

Прототип интерпретатора вычислительных моделей с поддержкой вложенных вычислительных моделей, циклов и массовых операций

Зимняя школа 2021

Руководитель: Перепёлкин Владислав Александрович

Выполнил: Парфенов Денис Романович, 4 курс ФИТ НГУ

29 января 2021г.

Предпосылки

Ранее был реализован прототип базы активных знаний, который позволял работать с вычислительными моделями, однако в нём отсутствовала возможность описывать циклы и операции над массивами.

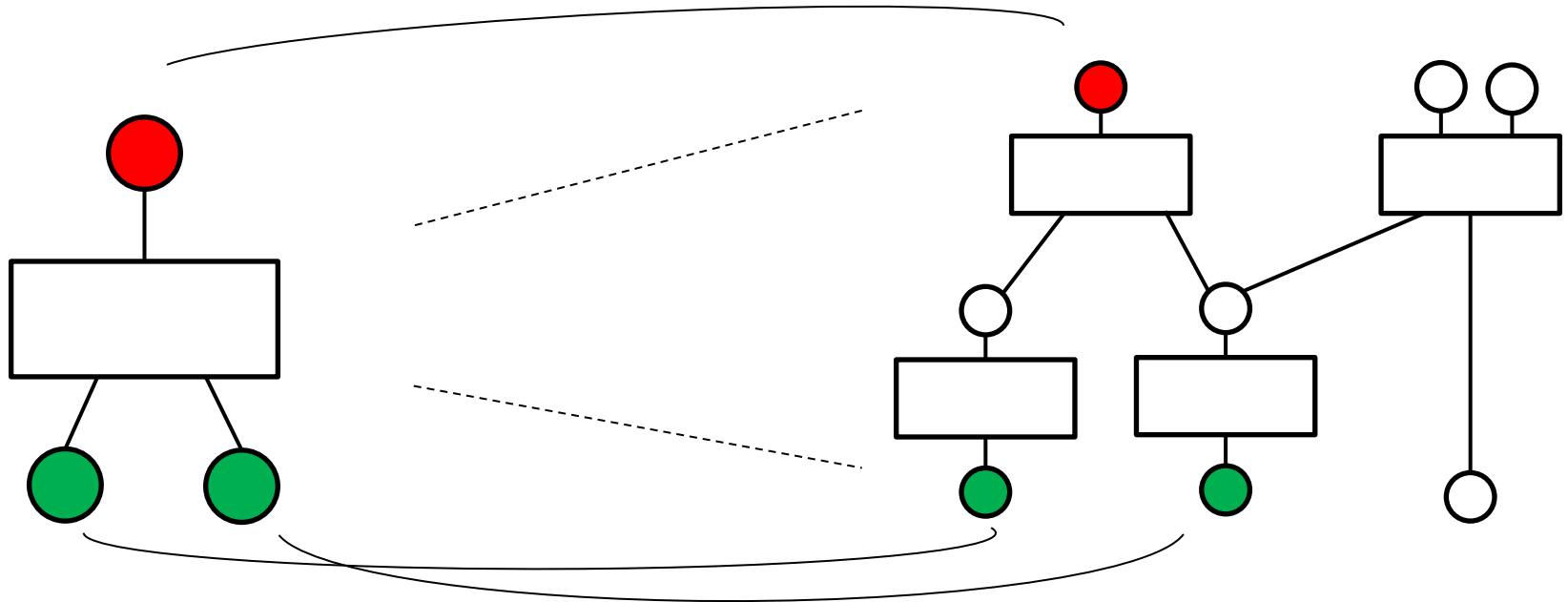
Также, был предложен способ работы с вложенными вычислительными моделями, частными случаями массовых операций.

Постановка задачи

Задача: разработать массовые операторы для языка описания вычислительных моделей и реализовать интерпретатор вычислительных моделей с их поддержкой.

Идея решения

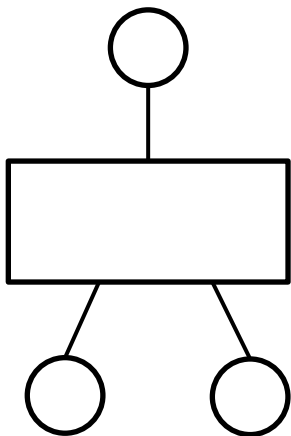
1. Вложенные вычислительные модели



Идея решения

2. Массовые операции

Были предложены 2 типа массовых операций – map и fold.

