

Computer Science Center
Yandex ML & Data Management Lab

Контрактное программирование
как способ задания и оптимизации вычислительных
графов для распределённых потоковых систем

Руководитель: Трофимов Артём
Студент: Стоян Андрей

СПб 2020

Содержание

1	Введение	3
1.1	Способы задания вычислительных графов	3
1.2	Недостатки	3
2	Проблема	3
3	Существующие решения	3
4	Идея	3
4.1	Контракты	3
4.2	Примеры	3
5	Вид реализации	4
6	Предварительные результаты	4
7	Дальнейшая работа	4
7.1	Запись контрактов	4
7.2	Алгоритмы перебора	4
7.3	Cost-модель	4
8	Вывод	4
9	Приложение А: развитие идеи	4
10	Приложение Б: используемый синтаксис языка Haskell	5
11	Список литературы	5

1 Введение

TODO роль распределённых потоковых вычислений в жизни
TODO ссылка на репу

1.1 Способы задания вычислительных графов

TODO композиция, потоковый SQL

1.2 Недостатки

TODO композиция, потоковый SQL

2 Проблема

TODO автоматическое построение и оптимизация

3 Существующие решения

TODO поресёрчить program synthesis
TODO computational graph synthesis
TODO model generation (synthesis) (из той же оперы, что и model checking)

4 Идея

TODO задание графа: CoGraph, Graph

4.1 Контракты

TODO Контракты

4.2 Примеры

TODO примеры в терминах контрактов, получающиеся и не получающиеся графы из такого описания

5 Вид реализации

TODO Apache Beam, Python API
TODO хаскельный прототип

6 Предварительные результаты

TODO Предварительные результаты

7 Дальнейшая работа

Развивать проект планируется в следующих направлениях:

- Поиск более удобных и выразительных нотаций для записи контрактов.
- Поиск более эффективных алгоритмов перебора конкретных графов.
- Подбор функций оценки эффективности.

7.1 Запись контрактов

TODO Запись контрактов

7.2 Алгоритмы перебора

TODO Алгоритмы перебора

7.3 Cost-модель

TODO Cost-модель
TODO ещё чего?

8 Вывод

TODO conclusion

9 Приложение А: развитие идеи

TODO удалить нумерацию
TODO appendix A

10 Приложение Б: используемый синтаксис языка Haskell

TODO remove numeration

TODO appendix B

11 Список литературы

[1] Leslie Lamport