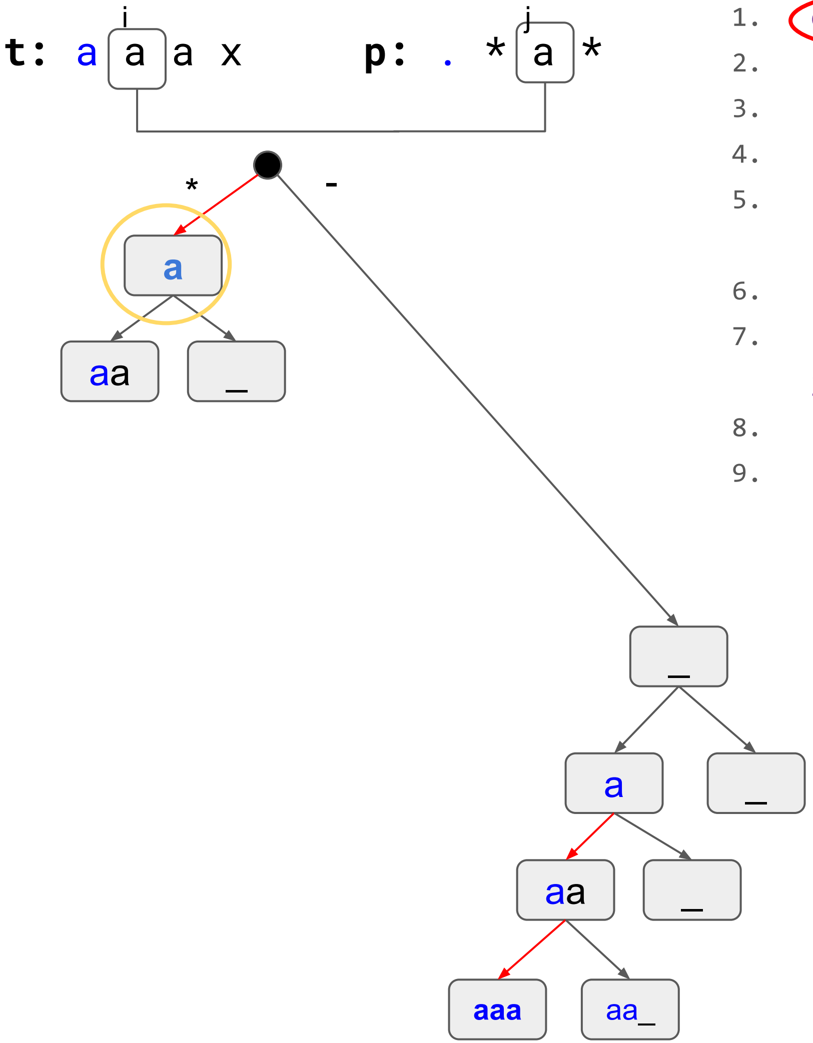


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

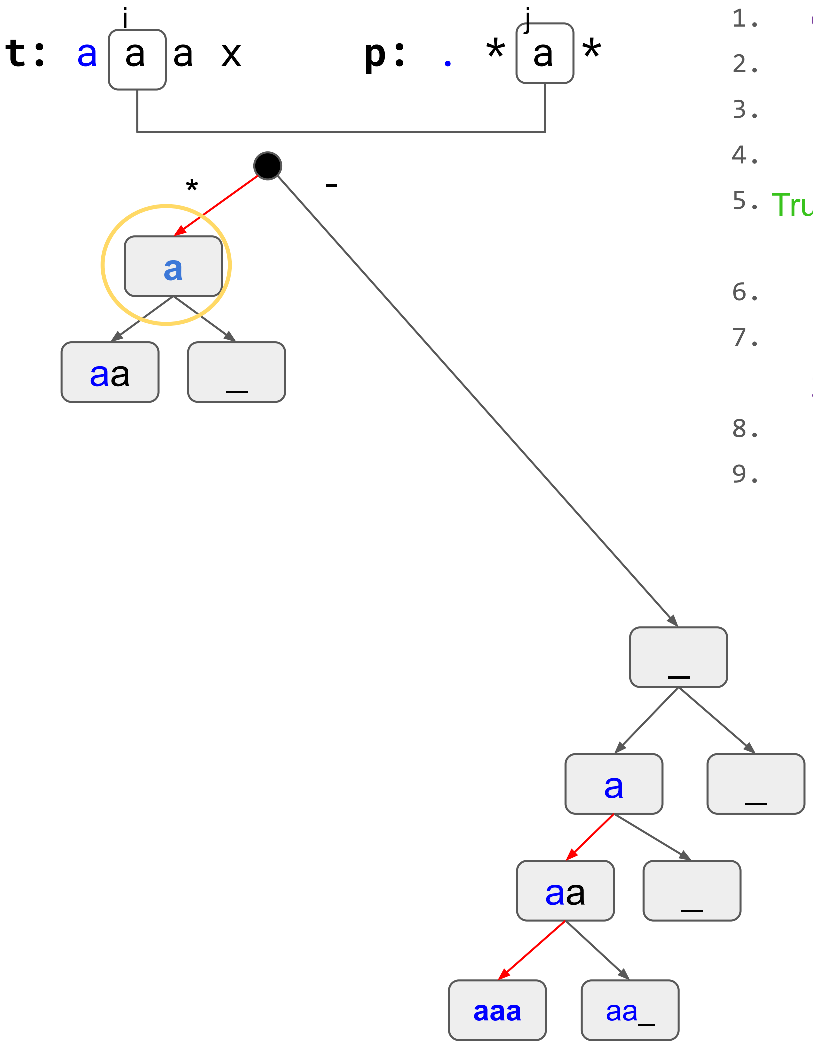
True



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

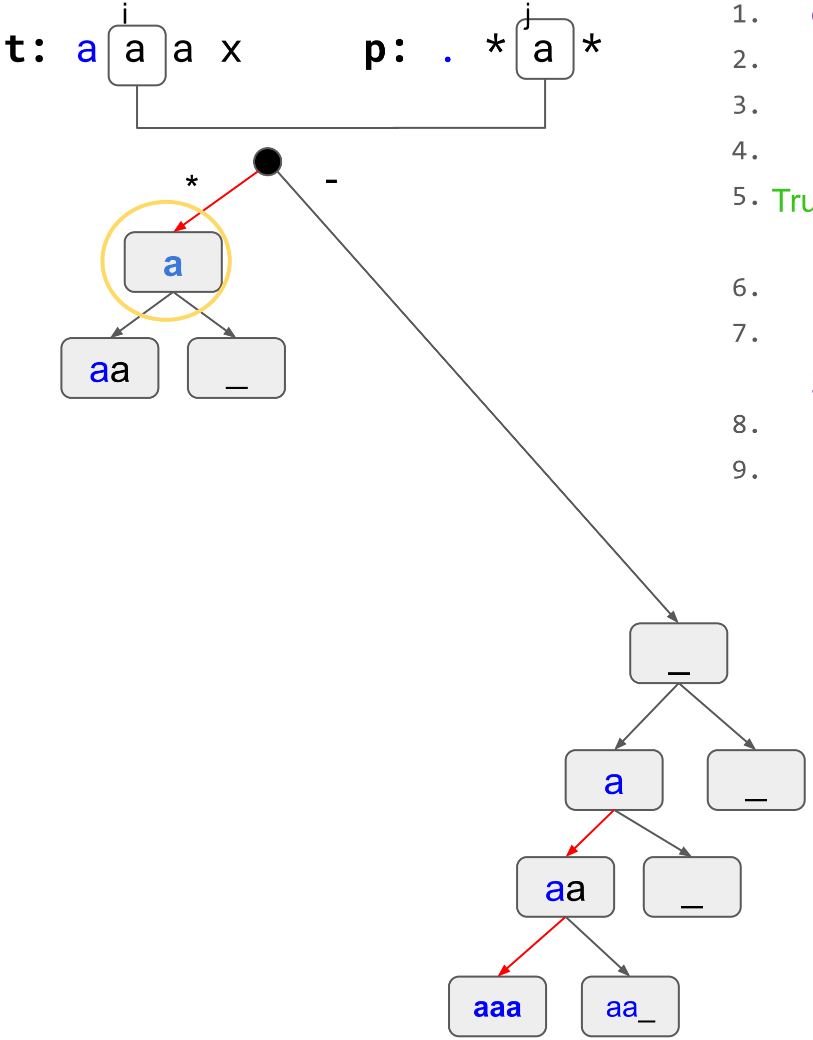


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



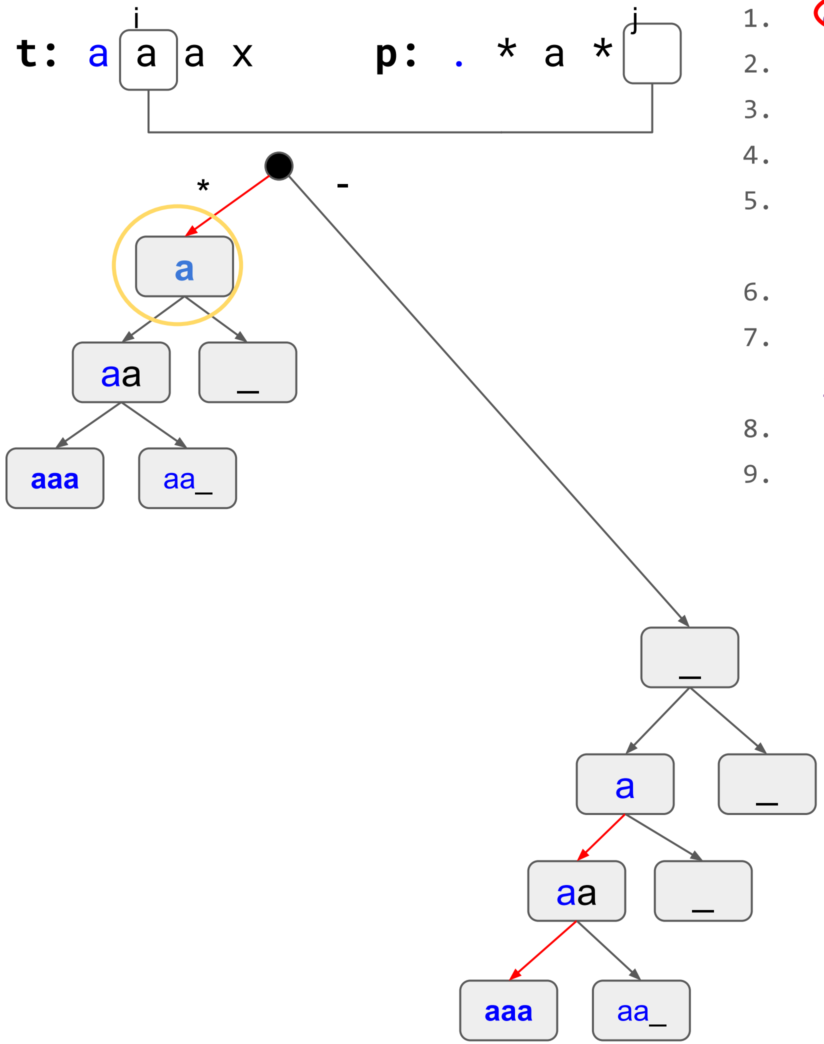
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

