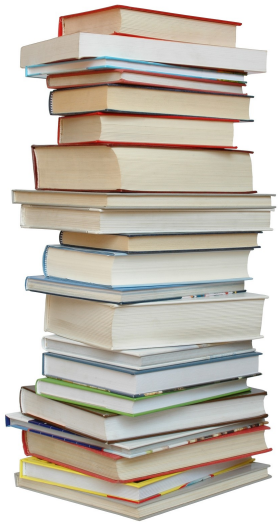
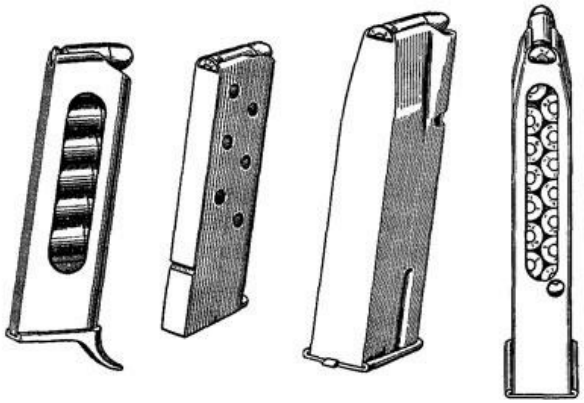
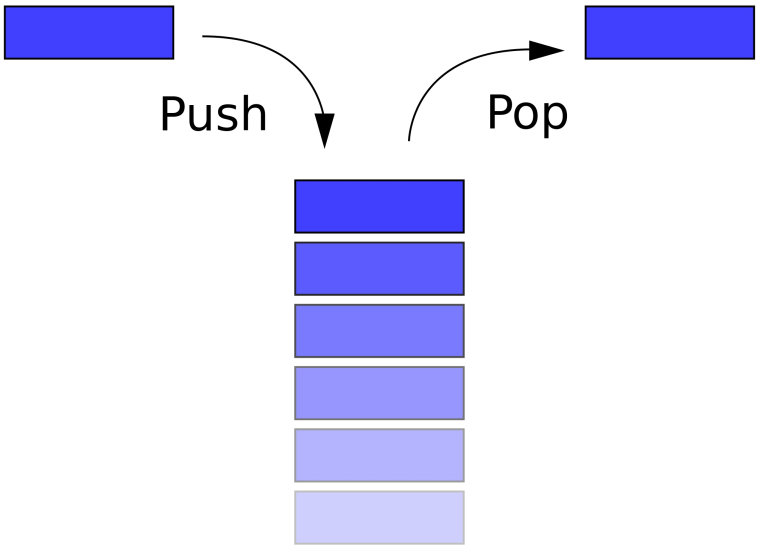
Стек

