

# Документация к построению ER и реляционных диаграмм

Григорьев Роман, Бермишев Владислав

15 января 2023 г.

# Содержание

<b>1</b>	<b>Описание грамматики</b>	<b>3</b>
1.1	Грамматика . . . . .	3
1.2	Дополнительные условия на грамматику . . . . .	3
<b>2</b>	<b>Построение диаграмм</b>	<b>3</b>
2.1	Возможности . . . . .	3
2.2	Установка и запуск . . . . .	4
2.3	Структура проекта . . . . .	4
2.4	Настраиваемые параметры . . . . .	4
<b>3</b>	<b>Примеры работы</b>	<b>5</b>

# 1 Описание грамматики

## 1.1 Грамматика

Программа представляет ER-диаграмму и реляционную диаграмму по грамматике, приведённой ниже:

`Grammar ::= Rule+`

`Rule ::= Entity_definition | Relation_definition | type_sybtype_relation_definition`

`Entity_definition ::= [Name] -> eps;  
([Name].attr = [List], [Name].ident = [List]) | ([Name].ident = [List],  
[Name].attr = [List])`

`[Name] ::= ([A-z][0-9])*`

`[List] ::= (([Name],)*[Name])`

`Relation_definition ::= [Name]'1 -> [Name]'2;  
[Name]'1.cardinality = [Cardinality],  
[Name]'2.cardinality = [Cardinality],  
([Name]'1.ident ::= [Name]'2.ident | [Name]'2.ident ::= [Name]'1.ident | eps)`

`[Cardinality] = (0|1|N) - (0|1|N)`

`[type_subtype_definition] ::= [Name]'1 -> ([Name]'i |)*[Name]'n;  
([Name]'i.attr = [List])^n,  
[Name]'1.subtypes = [inclusive | exclusive]`

## 1.2 Дополнительные условия на грамматику

- `Entity_definition` для конкретного `[Name]` не может встречаться больше 1 раза.
- Не более 1 определения `attr` и `ident` в `Entity_definition`.
- Нельзя определять `attr` и `ident` другого Entity отличного от определённого в `Entity_definition`.
- Обязательность задания 2 пар чисел кардинальности относящимся к этому Relation в `Relation_definition`.
- Не более 1 зависимости идентификаторов (`[Name]'1.ident ::= [Name]'2.ident`) в `Relation_definition`.
- Обязательность определения `[Name]'1` и `[Name]'2` до определения отношения. При указаний зависимости идентификаторов, обязательность непустого пересечения идентификаторов `[Name]'1` и `[Name]'2`.
- Не более 1 определения атрибутов в `type_subtype_template` для `[Name]'i`.
- Обязательность определения сущности `[Name]'1` до определения `type_sybtype_template`.
- Не более 1 определения типа связи шаблона (`[Name]'1.subtypes = ...`), значение по умолчанию будет `inclusive`.

# 2 Построение диаграмм

## 2.1 Возможности

Данная программа позволяет строить ER-диаграмму грамматики и преобразовывать её в реляционную. Диаграммы представляются в виде файлов с расширением `.svg`.

Также есть возможность автоматического извлечения таблицы кардинальностей из описания модели. Данная таблица представляется в виде файла с расширением `.csv`.

## 2.2 Установка и запуск

Для запуска программы вам понадобится язык программирования C++. Далее склонируйте репозиторий и установите библиотеку для автоматической визуализации диаграмм:

```
git clone https://github.com/VladBermishev/FormalLanguageTheory_Labs/lab5
cd lab5
sudo apt-get install libgraphviz-dev
```

Для работы программы необходимо создать конфигурационный файл и указать в нем параметры модели. Для запуска программы требуется собрать проект:

```
mkdir build ; cd build
cmake ../ -DBUILD=Release
cmake --build . --target lab5 -j 3
```

Пример запуска программы:

```
./lab5 < tests/1
or
./lab5 tests/1
```

Программа выведет ER-модель, реляционную и таблицу кардинальностей для нашей модели.

## 2.3 Структура проекта

```
FormalLanguageTheory_Labs/lab5/
- include/lab5/
    - attribute.h
    - entity.h
    - entity_relationship_diagram.h
    - relation.h
- tests/
    - 1
- CMakeLists.txt
- main.cpp
- readme.md
```

В репозитории в директории `tests` лежат конфигурационные файлы. В директории `include/lab5/`:

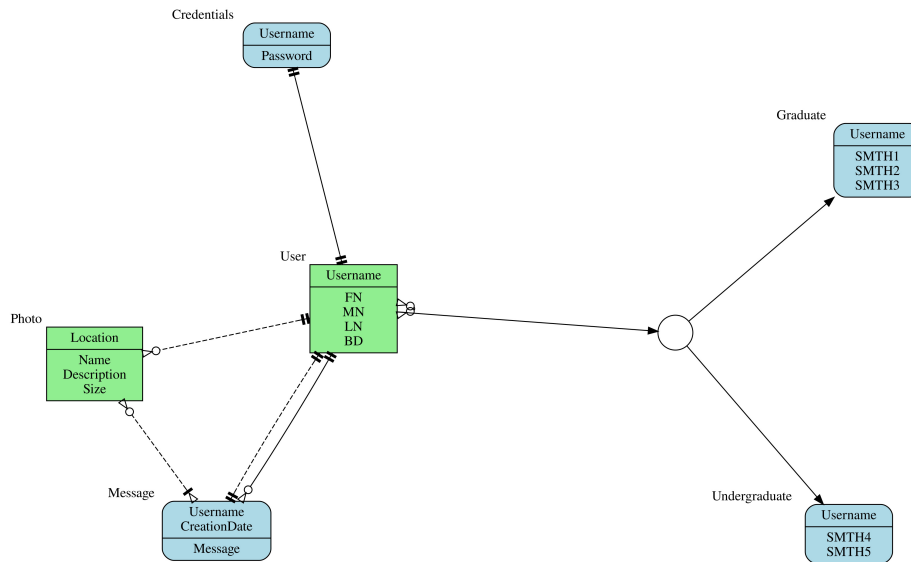
- в модуле `attribute.h` объявлен класс `Attribute`, описывающий атрибуты и индефекторы сущности.
- в модуле `entity.h` объявлен класс `Entity`, описывающий сущности.
- в модуле `entity_relationship_diagram.h` объявлен класс `EntityRelationshipDiagram`, описывающий класс `EntityRelationship` и `RelationDiagram`.
- в модуле `relation.h` объявлен класс `Relation`, описывающий отношения между сущностями.

В директории `tests/` представлены тесты к данной программе. Основной модуль - `main.cpp`, в нем находится код, использующийся для построения диаграм.

## 2.4 Настраиваемые параметры

Необязательным параметром является путь до входной грамматики.

Relationship		Cardinality		
Parent	Child	Type	MAX	MIN
User	User_User	Identifying	N-1	0-1
User_User	User	Identifying	1-N	1-0
User	Credentials	Identifying	1-1	1-1
User	Message	Identifying	N-1	0-1
User	Photo	Nonidentifying	N-1	0-1
Message	Message_Photo	Identifying	N-1	0-1
Message_Photo	Photo	Identifying	1-N	1-1
Message	User	Nonidentifying	1-1	1-1



### 3 Примеры работы

Запустим код, используя конфигурационный файл `tests\1`. На вывод мы получаем 3 файла:

- Таблицу кардинальностей нашей модели.
- ER-модель.
- Реляционную модель.