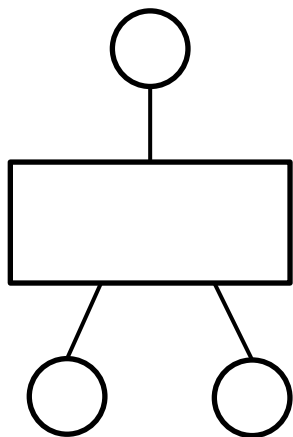


Условие	Операция
<i>Условие1</i>	<i>Операция1</i>
...	...
<i>УсловиеN</i>	<i>ОперацияN</i>

Идея решения

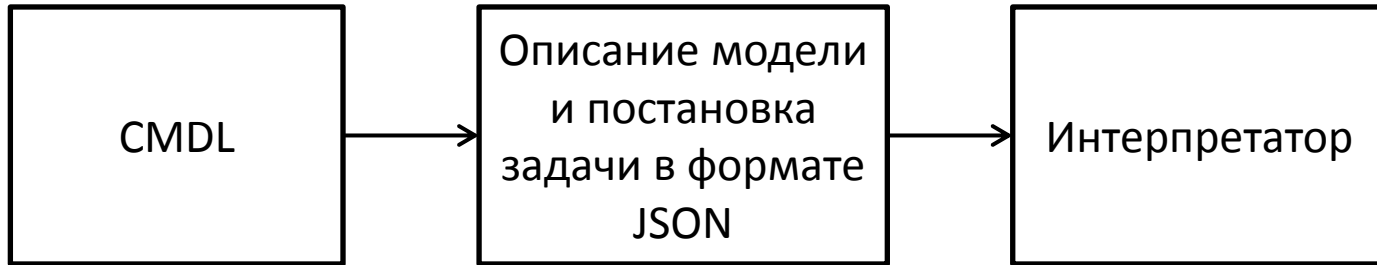
3. Циклы

Рассматриваются 2 типа циклов – while и for.

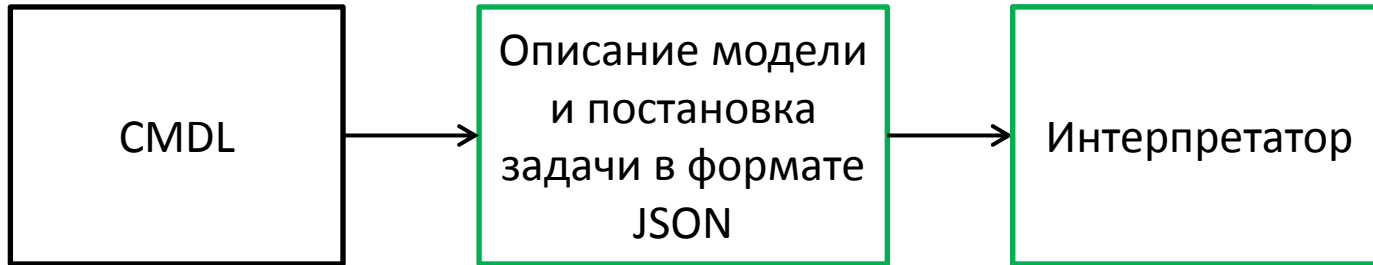


Условие	Операция
<i>Условие1</i>	<i>Операция1</i>
<i>...</i>	<i>...</i>
<i>УсловиеN</i>	<i>ОперацияN</i>