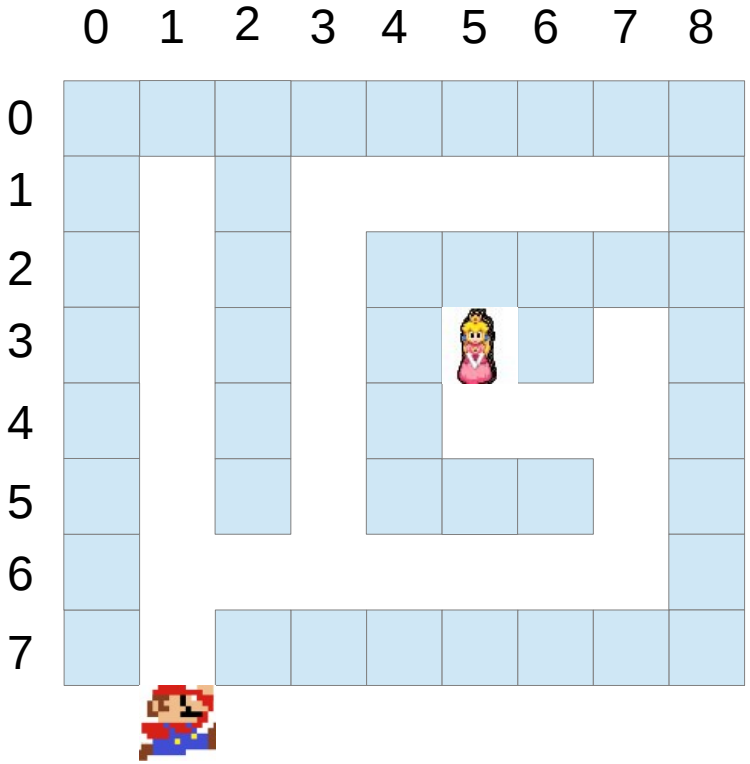
Стек

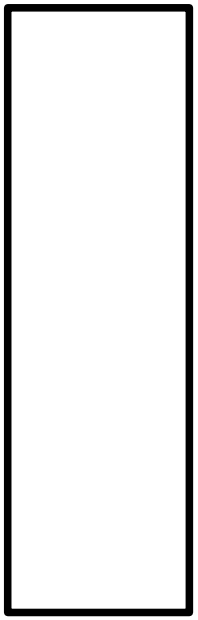






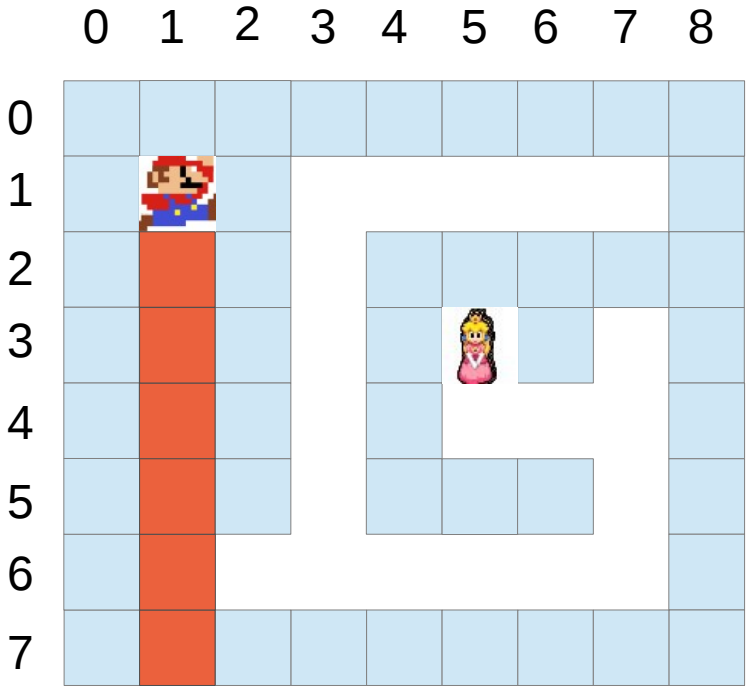
Применение стека

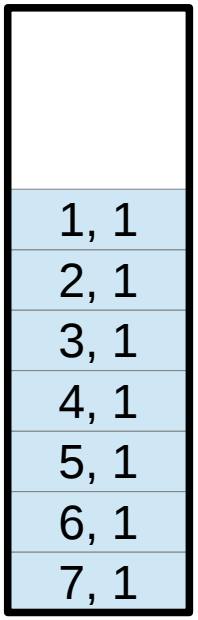




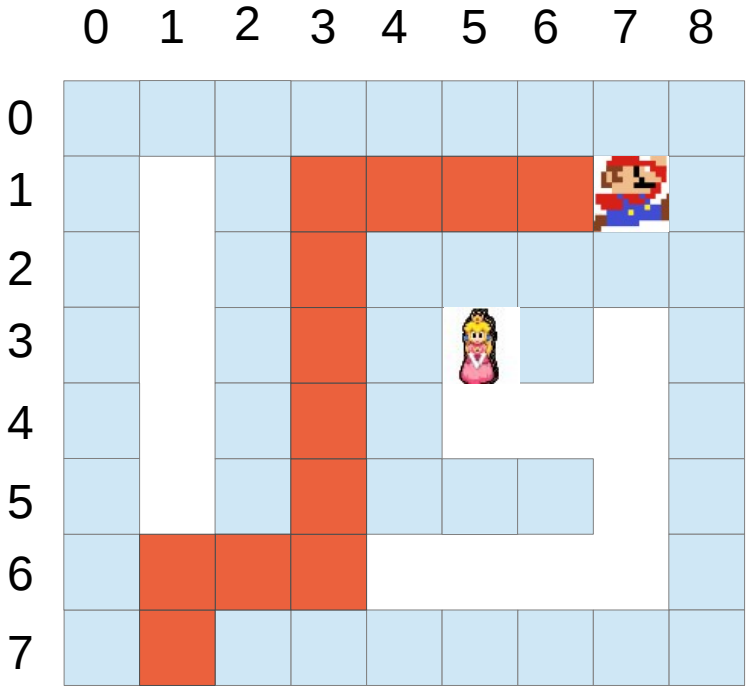
Применение стека

Будем запоминать пройденный путь в стеке





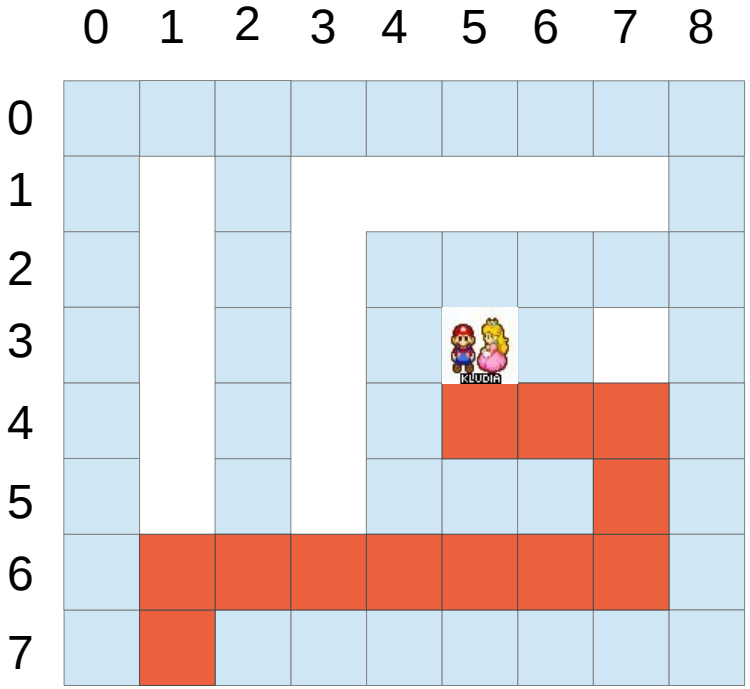
Применение стека

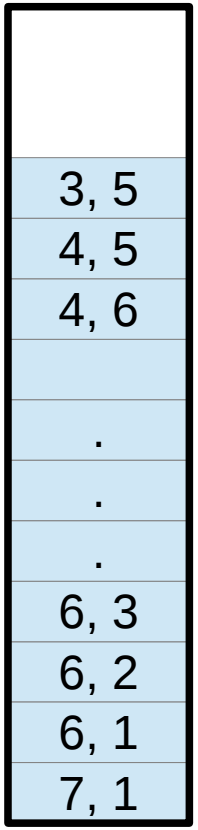


1, 7
1, 6
1, 5
.
.
.
.
6, 3
6, 2
6, 1
7, 1

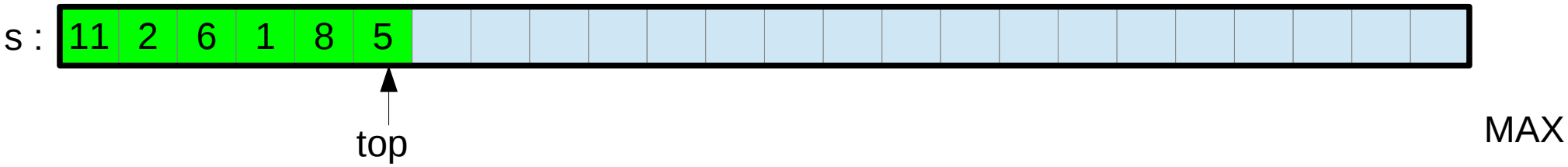
Применение стека

Теперь выбраться из лабиринта не проблема!



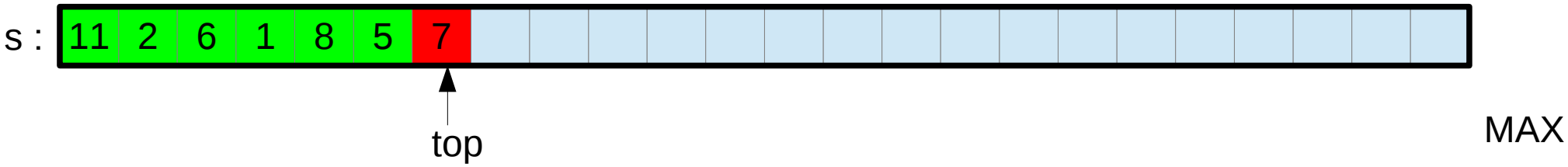


Стек на основе массива



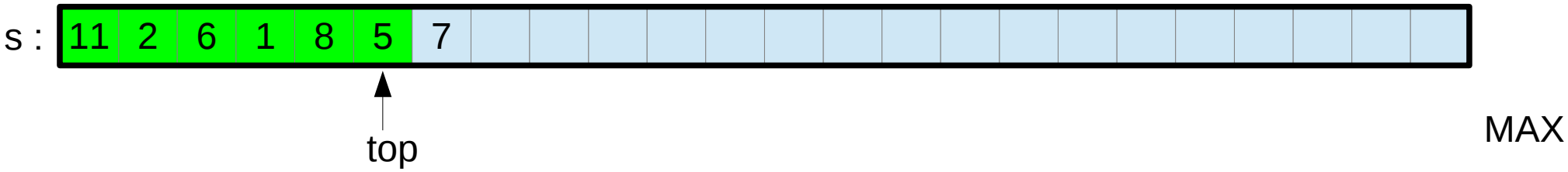

```
const int MAX = 24;

int s[MAX];

int top = -1; // изначально top никуда не указывает


bool empty()
{
    return (top == -1);
}
```

Добавление элемента



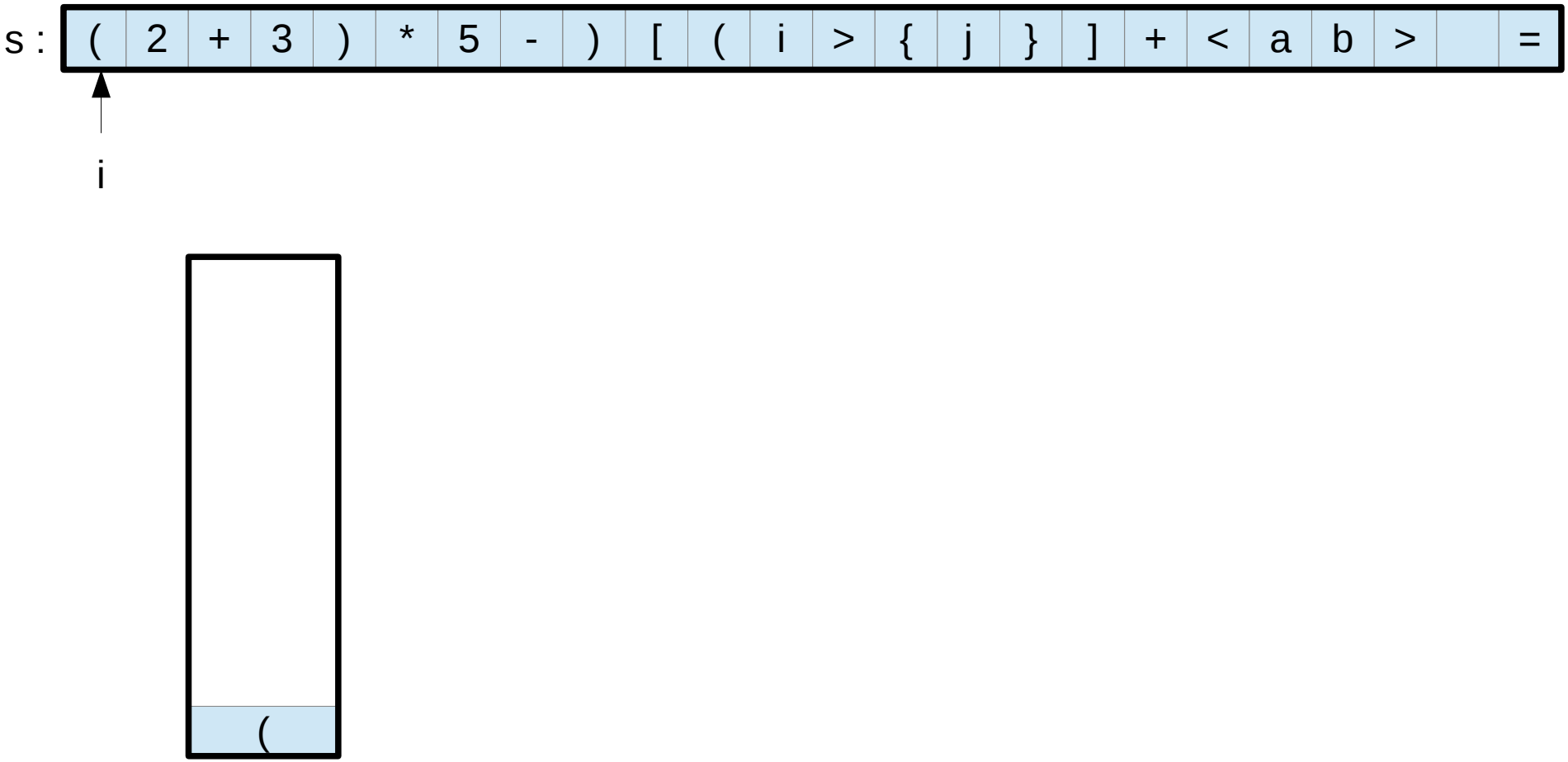

```
void push(int x)
{
    top++;  // top = top + 1
    s[top] = x;
}
```