True

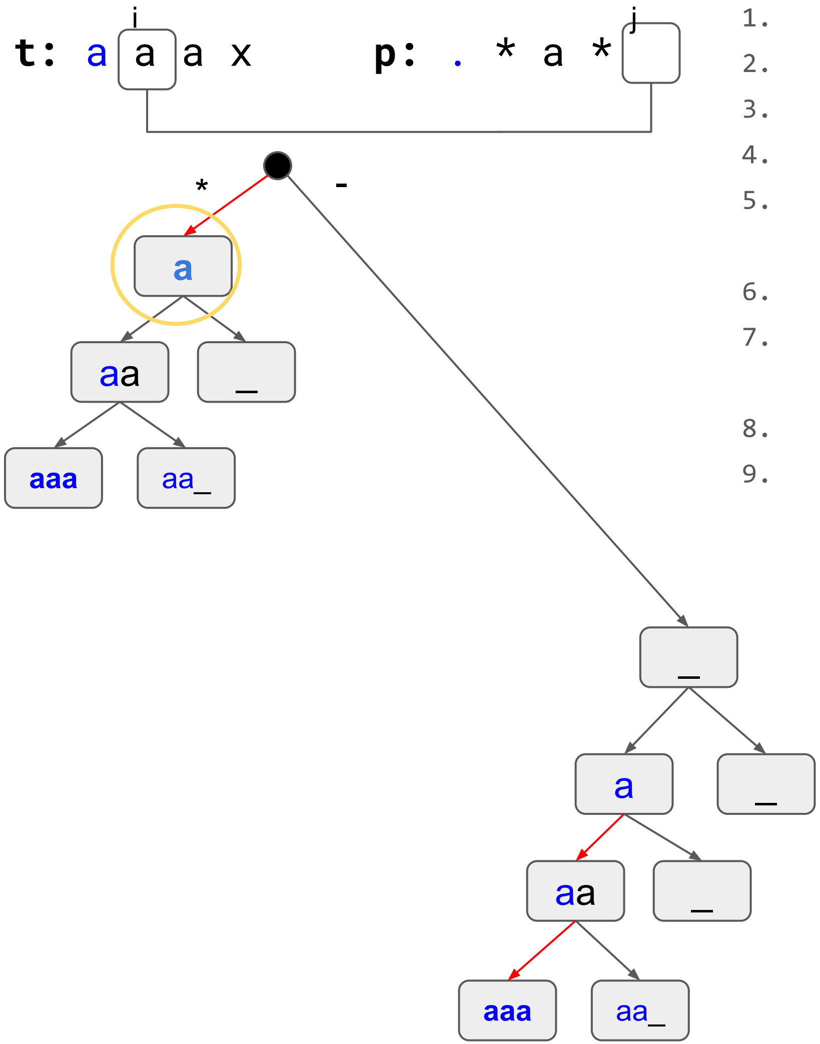


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
        return ( i == len(text) )
    else:
        first_match = i < len(text) and pattern[j] in {text[i], '.'}
        if (j+1 < len(pattern) and pattern[j+1] == '*'):
            return match(i, j+2) or (first_match and match(i+1, j))
