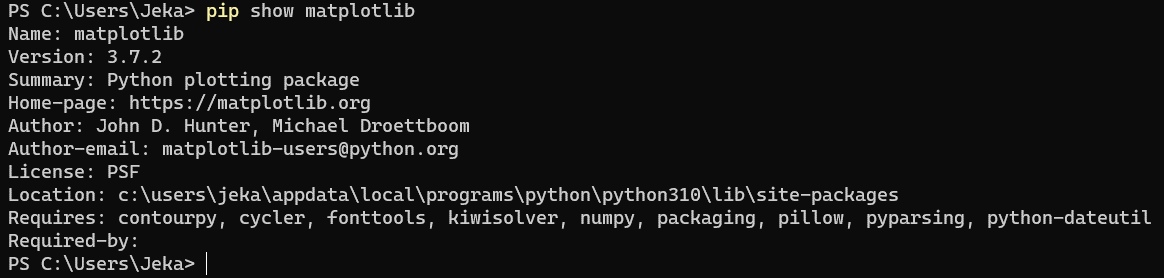
**Практическая работа 2**

**Задание 1:**

Вывести служебную информацию о пакете matplotlib (Python). Разобрать основные элементы содержимого файла со служебной информацией из пакета. Как получить пакет без менеджера пакетов, прямо из репозитория?



1. **Name**: Название пакета (matplotlib в данном случае).
2. **Version**: Версия пакета (в данном примере - 3.7.2).
3. **Summary**: Краткое описание пакета (здесь указано, что это библиотека для построения графиков в Python).
4. **Home-page**: Ссылка на домашнюю страницу пакета (в данном примере, это <https://matplotlib.org>).
5. **Author**: Авторы пакета.
6. **Author-email**: Адреса электронной почты авторов (если указаны).
7. **License**: Лицензия, под которой распространяется пакет.
8. **Location**: Местоположение файлов пакета на вашей системе.
9. **Requires**: Другие пакеты, от которых зависит matplotlib.
10. **Required-by**: Пакеты, которые требуют matplotlib в качестве зависимости.

Эти данные предоставляют важную информацию о пакете, включая его версию, авторство, зависимости и т.д.

Для того чтобы получить пакет прямо из репозитория без использования менеджера пакетов, мы можем воспользоваться Git для клонирования репозитория с исходным кодом.

Вот как это можно сделать:

1. **Установите Git (если у нас его еще нет)**:
   * Если у нас еще не установлен Git, нужно скачать и установить его с официального сайта: <https://git-scm.com/downloads>.
2. **Клонируйте репозиторий**:

Предположим, мы хотим получить пакет matplotlib.

* + Переходим в командную строку (терминал) и выполняем следующую команду:

git clone https://github.com/matplotlib/matplotlib.git

Это клонирует репозиторий с исходным кодом Matplotlib на наш компьютер.

**Переходим в директорию с кодом**:

После успешного клонирования репозитория, переходим в созданную директорию. Например, для Matplotlib:

cd matplotlib

**Устанавливаем пакет вручную**:

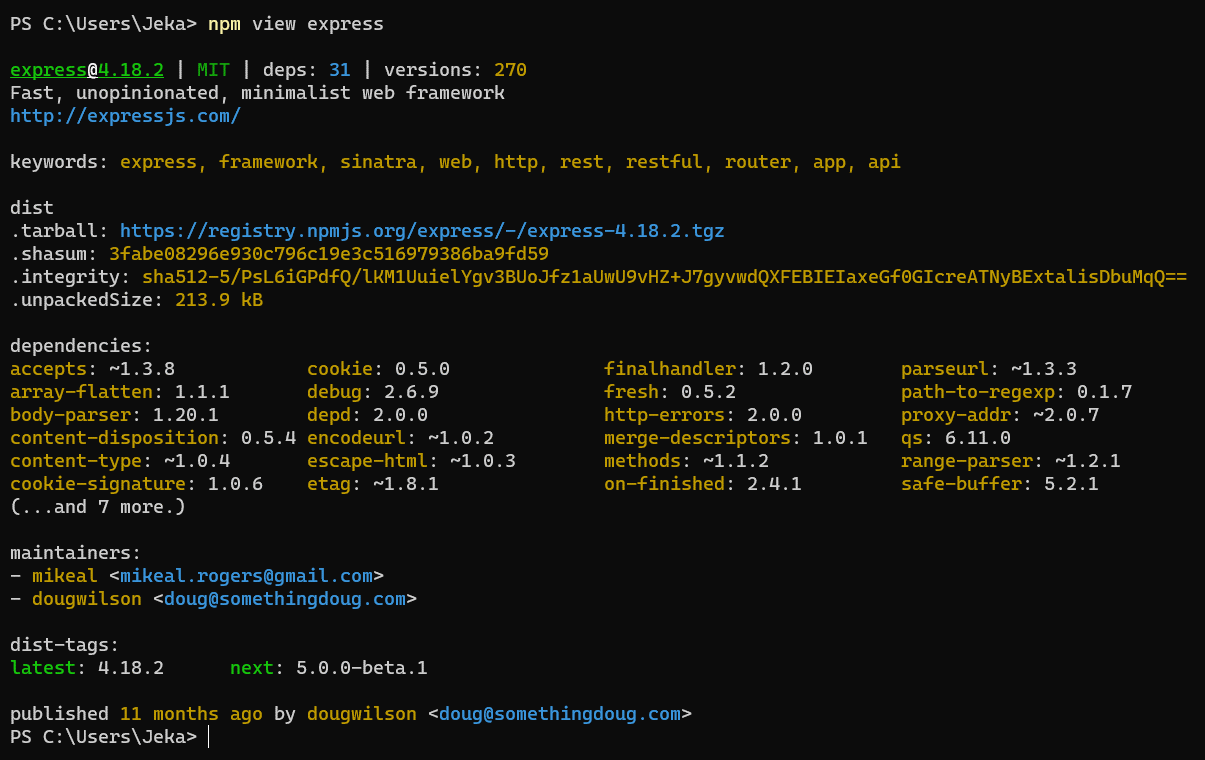
В этом случае, мы должны выполнить команду python setup.py install для установки пакета:

python setup.py install

Эти шаги позволяют нам получить пакет напрямую из его репозитория без использования менеджера пакетов. Но нужно иметь в виду, что при этом мы берем на себя ответственность за управление зависимостями и обновлениями самостоятельно. Использование менеджера пакетов, такого как pip, предпочтительнее, так как он автоматизирует этот процесс и управляет зависимостями.

**Задание 2:**

Вывести служебную информацию о пакете express (JavaScript). Разобрать основные элементы содержимого файла со служебной информацией из пакета. Как получить пакет без менеджера пакетов, прямо из репозитория?



1. **Название и версия пакета**:
   * Название: express
   * Версия: 4.18.2
   * Лицензия: MIT
2. **Описание**:
   * Описание пакета: "Fast, unopinionated, minimalist web framework"
   * Сайт: <http://expressjs.com/>
3. **Ключевые слова**:
   * express, framework, sinatra, web, http, rest, restful, router, app, api
4. **Информация о дистрибутиве**:
   * .tarball: Ссылка на архив с пакетом.
   * .shasum: Контрольная сумма для проверки целостности архива.
   * .integrity: Криптографически подписанная целостность архива.
   * .unpackedSize: Размер распакованного пакета.
5. **Зависимости**:
   * Перечень пакетов, от которых зависит express, включая их версии.
6. **Список поддерживающих разработчиков**:
   * Имена и электронные адреса разработчиков, ответственных за поддержку и развитие пакета.
7. **Дистрибутивные теги**:
   * latest: Последняя версия (4.18.2).
   * next: Следующая версия, на момент вывода - 5.0.0-beta.1.
8. **Дата публикации**:
   * Данные о том, когда пакет был опубликован в реестре пакетов npm.