Извлечение элемента

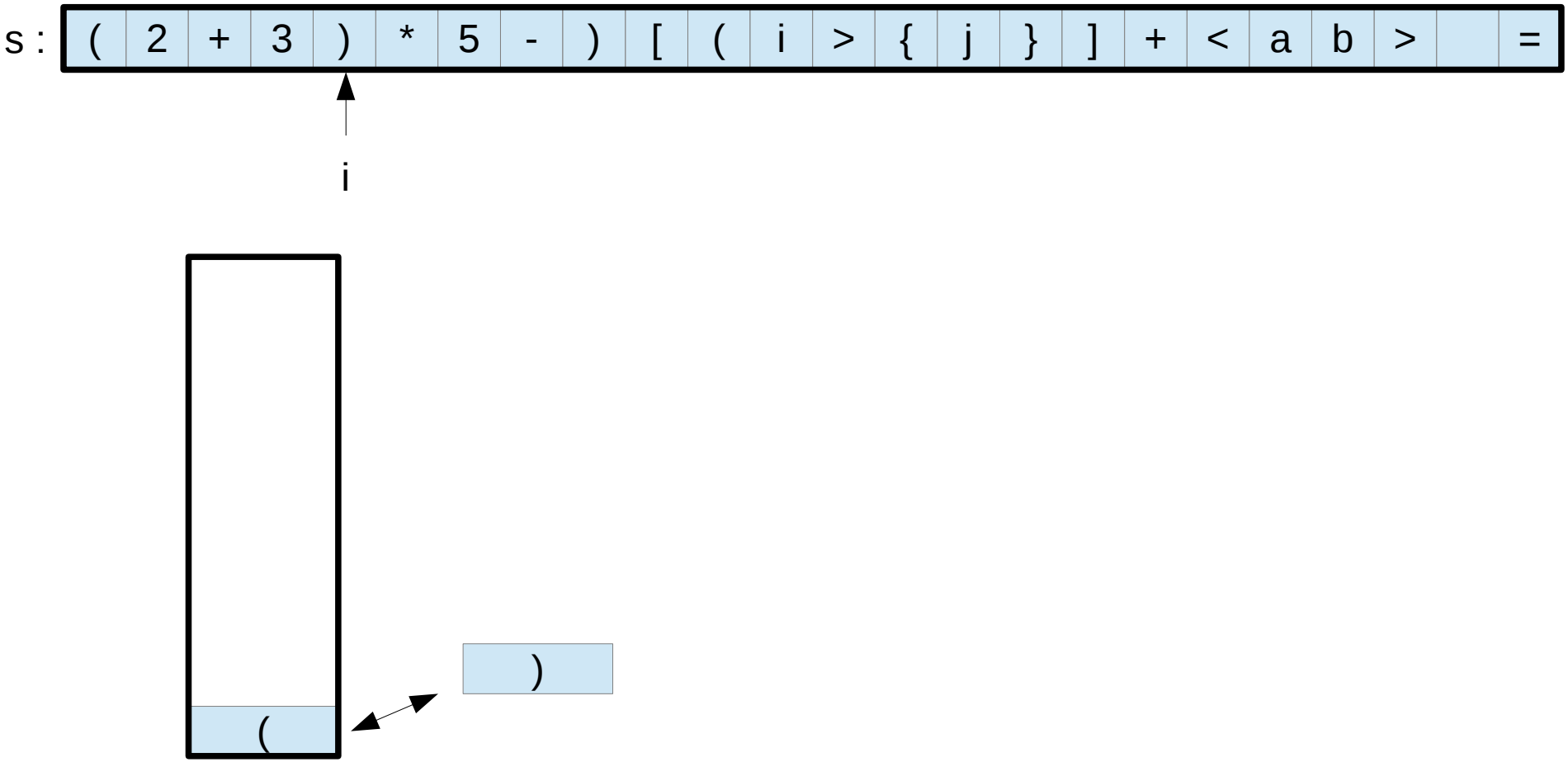



```
int pop()
{
    int x = s[top];
    top--;  // top = top - 1
    return x;
}
```

