



# **Структурированная нормализация текста с использованием недетерминированных FST**

Tinkoff.AI Speech Meetup #3

Владимир Марков

15.05.2024

- Нормализация текста
- Применение нормализации
- Способы нормализации

# **Нормализация - преобразование текста в его стандартную форму**



Нормализация текста



Применение нормализации



Способы нормализации

# Нормализация - преобразование текста в его стандартную форму



## Нормализация включает

- Исправление ошибок
- Расшифровка сокращений
- Преобразование чисел и символов
- Интерпретация дат, времени

- Нормализация текста
- Применение нормализации
- Способы нормализации

# Нормализация - преобразование текста в его стандартную форму



## Нормализация включает

- Исправление ошибок
- Расшифровка сокращений
- Преобразование чисел и символов
- Интерпретация дат, времени



## Цель нормализации

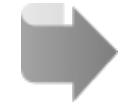
Убедиться, что система  
интерпретирует и произносит текст  
так, как это делал бы человек



**Нормализация текста**

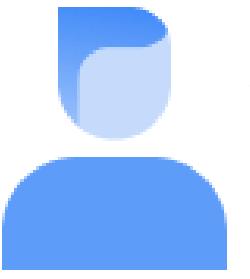


Применение нормализации



Способы нормализации

Олег, сколько времени?





**Нормализация текста**



Применение нормализации



Способы нормализации



Олег, сколько времени?



Сейчас двенадцать тридцать пять





Нормализация текста



Применение нормализации



Способы нормализации

Олег, сколько времени?



Сейчас двенадцать тридцать пять



**1. Вход:** Сколько времени?



Нормализация текста



Применение нормализации



Способы нормализации

Олег, сколько времени?



Сейчас двенадцать тридцать пять



- 1. Вход:** Сколько времени?
- 2. Подготовка ответа:** Сейчас 12:35



Нормализация текста



Применение нормализации

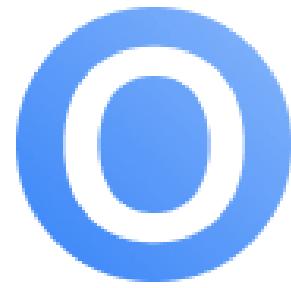


Способы нормализации

Олег, сколько времени?



Сейчас двенадцать тридцать пять



- 1. Вход:** Сколько времени?
- 2. Подготовка ответа:** Сейчас 12:35
- 3. Нормализация:** Сейчас двенадцать тридцать пять



Нормализация текста

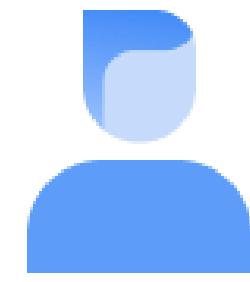


Применение нормализации



Способы нормализации

Олег, сколько времени?



Сейчас двенадцать тридцать пять



- 1. Вход:** Сколько времени?
- 2. Подготовка ответа:** Сейчас 12:35
- 3. Нормализация:** Сейчас двенадцать тридцать пять
- 4. Озвучивание:**



Нормализация текста



Применение нормализации



Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ



Нормализация текста



Применение нормализации

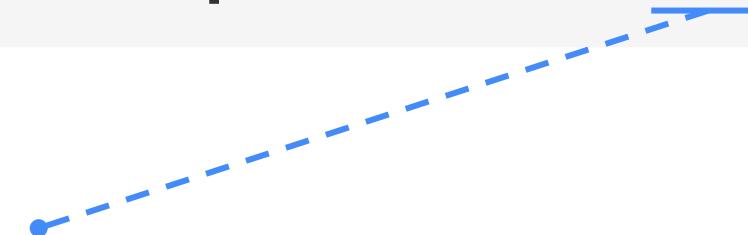


Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ

двенадцать  
двенадцати  
двенадцатью





Нормализация текста



Применение нормализации

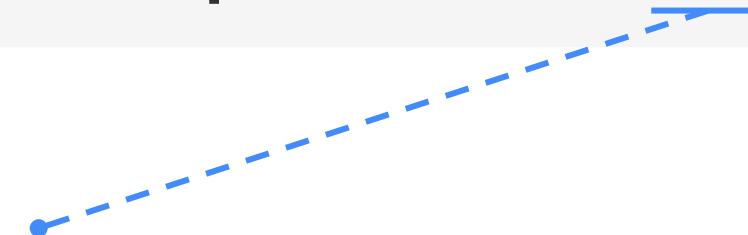


Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ

двенадцать  
двенадцати  
двенадцатью





Нормализация текста



Применение нормализации



Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ

двенадцать  
двенадцати  
двенадцатью

двенадцать  
двенадцати  
двенадцатью



Нормализация текста



Применение нормализации



Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ

двенадцать  
двенадцати  
двенадцатью

двенадцать  
двенадцати  
двенадцатью

дополнительные  
дополнительных  
дополнительными



Нормализация текста



Применение нормализации



Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ

двенадцать  
двенадцати  
двенадцатью

двенадцать  
двенадцати  
двенадцатью

дополнительные  
дополнительных  
дополнительными

гигабайт  
гигабайта  
гигабайтами



Нормализация текста



Применение нормализации



Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ

двенадцать  
двенадцати  
двенадцатью

двенадцать  
двенадцати  
двенадцатью

дополнительные  
дополнительных  
дополнительными

гигабайт  
гигабайта  
гигабайтами



Нормализация текста



Применение нормализации



Способы нормализации

## Текст от пользователя

Завтра после 12 у вас появятся 12 доп. ГБ

двенадцать  
двенадцати  
двенадцатью

двенадцать  
двенадцати  
двенадцатью

дополнительные  
дополнительных  
дополнительными

гигабайт  
гигабайта  
гигабайтами

## Текст после нормализации

Завтра после **двенадцати** у вас появятся  
**двенадцать дополнительных гигабайт**



Нормализация текста



**Применение  
нормализации**



Способы нормализации



## Преобразование текста в речь

- Голосовые ассистенты
- Чтение новостей, сайтов
- GPS навигаторы
- Колл-центры, службы поддержки
- Озвучивание YouTube роликов



Нормализация текста



**Применение  
нормализации**



Способы нормализации

 Нормализация текста **Применение  
нормализации** Способы нормализации

## Преобразование текста в речь

- Голосовые ассистенты
- Чтение новостей, сайтов
- GPS навигаторы
- Колл-центры, службы поддержки
- Озвучивание YouTube роликов



## Анализ и обработка текста

- Data mining
- Поисковые системы
- Перевод текстов
- Распознавание речи



Нормализация текста



Применение нормализации



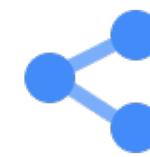
**Способы нормализации**



## Нейронные сети (seq2seq)

- Нормализация текста
- Применение нормализации
- Способы нормализации

- Сложные преобразования
- Нет возможности контролировать
- Большие вычислительные ресурсы

 Нормализация текста Применение нормализации Способы нормализации

## Нейронные сети (seq2seq)

- Сложные преобразования
- Нет возможности контролировать
- Большие вычислительные ресурсы



## Регулярные выражения

- Простота в использовании
- Нет понимания контекста
- Только простые преобразования



## Нейронные сети (seq2seq)

- Сложные преобразования
- Нет возможности контролировать
- Большие вычислительные ресурсы

→ Нормализация текста

→ Применение нормализации

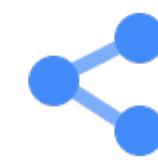
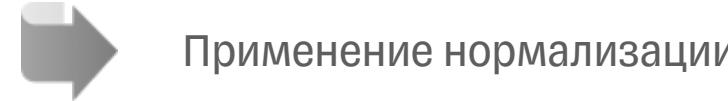
→ Способы нормализации



## Регулярные выражения

- Простота в использовании
- Нет понимания контекста
- Только простые преобразования

```
(?:(?:^|((0?[1-9])|(1[2][0-9]|3[01]))\.(0[1-9]|1[0-2])\.(0[1-9]|1[8][0-9]|9[0-9])|([1-9][0-9]{3}|4[0-8][0-9]{2}|49[0-8][0-9]|499[0-9]))|0?[1-9]|1[2][0-9]|3[01]) ([января|февраля|марта|апреля|мая|июня|июля|августа|сентября|октября|ноября|декабря])( (0[1-9]|1[8][0-9]|9[0-9])|([1-3][0-9]{3}|4[0-8][0-9]{2}|49[0-8][0-9]|499[0-9]))?)$|^(:|[1-9]\d{0,2})(( ?\d\d\d){0,3})?|0)$|^(:|[1-9]\d{0,2})(( ?\d\d\d){0,3})?|0)(,(0?[1-9]|1[8][0-9])?)? ?[$\pounds$\e]$|^(:|[1-9]\d{0,2})(( ?\d\d\d){0,3})?|0),\d{1,3}$|^(:|[01]\d|2[0-4]|\d):(0[1-9]|1[4][0-9]|5[0-9])$|^([1-9]\d{0,2})(( ?\d\d\d){0,3})?|0)(?:,\d{1,3})? ?(:|[ГБ|Гб|Gb|GB])|(:|[ГБ|Гб|Gb|GB])$)
```



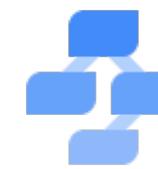
## Нейронные сети (seq2seq)

- Сложные преобразования
- Нет возможности контролировать
- Большие вычислительные ресурсы



## Регулярные выражения

- Простота в использовании
- Нет понимания контекста
- Только простые преобразования



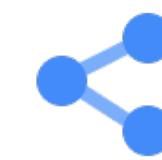
## Конечный автомат с выходом (FST)

- Простота в использовании
- Быстро работает
- Достаточно сложные преобразования

→ Нормализация текста

→ Применение нормализации

→ Способы нормализации



## Нейронные сети (seq2seq)

- Сложные преобразования
- Нет возможности контролировать
- Большие вычислительные ресурсы



## Регулярные выражения

- Простота в использовании
- Нет понимания контекста
- Только простые преобразования



## Конечный автомат с выходом (FST)

- Простота в использовании
- Быстро работает
- Достаточно сложные преобразования



## Комбинированный

- Высокое качество нормализации

Практический пример #1

# Нормализация времени

## Задача

Перевести время из числового формата в текст

Например:

01:00 ⇒ один час

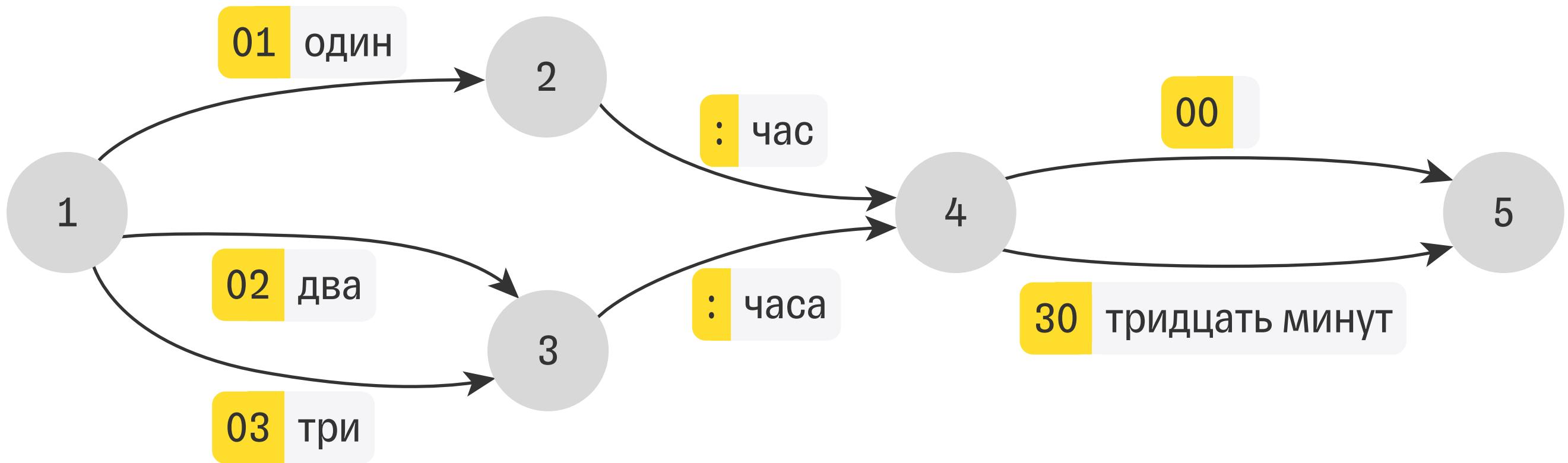
01:30 ⇒ один час тридцать минут

02:00 ⇒ два часа

03:30 ⇒ три часа тридцать минут

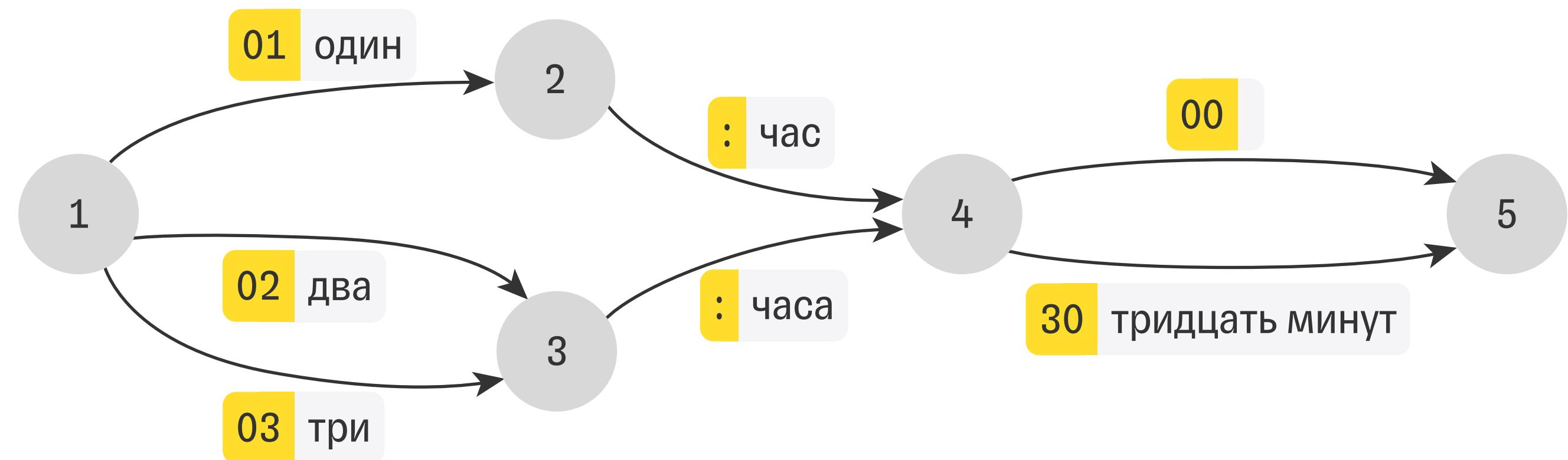
Практический пример #1

## Нормализация времени



Практический пример #1

## Нормализация времени



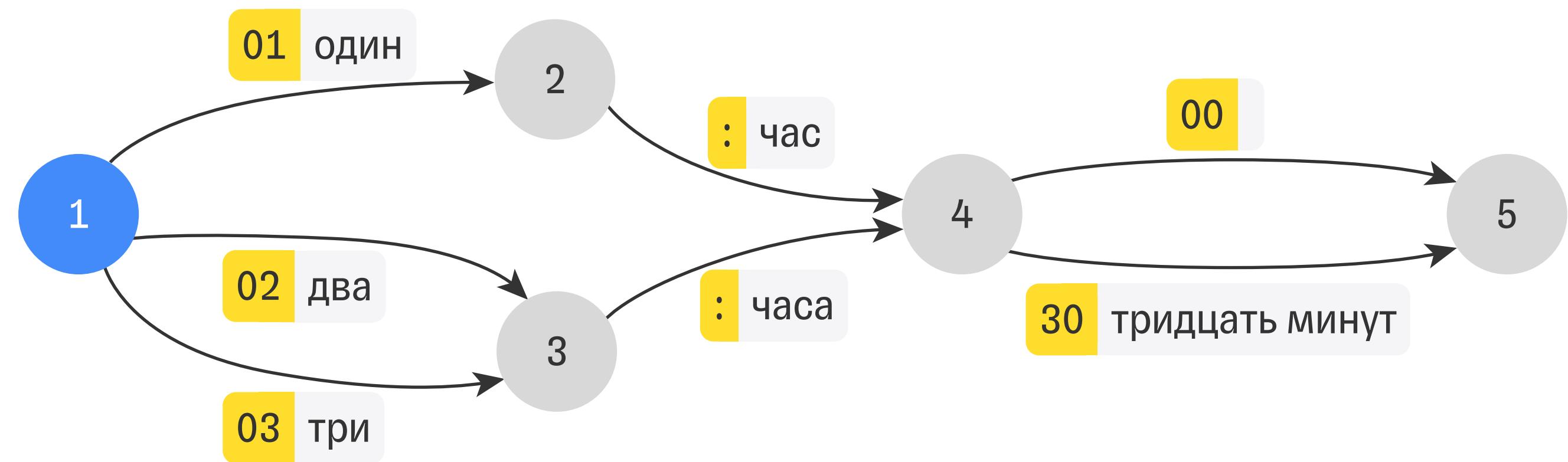
Входная строка

01:00

Выходная строка

Практический пример #1

## Нормализация времени



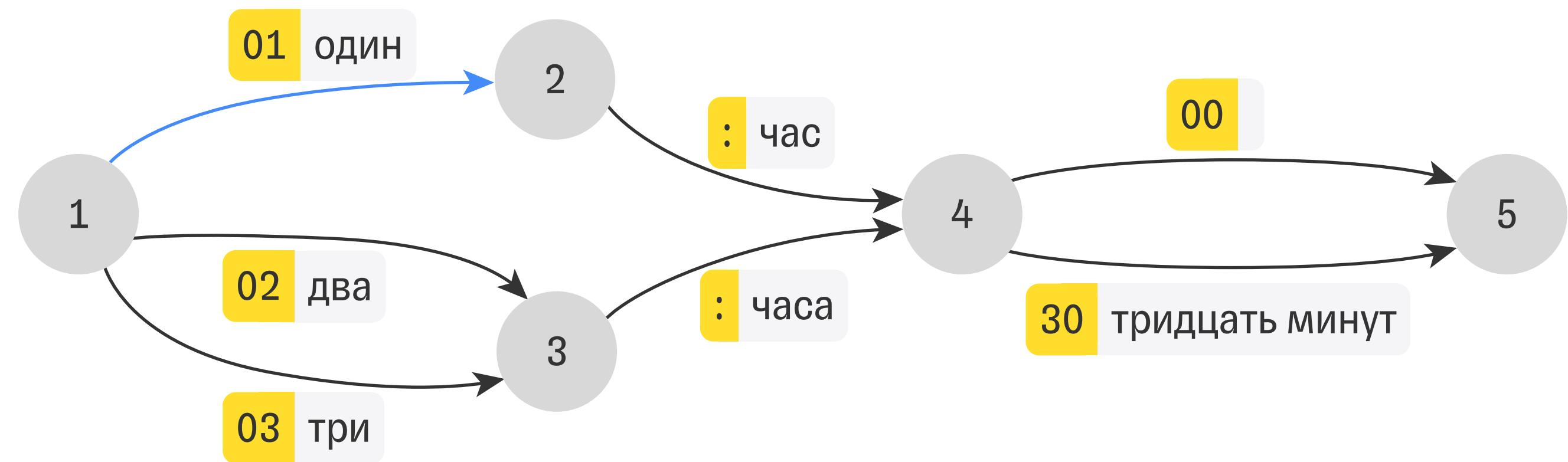
Входная строка

01:00

Выходная строка

Практический пример #1

## Нормализация времени



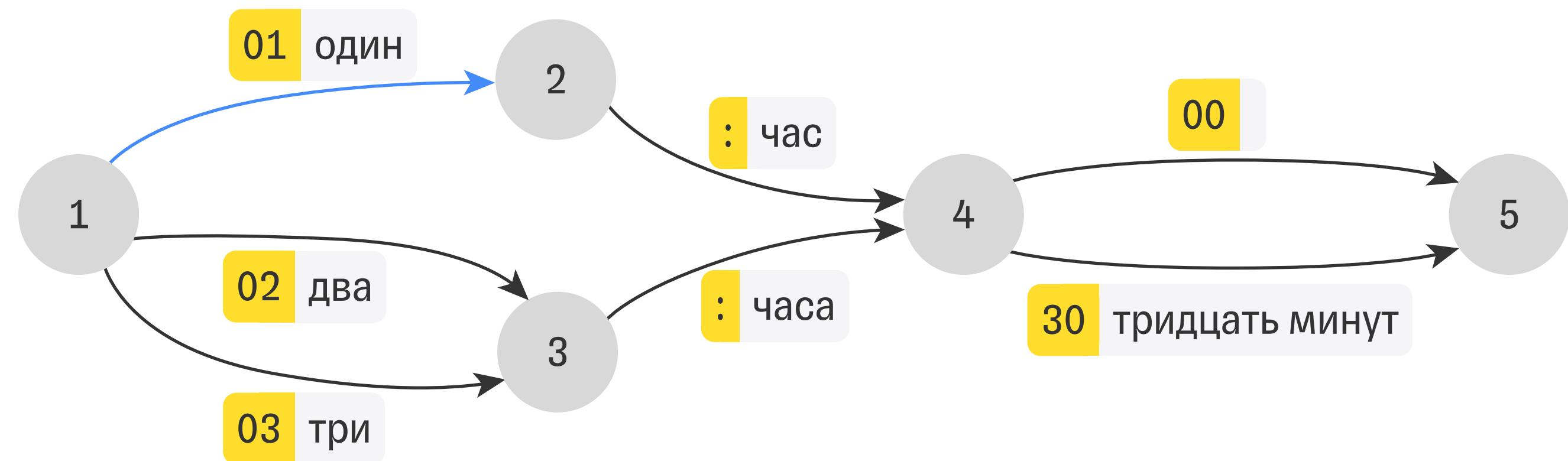
Входная строка

01:00

Выходная строка

Практический пример #1

## Нормализация времени



Входная строка

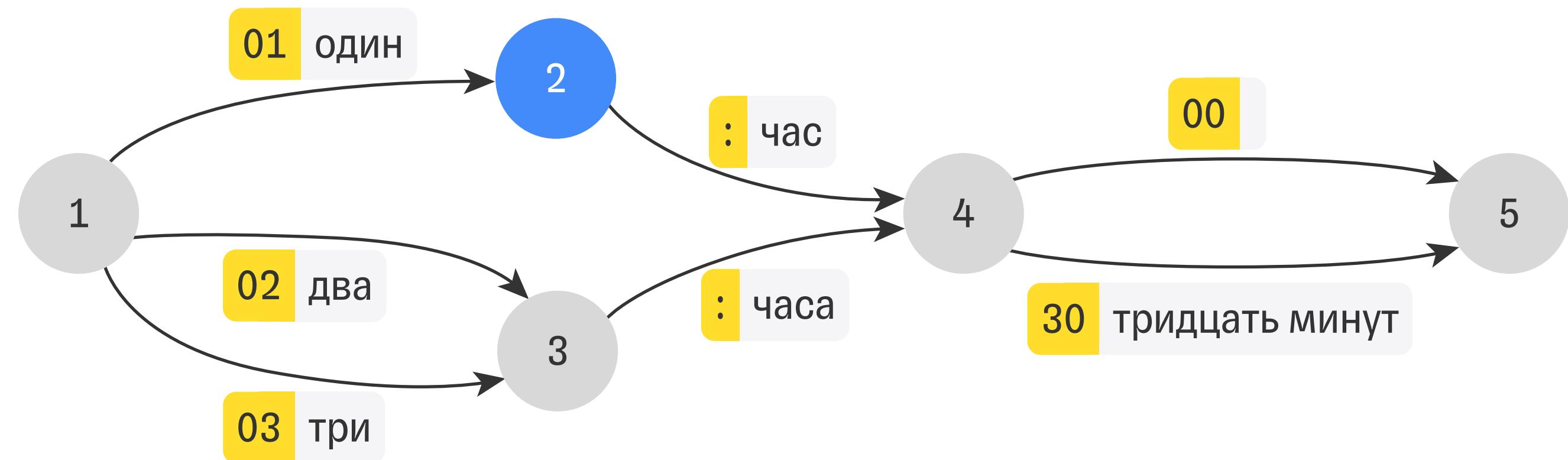
01:00

Выходная строка

один

Практический пример #1

## Нормализация времени



Входная строка

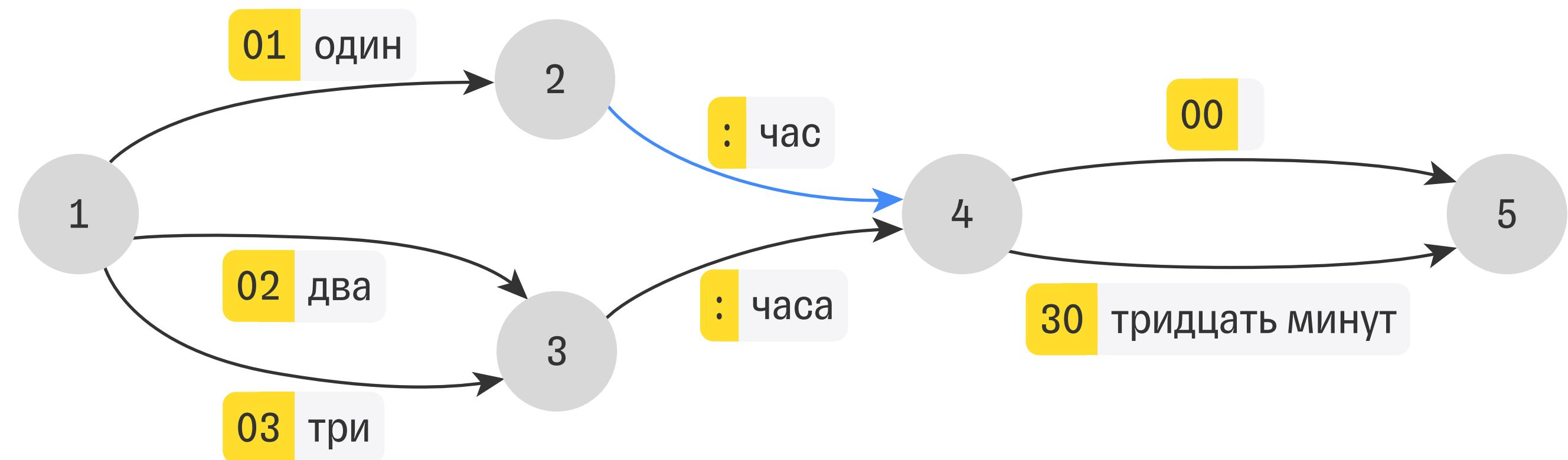
01:00

Выходная строка

один

Практический пример #1

## Нормализация времени



Входная строка

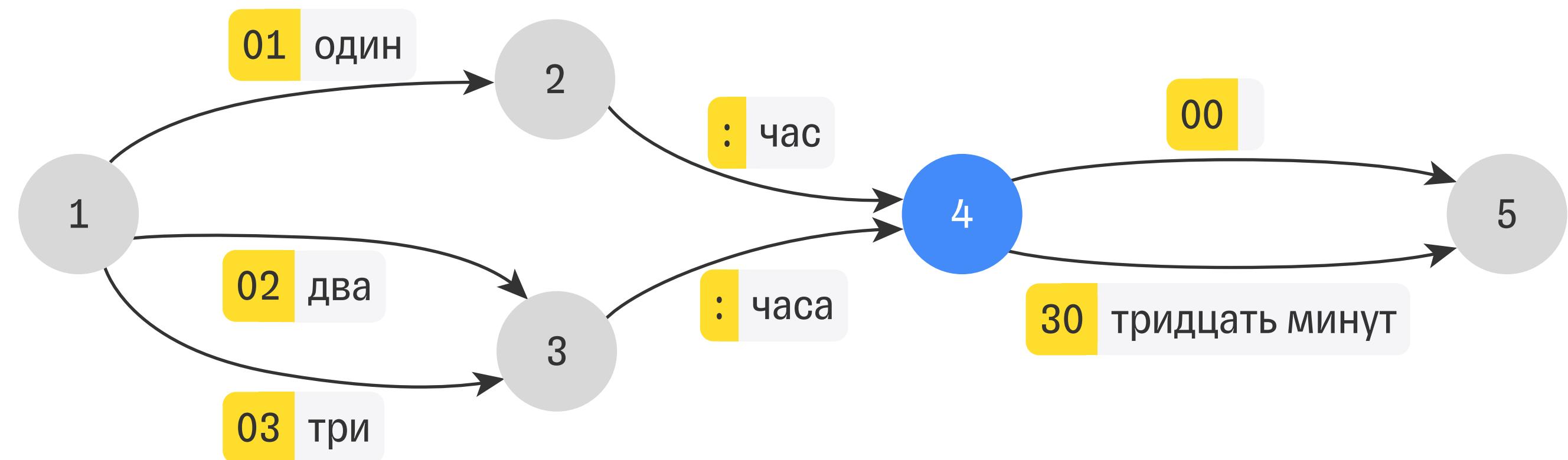
01:00

Выходная строка

один час

Практический пример #1

## Нормализация времени



Входная строка

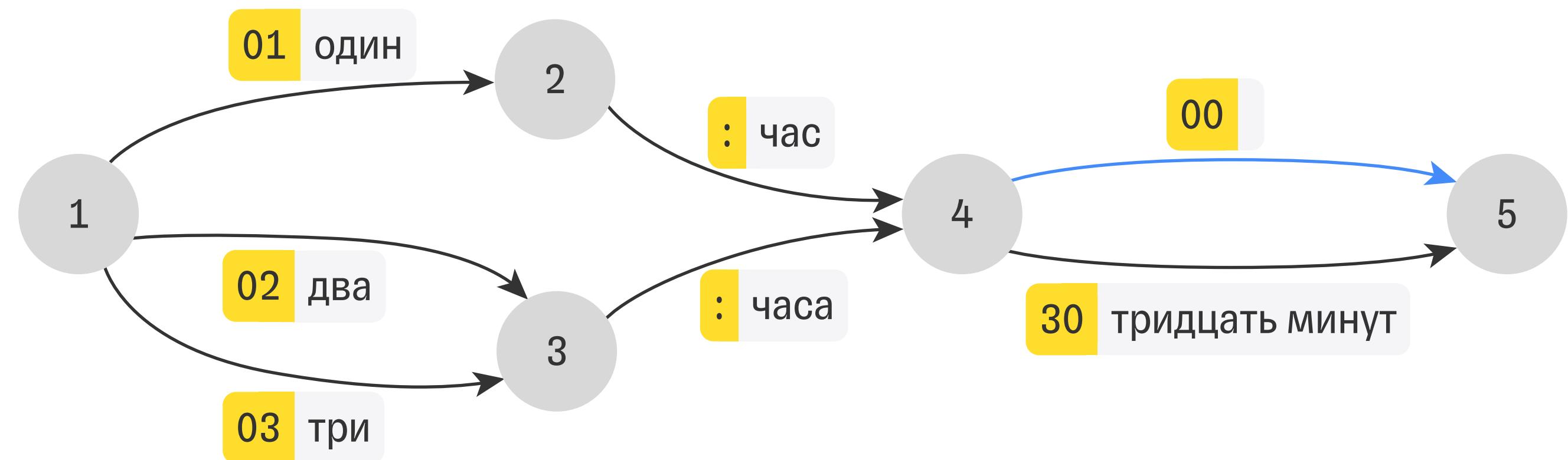
01:00

Выходная строка

один час

Практический пример #1

## Нормализация времени



Входная строка

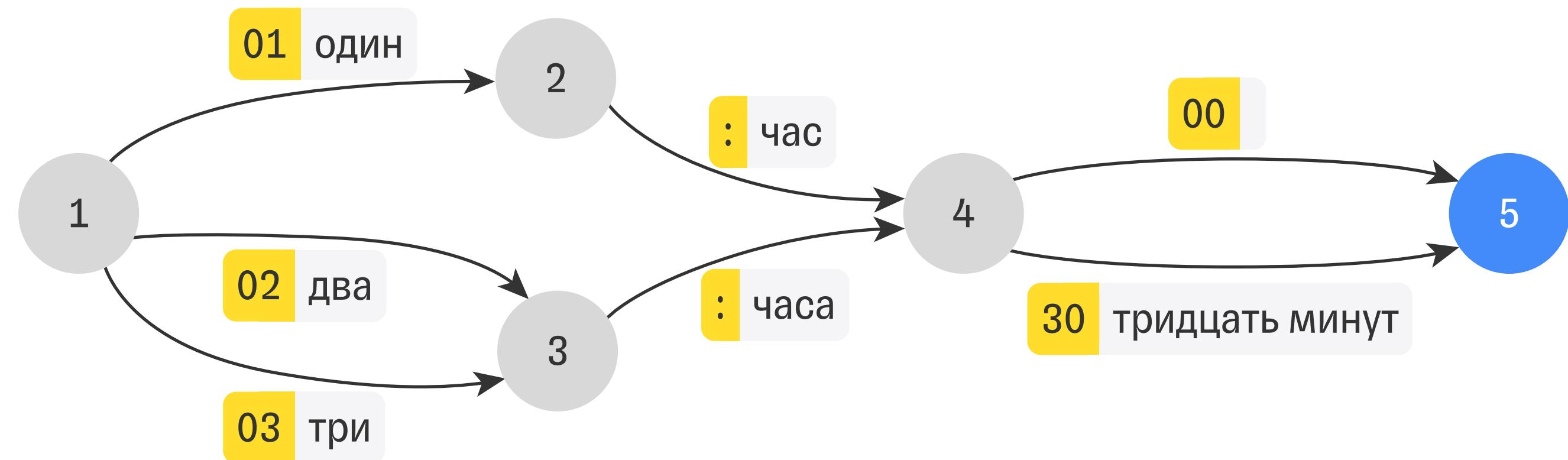
01:00

Выходная строка

один час

Практический пример #1

## Нормализация времени



Входная строка

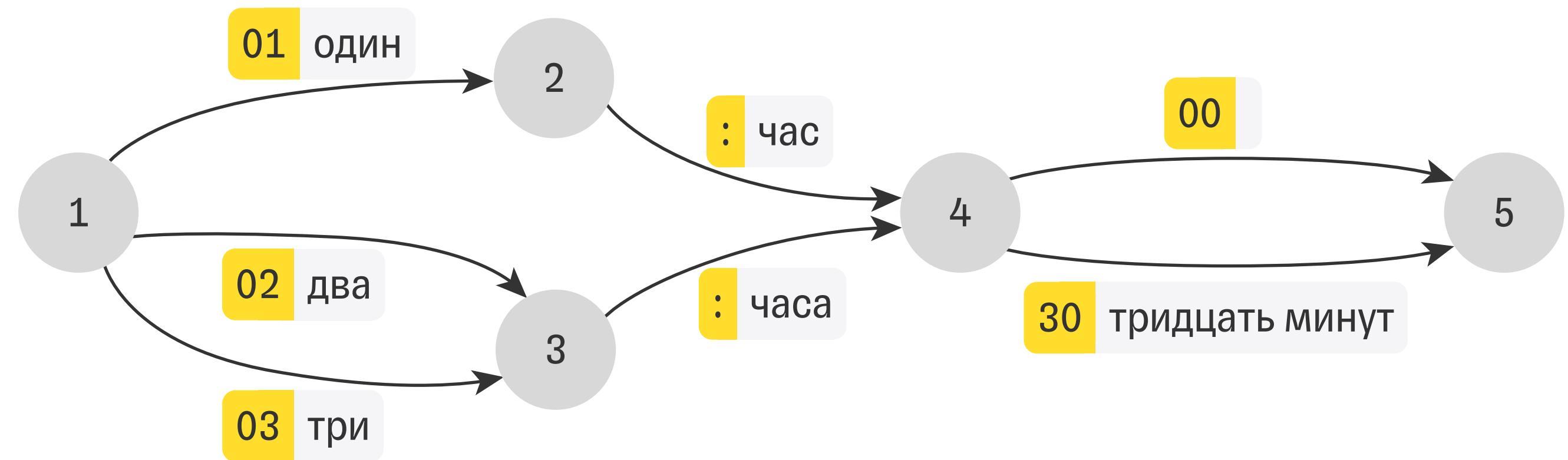
01:00

Выходная строка

один час

Практический пример #1

## Нормализация времени



Входная строка

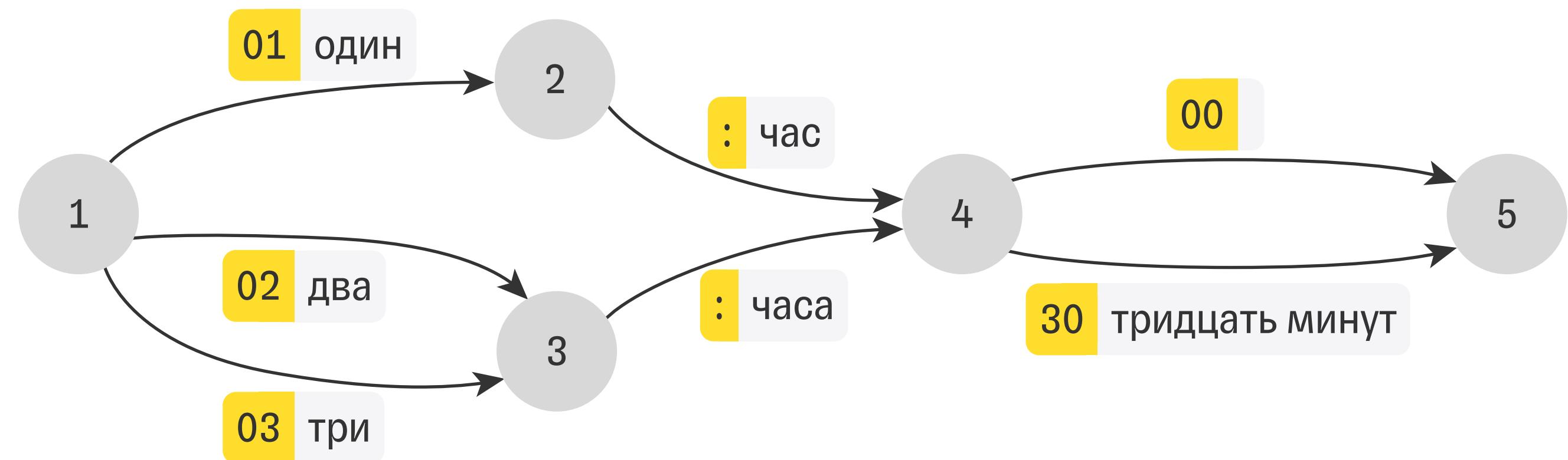
01:00

Выходная строка

один час

Практический пример #1

## Нормализация времени



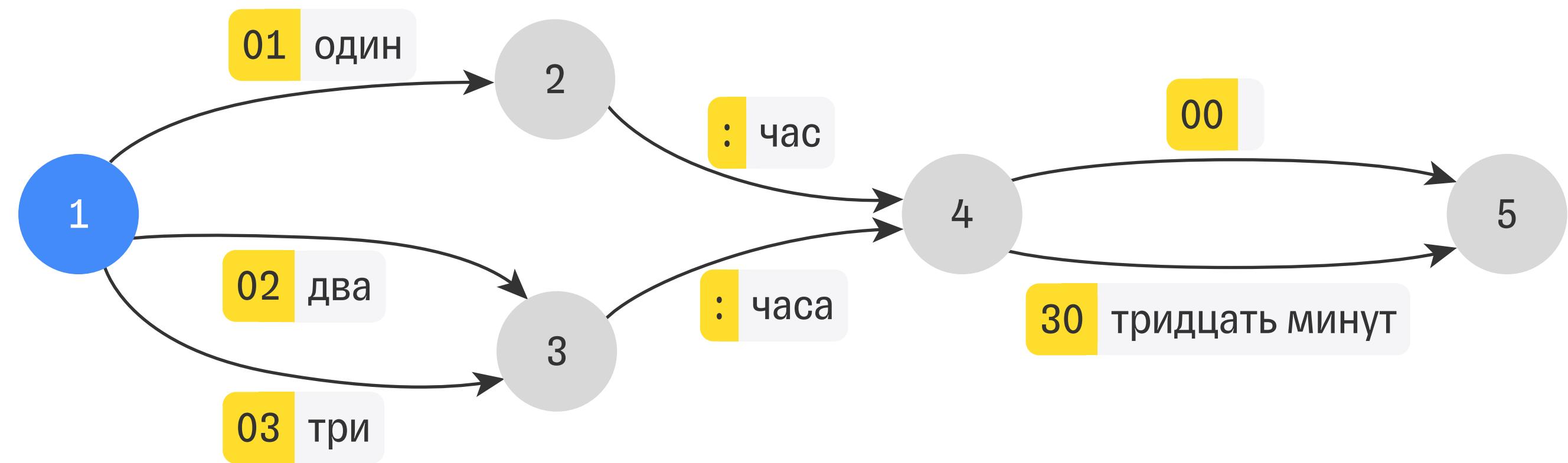
Входная строка

02:30

Выходная строка

Практический пример #1

## Нормализация времени



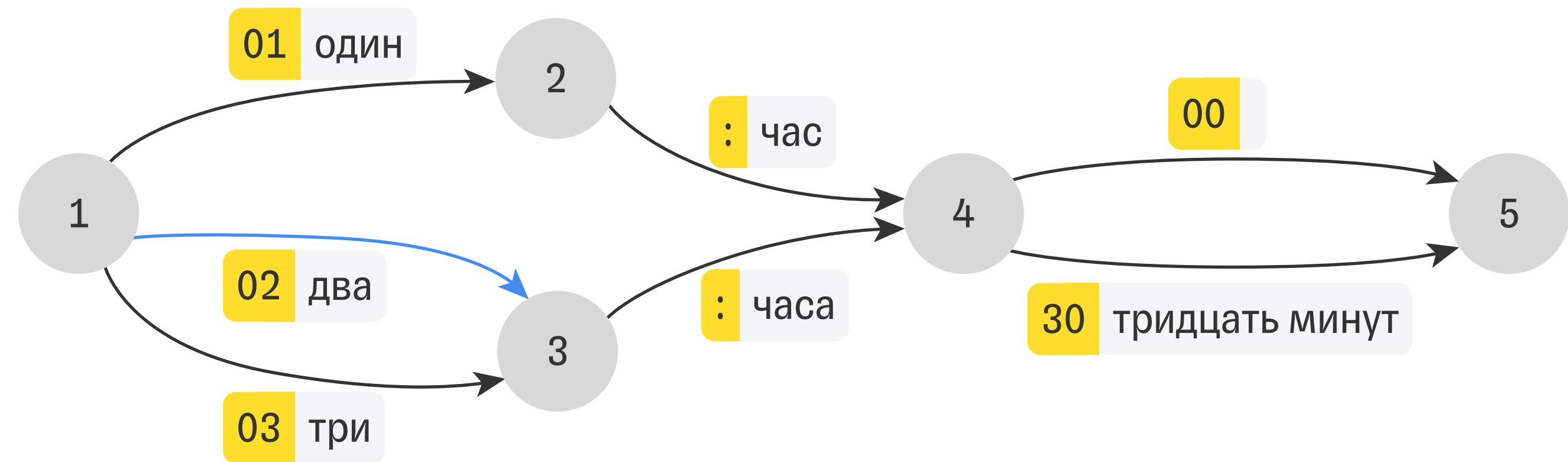
Входная строка

02:30

Выходная строка

Практический пример #1

## Нормализация времени



Входная строка

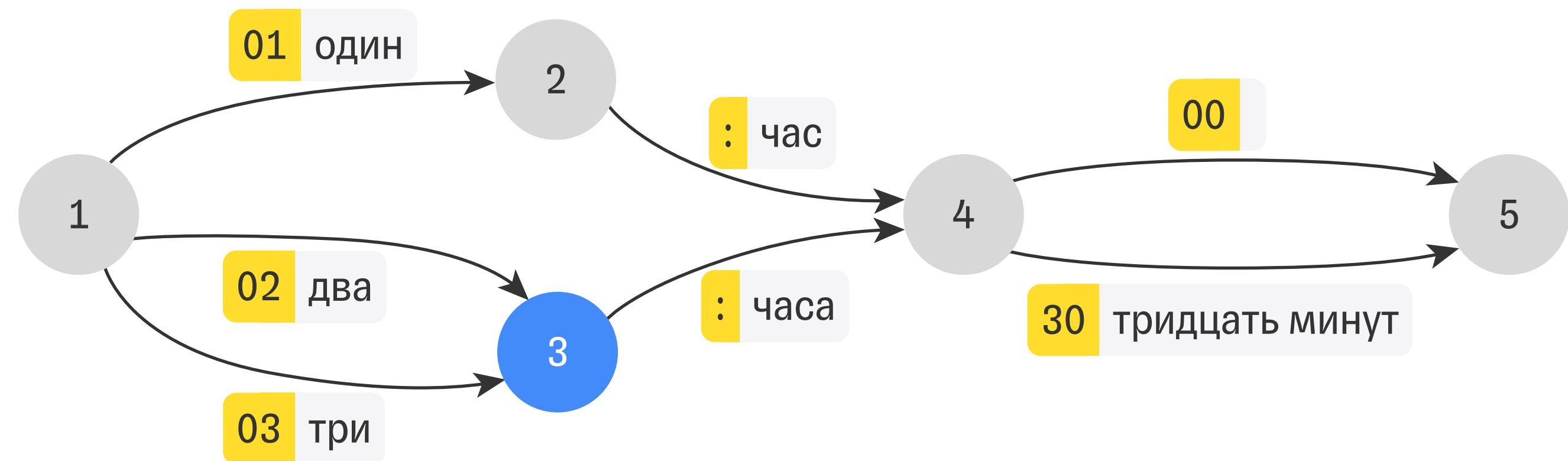
02:30

Выходная строка

два

Практический пример #1

## Нормализация времени



Входная строка

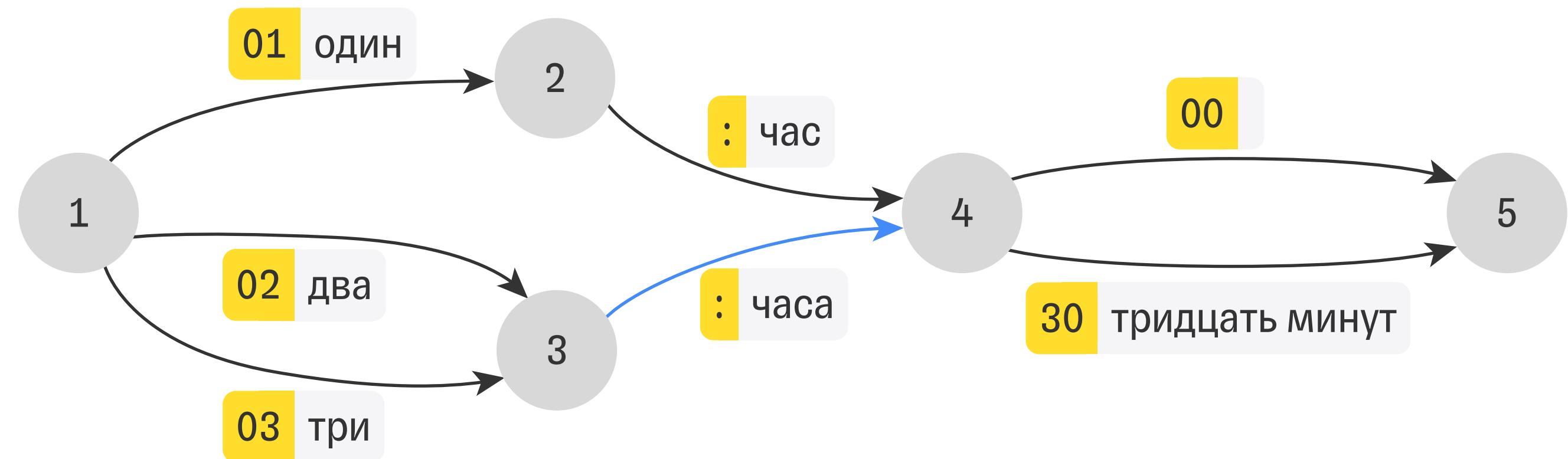
02:30

Выходная строка

два

Практический пример #1

## Нормализация времени



Входная строка

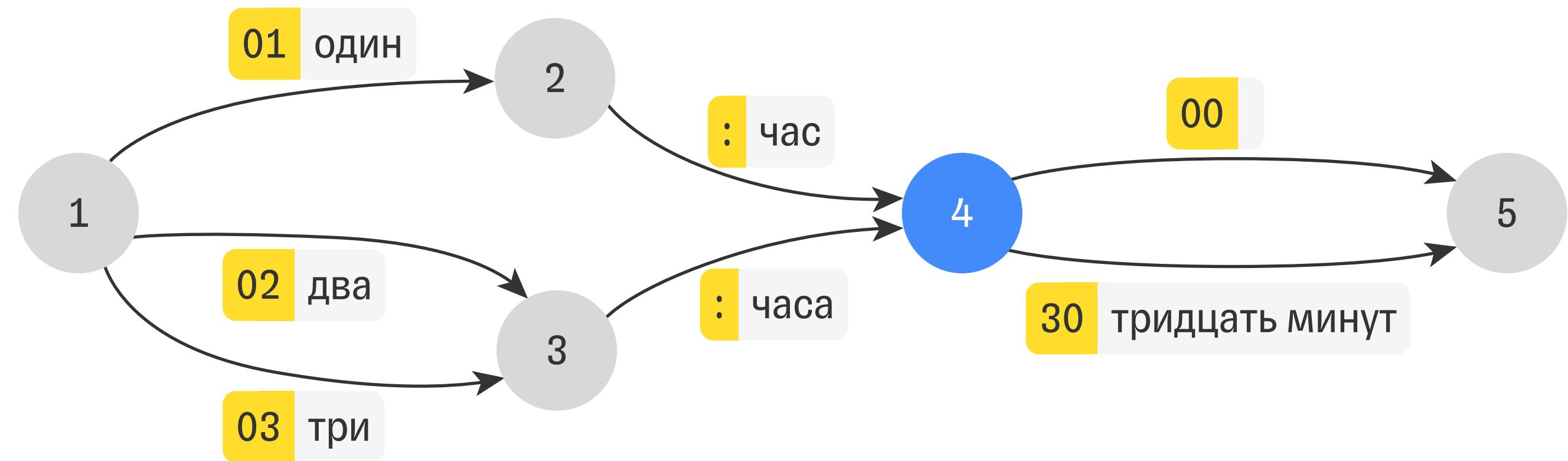
02:30

Выходная строка

два **часа**

Практический пример #1

## Нормализация времени



Входная строка

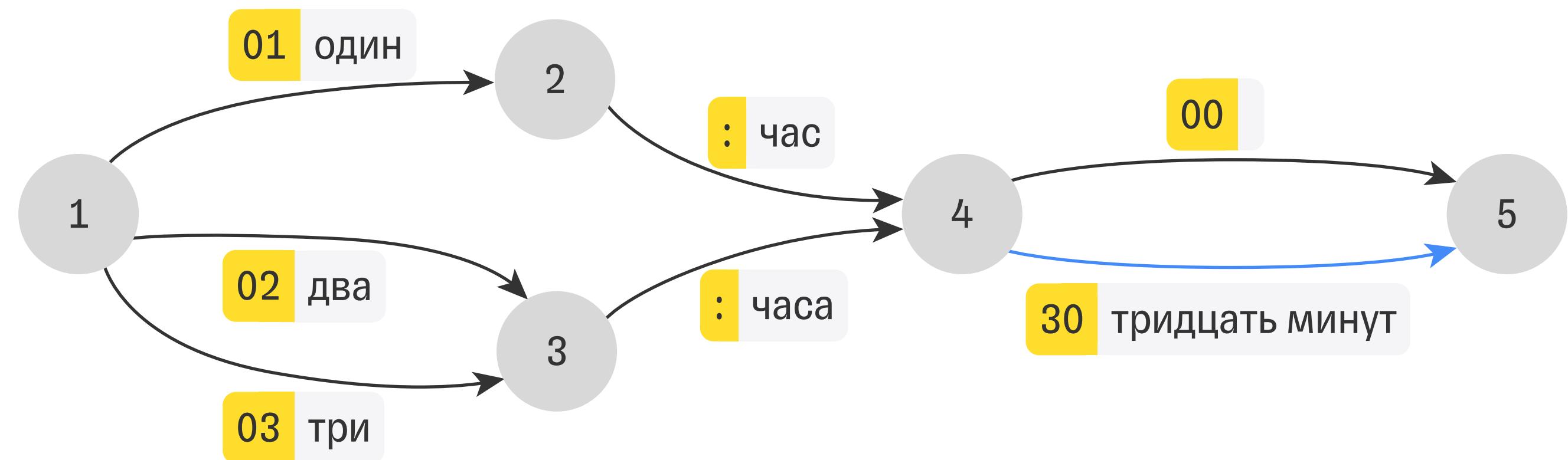
02:30

Выходная строка