Чтобы получить пакет без использования менеджера пакетов, нужно перейти к репозиторию express на GitHub или к источнику кода, указанному в .tarball, и скачать необходимую версию вручную. Затем мы можем разархивировать архив и использовать пакет по своему усмотрению. Однако такой метод установки не рекомендуется, так как менеджеры пакетов автоматически решают зависимости и обновления.

**Задание 3:**

Сформировать graphviz-код и получить изображения зависимостей matplotlib и express.

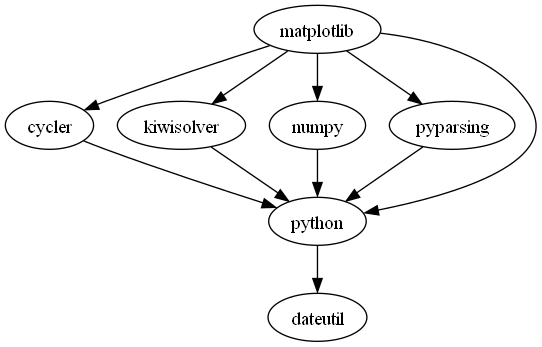


Рисунок 1 – зависимости библиотеки matplotlib в виде изображения

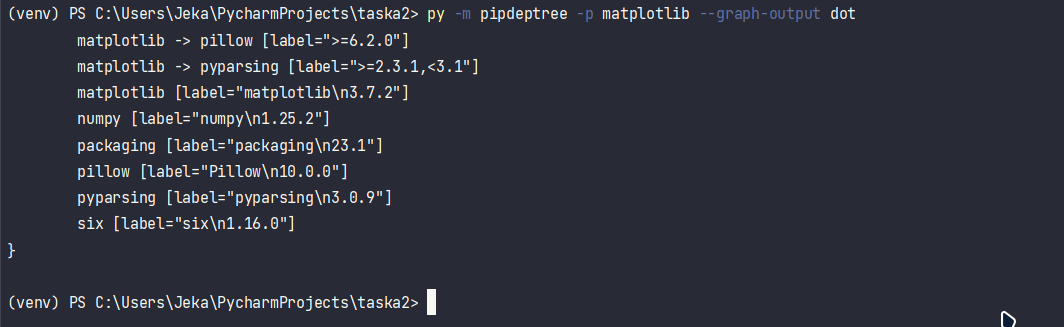


Рисунок 2 – зависимости библиотеки matplotlib в виде graphviz-кода

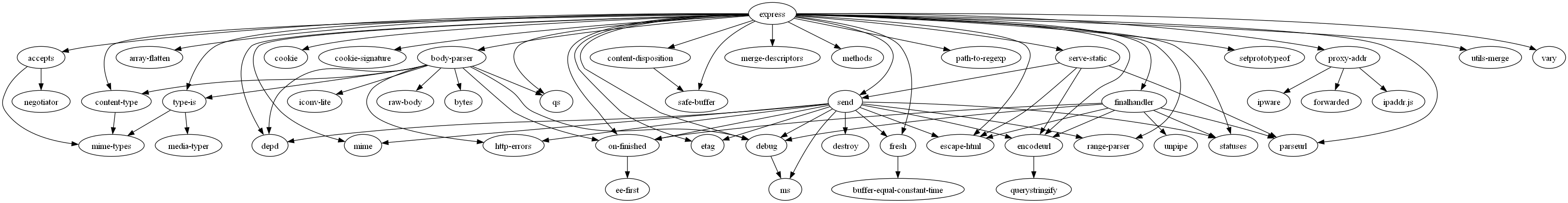


Рисунок 3 – зависимости библиотеки express в виде изображения

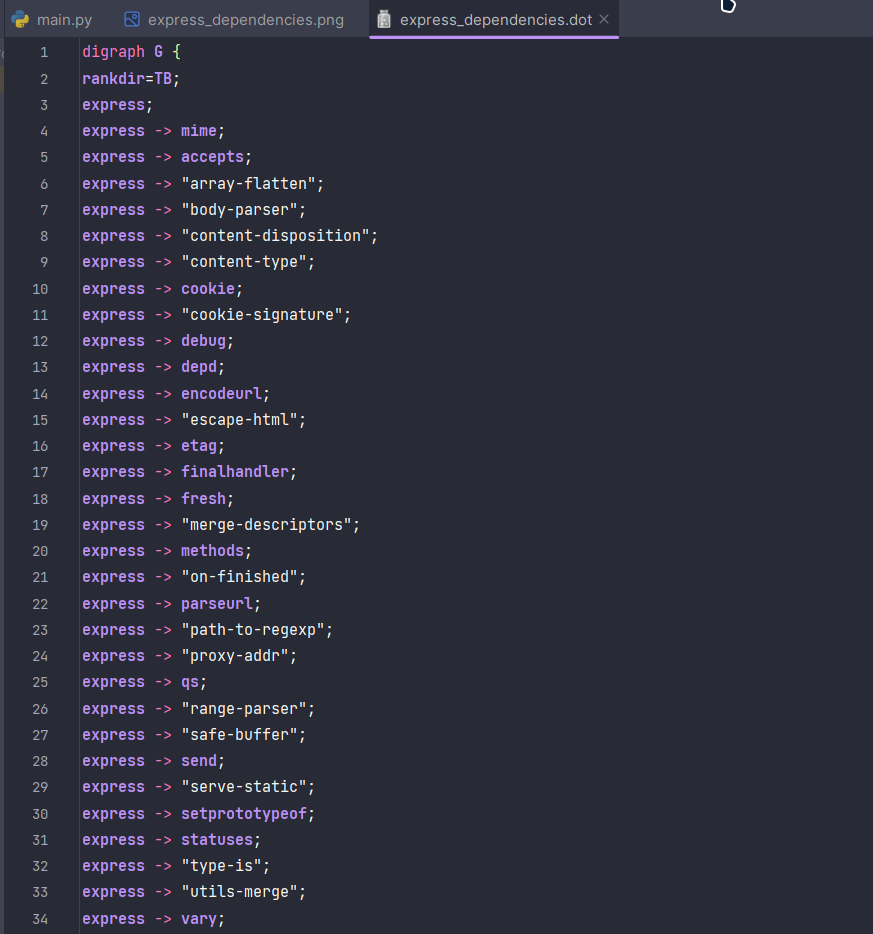
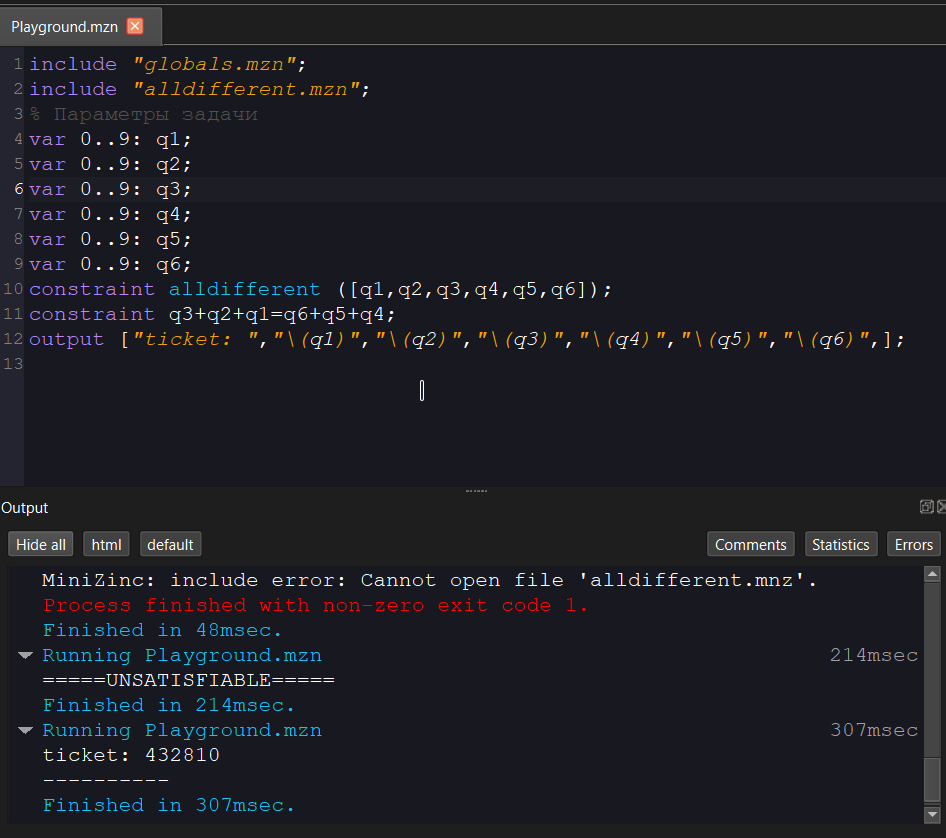