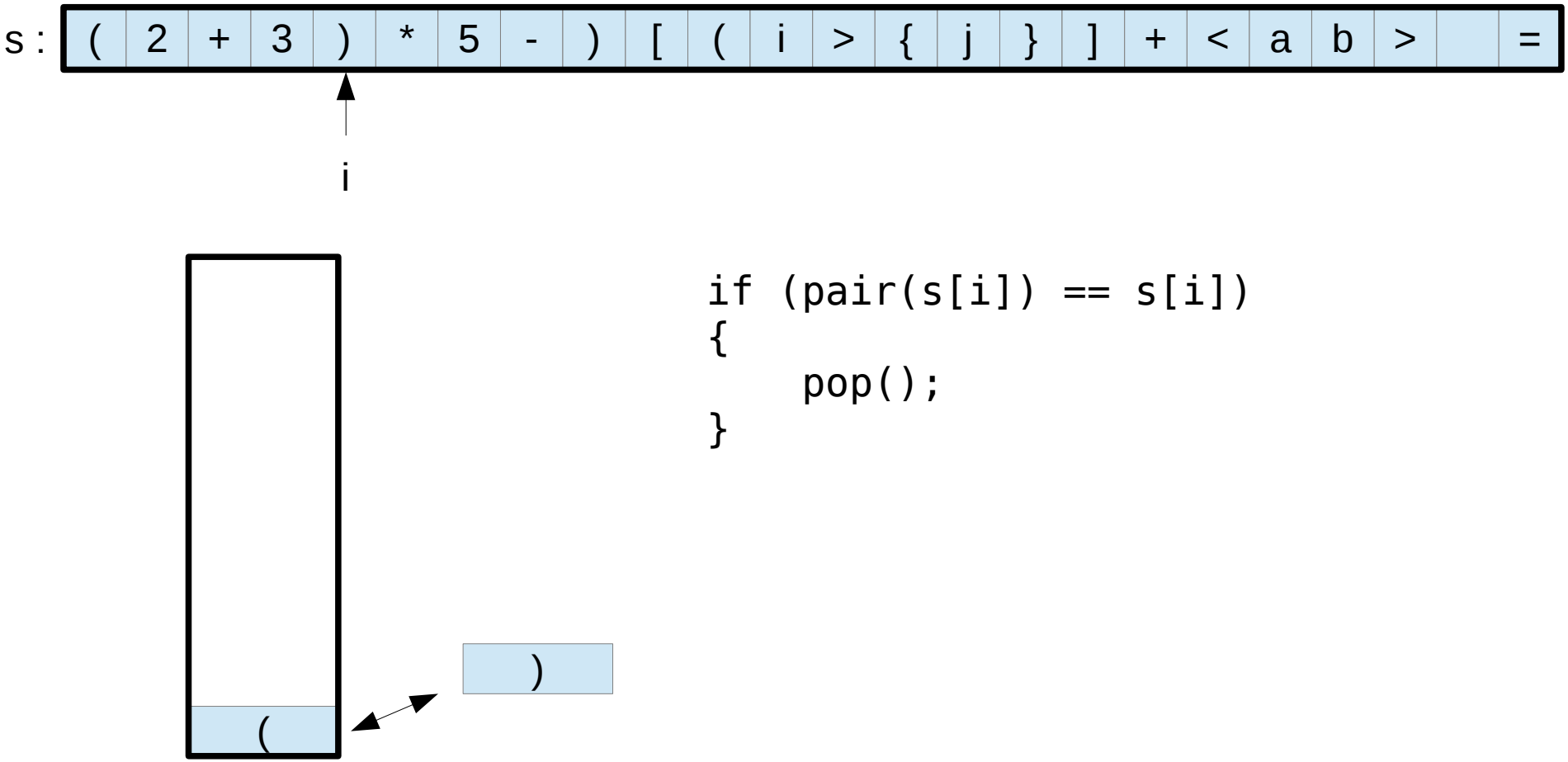
Проверка скобок



Проверка скобок

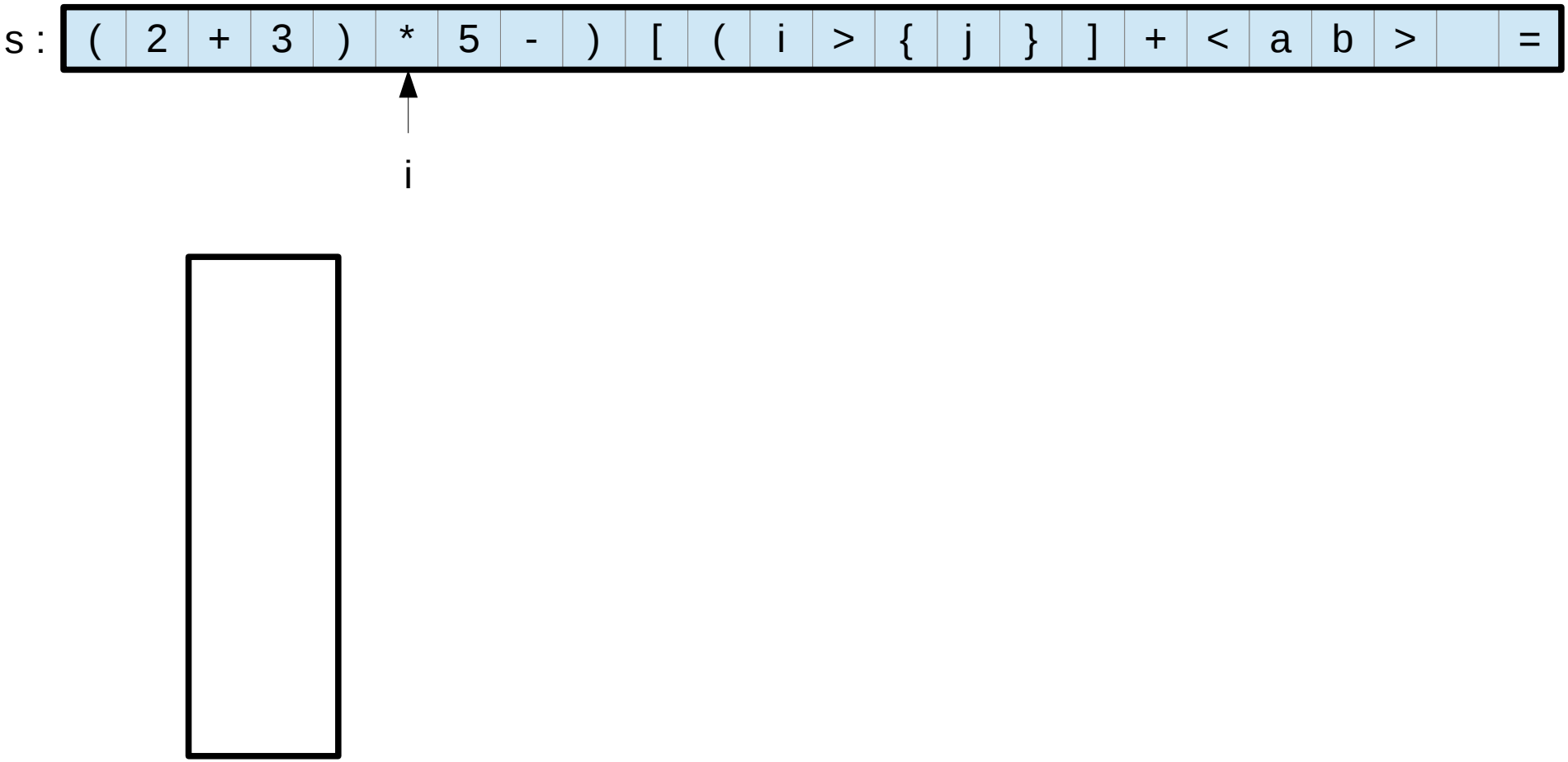


Проверка скобок

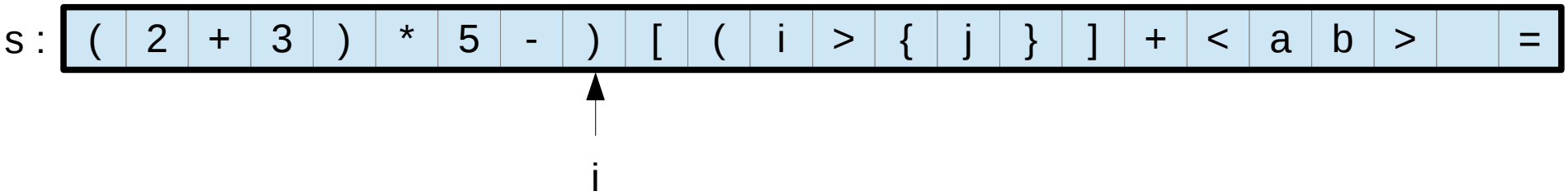


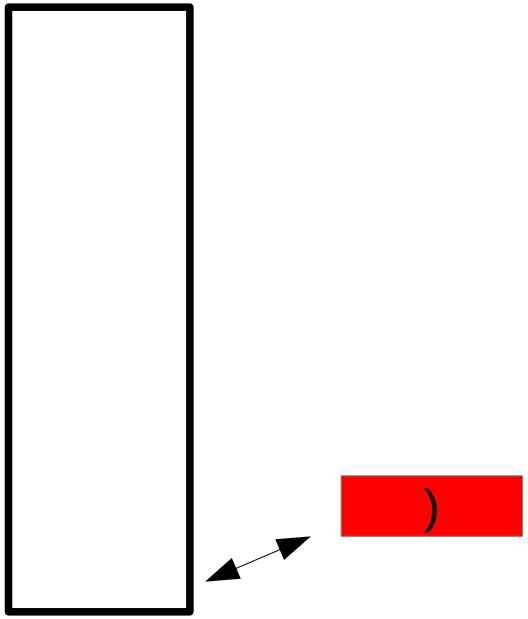

```
if (pair(s[i]) == s[i])
{
    pop();
}
```

Проверка скобок



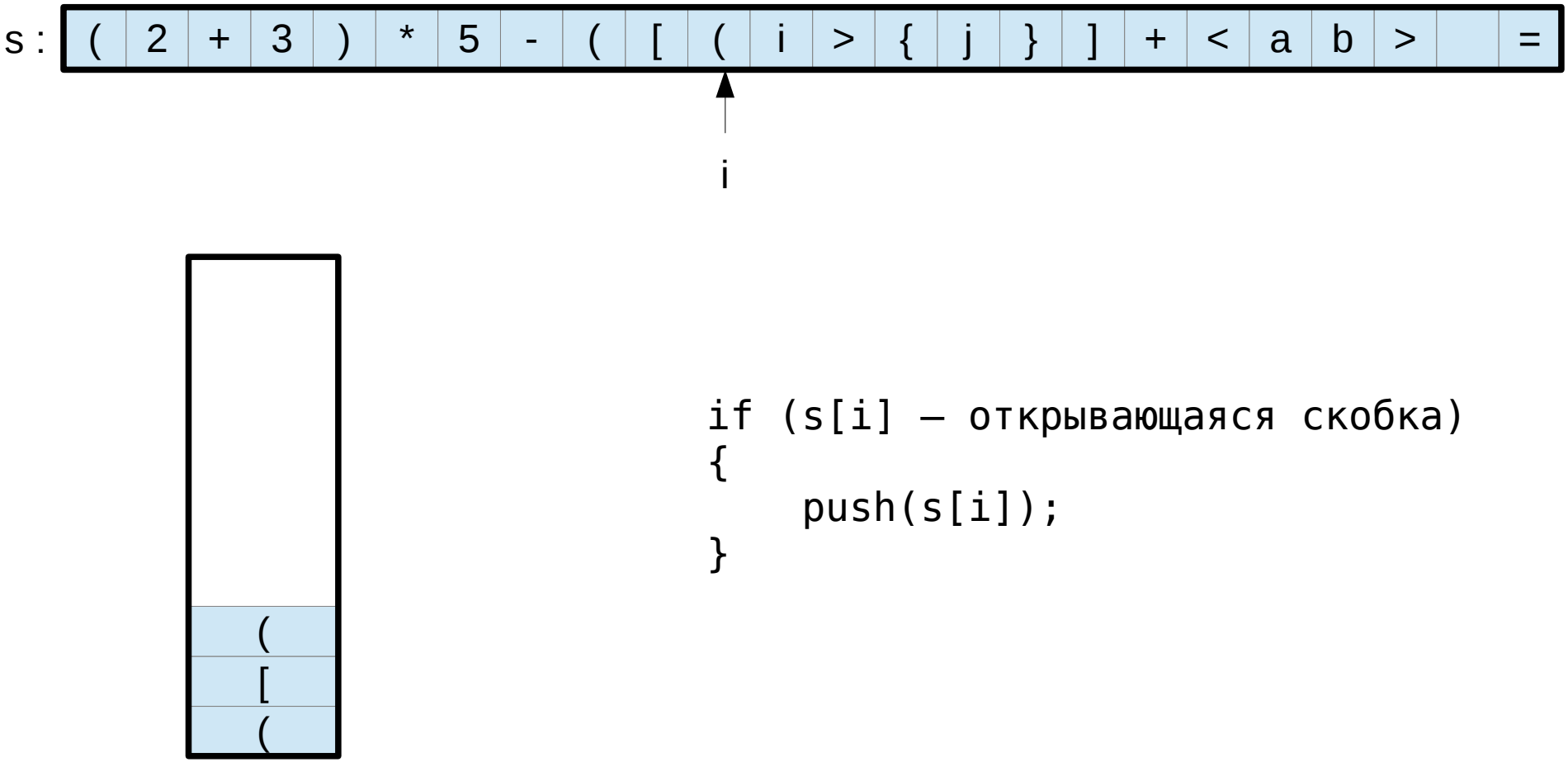
Проверка скобок





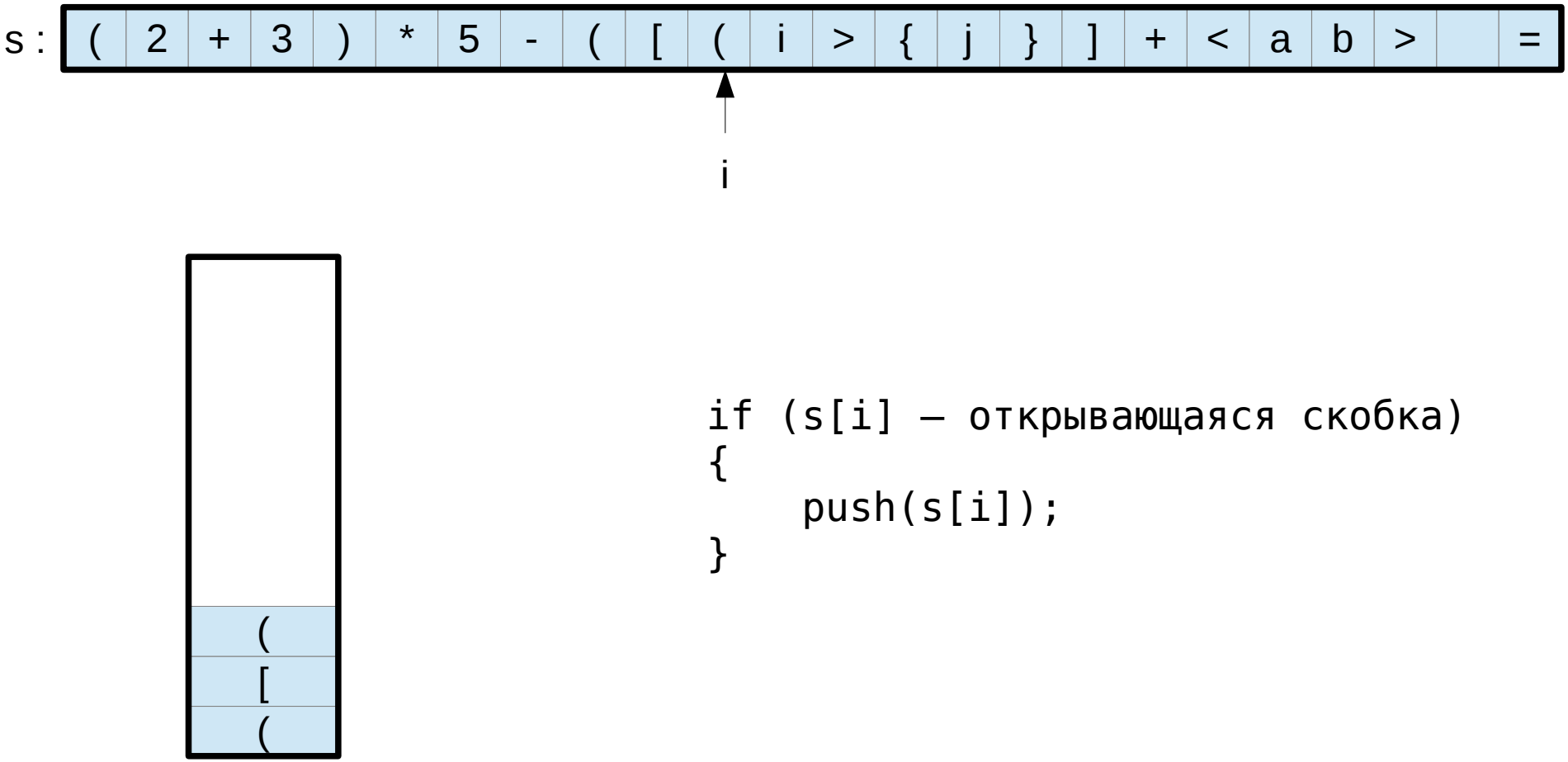

```
if (empty())
{
    return false;
}
```