два часа

Практический пример #1

## Нормализация времени



Входная строка

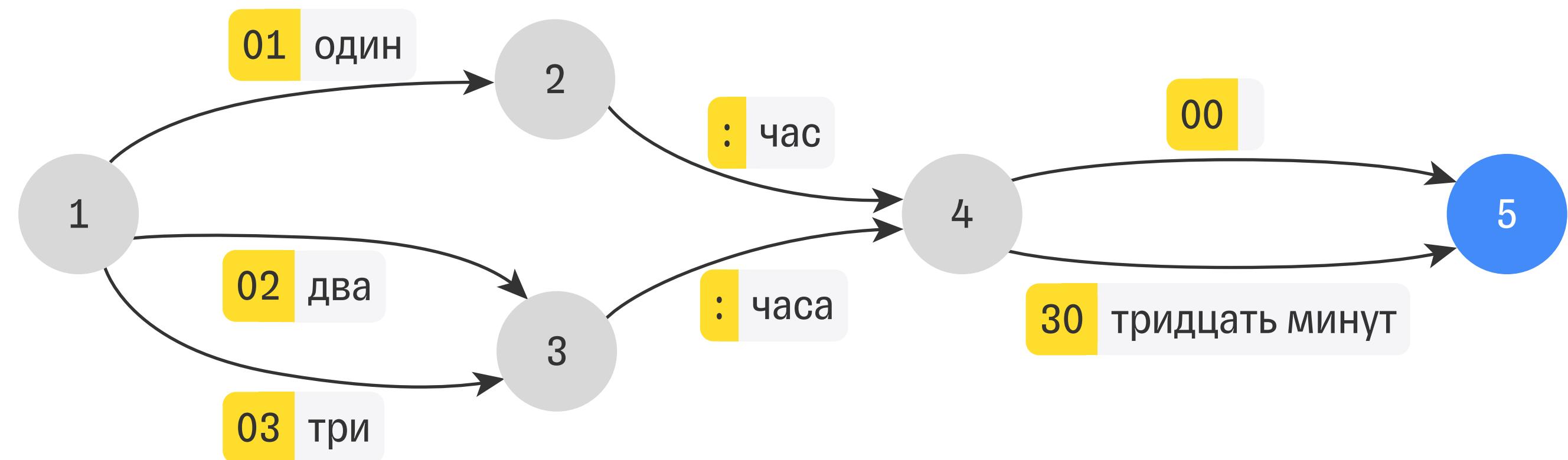
02:30

Выходная строка

два часа тридцать минут

Практический пример #1

## Нормализация времени



Входная строка

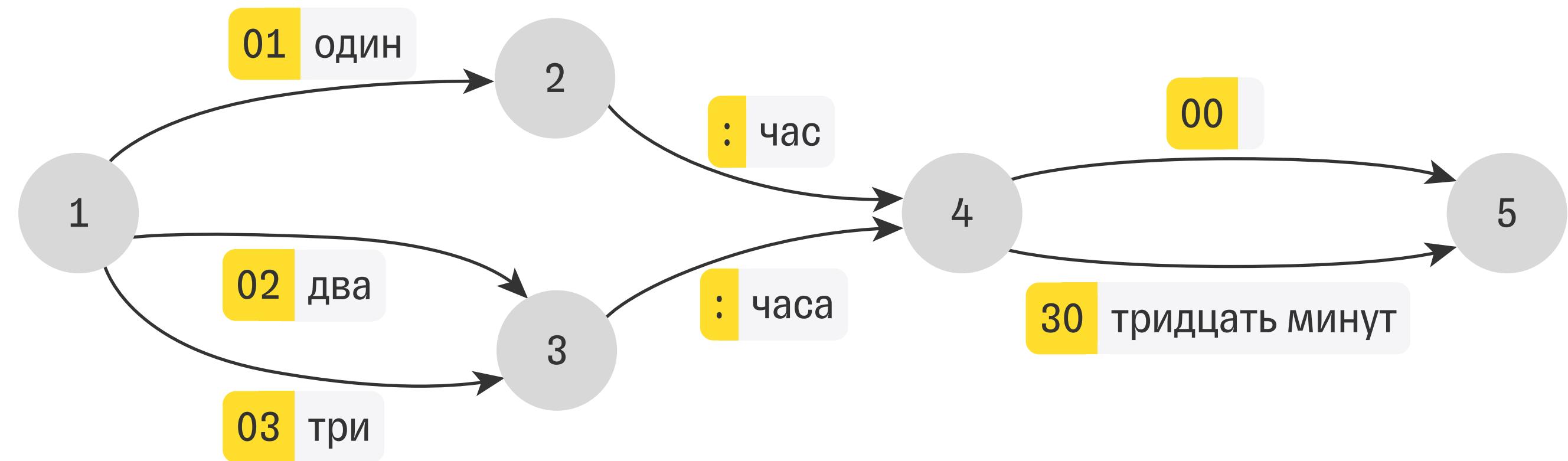
02:30

Выходная строка

два часа тридцать минут

Практический пример #1

## Нормализация времени



Входная строка

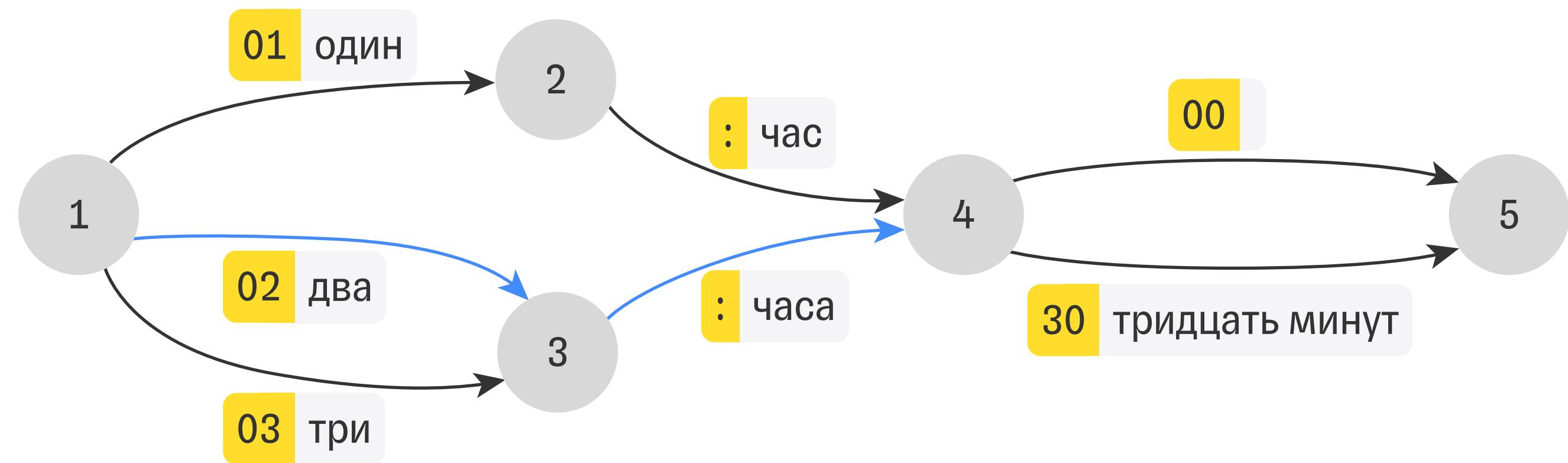
02:30

Выходная строка

два часа тридцать минут

Практический пример #1

## Нормализация времени



Входная строка

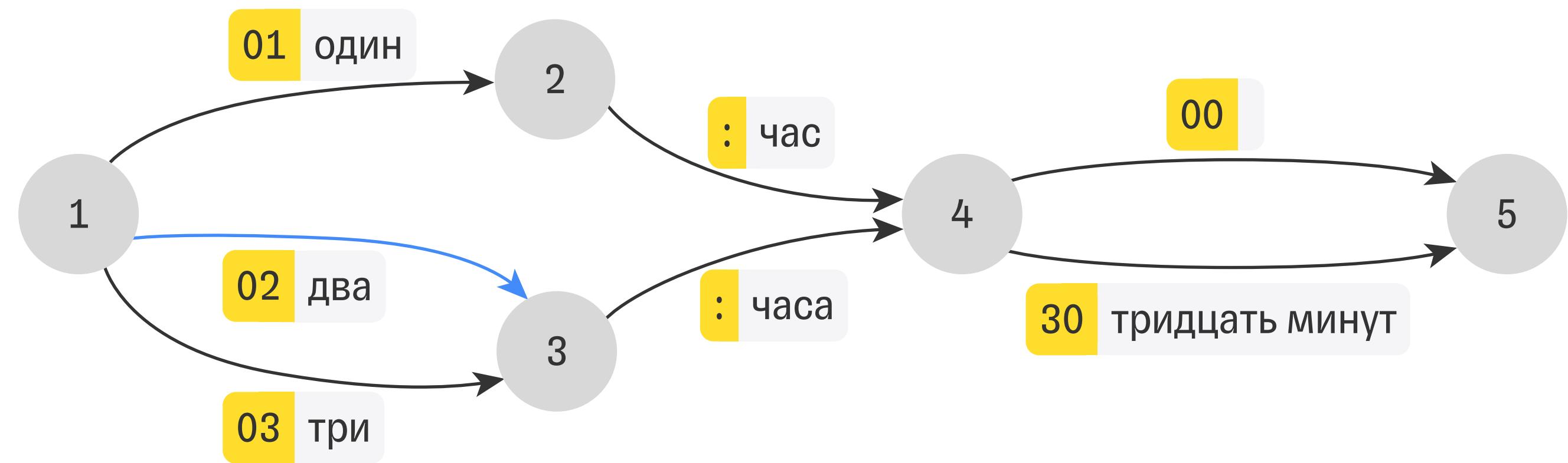
02:30

Выходная строка

два **часа** тридцать минут

Практический пример #1

## Нормализация времени



Входная строка

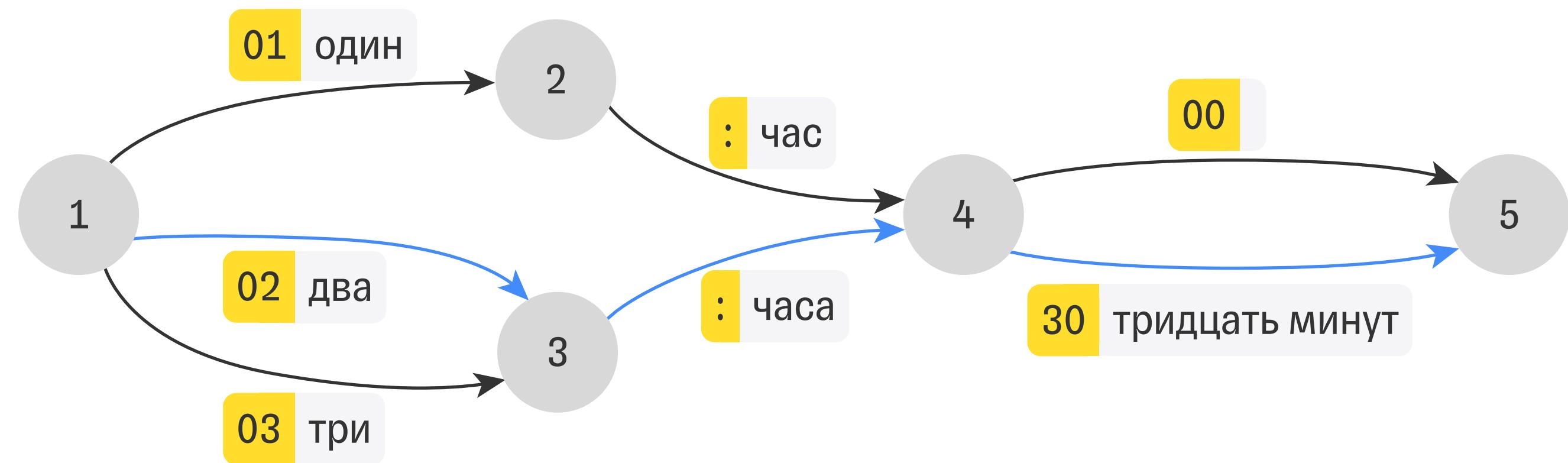
02:30

Выходная строка

два часа тридцать минут

Практический пример #1

## Нормализация времени



Входная строка

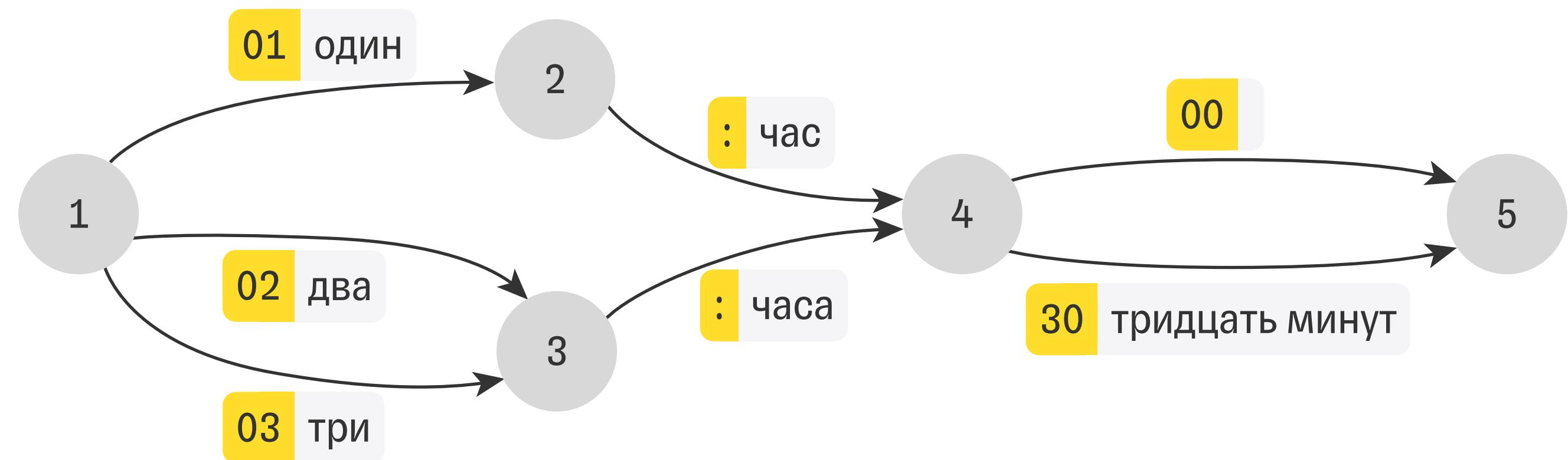
02:30

Выходная строка

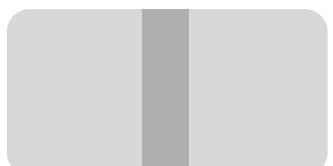
два часа тридцать минут

Практический пример #1

## Нормализация времени



Входная строка

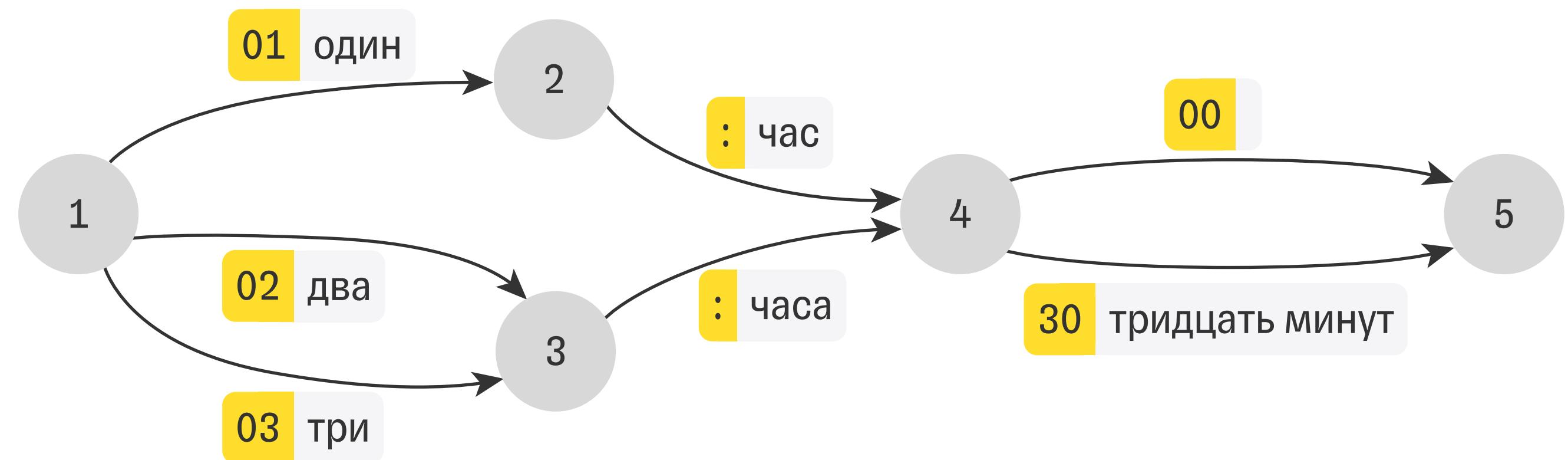


Выходная строка

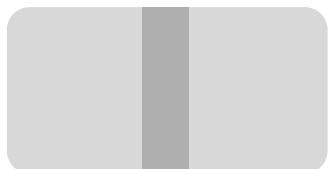


Практический пример #1

## Нормализация времени



Входная строка

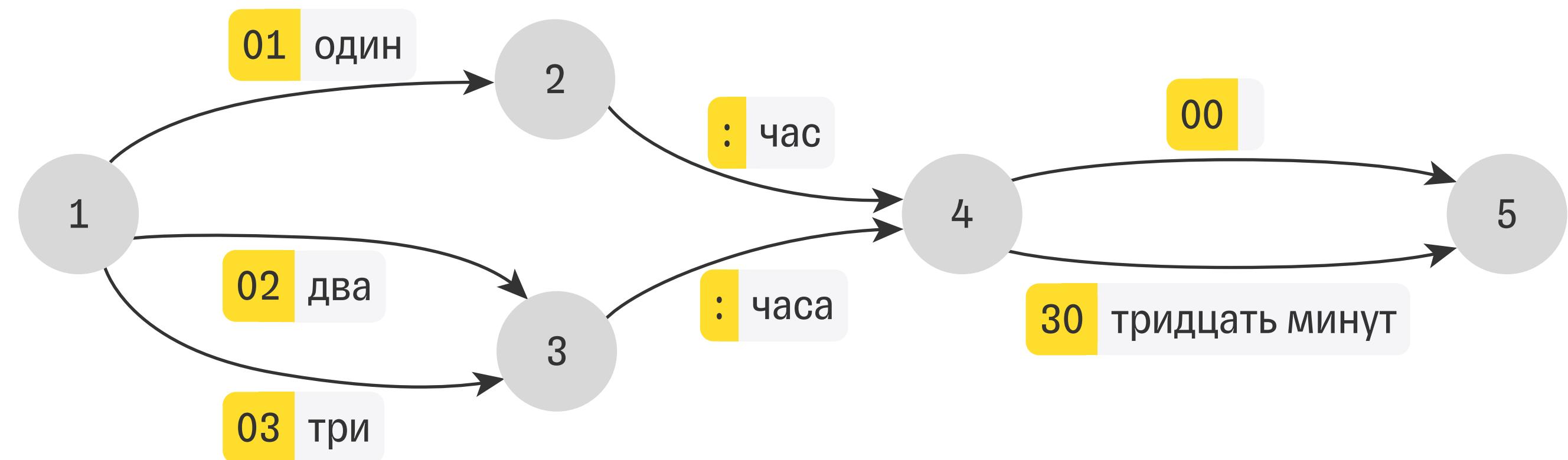


Выходная строка

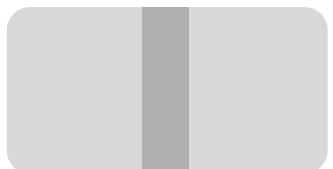


Практический пример #1

## Нормализация времени



Входная строка

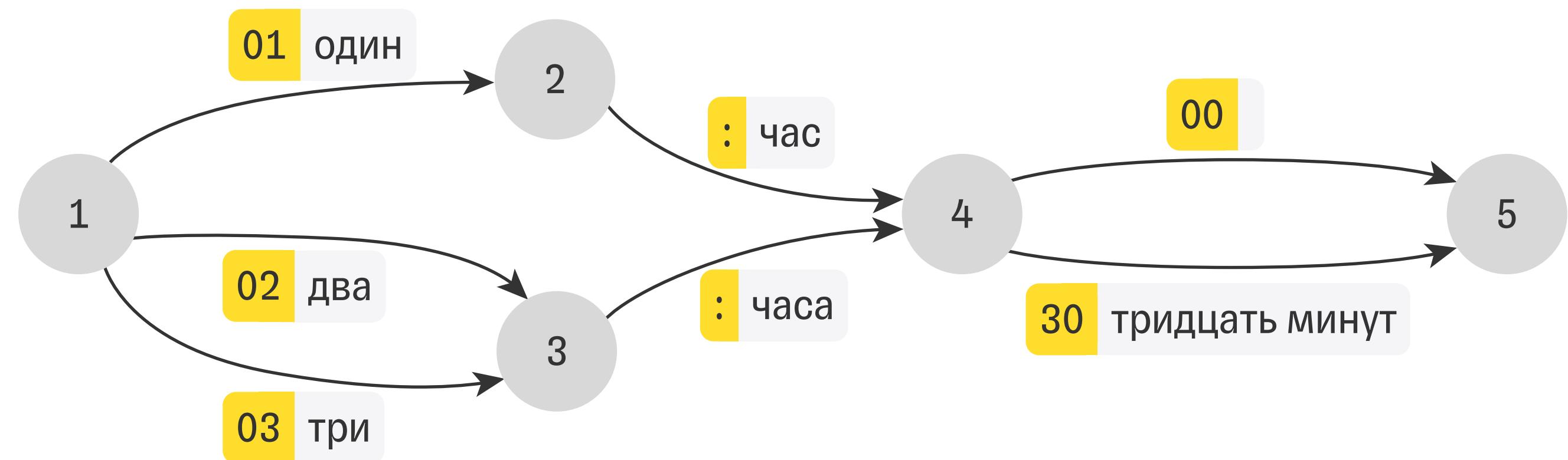


Выходная строка

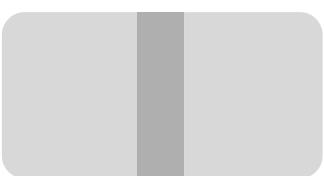


Практический пример #1

## Нормализация времени



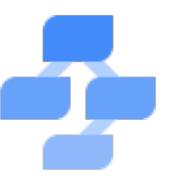
Входная строка



Выходная строка



## Нормализация времени



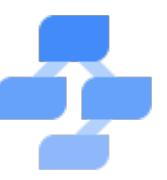
### PyNini – инструмент для создания FST

- Python библиотека
- Предназначена для работы с конечными автоматами (FST)
- Позволяет строить, оптимизировать и применять конечные автоматы для NLP задач
- Основана на OpenFST – де-факто стандарт для работы с FST

## Нормализация времени



Google Colab



### PyNini – инструмент для создания FST

- Python библиотека
- Предназначена для работы с конечными автоматами (FST)
- Позволяет строить, оптимизировать и применять конечные автоматы для NLP задач
- Основана на OpenFST – де-факто стандарт для работы с FST

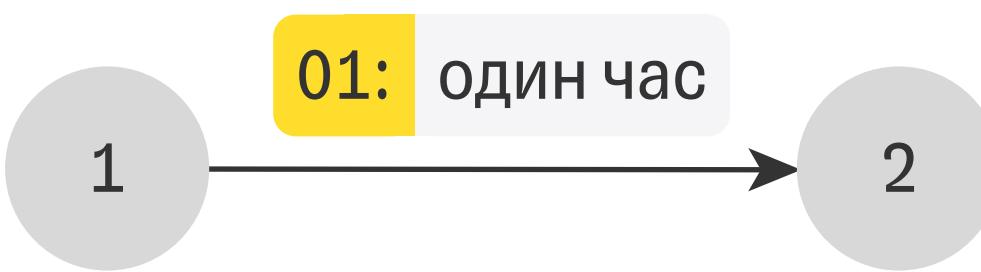
```
import pynini
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени



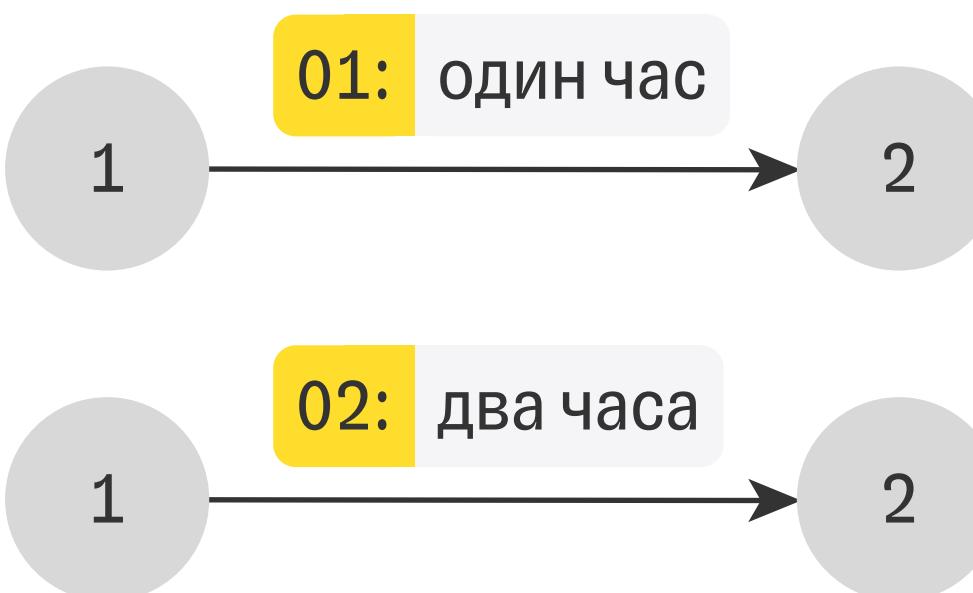
Google Colab