        else:
            return first_match and match(i+1, j+1)
```

True

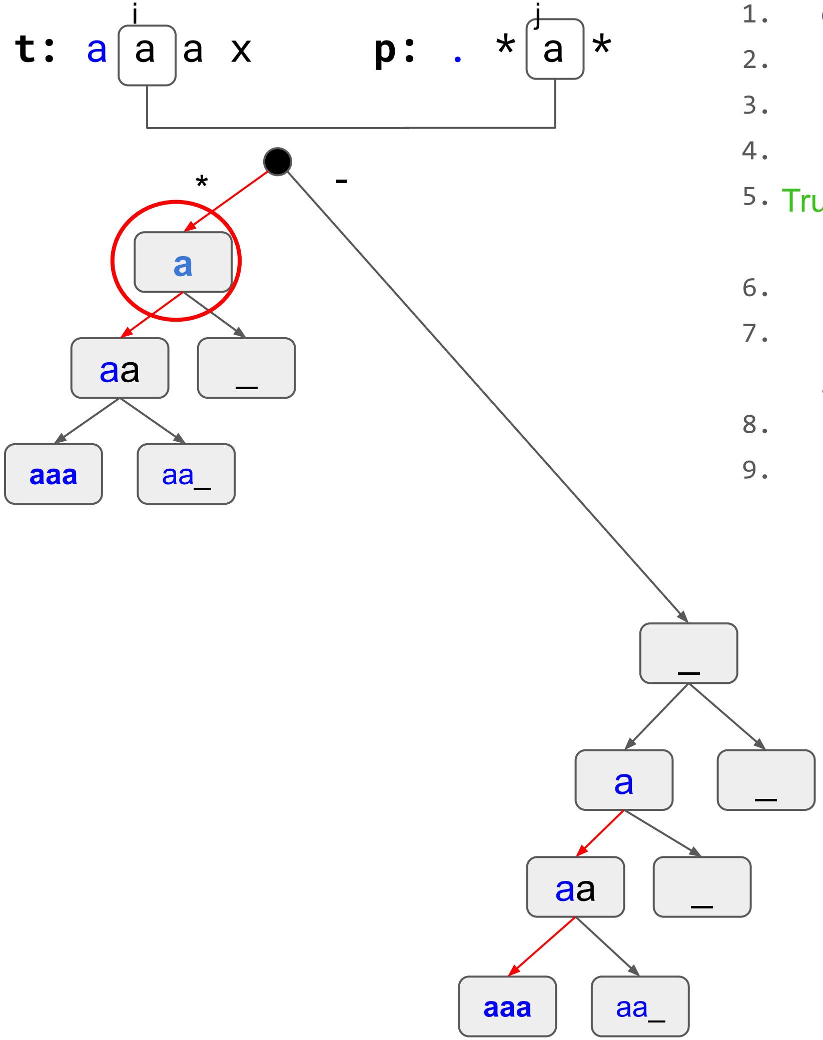


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

False

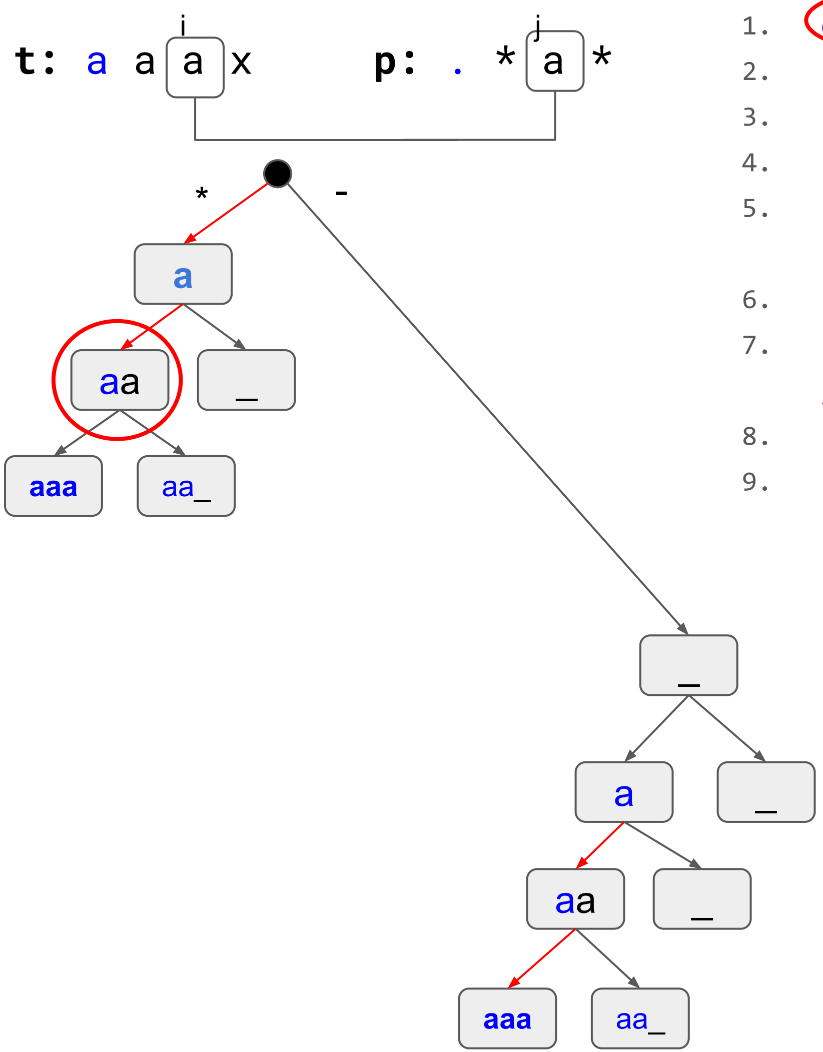


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

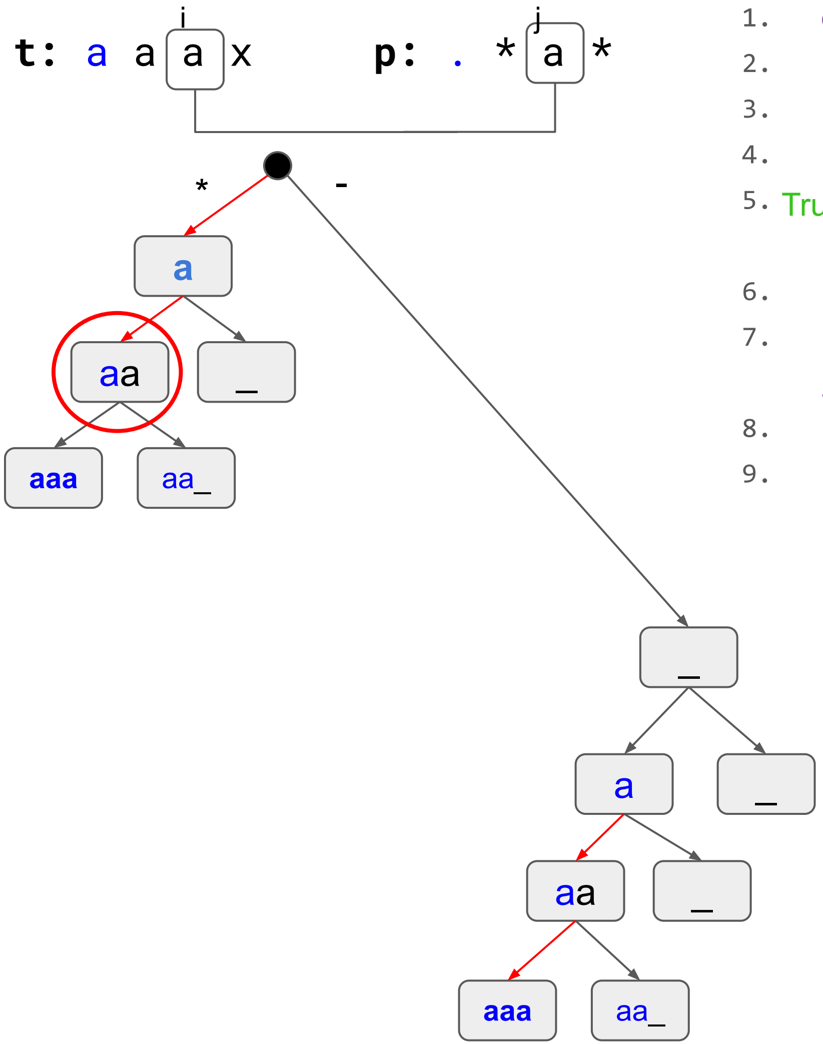
True

False

True