Проверка скобок



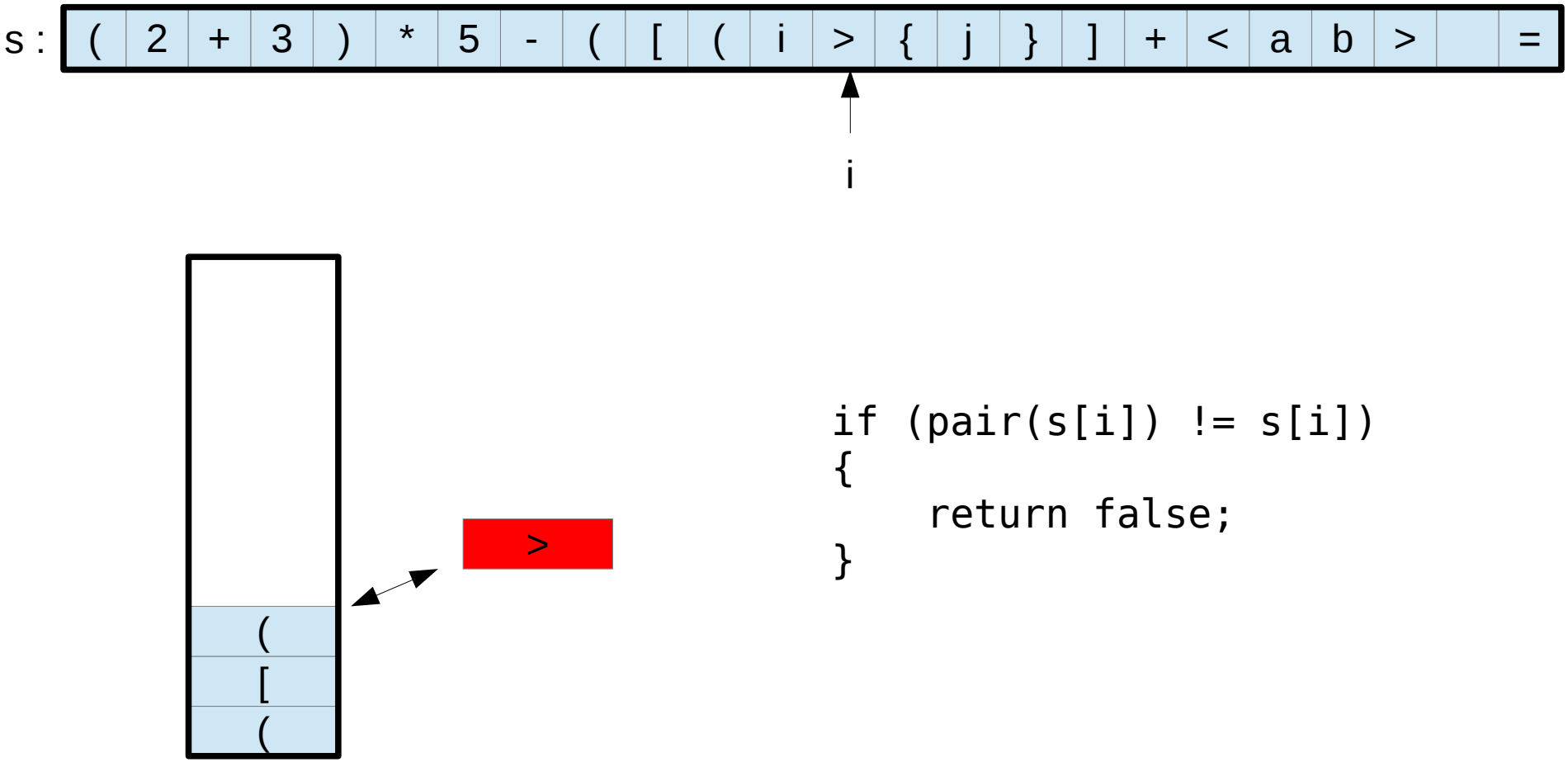

```
if (s[i] – открывающаяся скобка)
{
    push(s[i]);
}
```

Проверка скобок



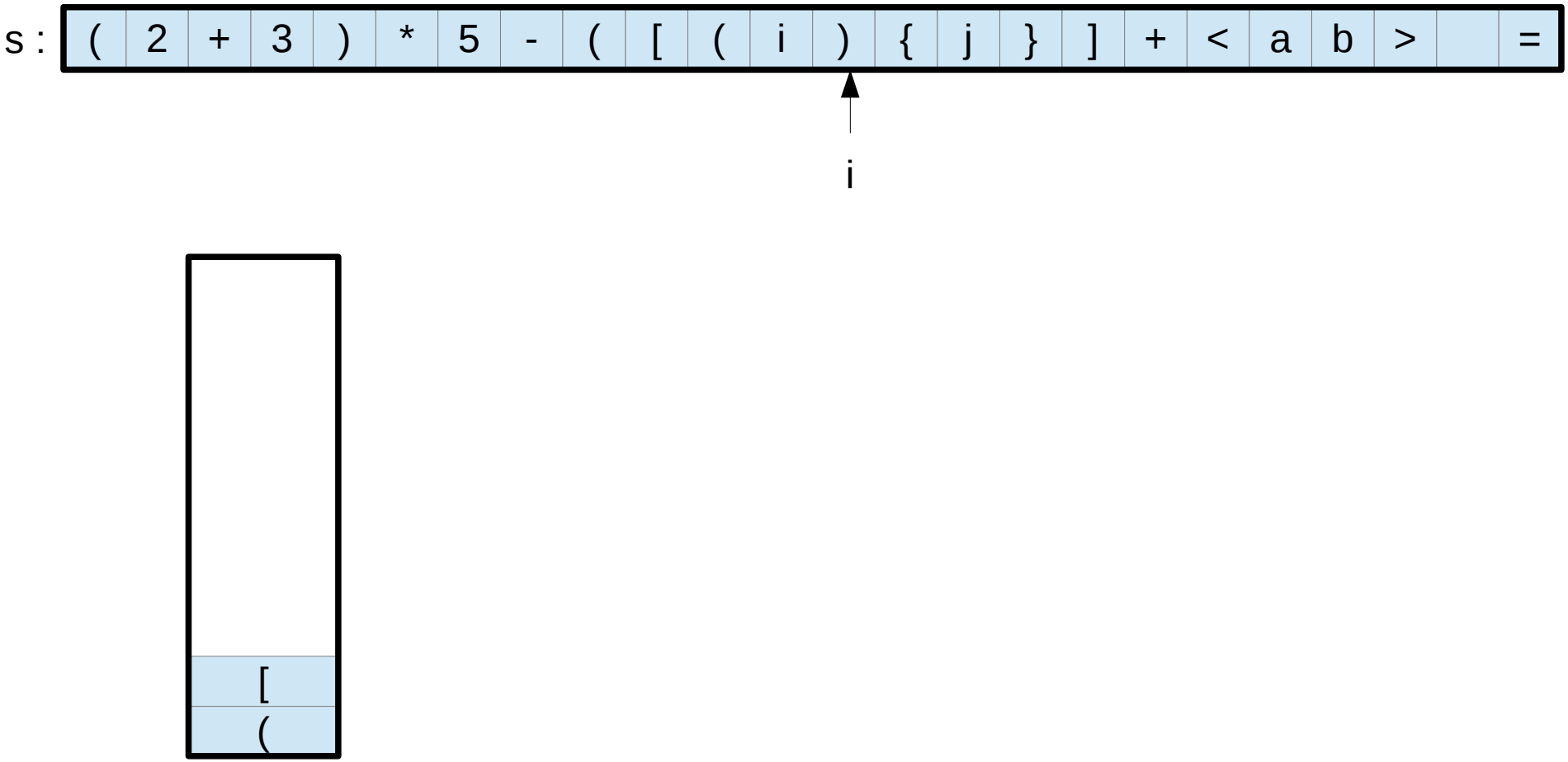

```
if (s[i] – открывающаяся скобка)
{
    push(s[i]);
}
```