```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени

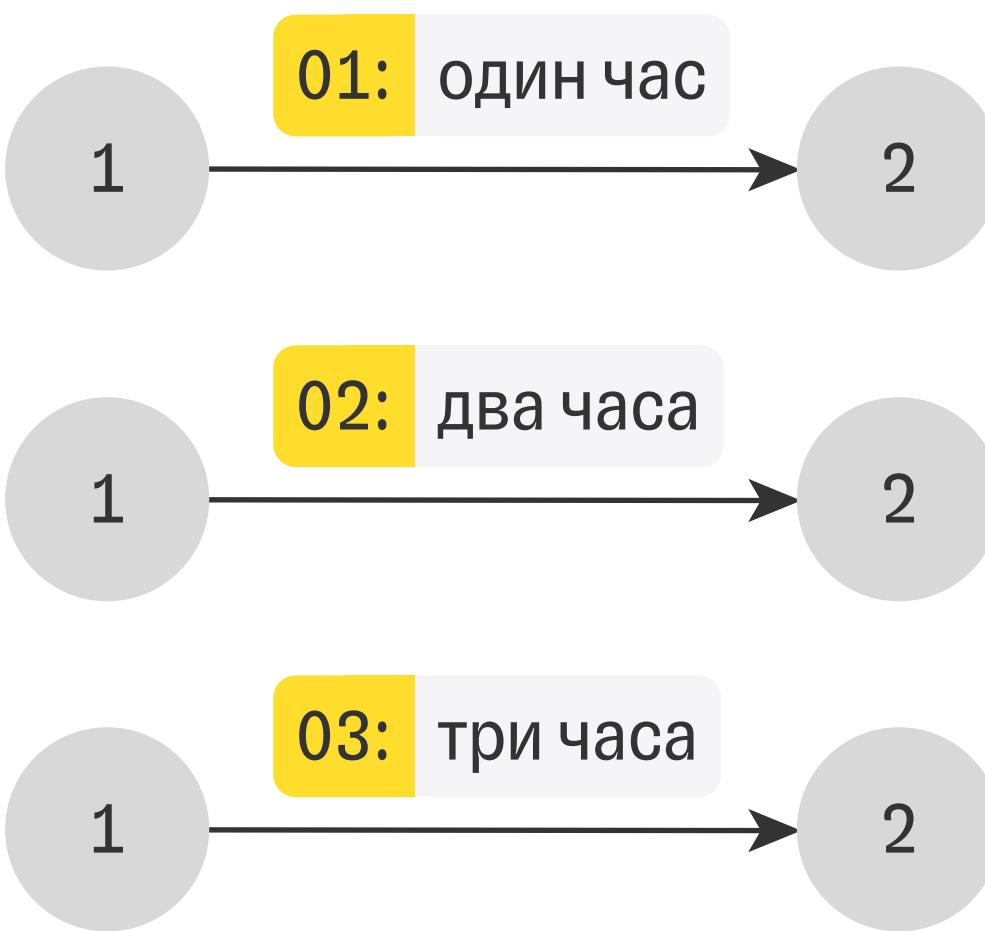


Google Colab

```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени

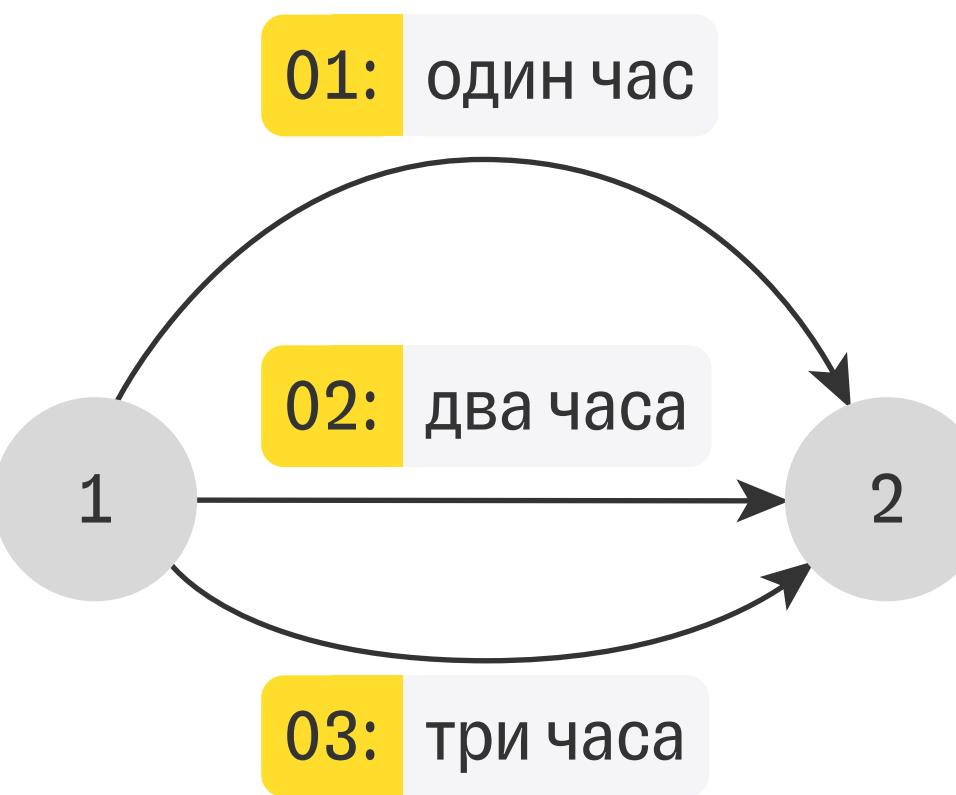


Google Colab

```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени



Google Colab

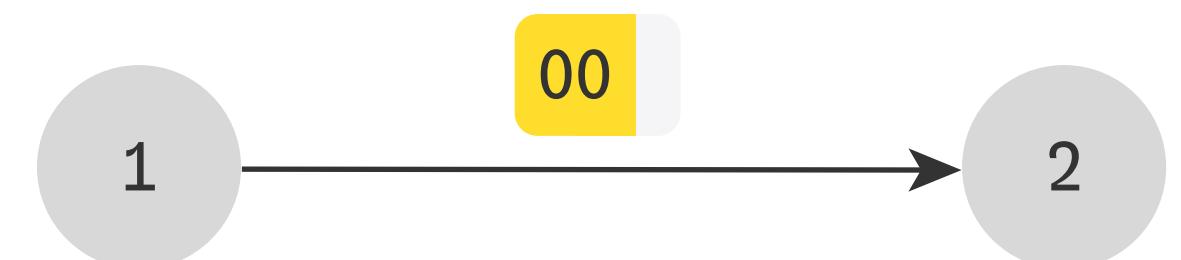
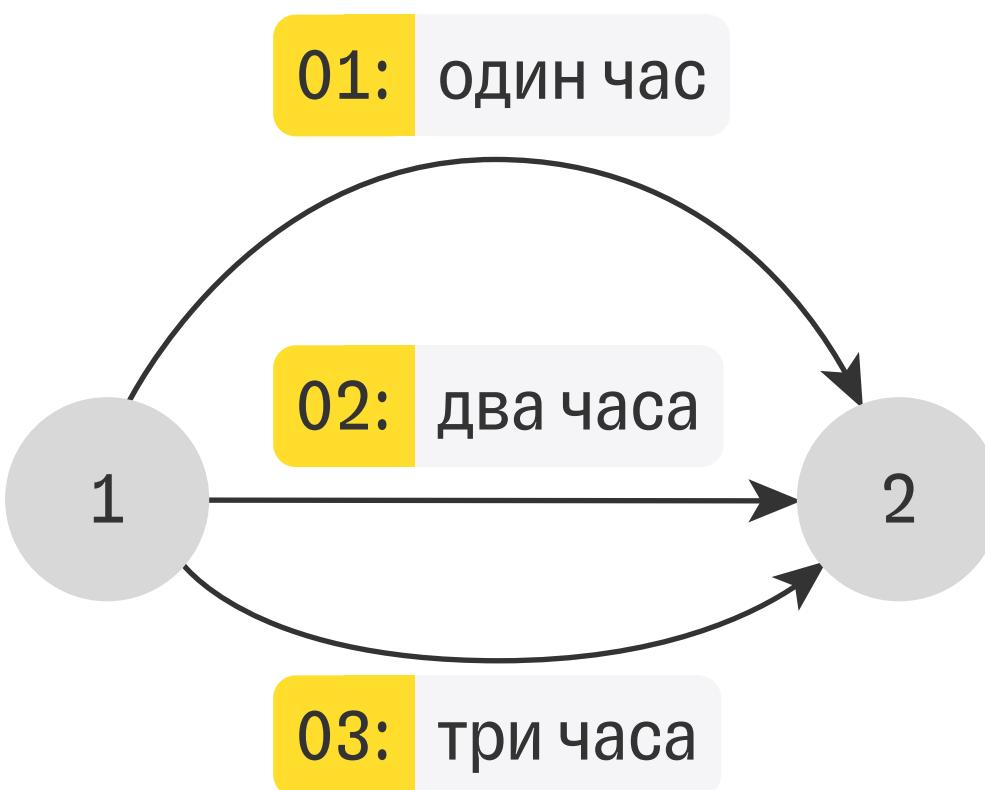
```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))

minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))

fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени

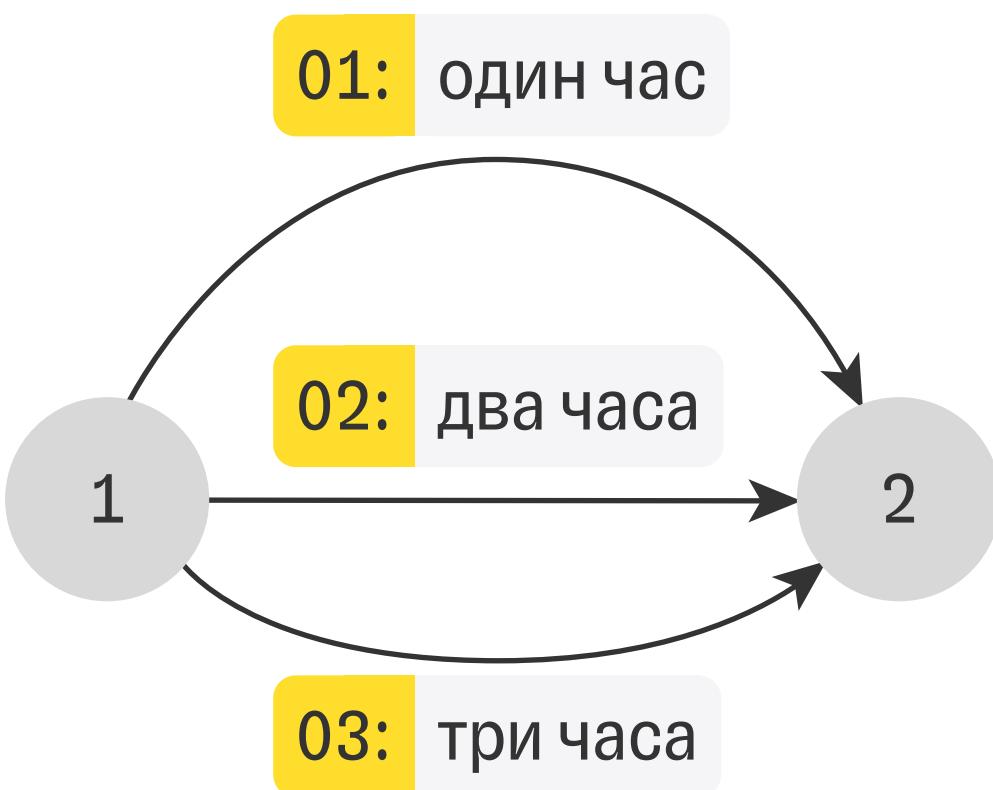


Google Colab

```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени

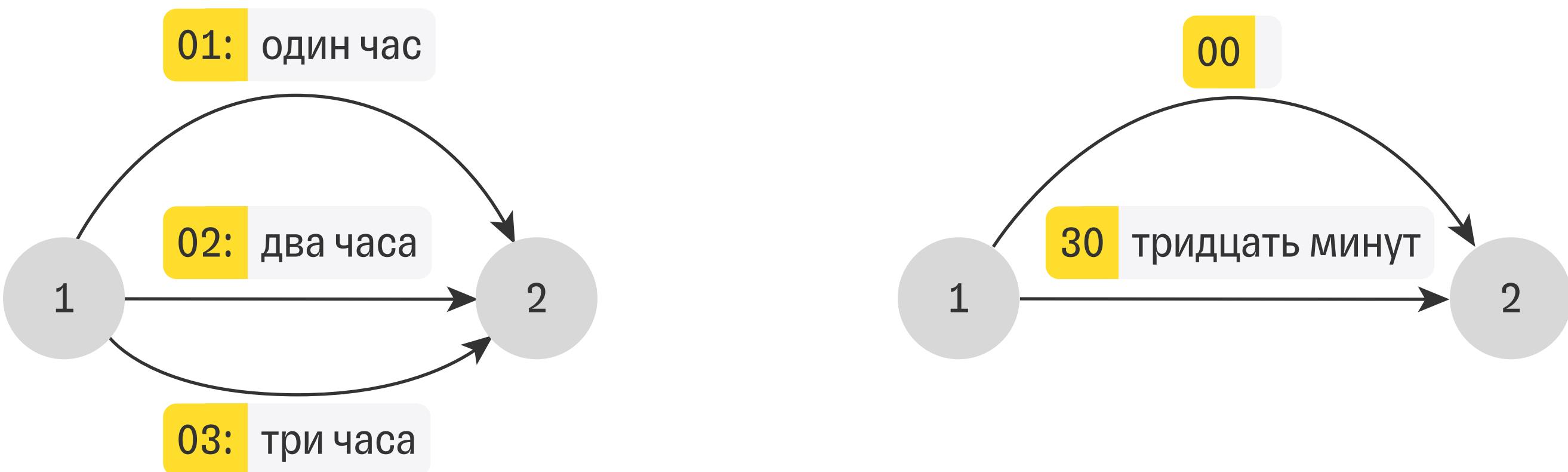


Google Colab

```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени

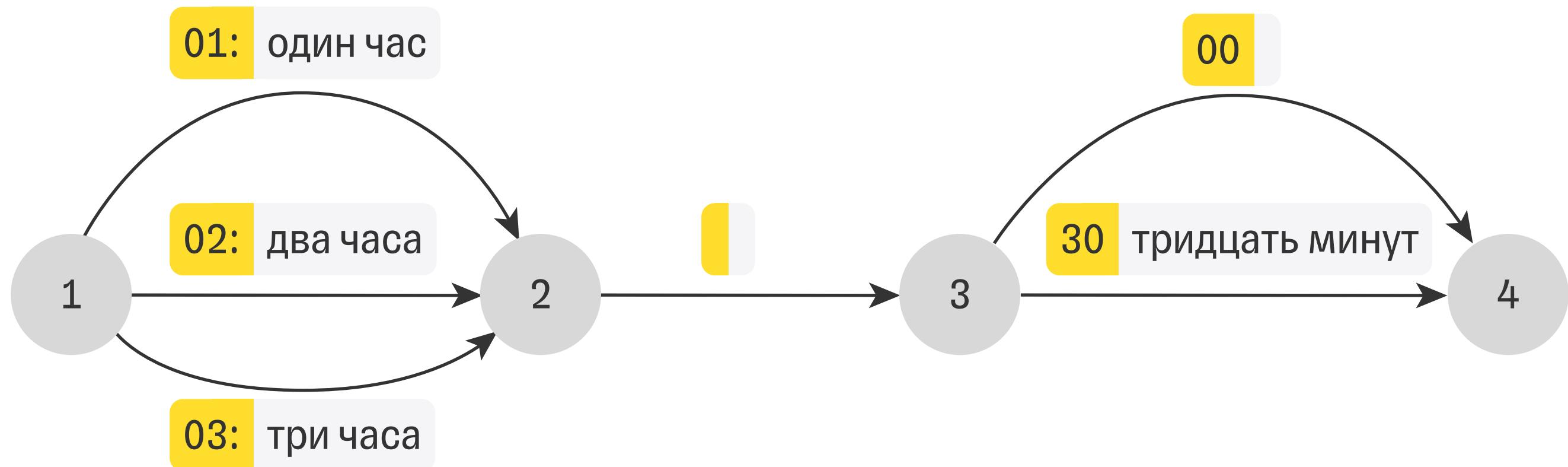


Google Colab

```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени

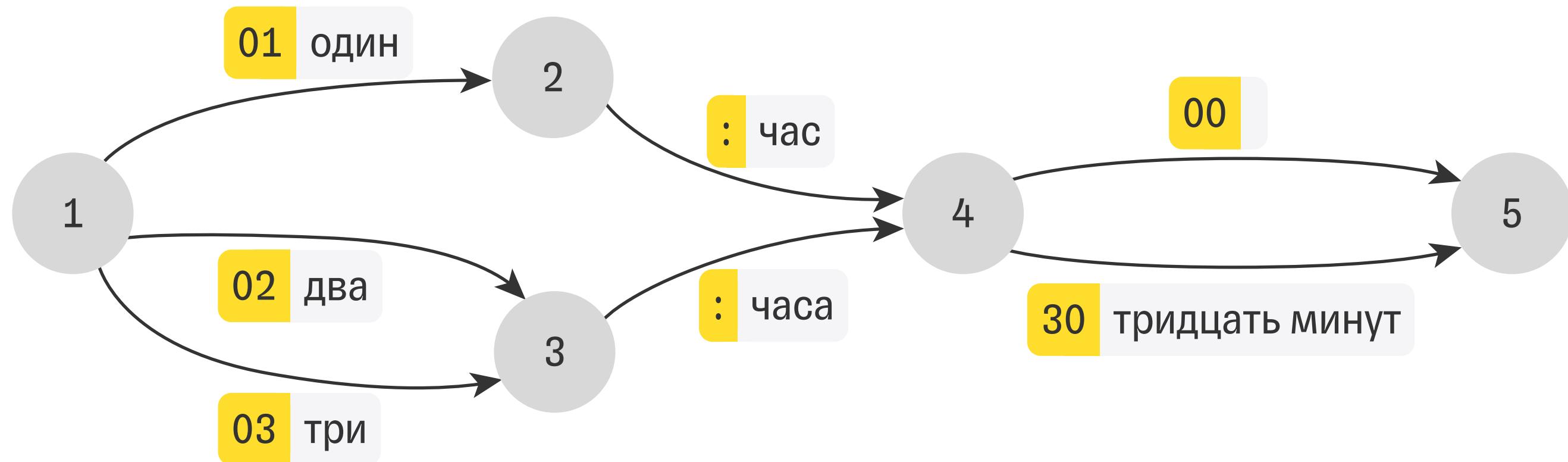


Google Colab

```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #1

## Нормализация времени



Google Colab