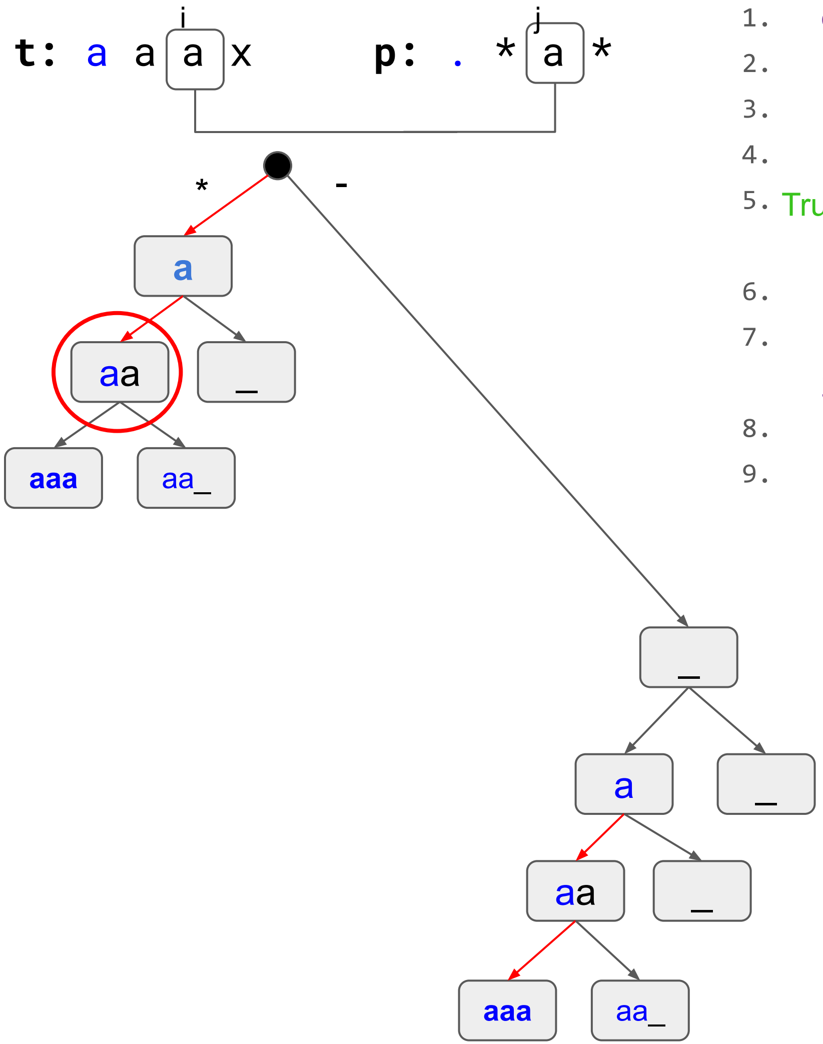


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

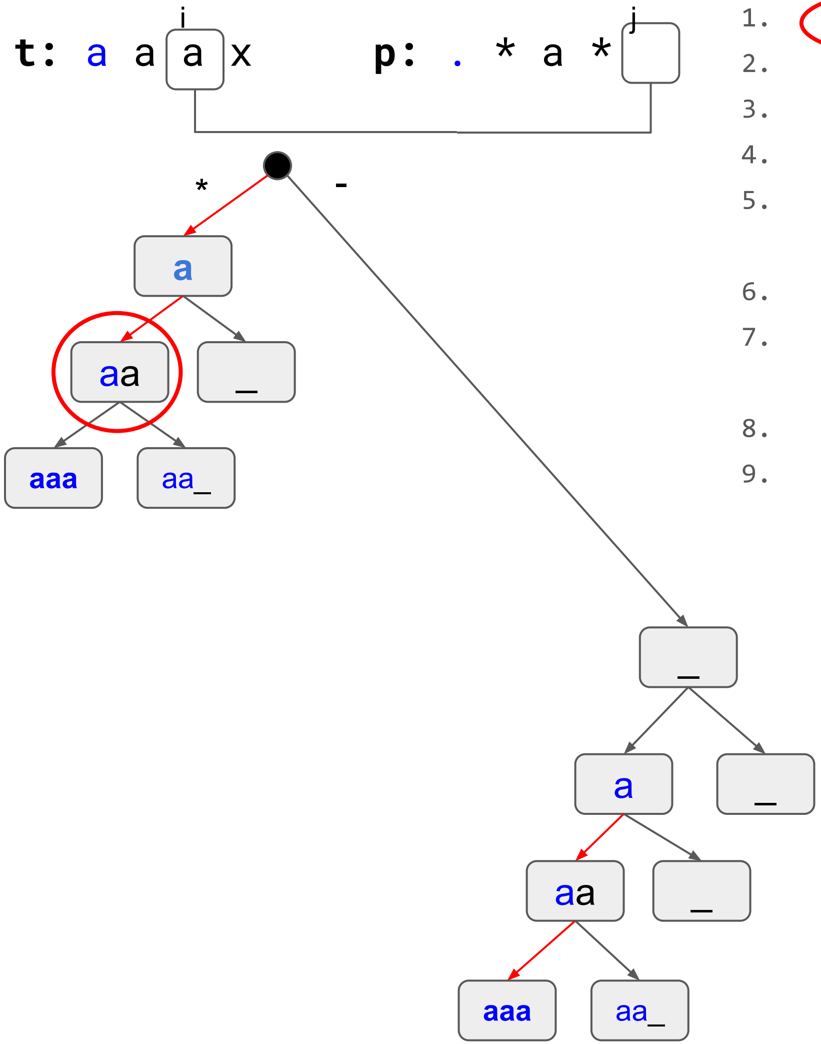


True

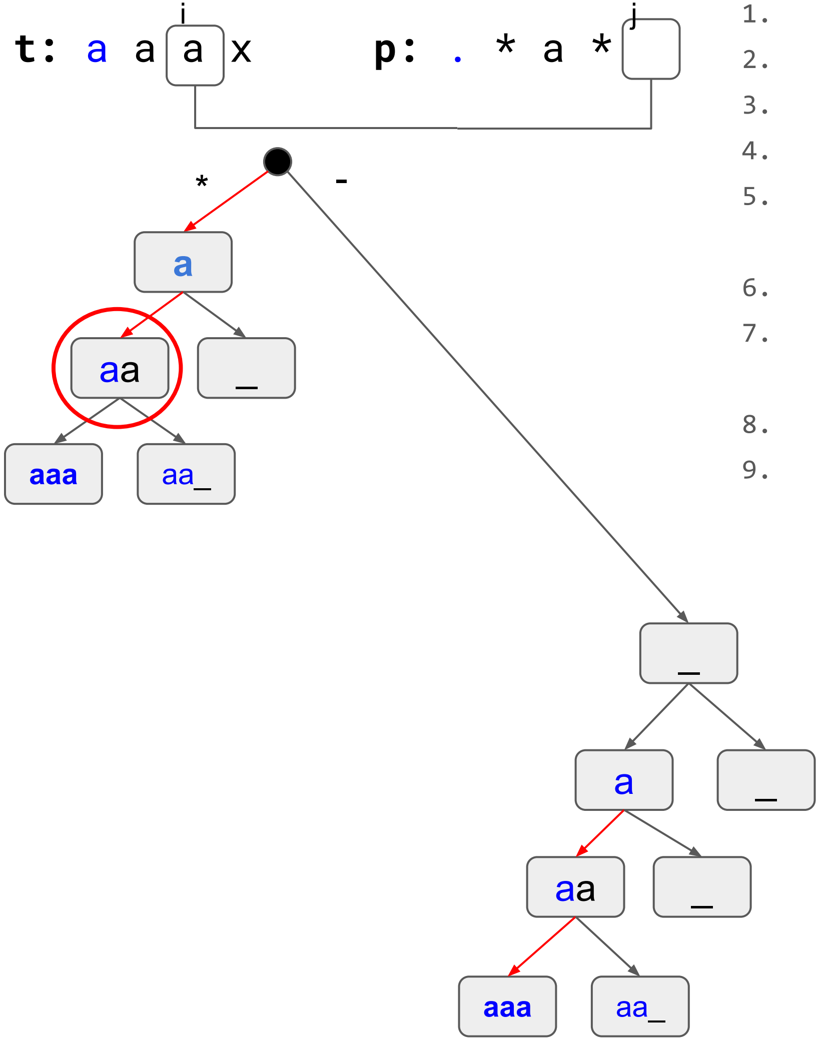
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
True  first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

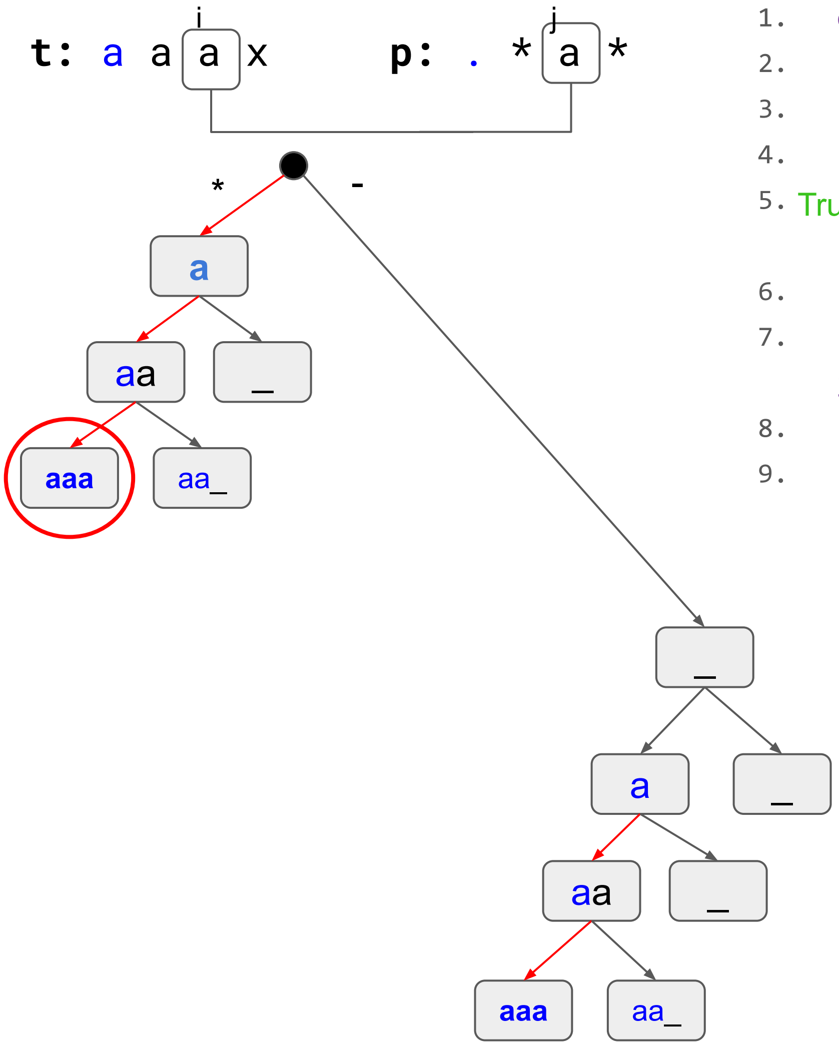