Рисунок 4 – зависимости библиотеки matplotlib в виде graphviz-кода

**Задание 4:**

Изучить основы программирования в ограничениях. Установить MiniZinc, разобраться с основами его синтаксиса и работы в IDE.

Решить на MiniZinc задачу о счастливых билетах. Добавить ограничение на то, что все цифры билета должны быть различными (подсказка: используйте all\_different). Найти минимальное решение для суммы 3 цифр.



**Задание 5:**

Решить на MiniZinc задачу о зависимостях пакетов для рисунка, приведенного ниже.

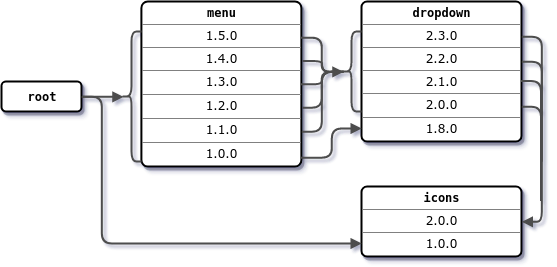
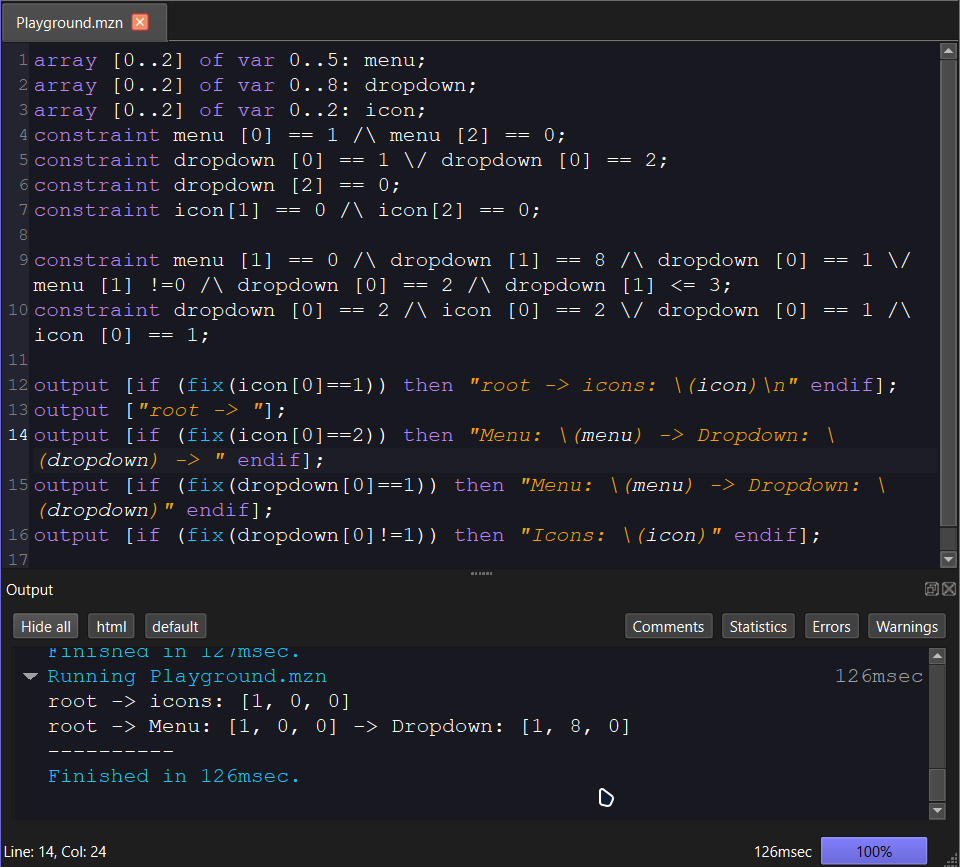