```
hours = pynini.union(
    pynini.cross("01:", "один час"),
    pynini.cross("02:", "два часа"),
    pynini.cross("03:", "три часа"))
minutes = pynini.union(
    pynini.cross("00", ""),
    pynini.cross("30", " тридцать минут"))
fst = pynini.concat(hours, minutes).optimize()
```

Практический пример #2

# Нормализация времени 2

## Задача

Перевести время из 12-часового формата в 24-часовой

Например:

12:15 AM  $\Rightarrow$  00:15

05:34 AM  $\Rightarrow$  05:34

12:52 PM  $\Rightarrow$  12:34

05:21 PM  $\Rightarrow$  17:21

Практический пример #2

# Нормализация времени 2

## Задача

Перевести время из 12-часового формата в 24-часовой

Например:

12:15 AM  $\Rightarrow$  00:15

05:34 AM  $\Rightarrow$  05:34

12:52 PM  $\Rightarrow$  12:34

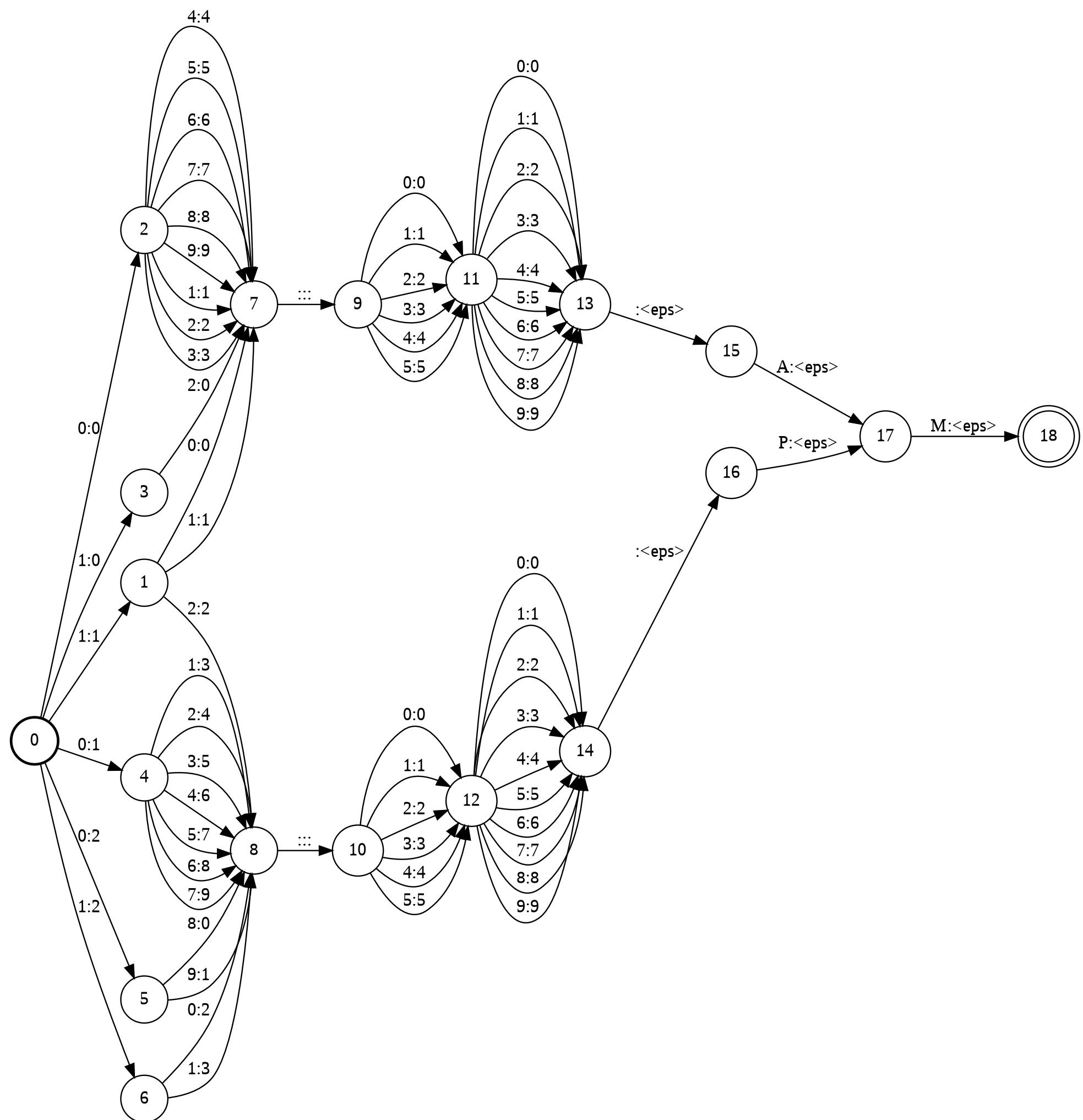
05:21 PM  $\Rightarrow$  17:21

Практический пример #2

## Нормализация времени 2

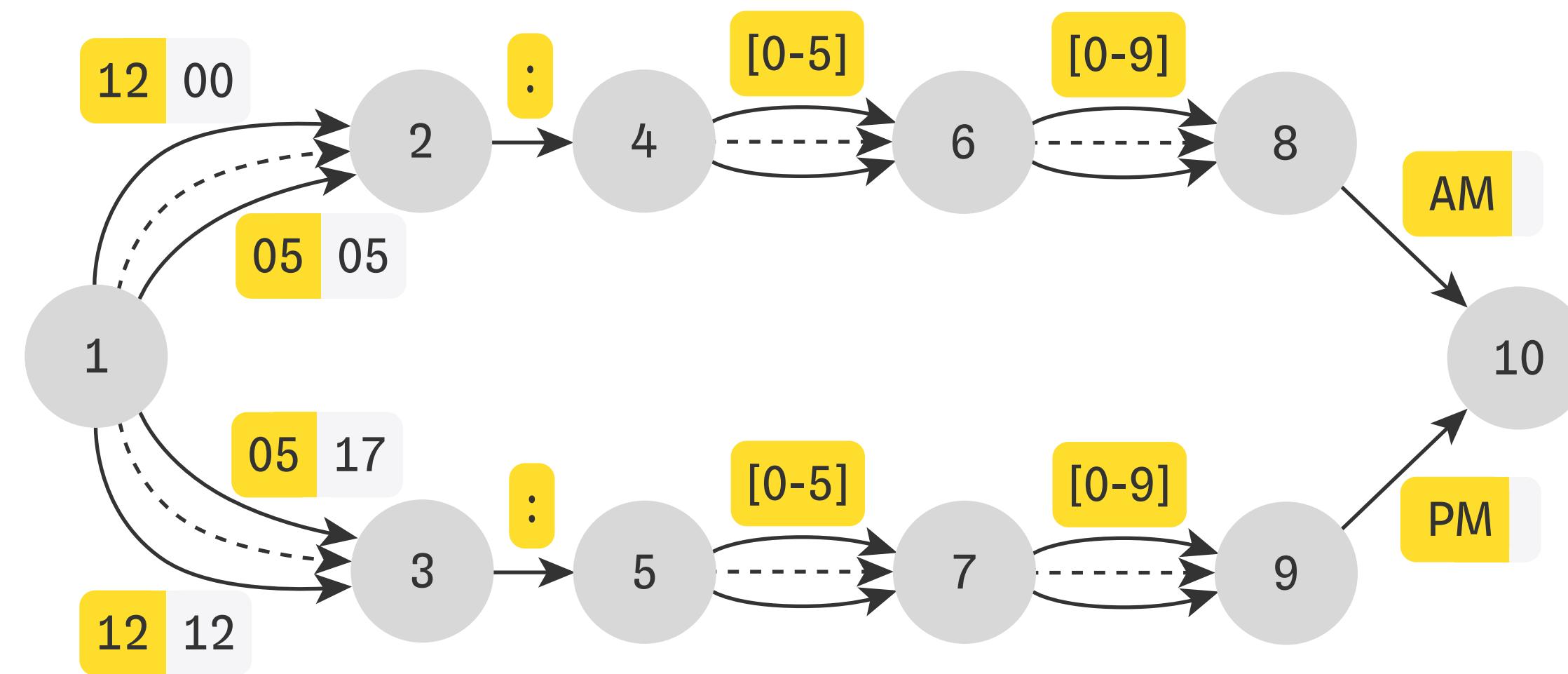


Google Colab



Практический пример #2

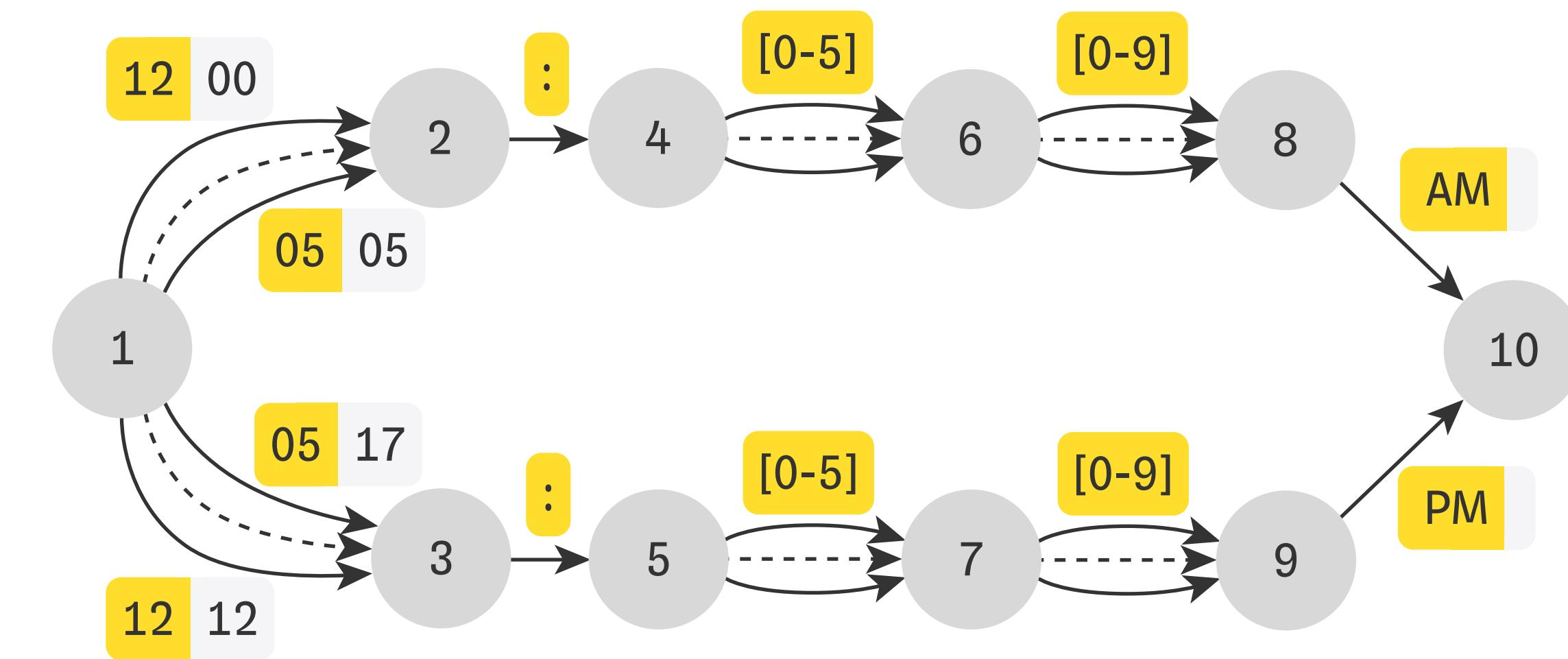
## Нормализация времени 2



Google Colab

Практический пример #2

## Нормализация времени 2



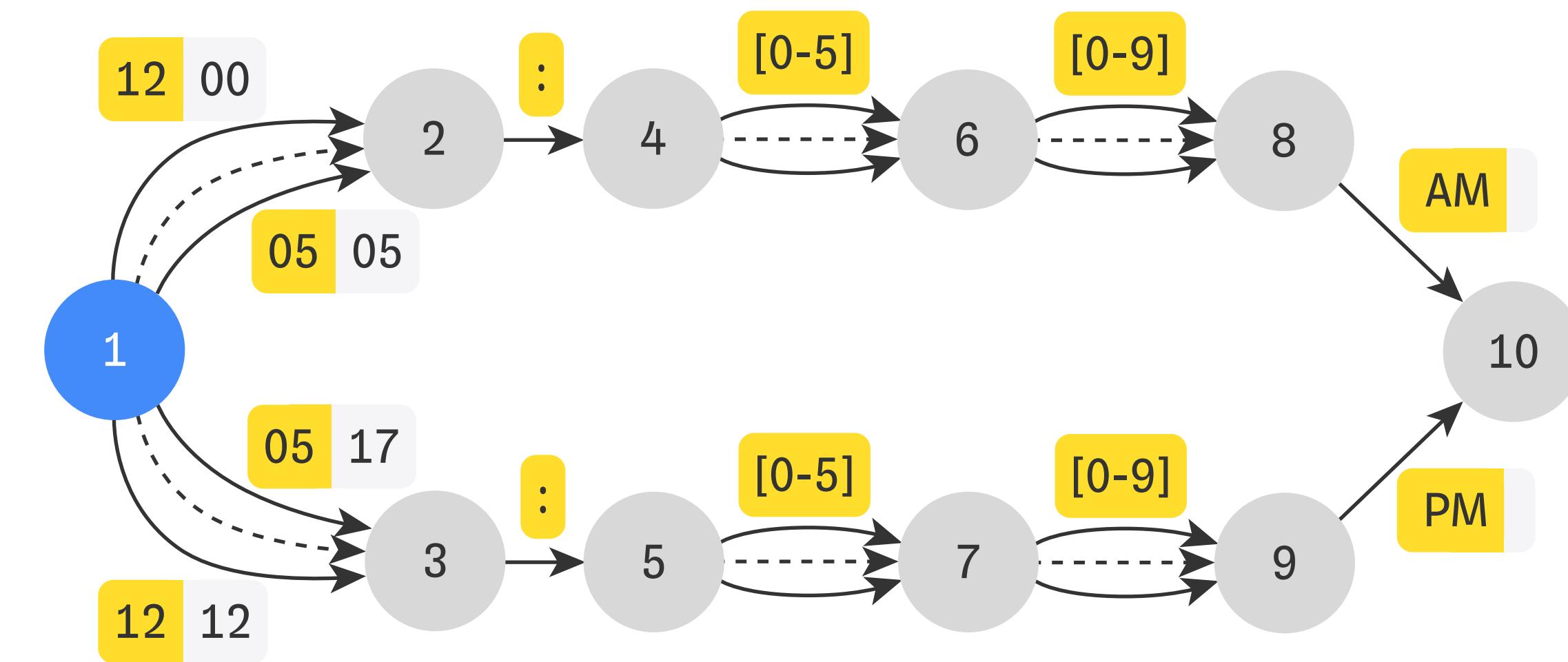
Входная строка

05:34 AM

Выходная строка

Практический пример #2

## Нормализация времени 2



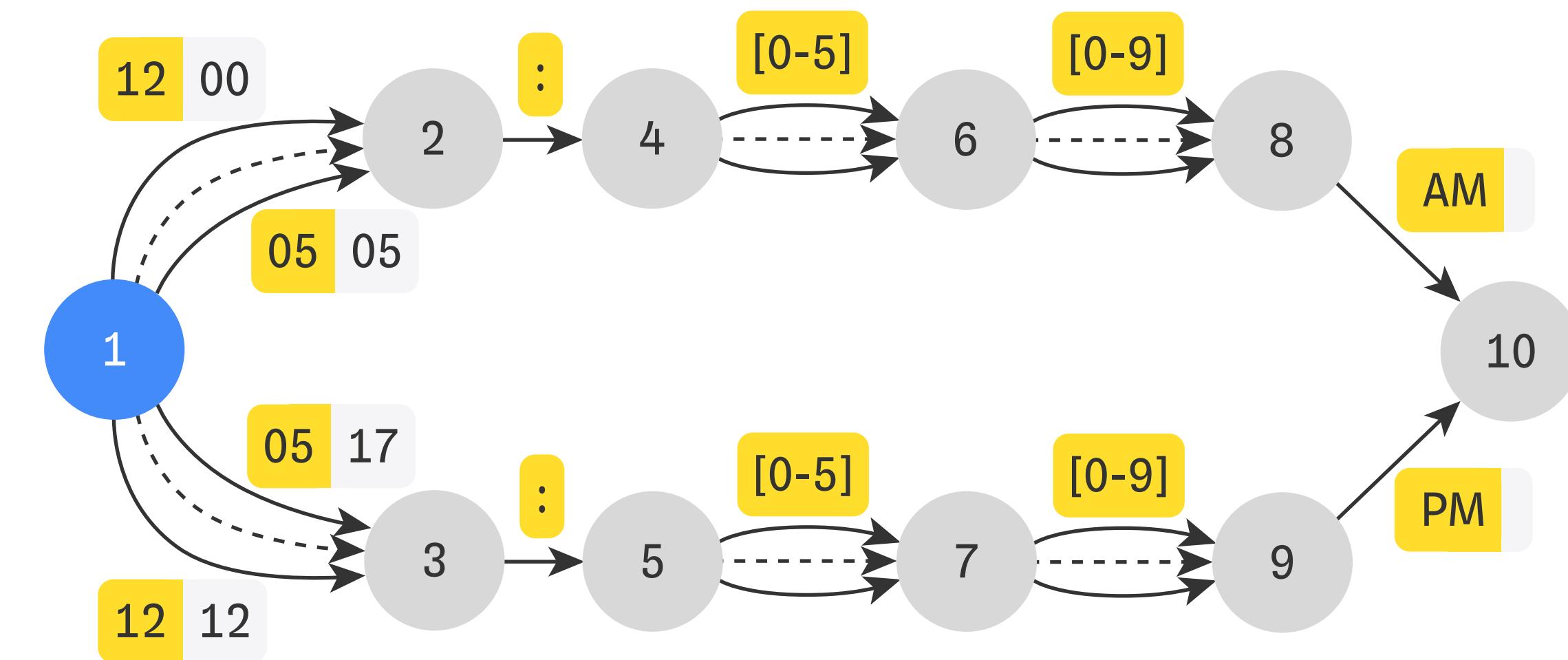
Входная строка

05:34 AM

Выходная строка

Практический пример #2

## Нормализация времени 2



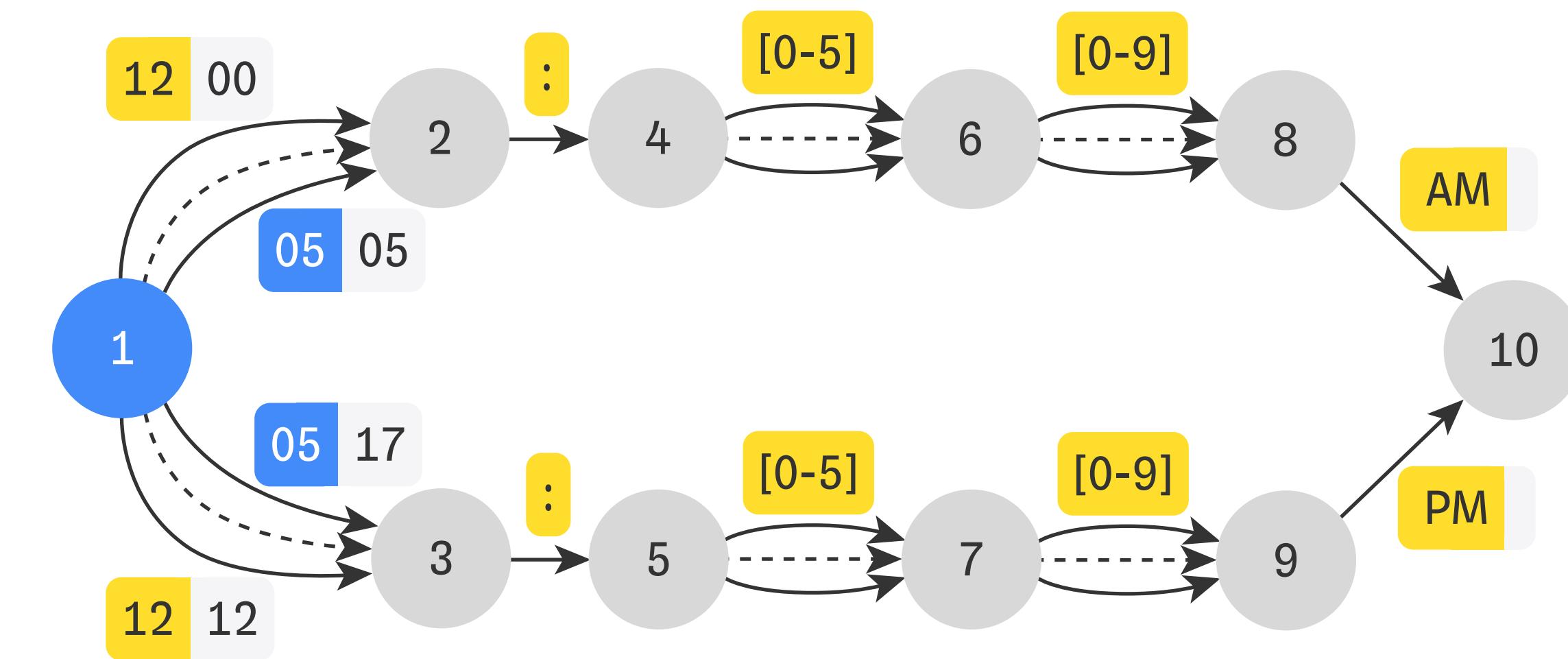
Входная строка

05:34 AM

Выходная строка

Практический пример #2

## Нормализация времени 2



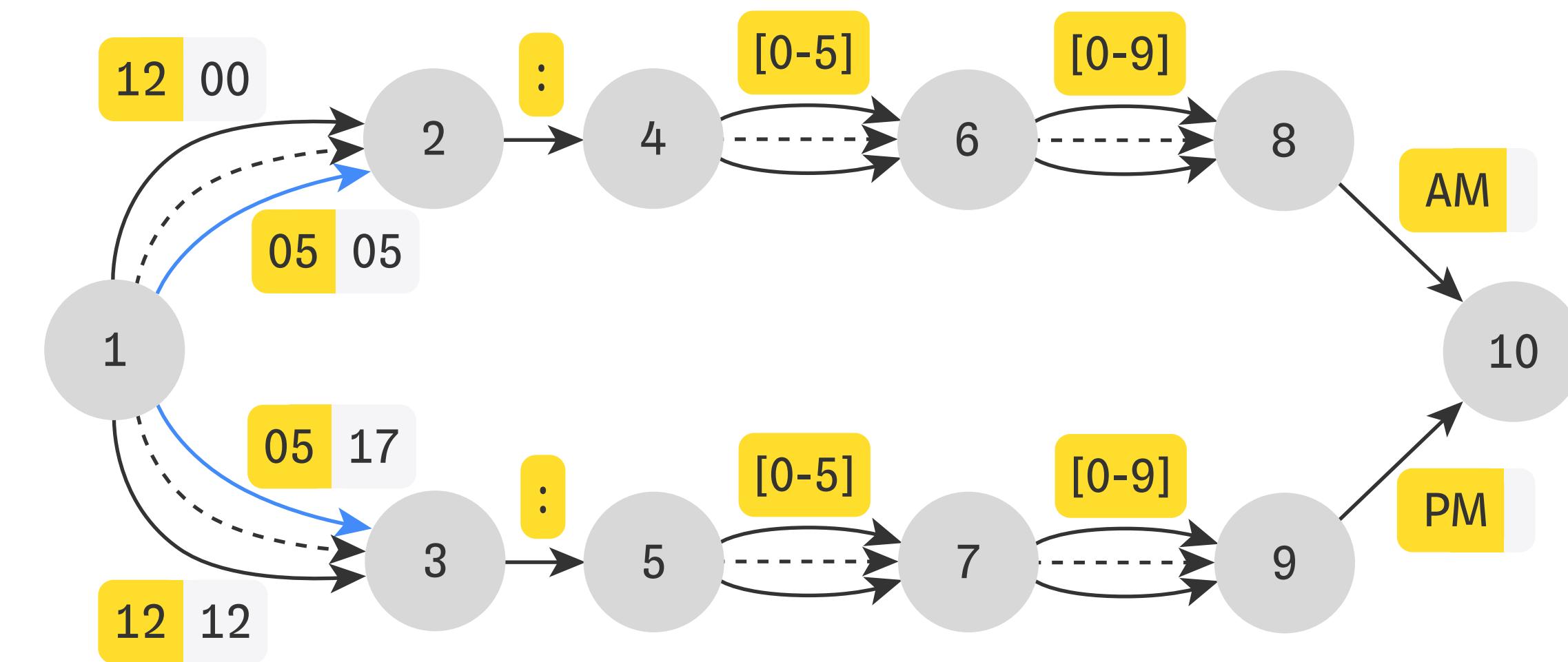
Входная строка

05:34 AM

Выходная строка

Практический пример #2

## Нормализация времени 2



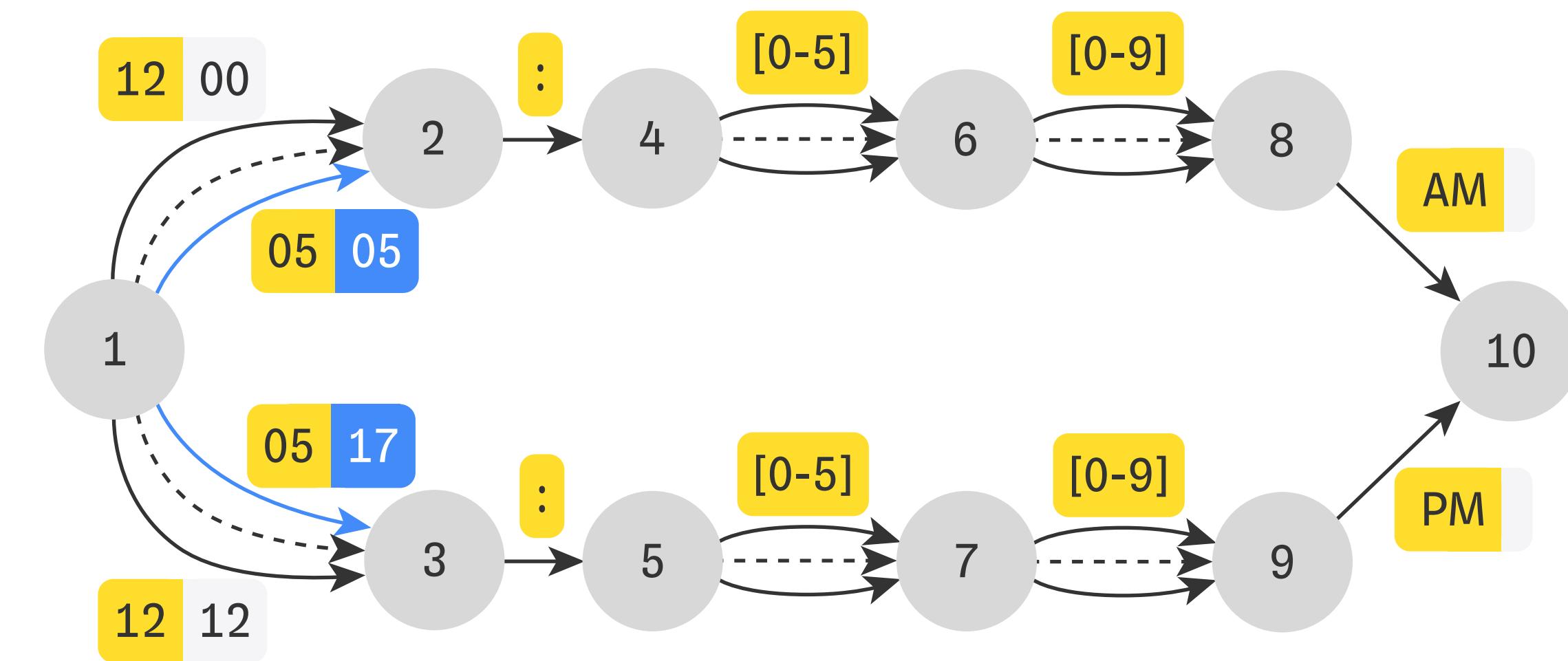
Входная строка

05:34 AM

Выходная строка

Практический пример #2

## Нормализация времени 2



Входная строка

05:34 AM

Выходная строка

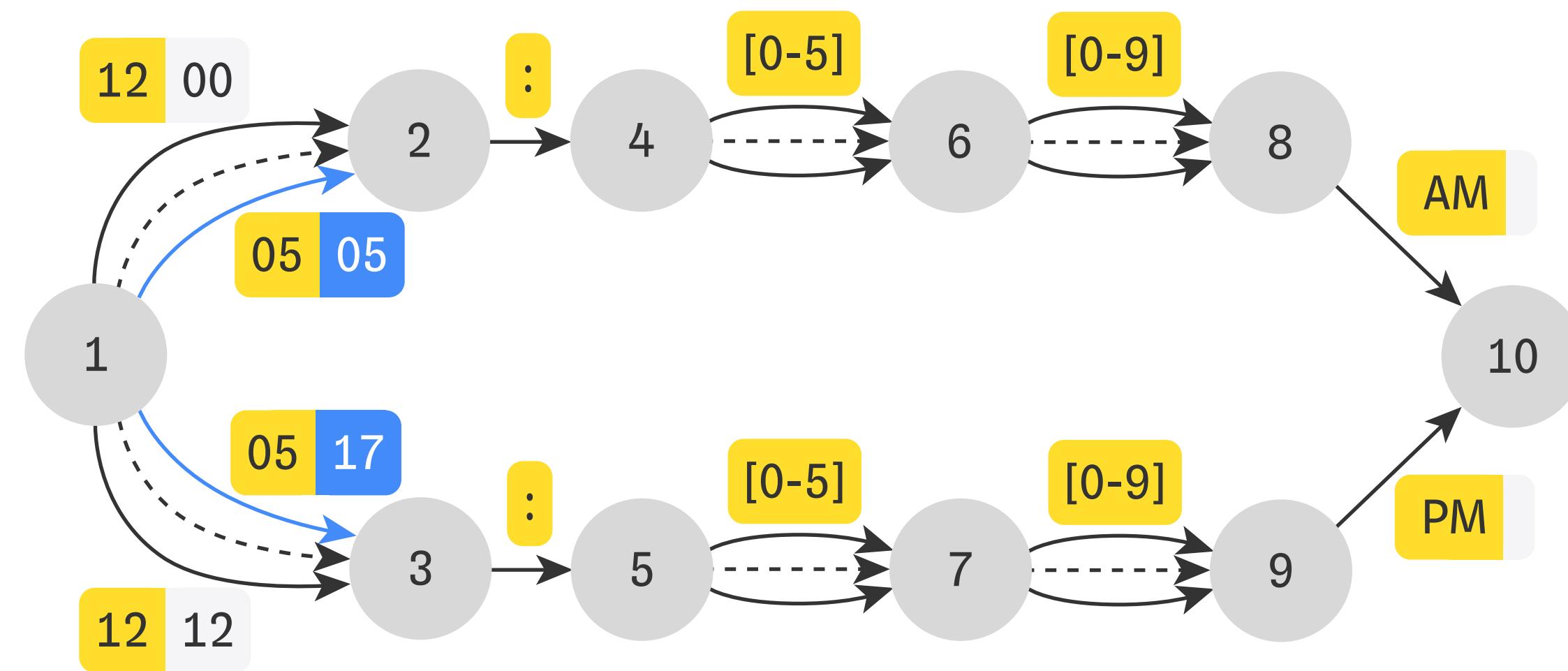