```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

False

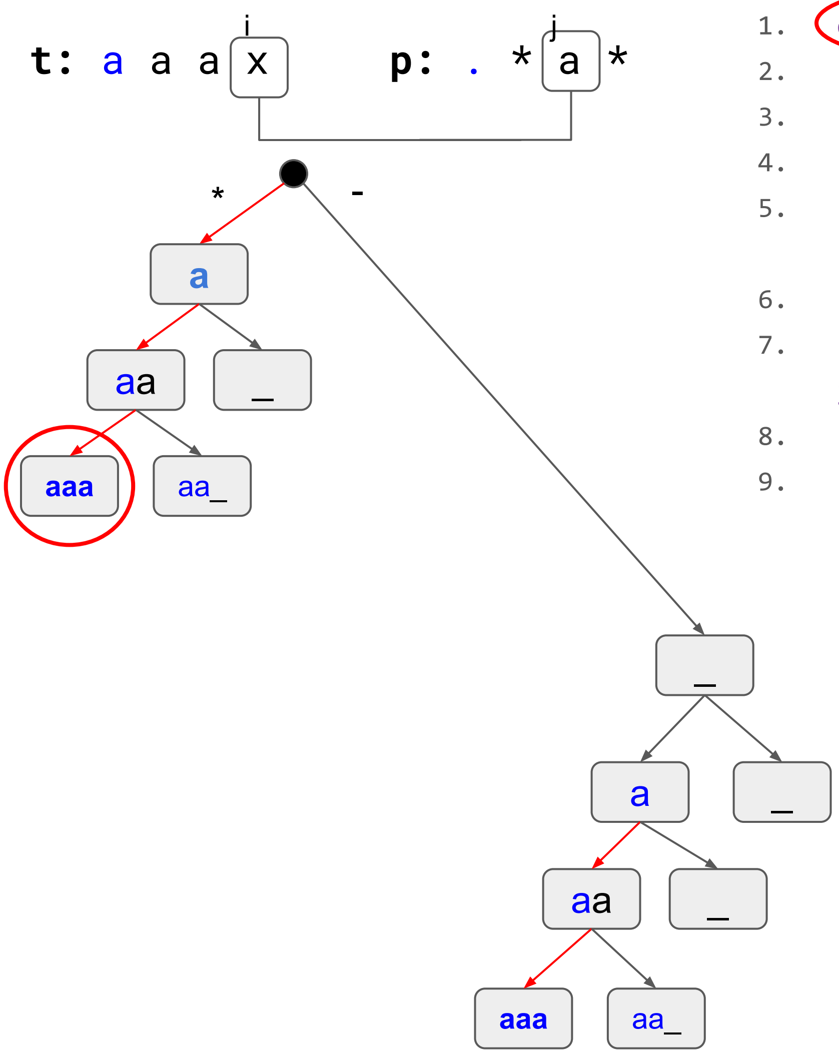


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

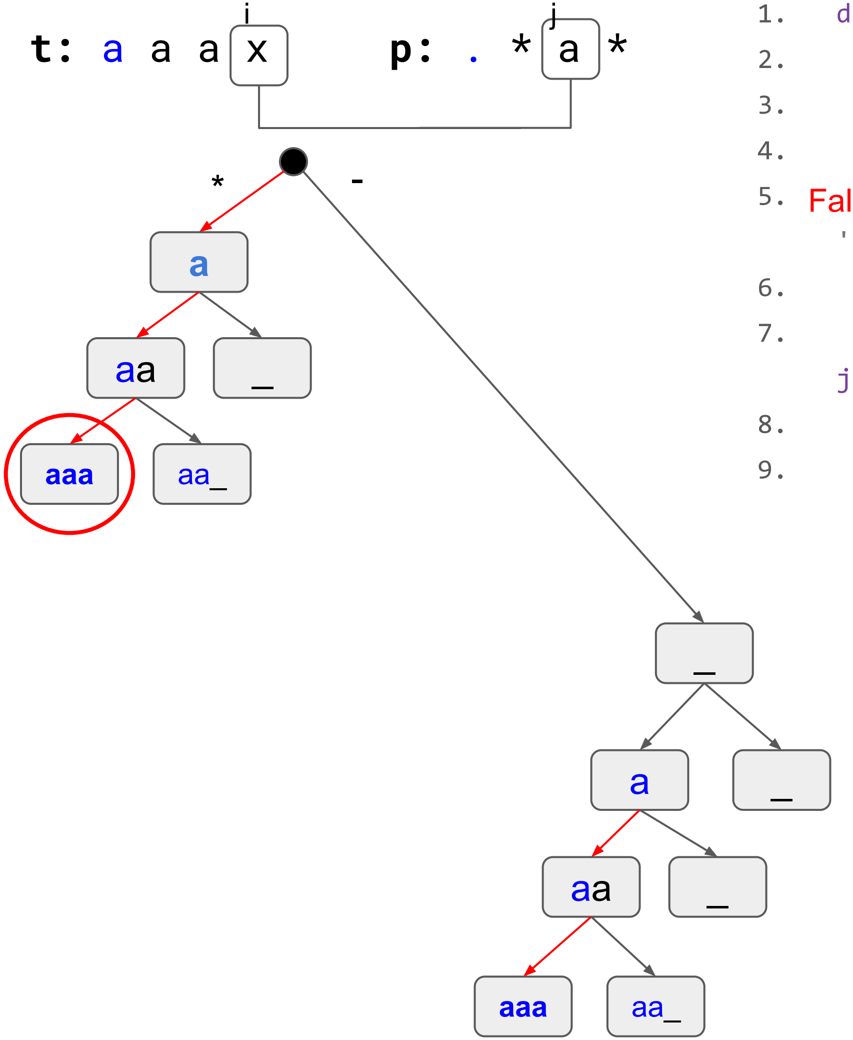
True

False

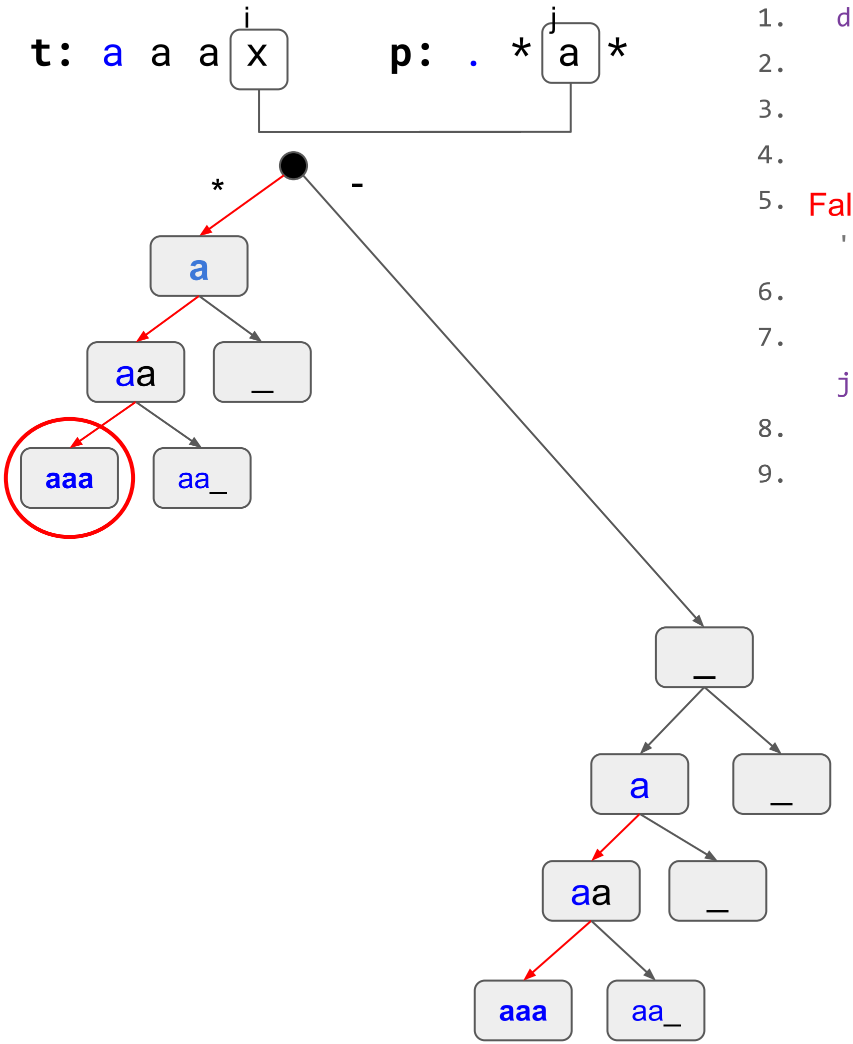
True



```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

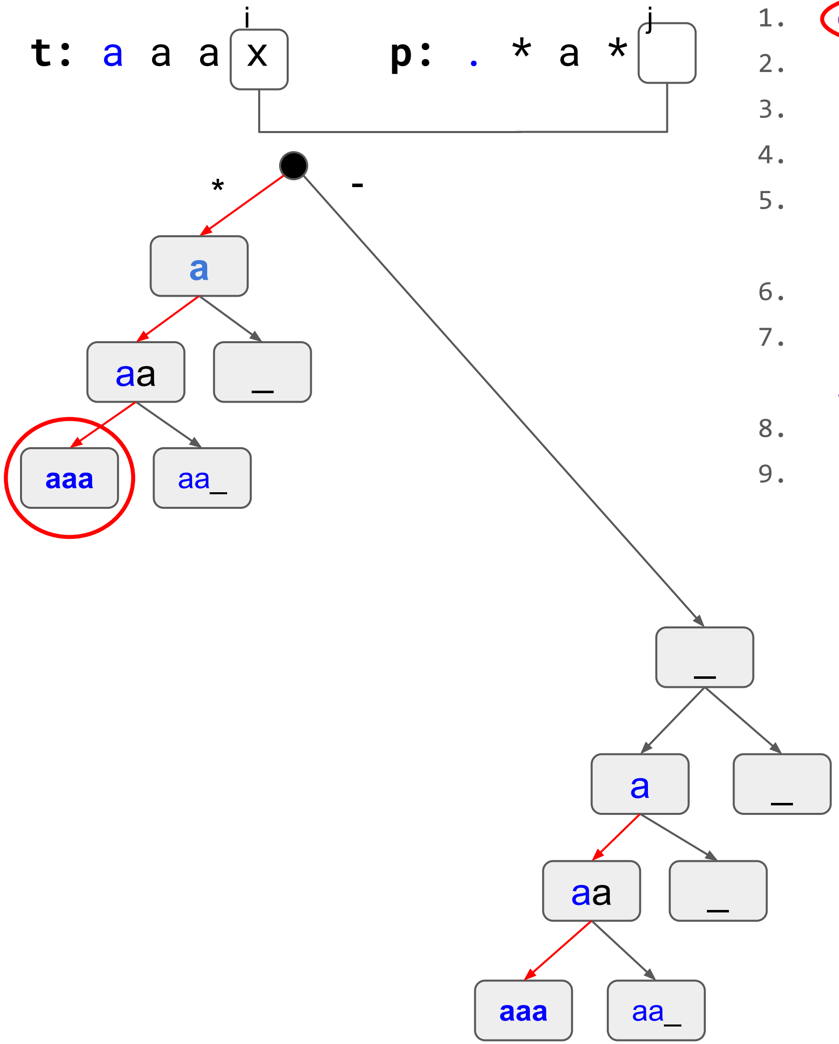


```