Проверка скобок

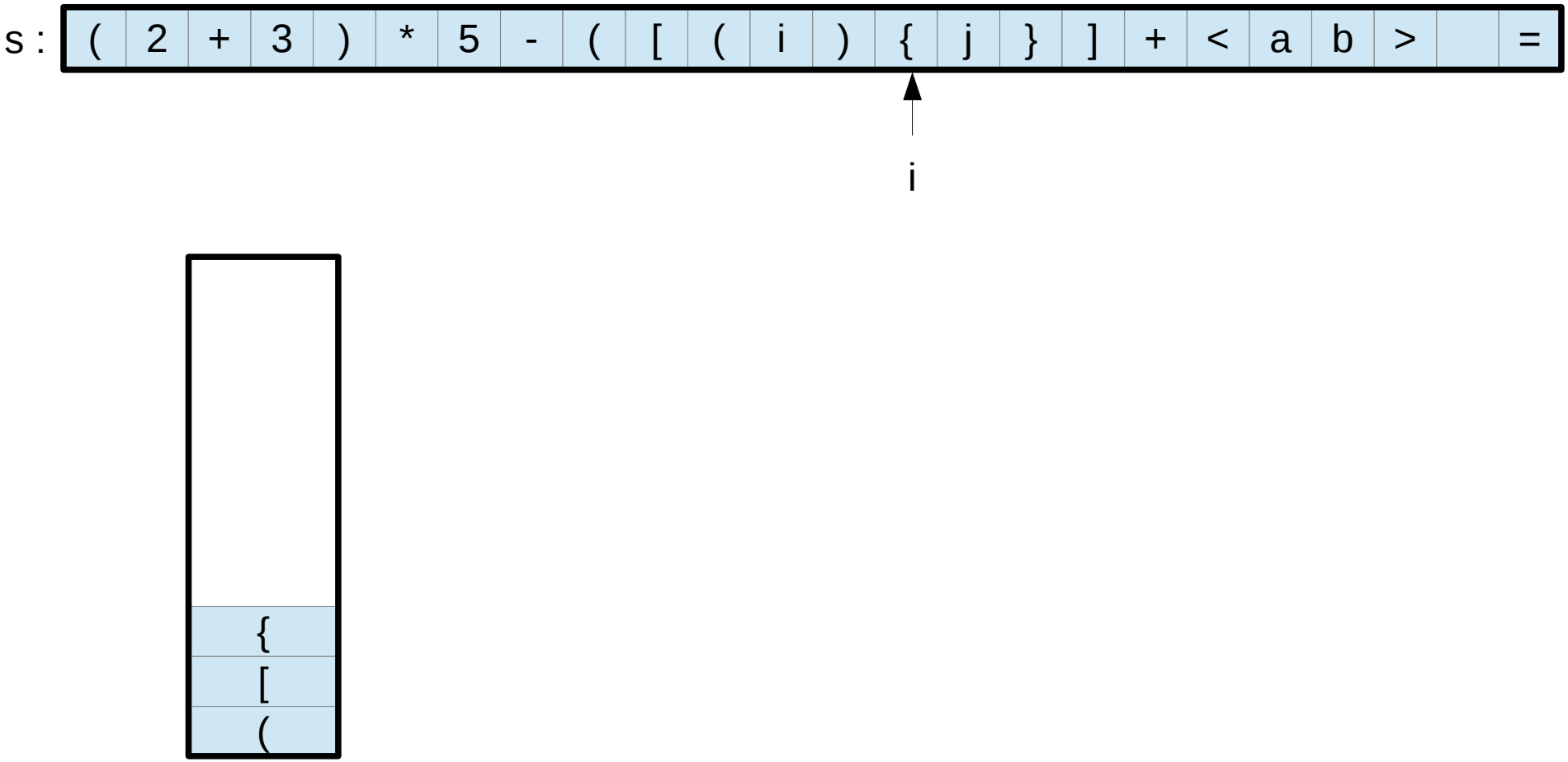



```
if (pair(s[i]) != s[i])
{
    return false;
}
```