Рисунок 5 – зависимости пакетов



**Задание 6:**

Решить на MiniZinc задачу о зависимостях пакетов для следующих данных:

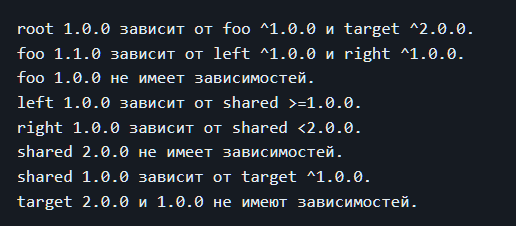
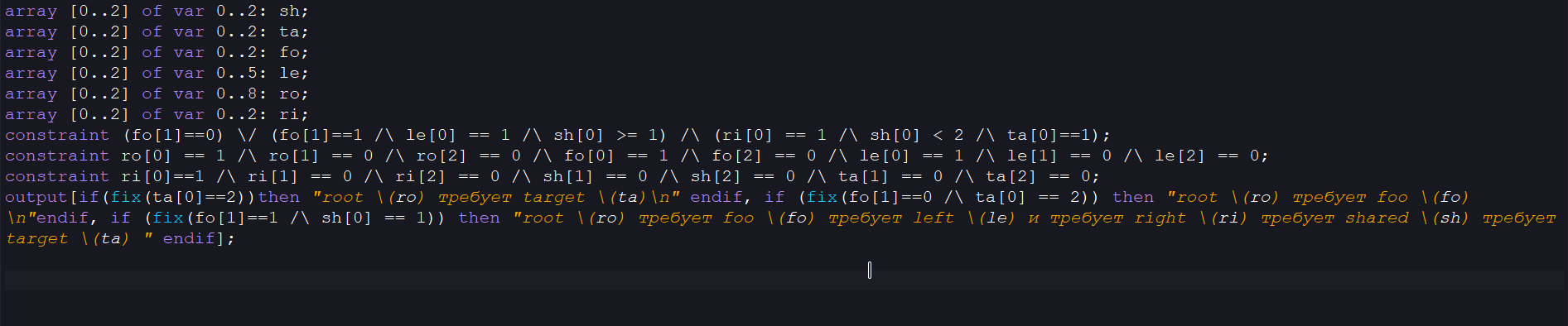


Рисунок 6 – данные для пакетов



Преобразованный для удобства чтения ответ:

root 1.0.0 требует target 2.0.0

(ИЛИ)

root 1.0.0 требует foo 1.0.0

(ИЛИ)

root 1.0.0 требует foo 1.1.0 , который требует left 1.0.0 (И) right 1.0.0, потребуется shared 1.0.0, который требует target 1.0.0.