??

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

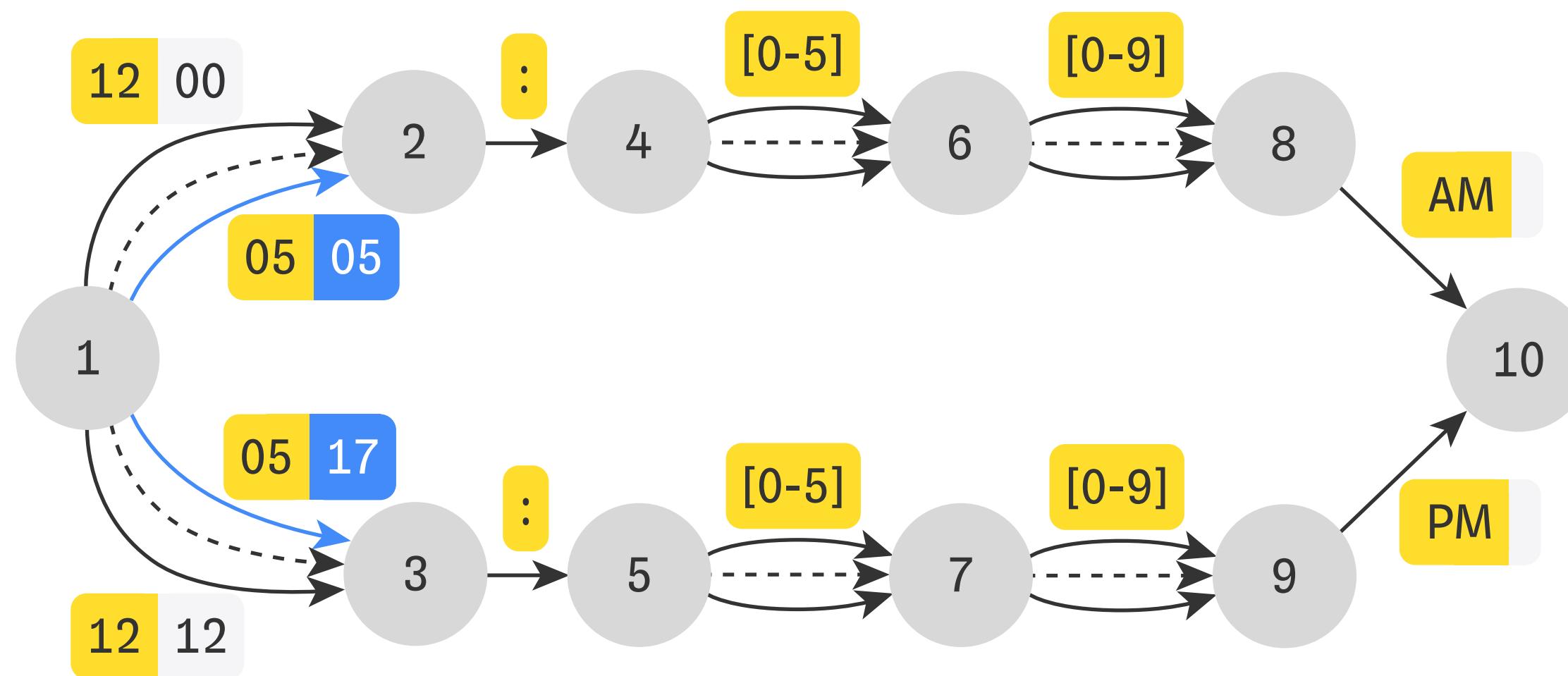
05 17

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05

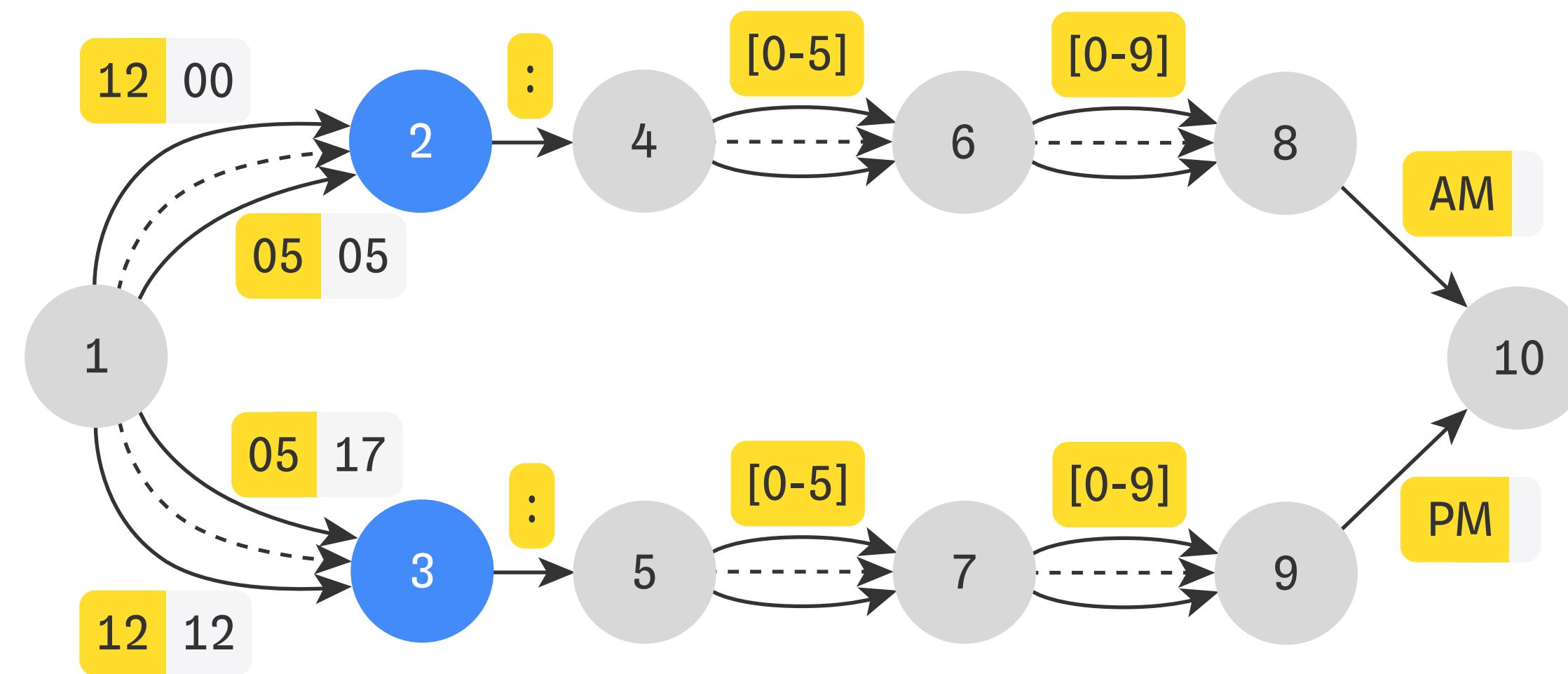
17

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05

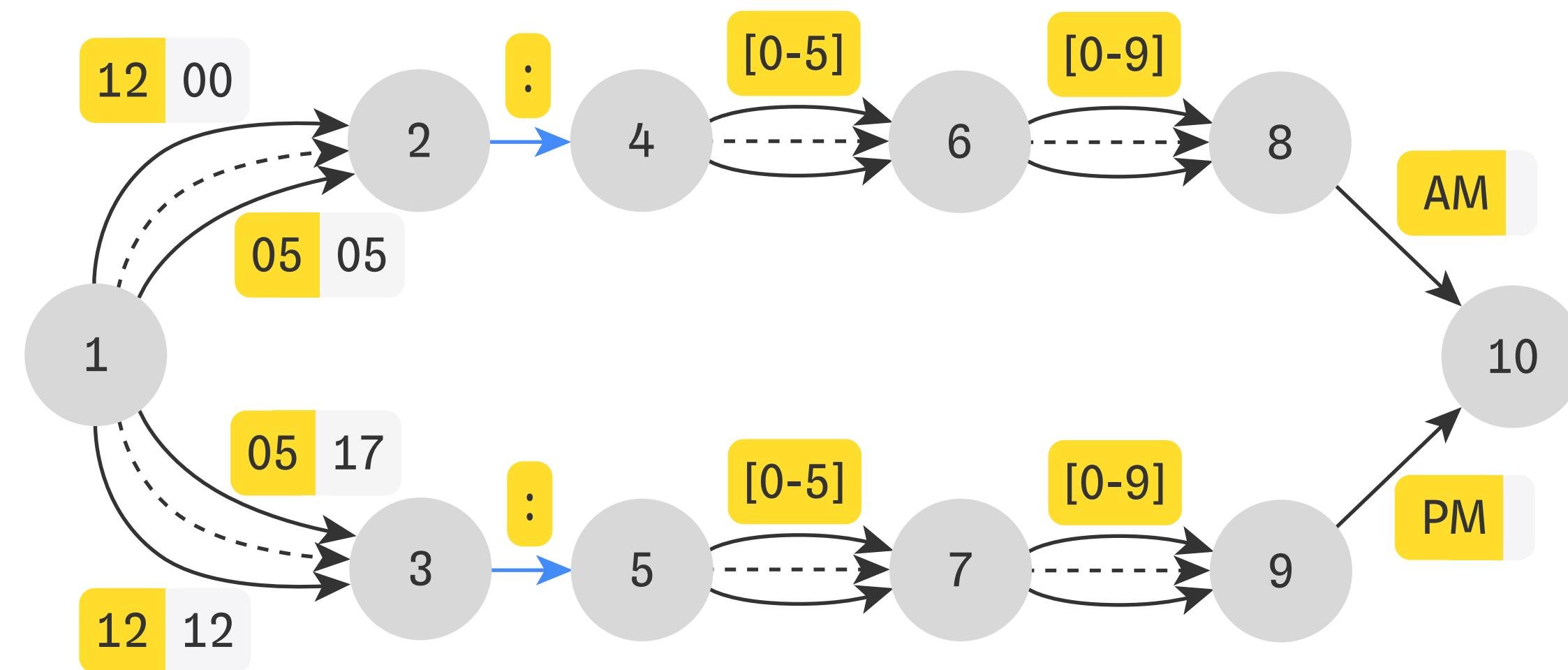
17

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05:

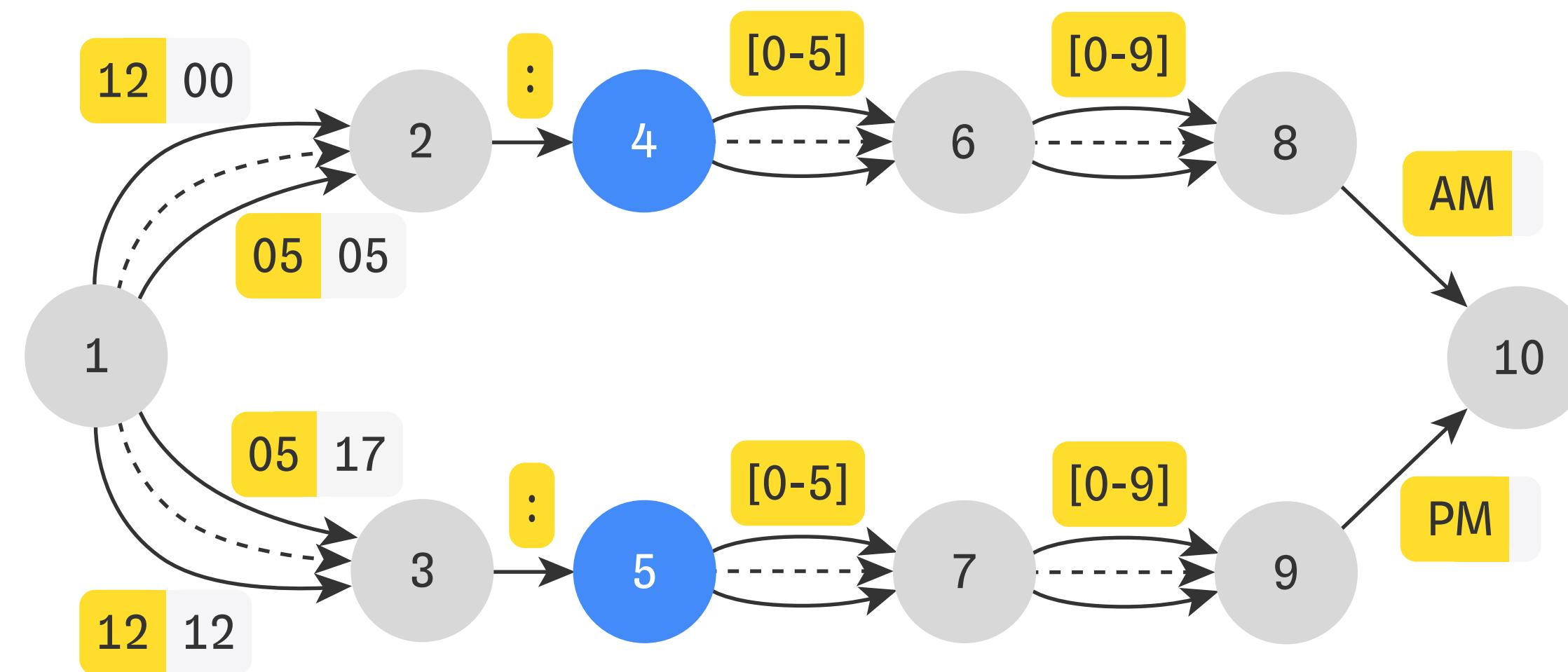
17:

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05:

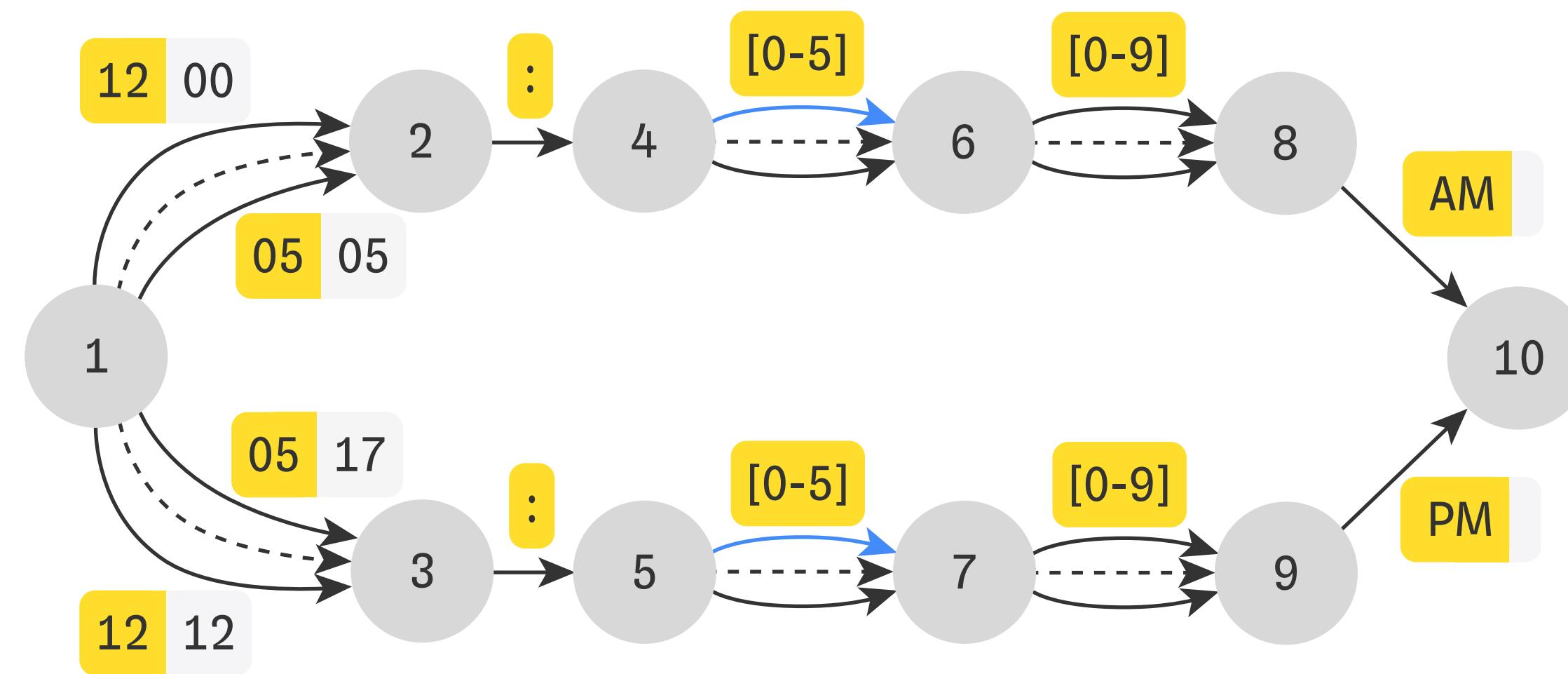
17:

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05:3

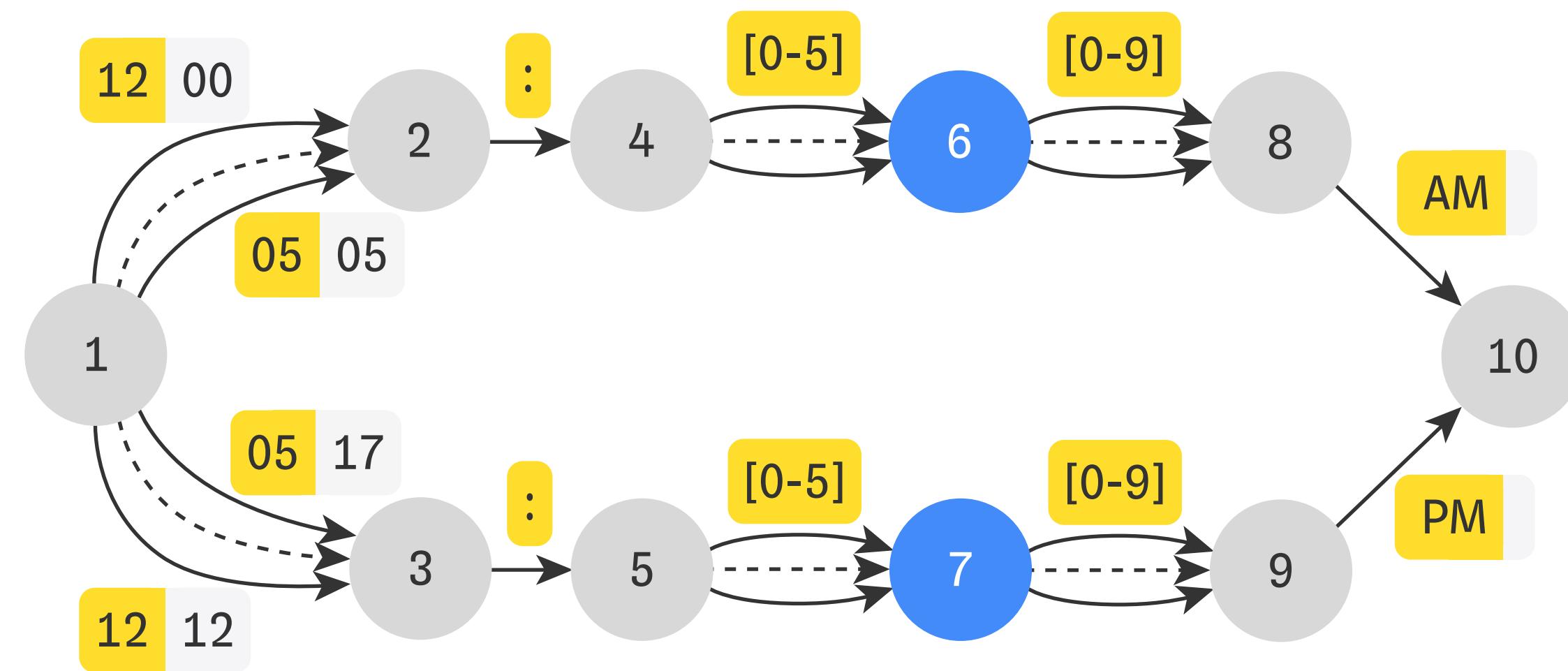
17:3

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05:3

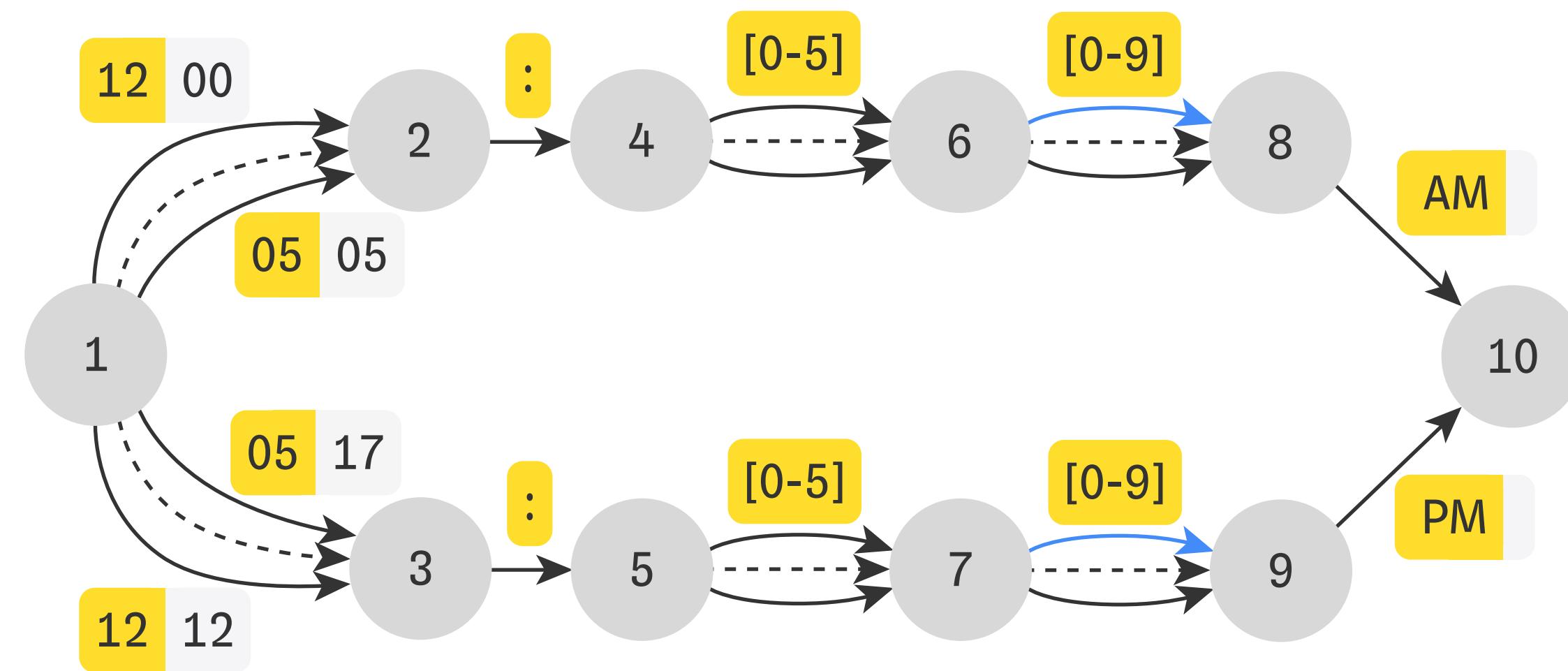
17:3

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:3<sub>4</sub> AM

Выходная строка

05:3<sub>4</sub>

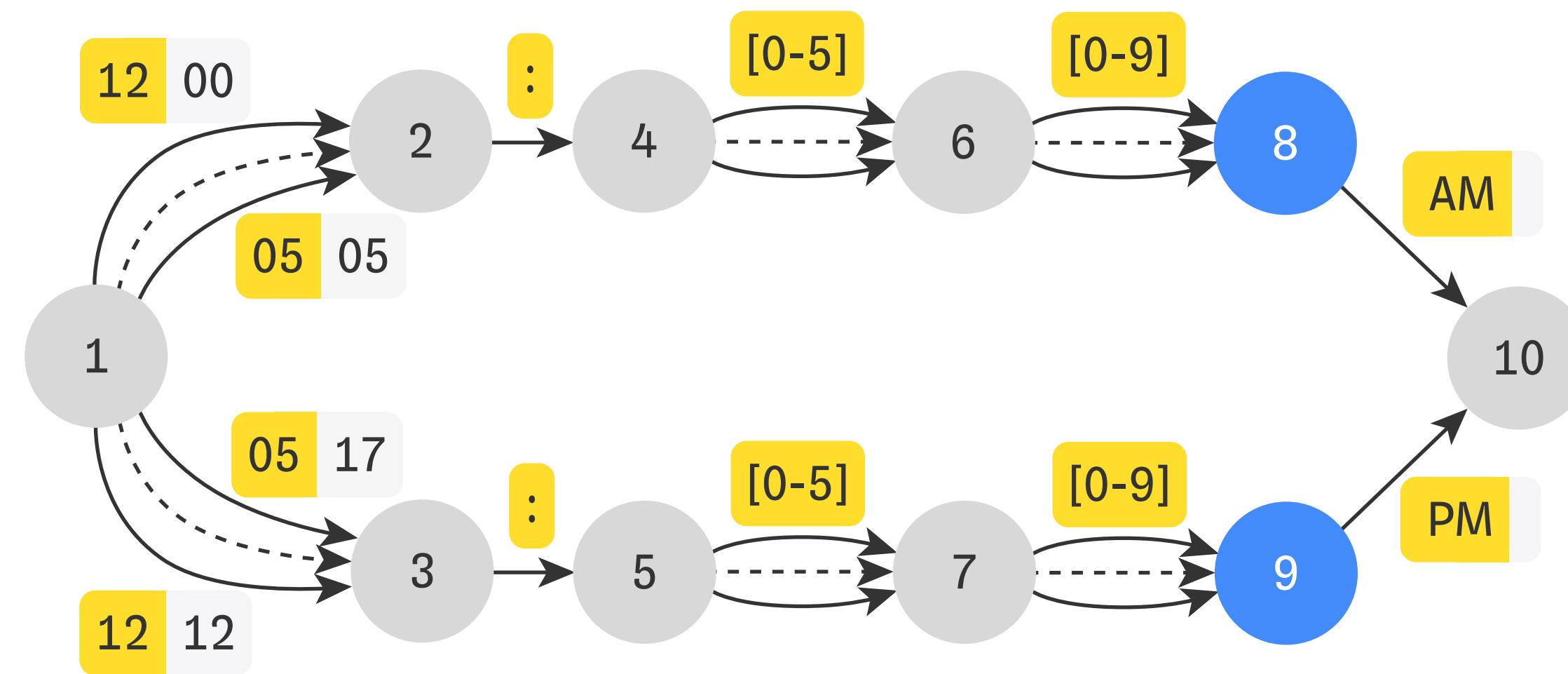
17:3<sub>4</sub>

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05:34

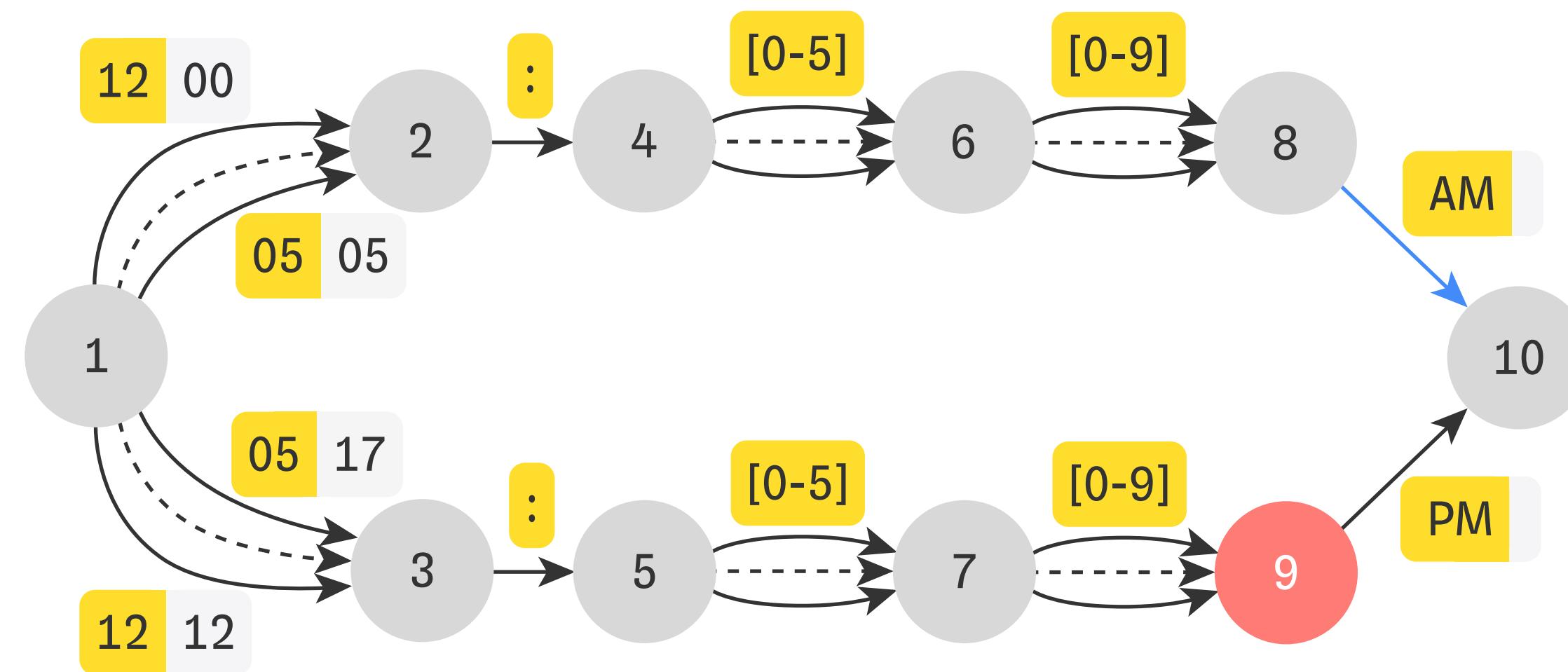
17:34

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

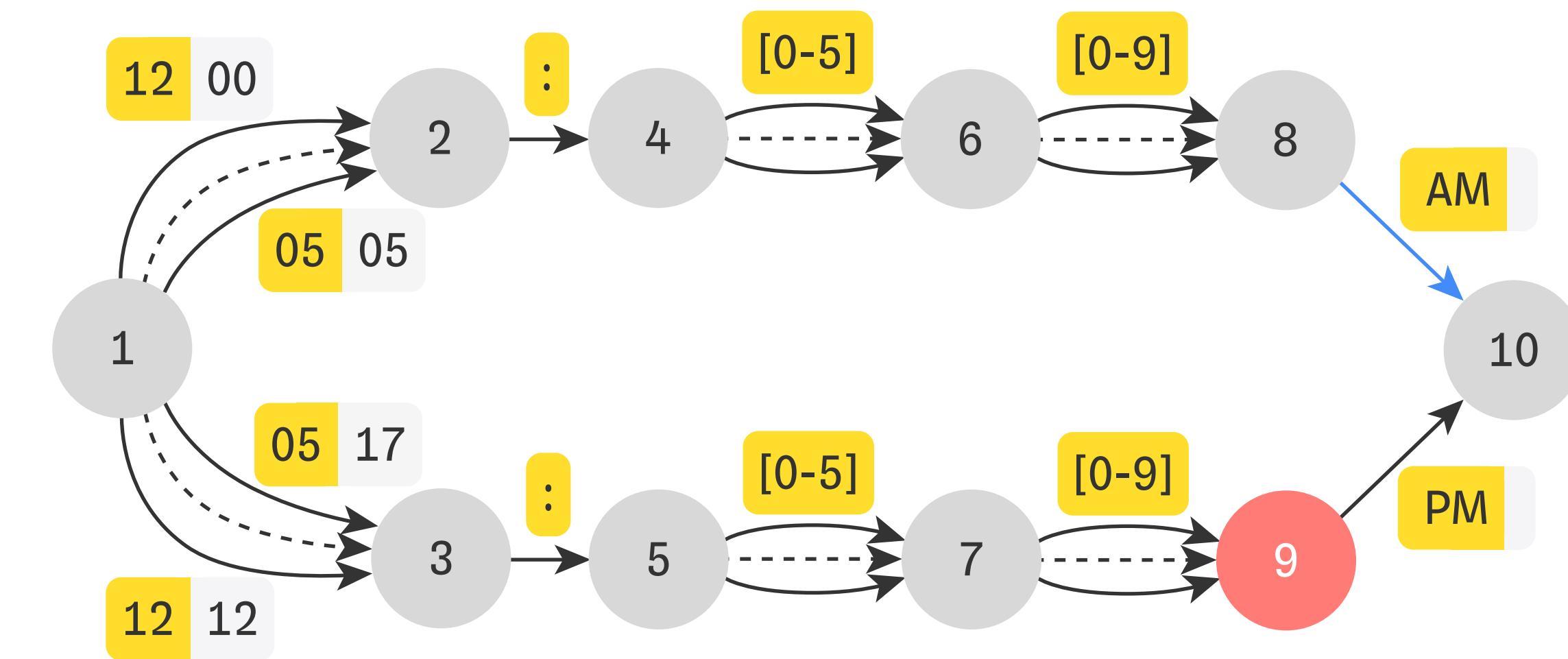
Выходная строка

05:34

17:34

Практический пример #2

## Нормализация времени 2



Входная строка

05:34 AM

Выходная строка

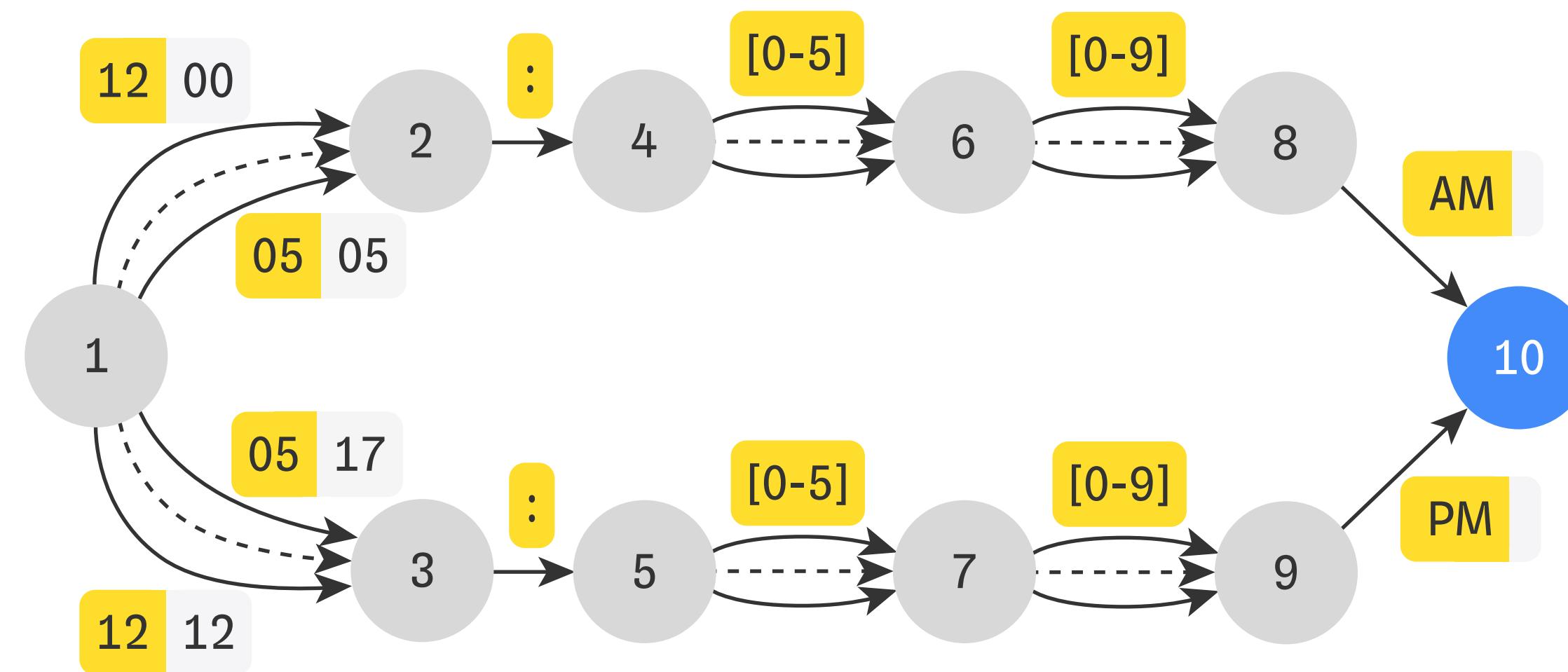
05:34

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

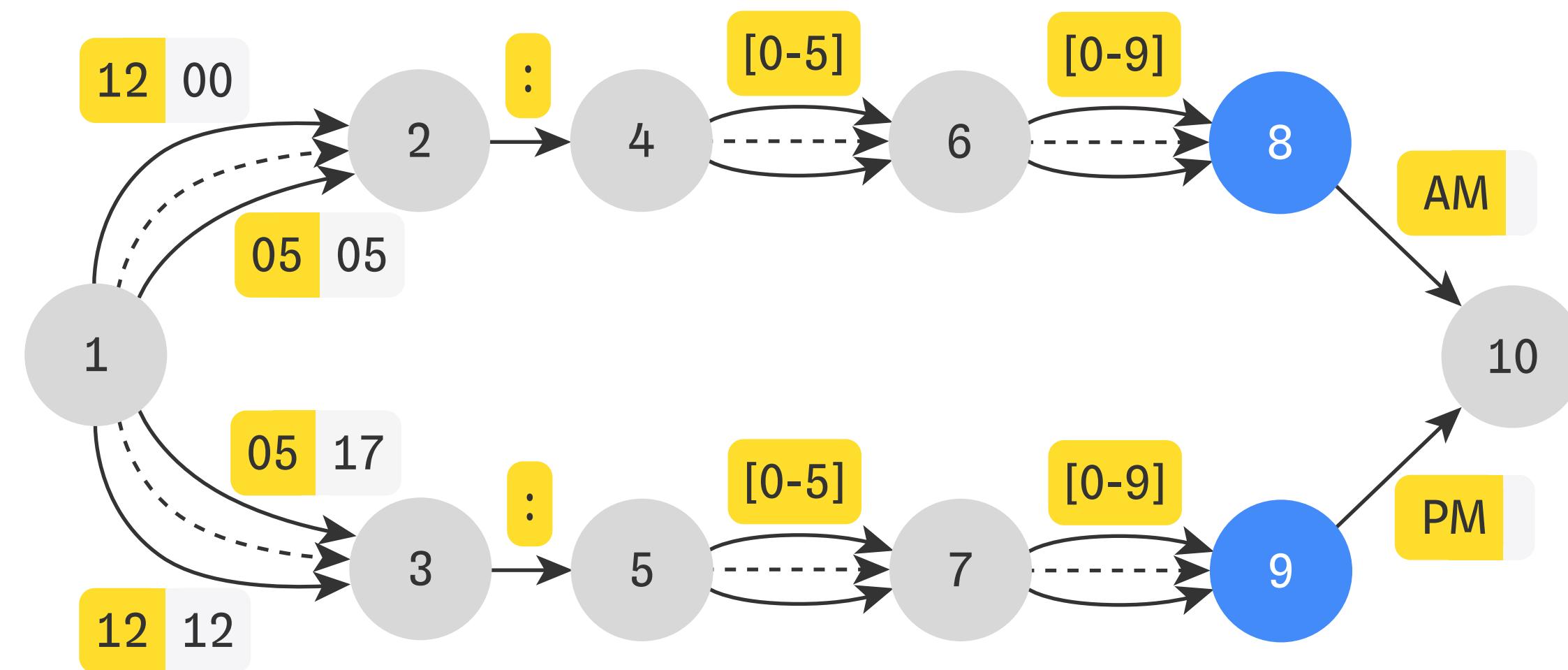
05:34

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 AM

Выходная строка

05:34

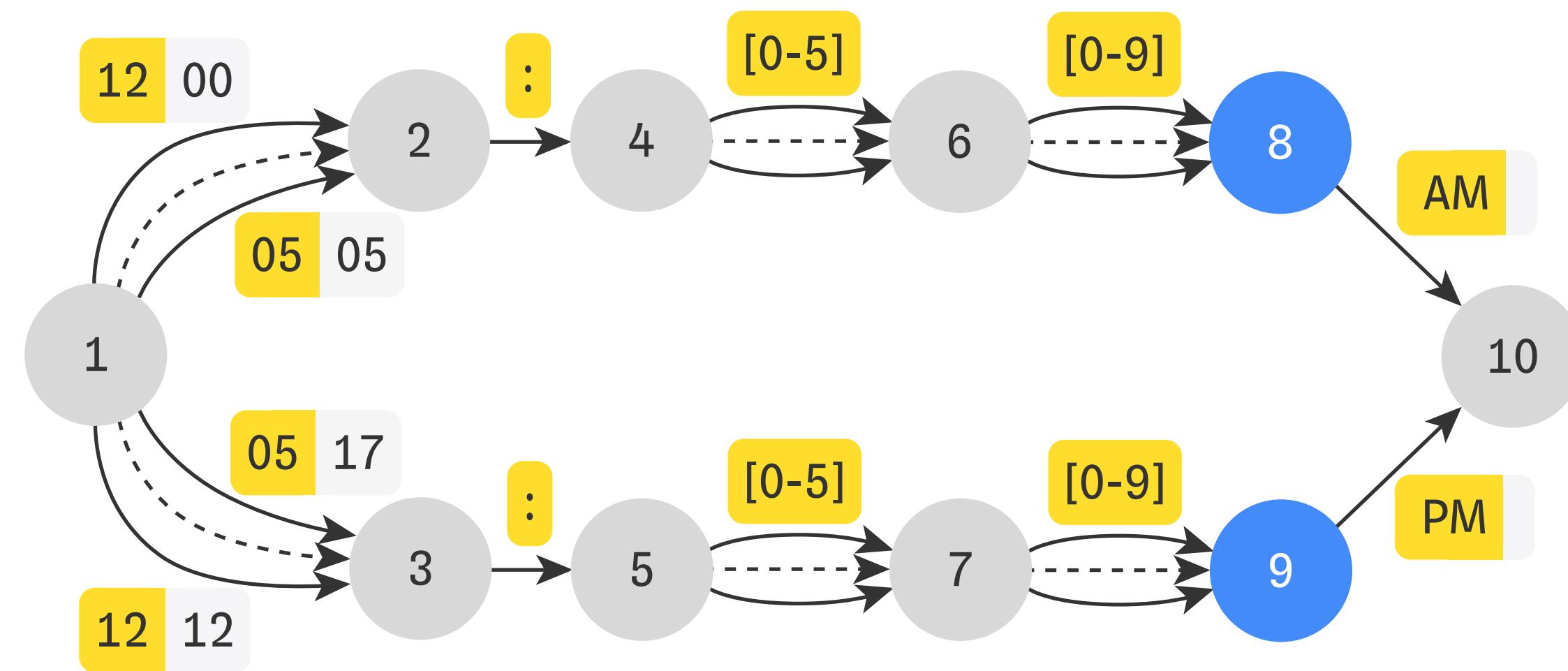
17:34

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 PM

Выходная строка

05:34

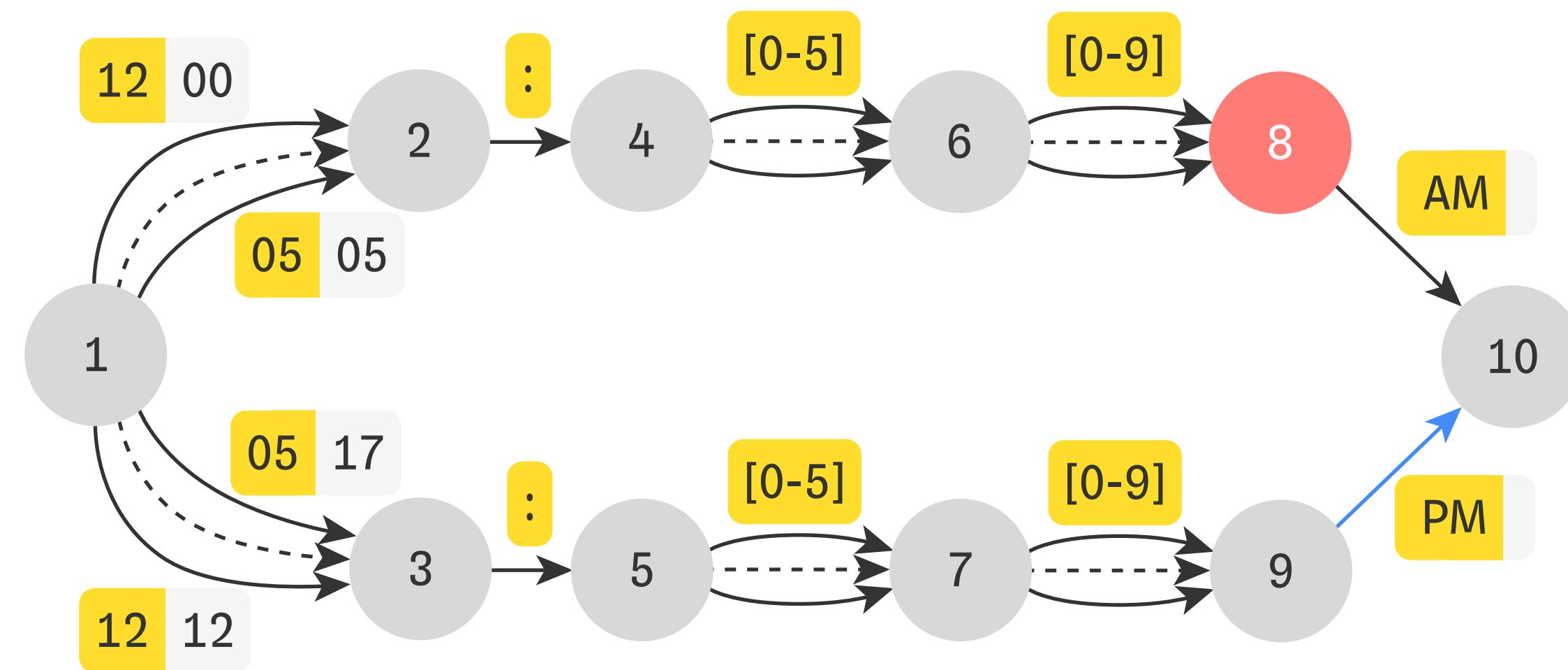
17:34

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 PM

Выходная строка

05:34

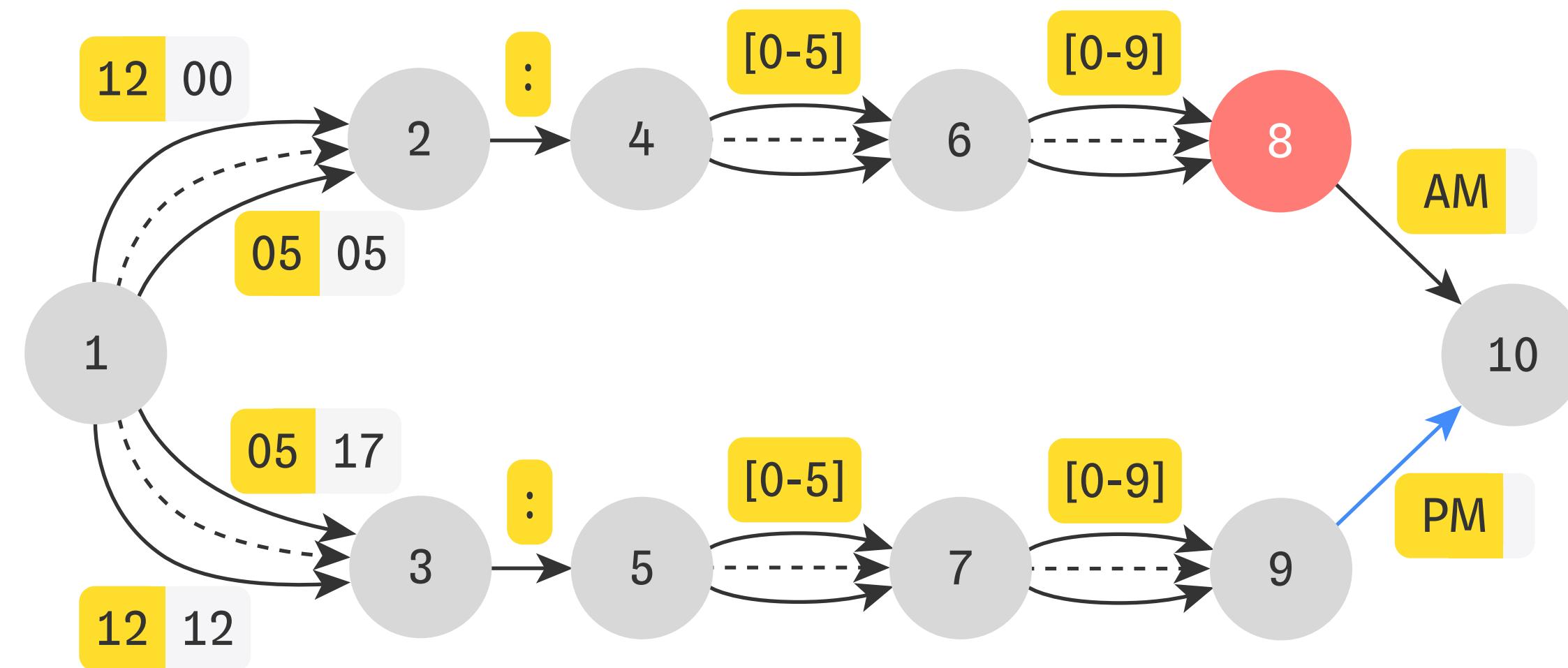
17:34

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 PM

Выходная строка