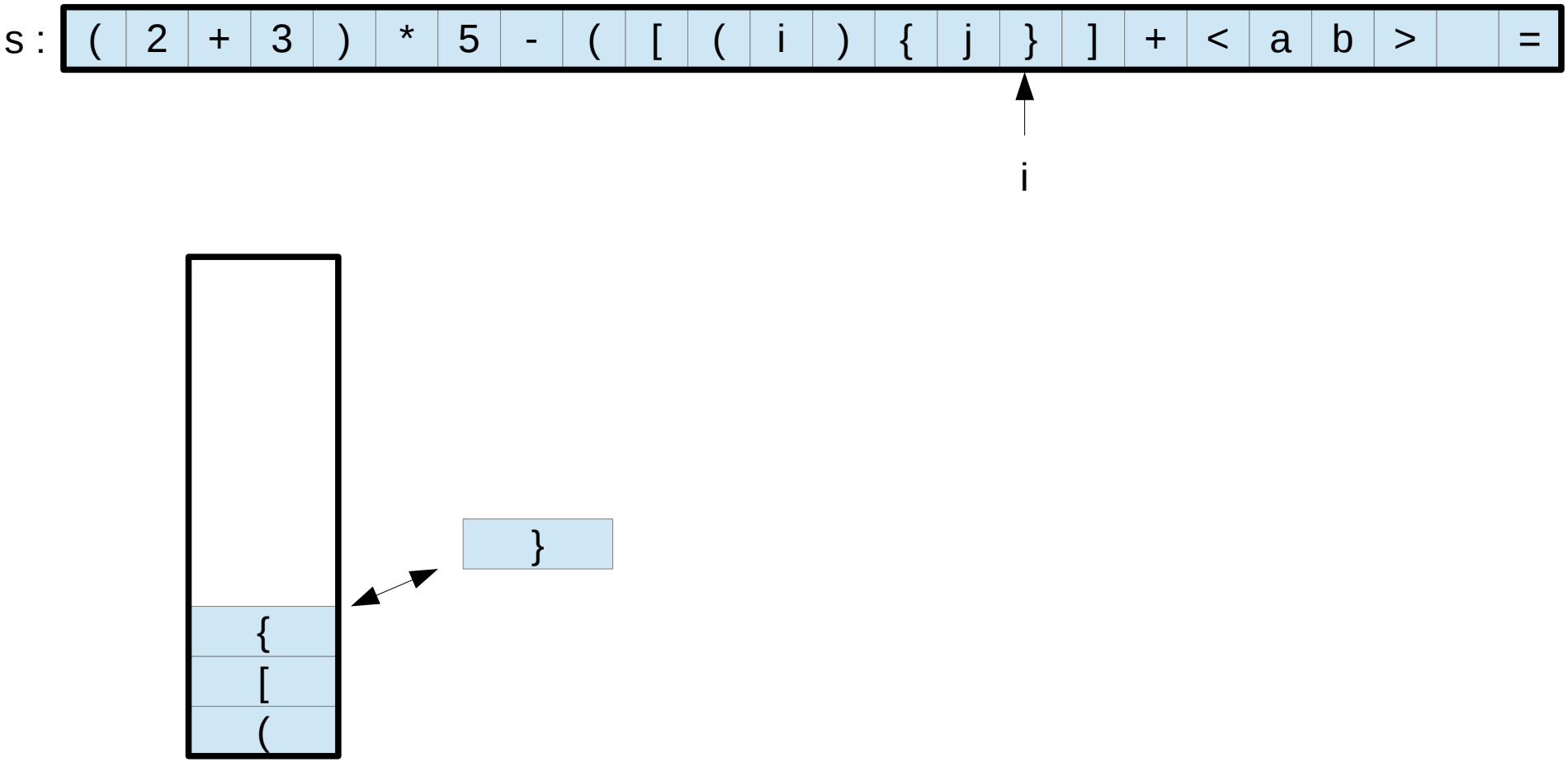
Проверка скобок



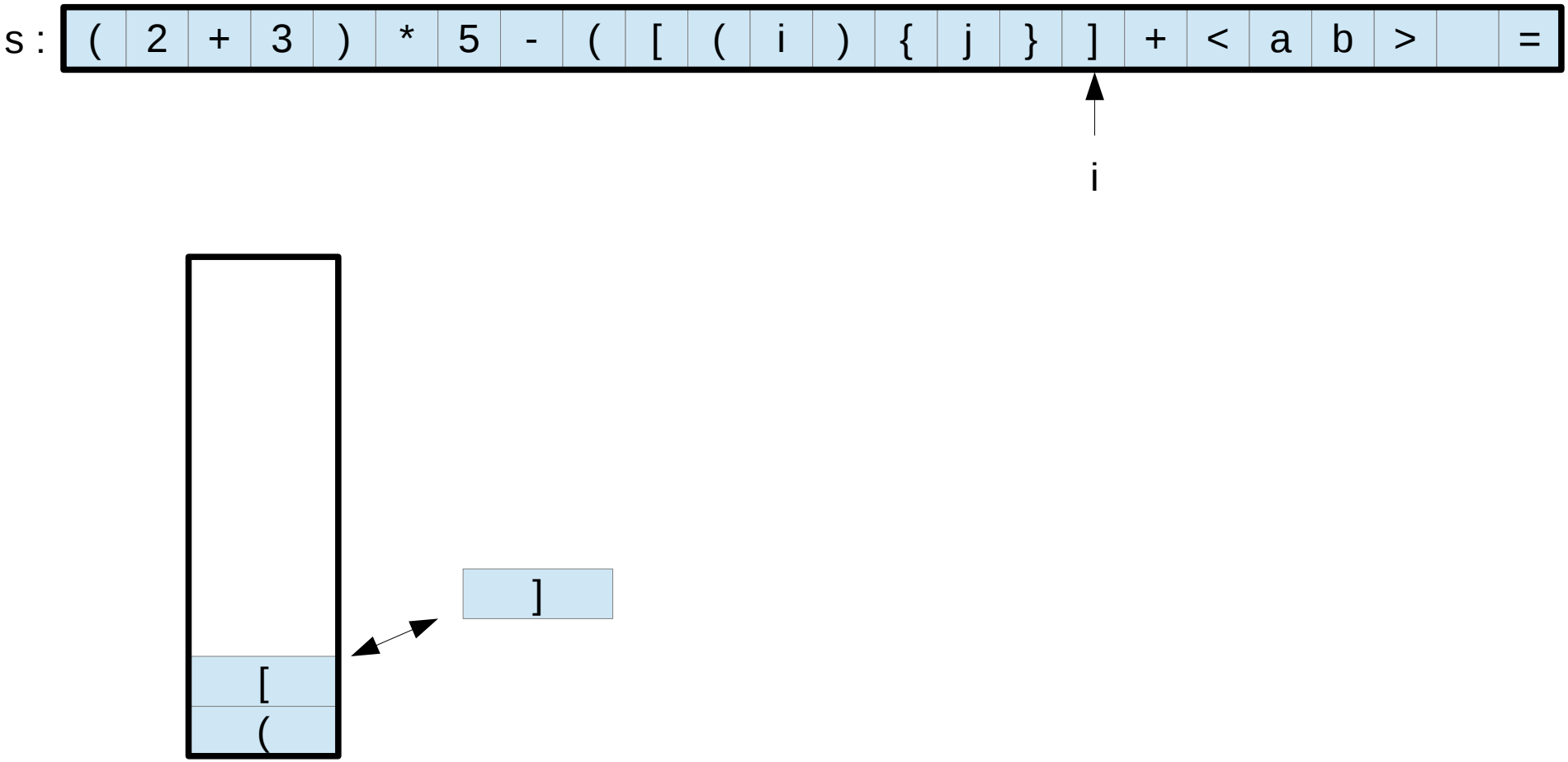
Проверка скобок



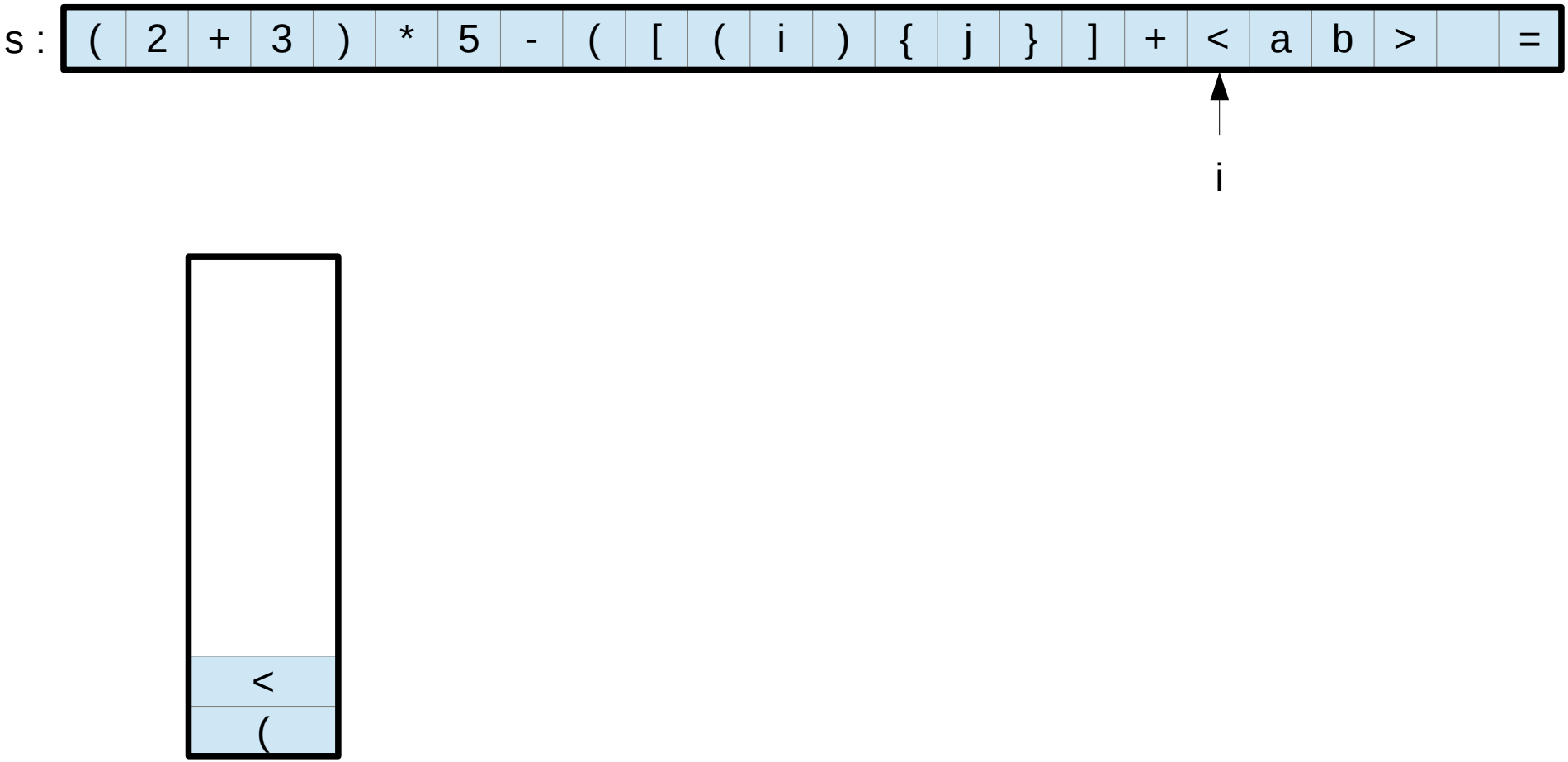
Проверка скобок



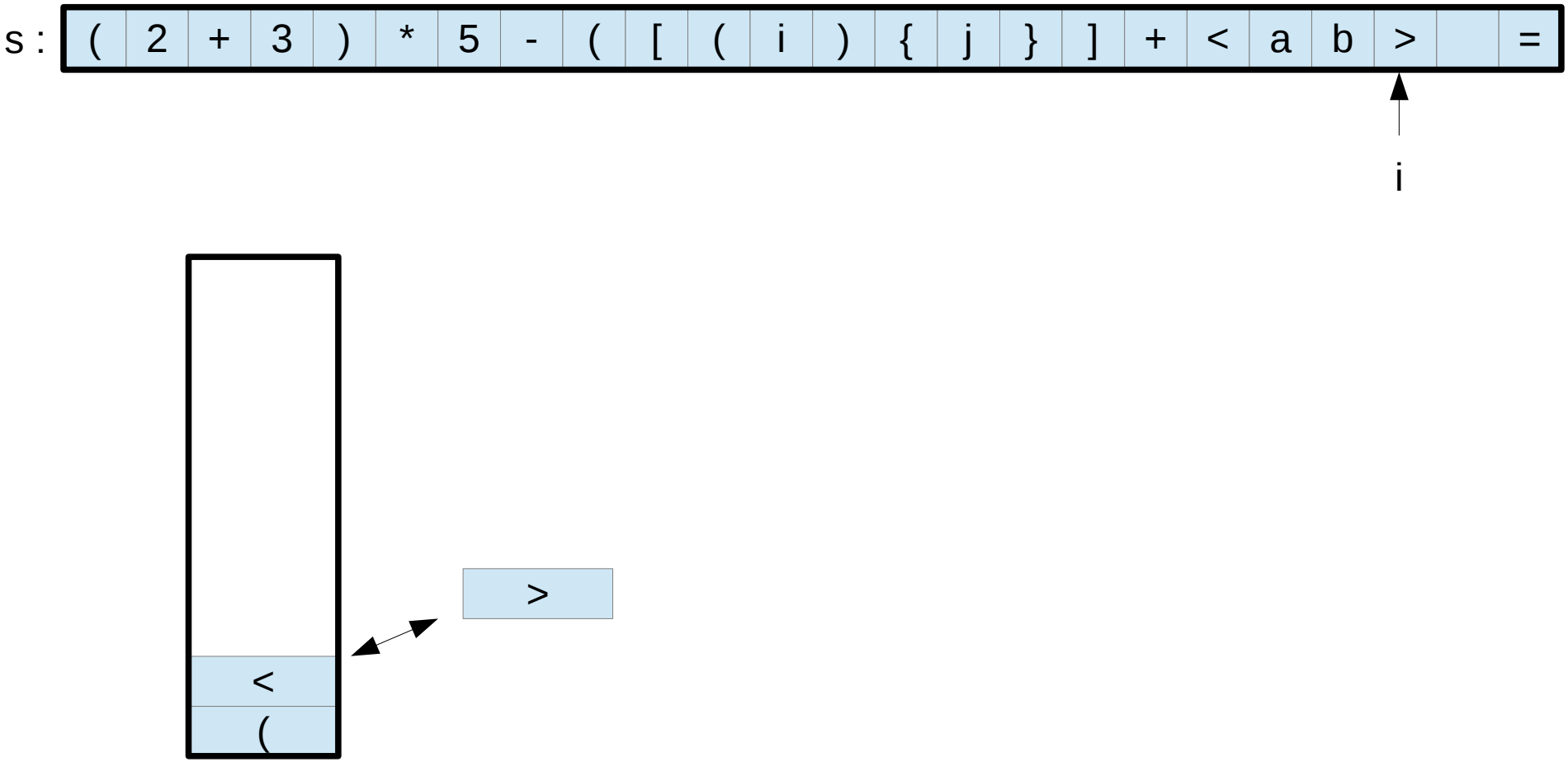
Проверка скобок



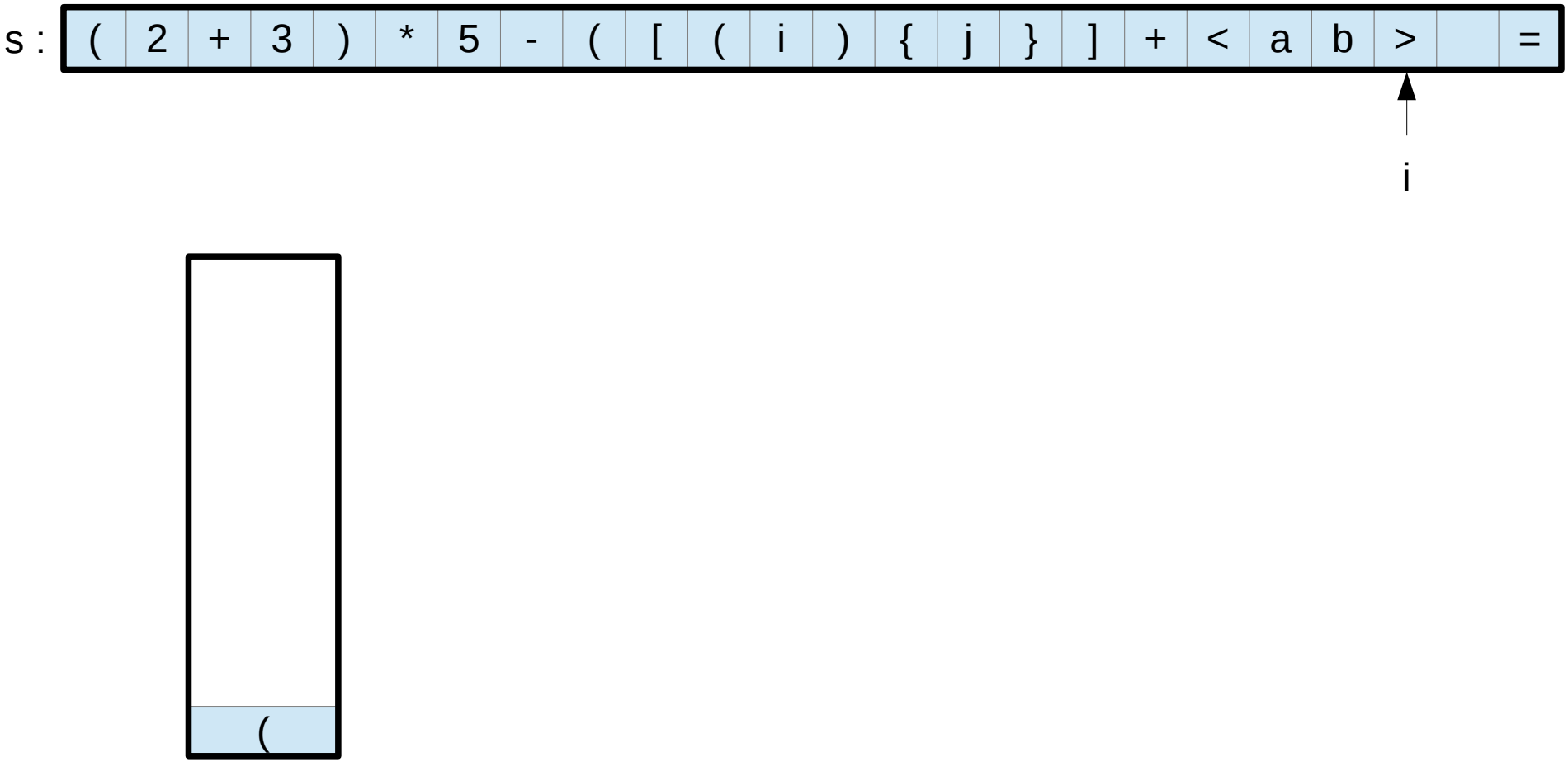
Проверка скобок



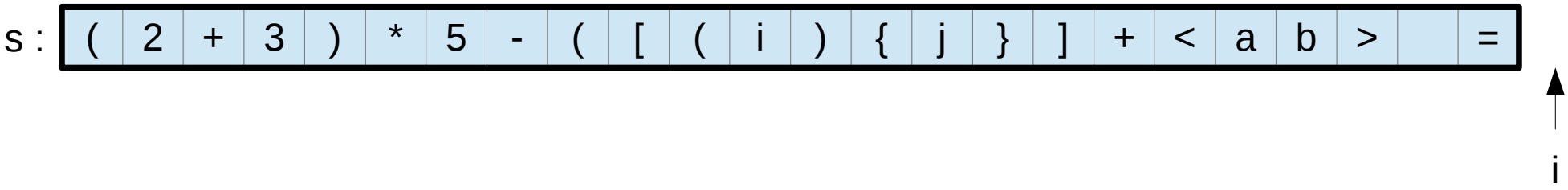
Проверка скобок