def match(i: int, j: int)-> bool:
  if (j == len(pattern)):
   return ( i == len(text) )
  else:
False first_match = i < len(text) and pattern[j] in {text[i], '.'}
   if (j+1 < len(pattern) and pattern[j+1] == '*'):
    return match(i, j+2) or (first_match and match(i+1, j))
   else:
    return first_match and match(i+1, j+1)
```

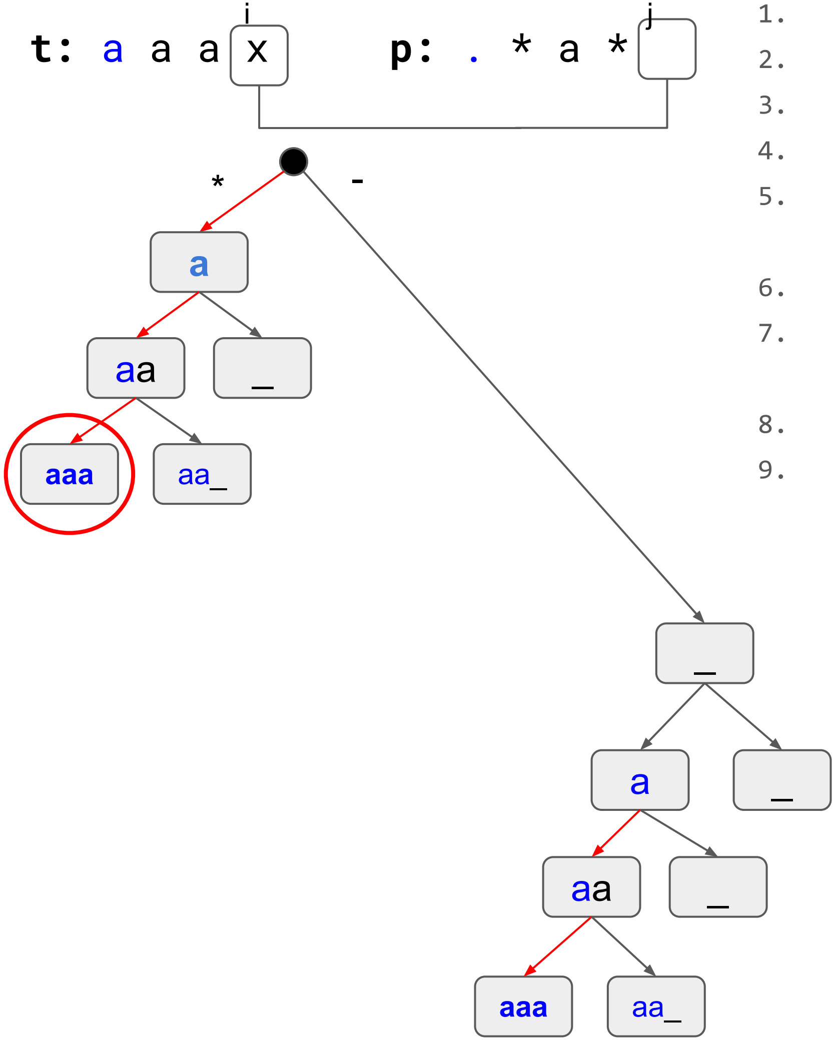


False