Реализация

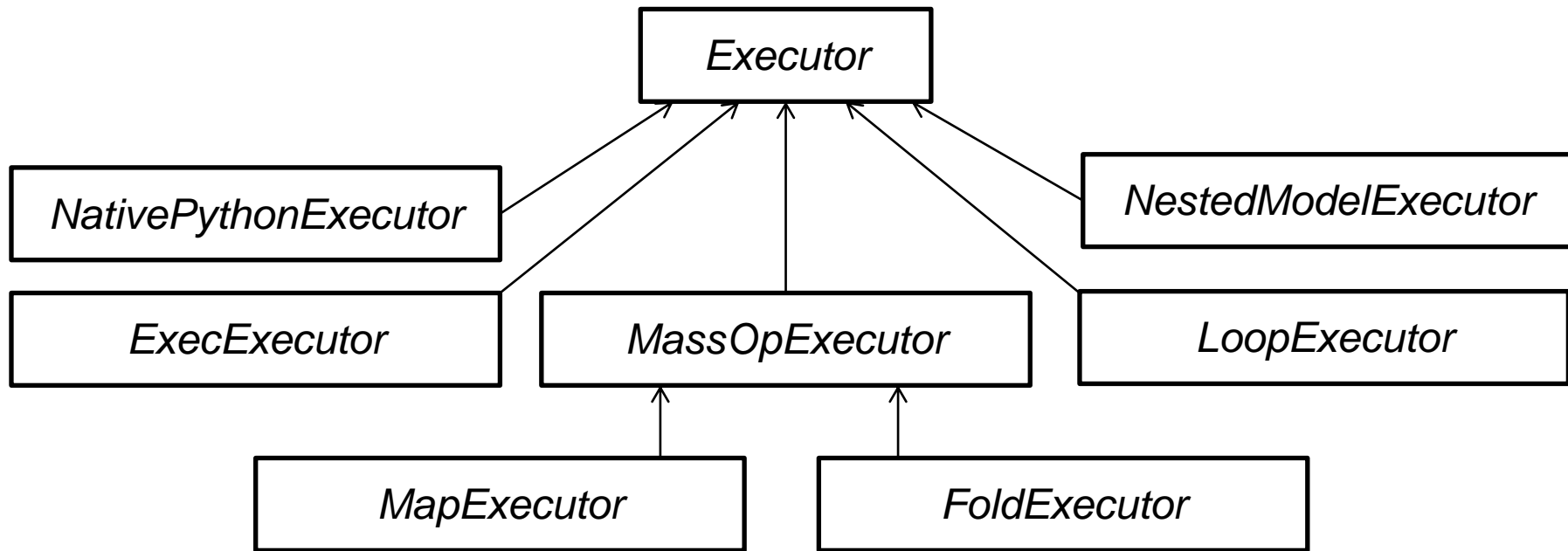


Реализация



Реализация

Executor (реализация операции) – иерархия классов



Реализация

Формат входного JSON-файла с моделью:

```
{
  "model": "name",
  "variables": [ { "name": "x" }, ... ],
  "executors": [ { "name": "ex1", "type": "exec", "path": ... } ... ],
  "operations": [
    {
      "name": "simple_sum",
      "type": "scalar",
      "in": ["x", "y"],
      "out": ["z"],
      "executor": {...}
    }, ...
  ]
}
```

Реализация

Формат входного JSON-файла с задачей:

```
{
  "in": [
    {
      "name": "x",
      "value": 5
    },
    {
      "name": "y",
      "value": 2
    }
  ],
  "out": ["z"]
}
```

Тестирование

```
{
  "model": "test_model",
  "variables": [
    {
      "name": "x"
    },
    {
      "name": "y"
    },
    {
      "name": "z"
    }
  ],
  "operations": [
    {
      "name": "sum",
      "type": "scalar",
      "in": ["x", "y"],
      "out": ["z"],
      "executor": {
        "model": "native",
        "name": "sum"
      }
    }
  ]
}
```

