

Инструкция по использованию системы Spoon Engine

1 О системе Spoon Engine

Spoon Engine представляет собой систему интерактивного взаимодействия с базой данной оптимальных контактных схем и And-Inverted Graph'ов для функций от 5 переменных. Возможности системы:

- Визуализация контактных схем и And-Inverter Graph'ов с помощью библиотеки Dracula Graph Library;
- Построение гистограмм сложности для данных классов дискретных управляющих систем;
- Отправка схем и графов в базу данных.

2 Регистрация в системе

Для отправки схем и улучшения существующих в базе требуется регистрация. Для регистрации со страницы авторизации перейдите по ссылке «Register» или по адресу <http://mks2.cs.msu.ru/register> и заполните все необходимые поля. После регистрации необходимо подтверждение учётной записи, до этого доступ к странице отправки будет отсутствовать. Для ускорения подтверждения рекомендуется сообщить о регистрации вконтакте <http://vk.com/zhvv1> либо по адресу электронной почты zhvv117@gmail.com.

3 Формат схем и отправка решений

Система позволяет отправлять файлы, содержащие описание контактных схем и And-Inverter Graph'ов в нижеописанном формате. Размер файла не должен превышать 300 МБ. После отправки происходит автоматическая проверка корректности и дальнейшая запись в базу данных оптимальных схем и графов. После прохождения проверки, в системе будет виден её результат: количество обновлённых схем в базе (колонка «Accepted»); количество схем, сложность которых оказалась не лучше, чем уже существующих в базе (колонка «Not better»); количество ошибок процедуры проверки корректности (колонка «Fail»). Ошибки при проверке корректности

могут быть повлечены несоответствием формата, а также неэквивалентностью указанной в описании функции алгебры логики и функцией, реализуемой схемой либо графом. Следует отметить, что в некоторых случаях несоответствия формату, при наличии хотя бы одной ошибки проверки корректности, вообще говоря, может быть просмотрен не весь отправленный файл. В связи с этим, для получения желаемых результатов требуется свести число этих ошибок к нулю.

3.1 Общий формат файла

Каждая схема или граф задаётся в файле пятью строками. Число описаний в файле ограничивается только размером файла (не более 300 МБ). Первая строка содержит десятичный код функции (mincode), получаемый из вектора-столбца значений функции простым переводом из двоичной системы в десятичную. Вторая строка содержит класс схем, это может быть либо строка «contact_circuit» для контактных схем, либо строка «aig» для And-Inverter Graph'ов. Далее, следует трёхстрочное описание схемы.

3.2 Формат описания контактных схем

В случае контактных схем в третьей строке описания задаётся число контактов N , то есть сложность данной схемы. Далее, в четвёртой строке, следуют записанные через пробел номера вершин S и F (нумерация осуществляется с нуля), являющихся входом и выходом схемы соответственно. В пятой строке содержатся N четвёрок чисел, разделителем которых также является пробельный символ. Каждая четвёрка $s f v e$ описывает один из N контактов схемы, где s и f — номера вершин, к которым присоединён контакт, v — номер переменной от 1 до 5, соответствующей пометке контакта, e определяет наличие или отсутствие отрицания у переменной, принимает значения либо 0, либо 1.

Пример описания контактной схемы.

```
6
111
7
0 1
0 2 1 1 2 3 4 1 3 4 5 0 4 1 2 1 2 5 4 0 5 4 5 1 3 5 3 1
```

3.3 Формат описания And-Inverter Graph'ов

В случае And-Inverter Graph'ов в третьей строке описания задаётся число вершин (или, что тоже самое, функциональных элементов «&») графа N , что является сложностью данного графа. В четвёртой строке задаётся выходная вершина графа, а также флаг наличия или отсутствия отрицания на выходе, флаг принимает значение 0 или 1. В пятой строке содержатся N четвёрок чисел, каждая из которых описывает одну из N вершин графа. Первые два числа четвёрки характеризуют

один из двух входов функционального элемента «&», вторые — другой вход. Первое число каждой пары показывает, выход какого из элементов подаётся на вход данному, второе определяет наличие отрицания на ребре между двумя этими элементами. Нумерация вершин осуществляется с единицы, причём первые 5 вершин соответствуют входным переменным функции x_1, \dots, x_5 .

Пример описания And-Inverter Graph'a.

```
6
111
6
11 0
4 0 5 1 4 1 5 0 6 1 7 1 8 0 3 0 2 1 9 1 1 1 10 0
```