```python
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

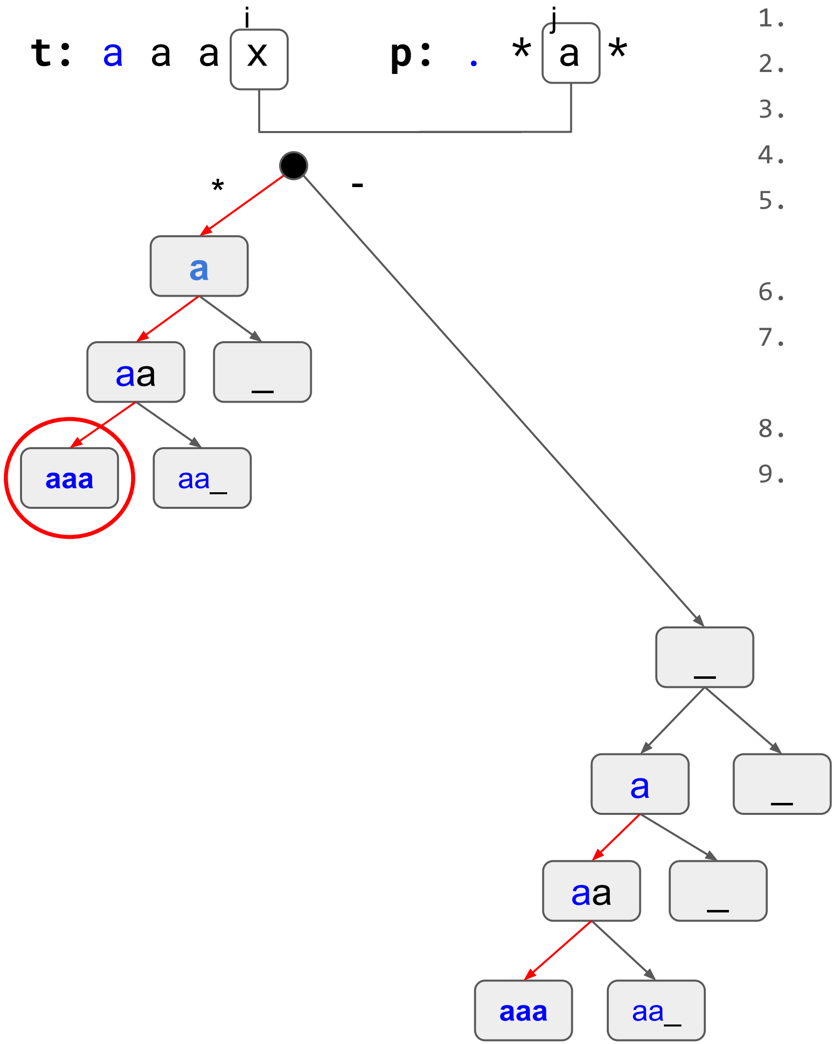


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```



```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

False

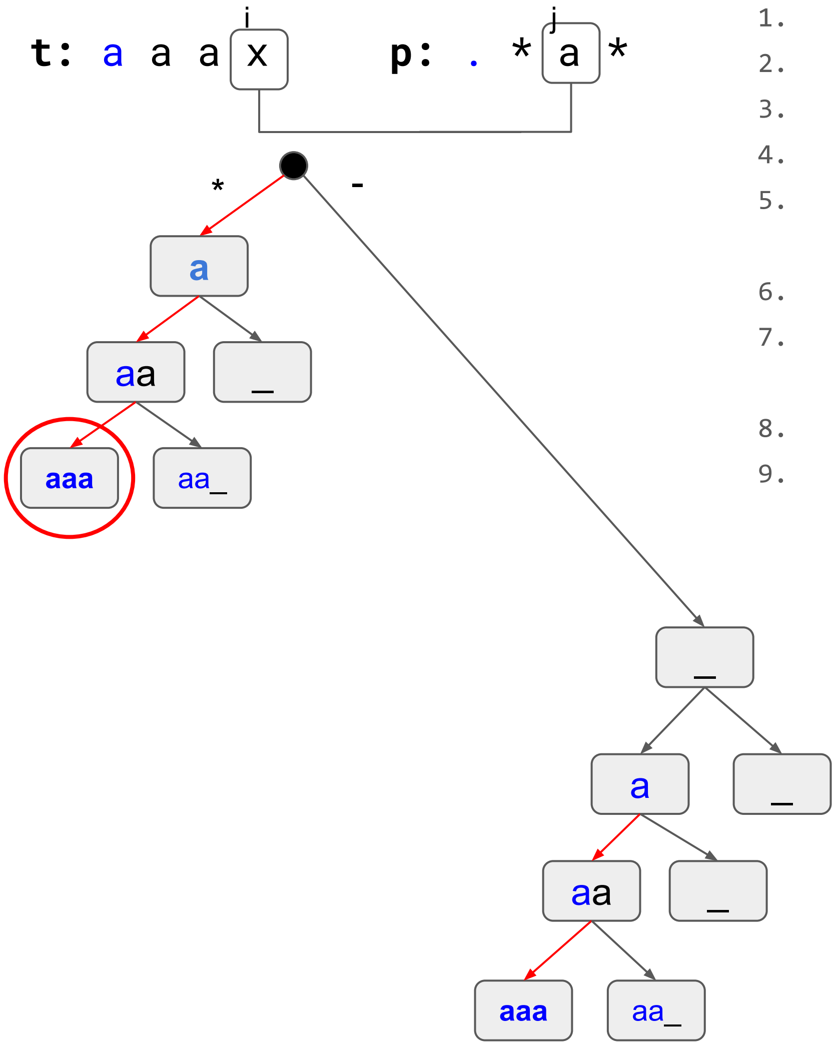