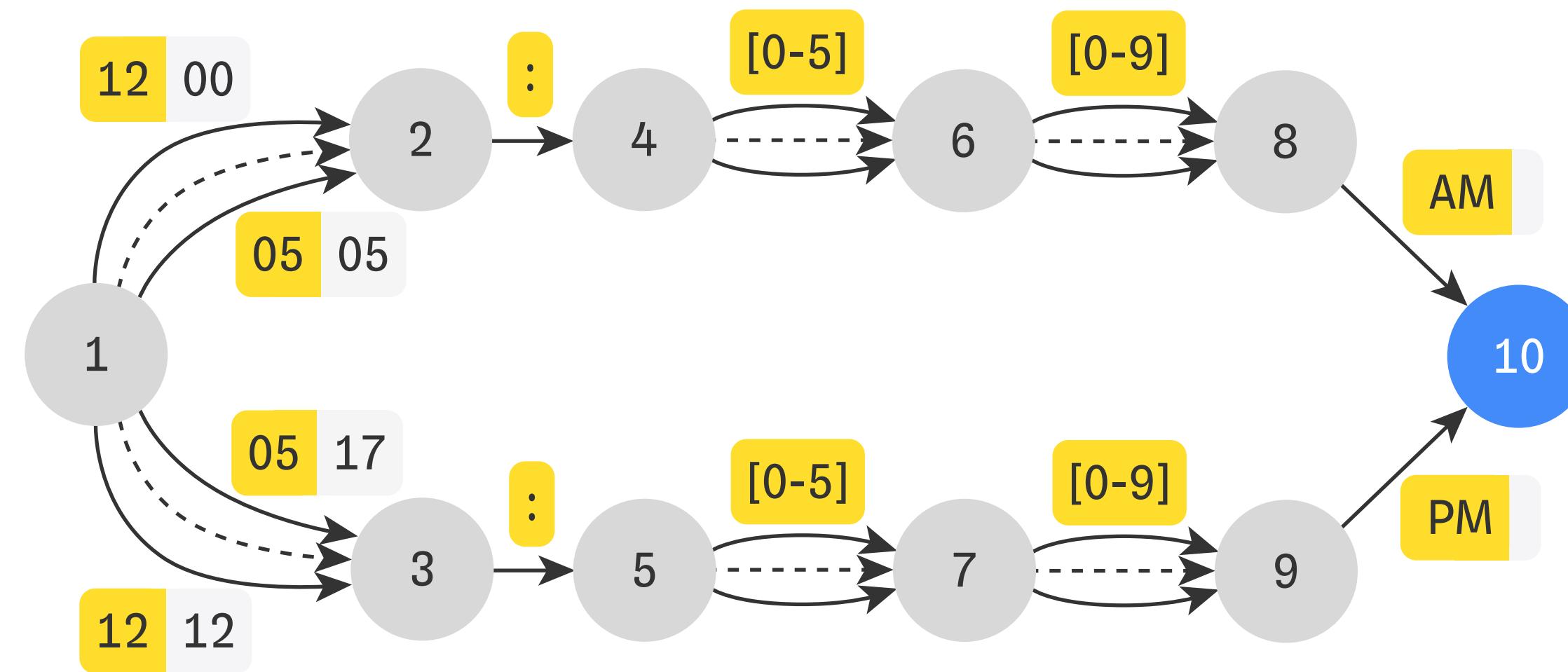
17:34

Практический пример #2

## Нормализация времени 2



Google Colab



Входная строка

05:34 PM

Выходная строка

17:34

Практический пример #3

# Нормализация времени final

## Задача

Создать универсальный FST для нормализации времени в любом месте предложения в любом формате

Например:

В 01:30 AM  $\Rightarrow$  В один час тридцать минут

Завтра в 03:00 в парке  $\Rightarrow$  Завтра в три часа в парке

Агент 02 ждет в 02:00  $\Rightarrow$  Агент 02 ждет в два часа

Практический пример #3

# Нормализация времени final

## Задача

Создать универсальный FST для нормализации времени в любом месте предложения в любом формате

Например:

В 01:30 AM  $\Rightarrow$  В один час тридцать минут

Завтра в 03:00 в парке  $\Rightarrow$  Завтра в три часа в парке

Агент 02 ждет в 02:00  $\Rightarrow$  Агент 02 ждет в два часа

Подзадачи:

1. Построить FST для нормализации времени в любой части текста
2. Построить FST для преобразования из 12 в 24 формат в любой части текста
3. Последовательно соединить оба FST

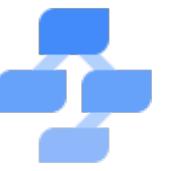
## Практический пример #3

# Нормализация времени final

1. Построить FST для нормализации времени в любой части текста



Google Colab

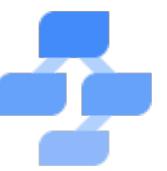


## CDRewrite – инструмент для перезаписи

- Создает контекстно-зависимые FST для правил перезаписи
- Принимает FST для перезаписи, правый и левый контекст, а также sigma star для движения по исходному тексту

## Нормализация времени final

1. Построить FST для нормализации времени в любой части текста



### CDRewrite – инструмент для перезаписи

- Создает контекстно-зависимые FST для правил перезаписи
- Принимает FST для перезаписи, правый и левый контекст, а также sigma star для движения по исходному тексту



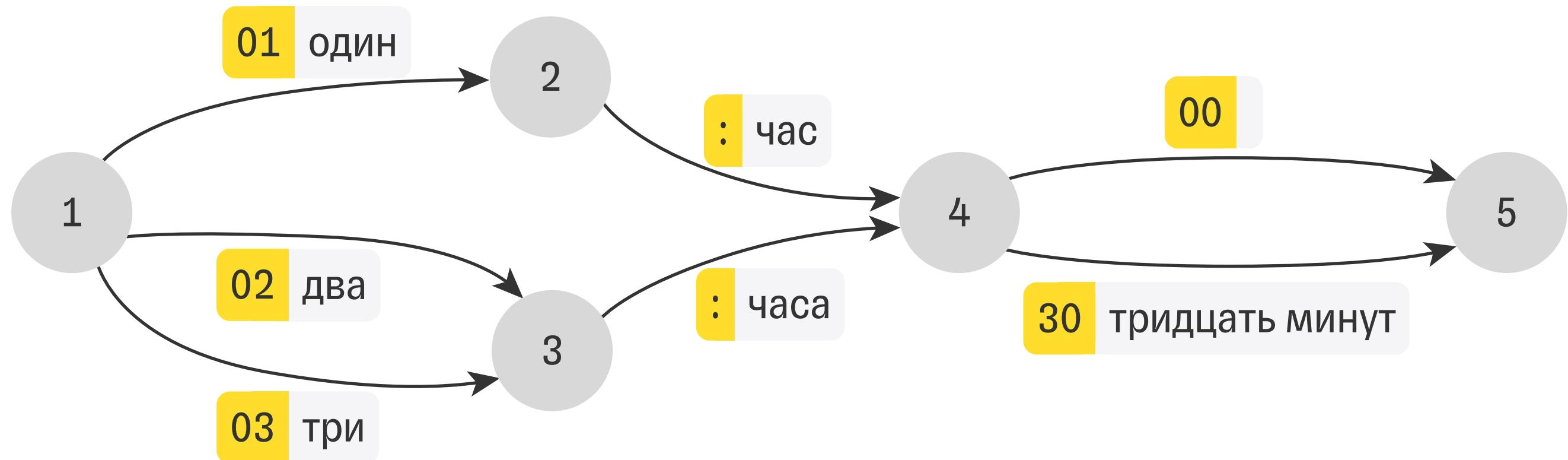
Google Colab