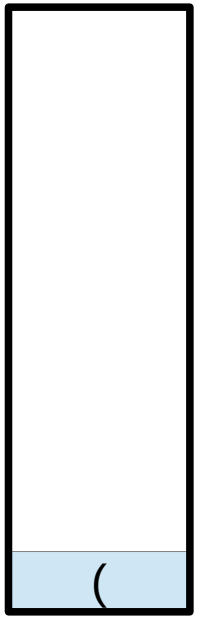


Проверка скобок



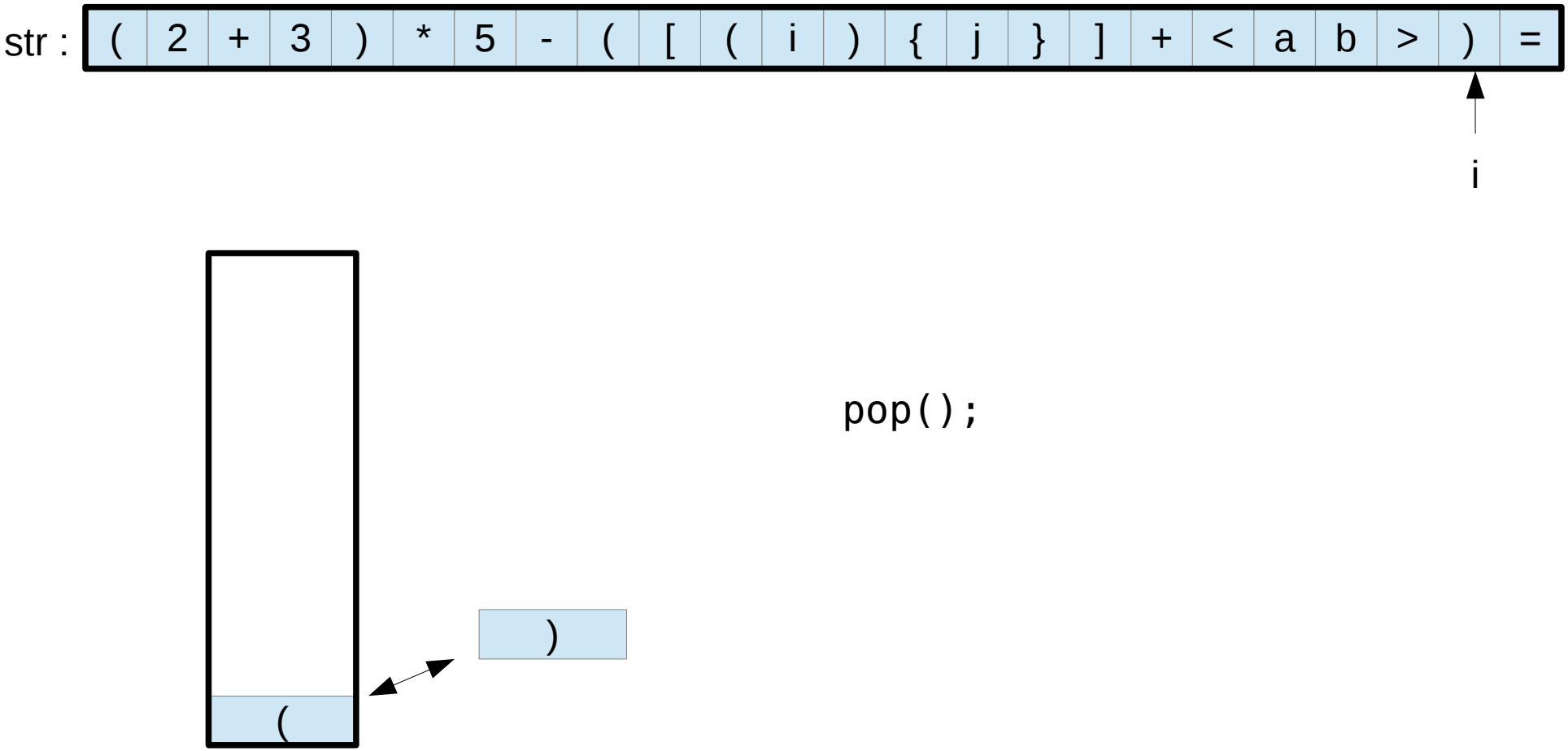
Проверка скобок






```
// после просмотра строки

if ( !empty() )
{
    return false;
}
```

Проверка скобок



Проверка скобок

s:

(	2	+	3	)	*	5	-	(	[	(	i	)	{	j	}	]	+	<	a	b	>	)	=

i

```
// после просмотра строки

if ( !empty() )
{
    return false;
}

// если стек пустой

return true;
```