```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i], '.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1, j))
      else:
        return first_match and match(i+1, j+1)
```

False

False

False

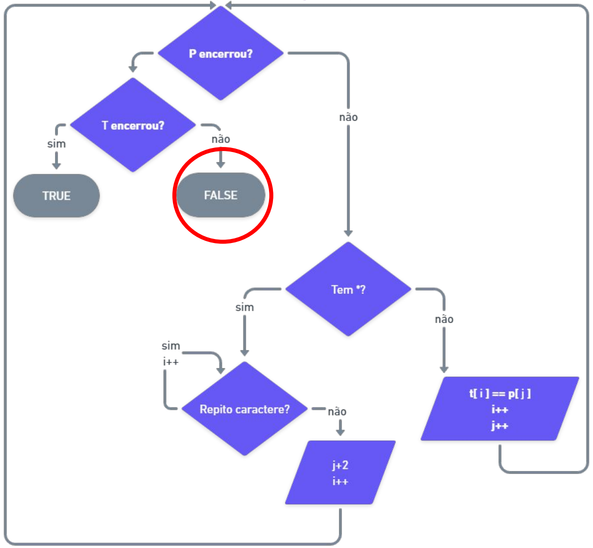


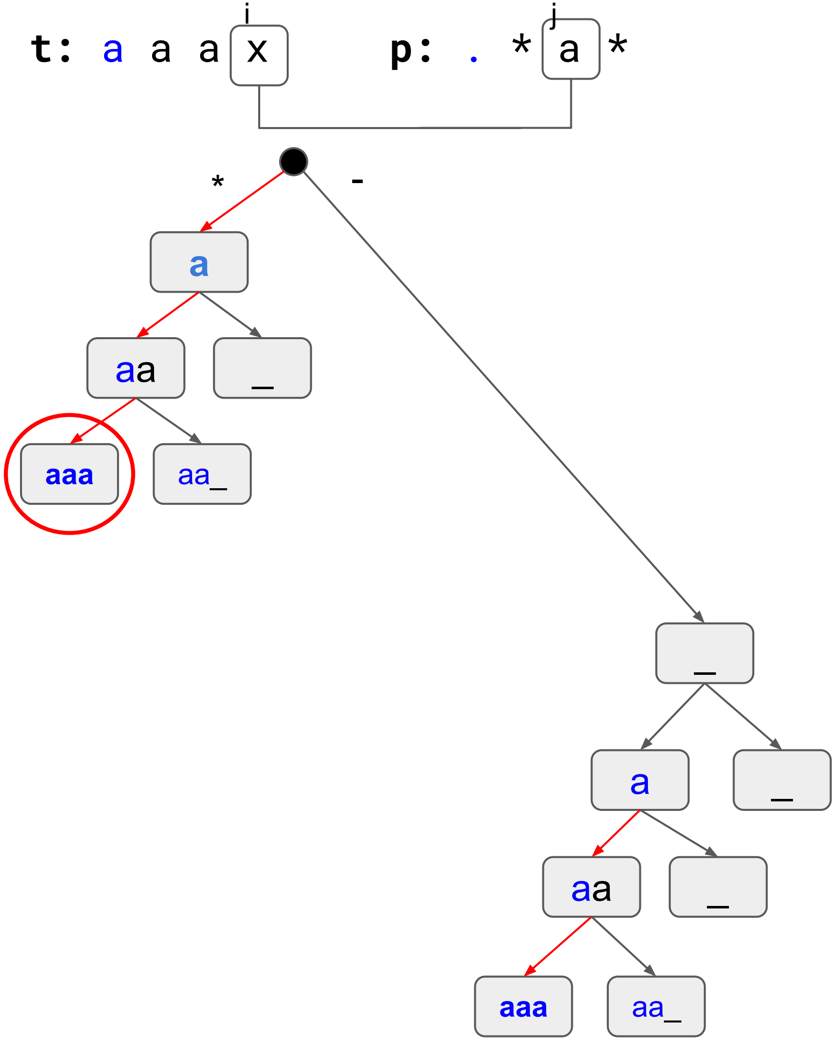
```
def match(i: int, j: int)-> bool:
    if (j == len(pattern)):
      return ( i == len(text) )
    else:
      first_match = i < len(text) and pattern[j] in {text[i],
'.'}
      if (j+1 < len(pattern) and pattern[j+1] == '*'):
        return match(i, j+2) or (first_match and match(i+1,
j))
      else:
        return first_match and match(i+1, j+1)
```

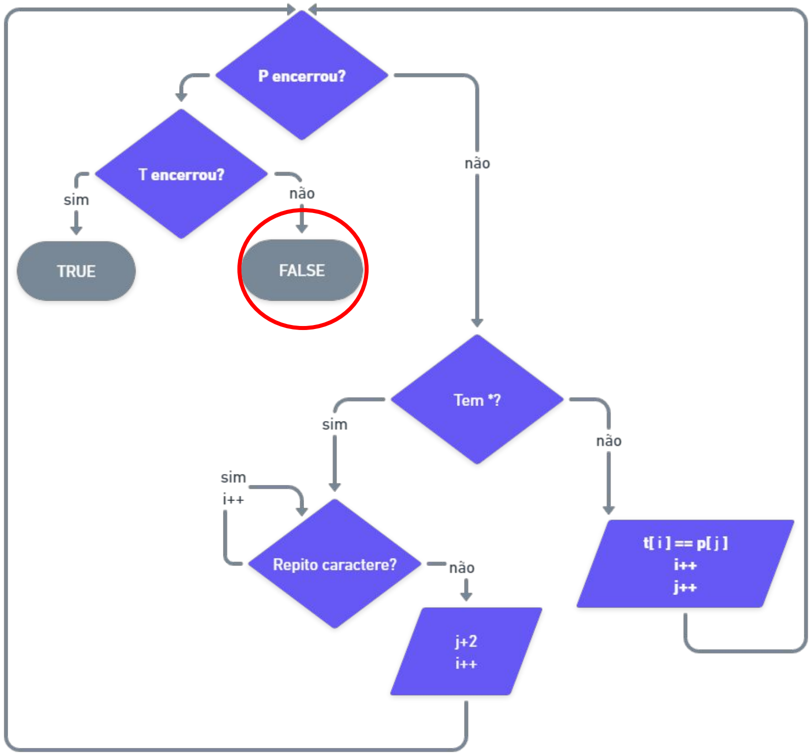
False

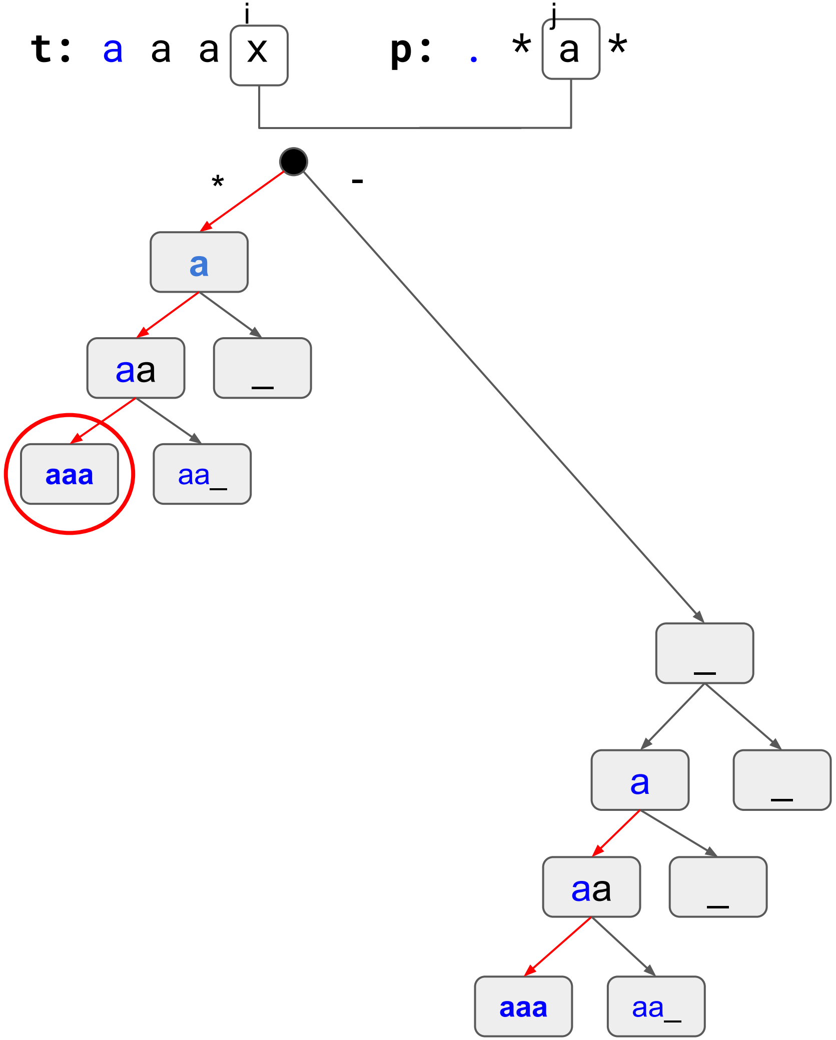
False

False

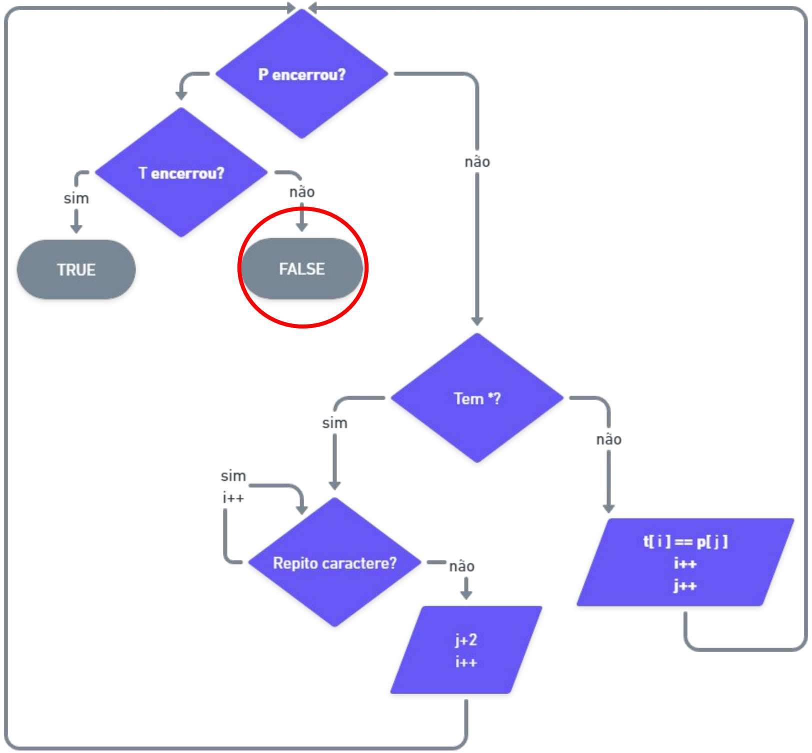



i
j
t: a a a x
p: . * a *
*
-
a
aa
_
aaa
aa_
_
a
_
aa
_
aaa
aa_


P encerrou?
T encerrou?
sim
não
TRUE
FALSE
não
Tem *?
sim
não
sim
i++
Repito caractere?
não
j+2
i++
t[i] == p[j]
i++
j++



i
t: a a a x
j
p: . * a *
*
-
a
aa
_
aaa
aa_
_
a
_
aa
_
aaa
aa_



P encerrou?
T encerrou?
sim
não
TRUE
FALSE
não
Tem *?
sim
não
sim
i++
Repito caractere?
não
j+2
i++
t[i] == p[j]
i++
j++