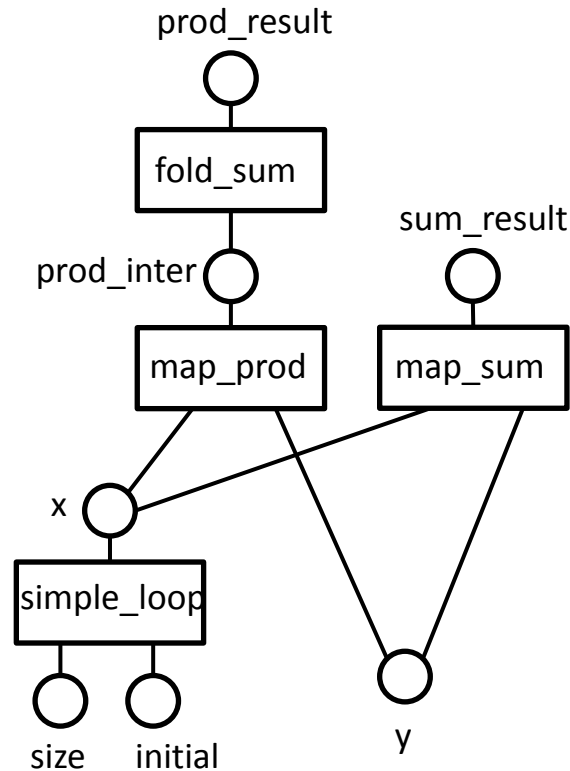
```
{
  "in": [
    {
      "name": "x",
      "value": 5
    },
    {
      "name": "y",
      "value": 2
    }
  ],
  "out": ["z"]
}
```

```
E:\NSU\CSC\Discr\lab2\venv\Scripts\python.exe E:/NSU/WS2021/ExecSys/main.py
{'x': 5, 'y': 2, 'z': 7}
```

```
Process finished with exit code 0
```

Тестирование

```
{  
  "in": [  
    {  
      "name": "size",  
      "value": 4  
    },  
    {  
      "name": "initial",  
      "value": [132]  
    },  
    {  
      "name": "y",  
      "value": [2, 1, 1, 0, 15]  
    }  
  ],  
  "out": ["prod_result"]  
}
```



```
E:\NSU\CSC\Discr\lab2\venv\Scripts\python.exe E:/NSU/WS2021/ExecSys/main.py  
{'size': 4, 'initial': [132], 'y': [2, 1, 1, 0, 15], 'x': [132, 0, 1, 2, 3],  
 'prod_inter': [264, 0, 1, 0, 45], 'prod_result': 310}
```

```
Process finished with exit code 0
```

Тестирование

```
"model": "test_model_vector",
"variables": [
  {
    "name": "x"
  },
  {
    "name": "y"
  },
  {
    "name": "prod_inter"
  },
  {
    "name": "prod_result"
  },
  {
    "name": "sum_result"
  },
  {
    "name": "initial"
  },
  {
    "name": "size"
  }
],
```

```
{
  "name": "map_prod",
  "type": "vector",
  "strat": "map",
  "in_vector": ["x", "y"],
  "in_scalar": [],
  "out": ["prod_inter"],
  "executors": [
    {
      "condition": {
        "model": "native",
        "name": "true"
      },
      "executor": {
        "model": "native",
        "name": "product"
      }
    }
  ]
},
```

```
{
  "name": "map_sum",
  "type": "vector",
  "strat": "map",
  "in_vector": ["x", "y"],
  "in_scalar": [],
  "out": ["sum_result"],
  "executors": [
    {
      "condition": {
        "model": "native",
        "name": "true"
      },
      "executor": {
        "model": "native",
        "name": "sum"
      }
    }
  ]
},
```

```
{
  "name": "fold_sum",
  "type": "vector",
  "strat": "fold",
  "in_vector": ["prod_inter"],
  "in_scalar": [],
  "out": ["prod_result"],
  "executors": [
    {
      "condition": {
        "model": "native",
        "name": "true"
      },
      "executor": {
        "type": "nested",
        "model": "test_model",
        "in": ["x", "y"],
        "out": ["z"]
      }
    }
  ]
},
```

```
{
  "name": "simple_loop",
  "type": "loop",
  "strat": "for",
  "in": ["size", "initial"],
  "out": ["x"],
  "limit": "size",
  "executors": [
    {
      "condition": {
        "model": "native",
        "name": "true"
      },
      "executor": {
        "type": "nested",
        "model": "test_append",
        "in": ["e", "arr_in"],
        "out": ["arr_out"]
      }
    }
  ]
},
```

Заключение

Бал реализован прототип системы исполнения и подтверждена работоспособность разработанных операторов.

В дальнейшем целесообразно усовершенствовать алгоритм планирования таким образом, чтобы идущие подряд массовые операции исполнялись более эффективно, а также сделать возможным параллельное исполнение независимых ветвей получившейся программы.