```
fst = ... # Построение FST нормализации чисел  
  
star = pynini.union(*"абвг...АБВГ...1234...!"#$...").closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

### Практический пример #3

## Нормализация времени final

1. Построить FST для нормализации времени в любой части текста



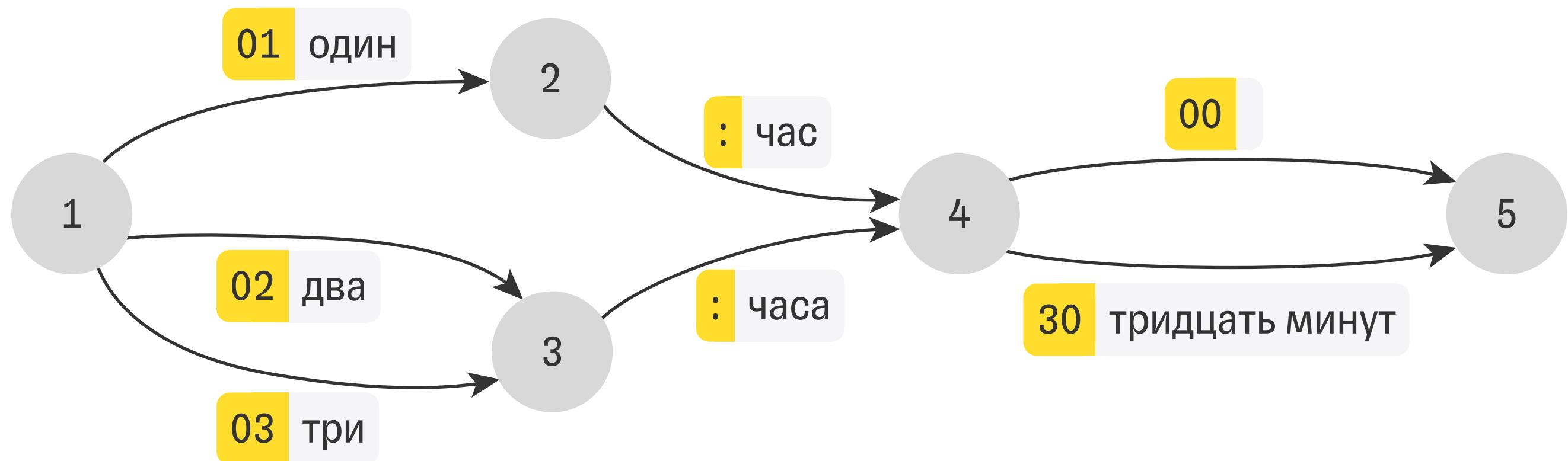
Google Colab

```
fst = ... # Построение FST нормализации чисел  
star = pynini.union(*"абвг...АБВГ...1234...!"#$...").closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

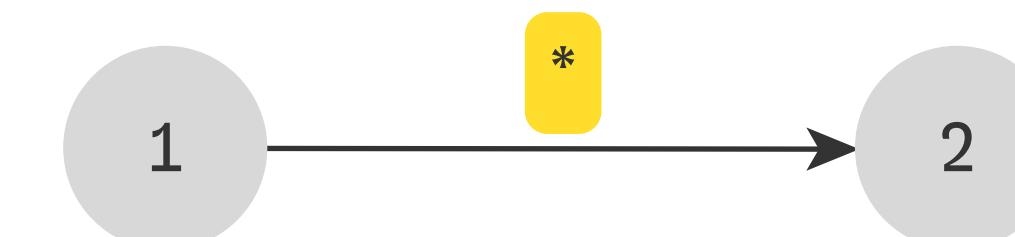
### Практический пример #3

## Нормализация времени final

1. Построить FST для нормализации времени в любой части текста



Google Colab

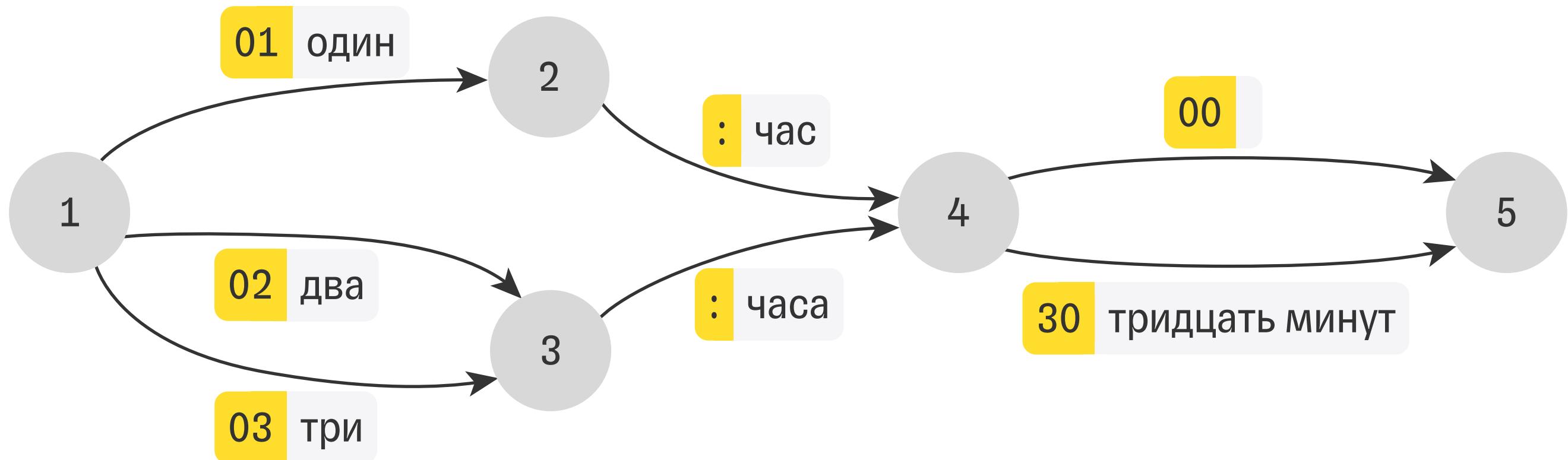


```
fst = ... # Построение FST нормализации чисел  
star = pynini.union(*"абвг...АБВГ...1234...!"#$...").closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

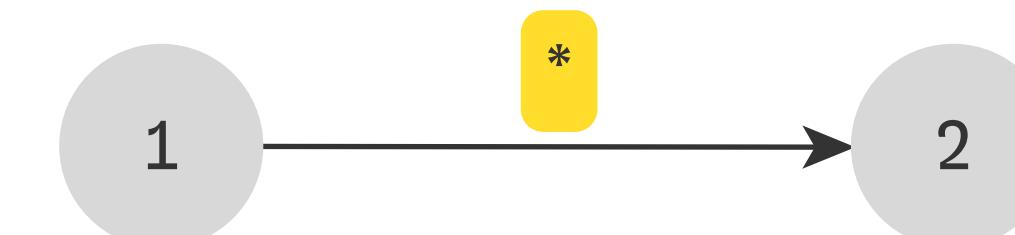
### Практический пример #3

## Нормализация времени final

1. Построить FST для нормализации времени в любой части текста



Google Colab



```
fst = ... # Построение FST нормализации чисел  
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

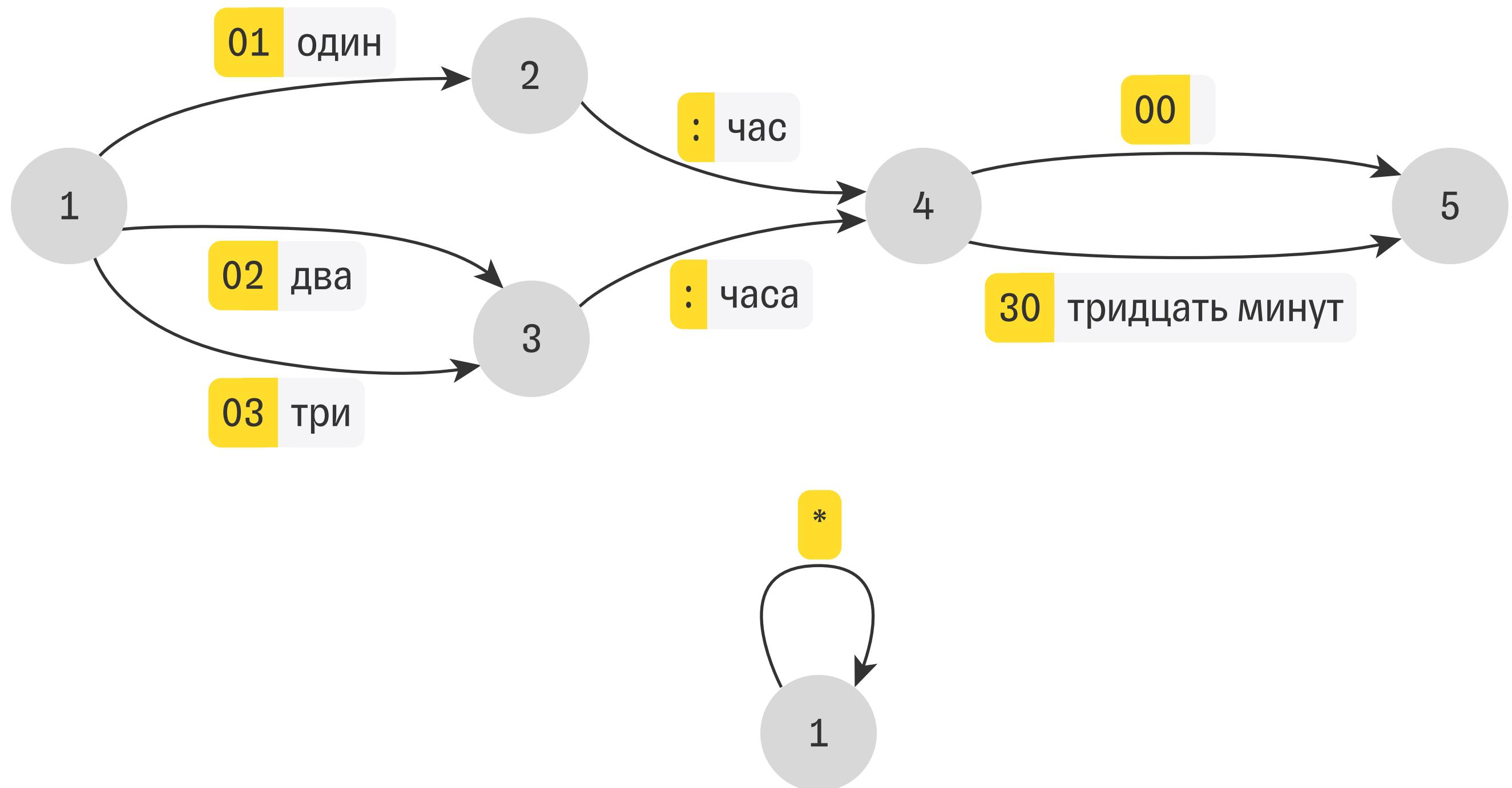
### Практический пример #3

## Нормализация времени final

1. Построить FST для нормализации времени в любой части текста



Google Colab



```
fst = ... # Построение FST нормализации чисел
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

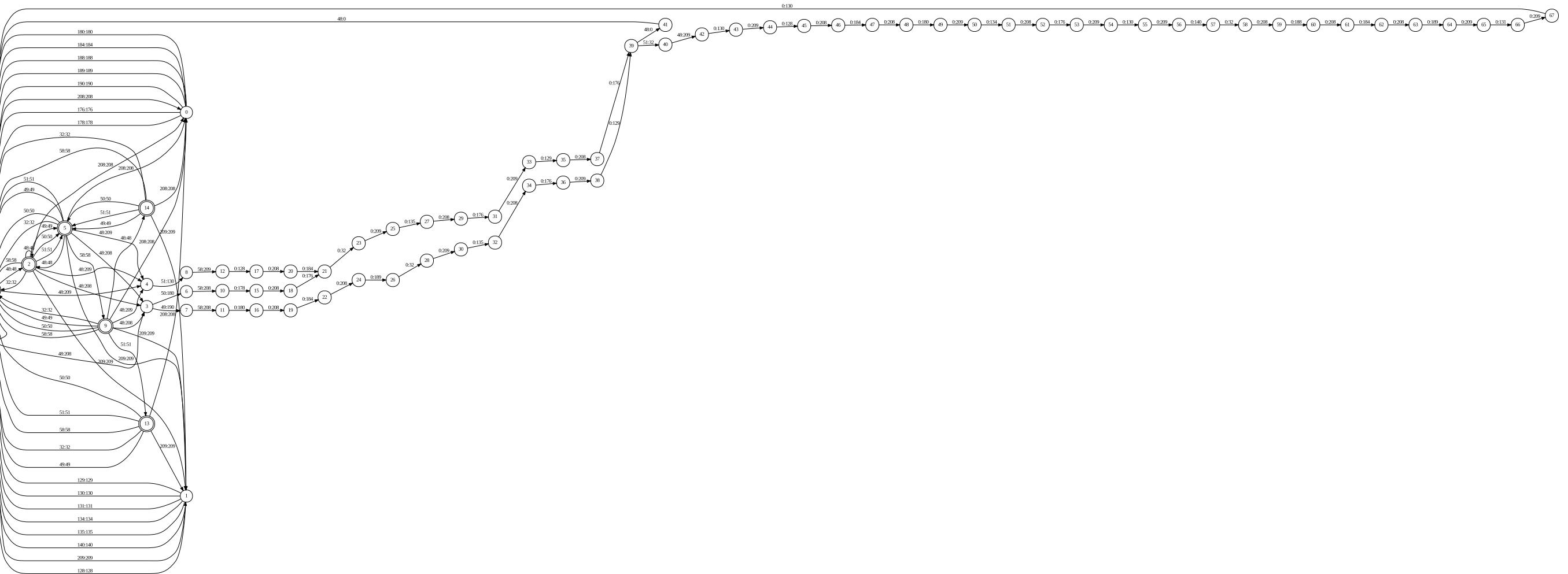
Практический пример #3

## Нормализация времени final

1. Построить FST для нормализации времени в любой части текста



Google Colab



```
fst = ... # Построение FST нормализации чисел
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

## Практический пример #3

# Нормализация времени final

2. Построить FST для преобразования из 12 в 24 формат в любой части текста



Google Colab

```
fst = ... # Построение FST конвертации времени

star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

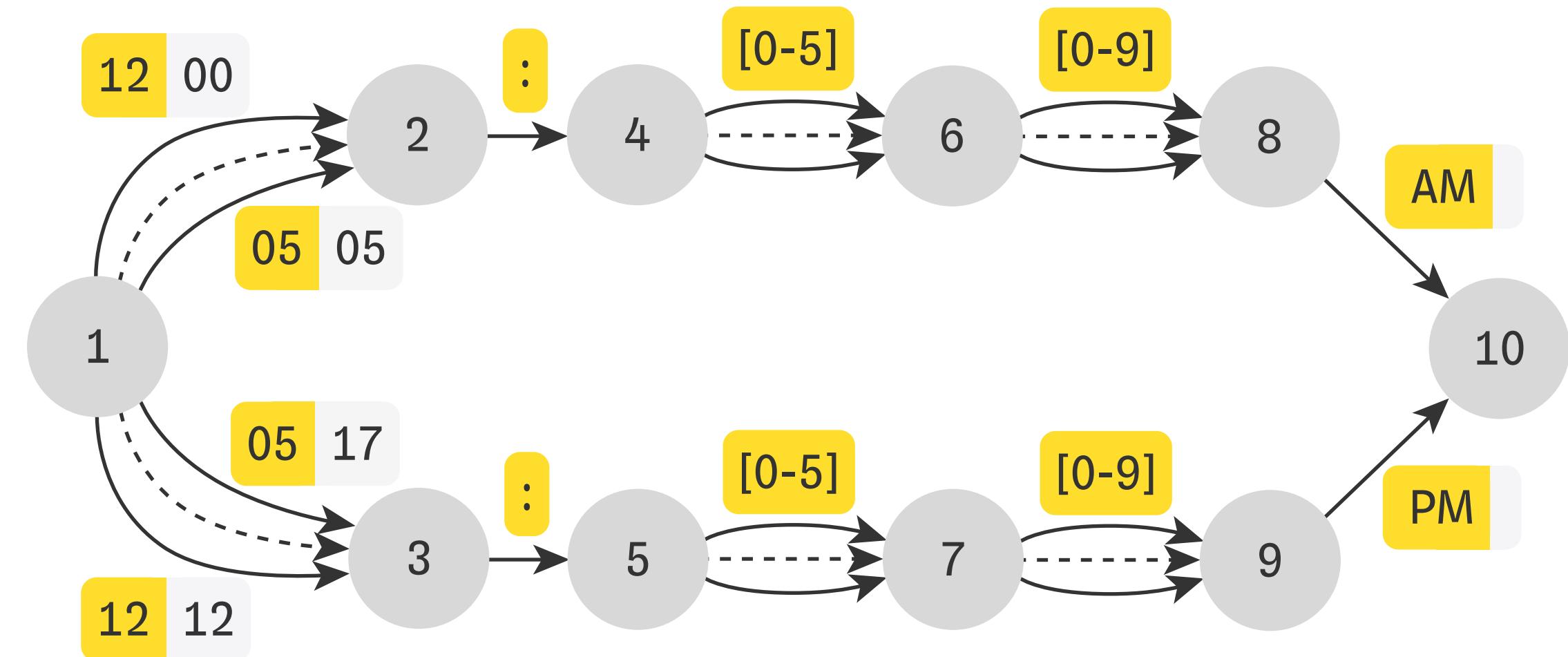
Практический пример #3

## Нормализация времени final

2. Построить FST для преобразования из 12 в 24 формат в любой части текста



Google Colab



```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

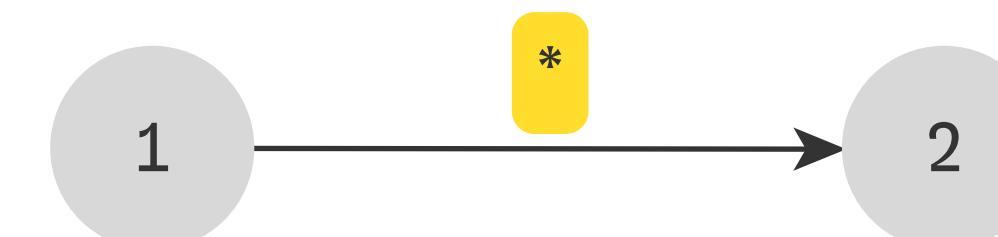
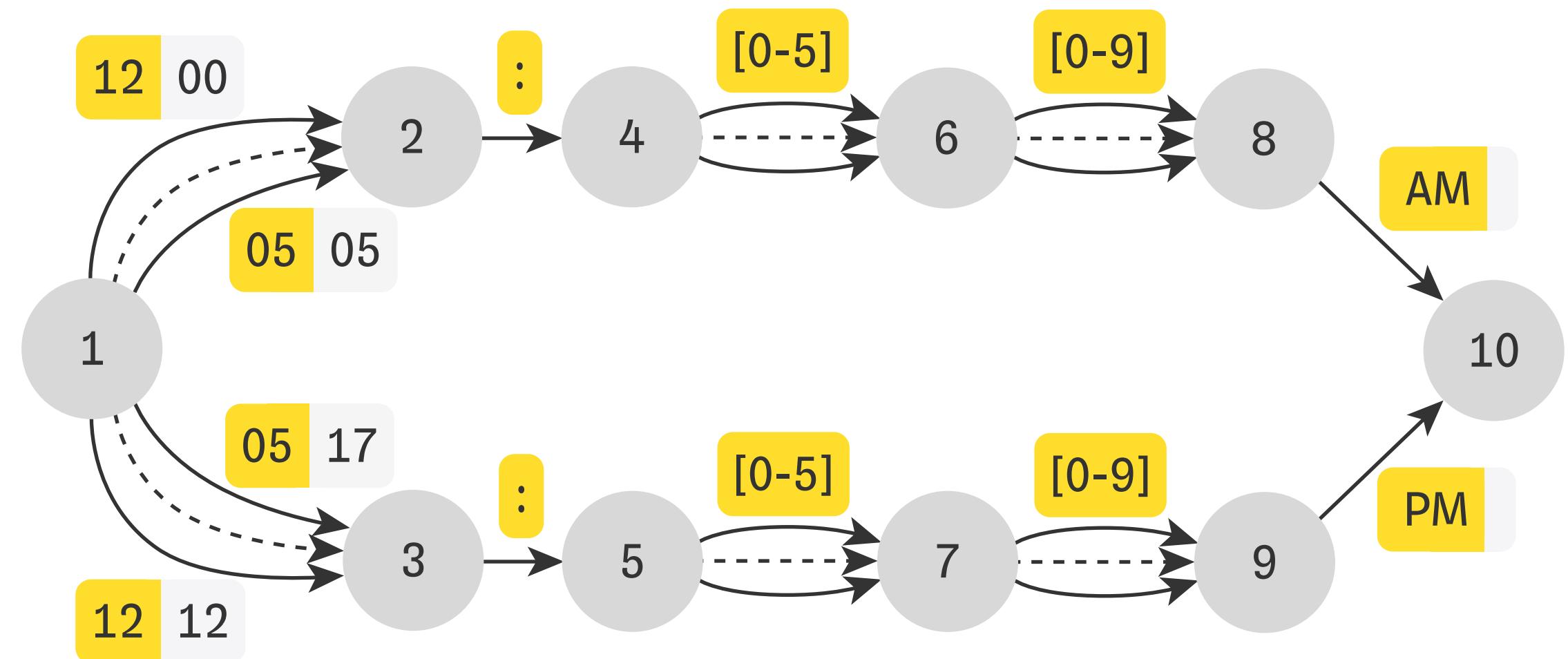
Практический пример #3

## Нормализация времени final

2. Построить FST для преобразования из 12 в 24 формат в любой части текста



Google Colab



```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

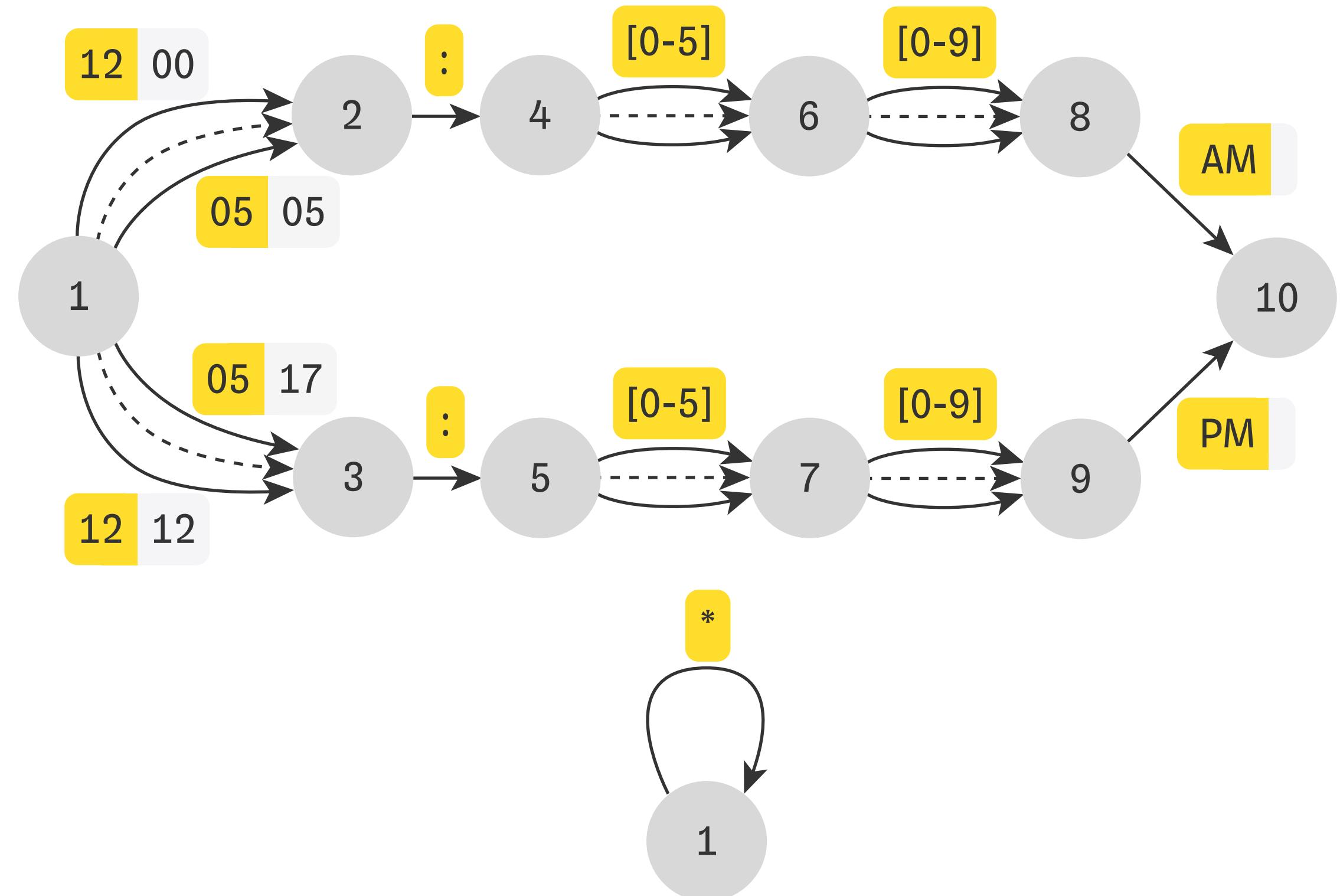
Практический пример #3

## Нормализация времени final

2. Построить FST для преобразования из 12 в 24 формат в любой части текста



Google Colab



```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

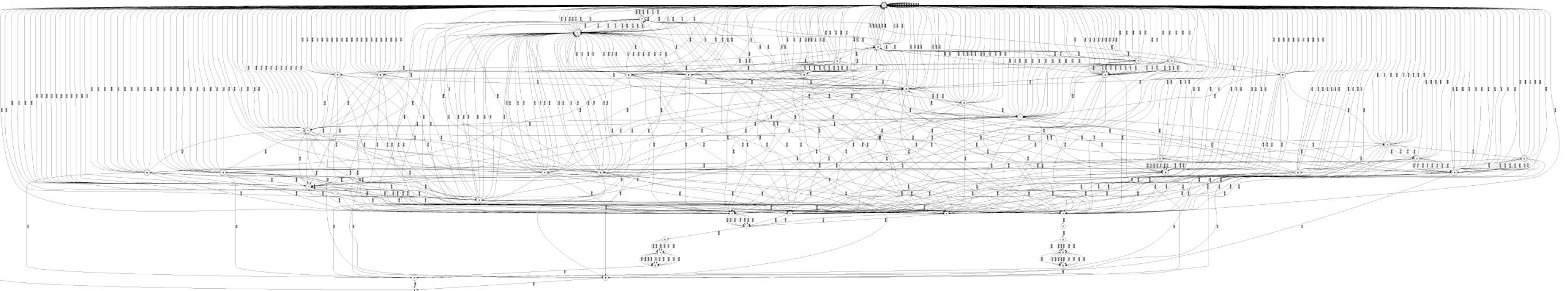
## Практический пример #3

# Нормализация времени final

2. Построить FST для преобразования из 12 в 24 формат в любой части текста



Google Colab



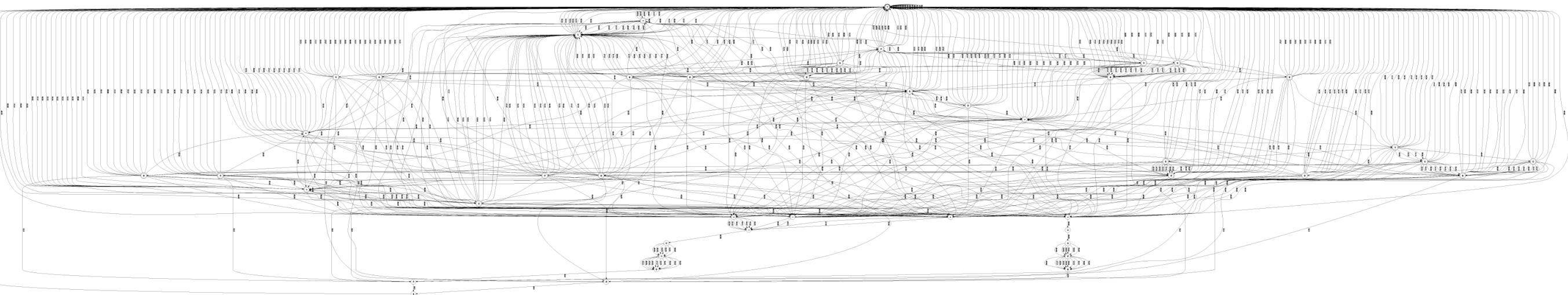
```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()  
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

## Практический пример #3

# Нормализация времени final

2. Построить FST для преобразования из 12 в 24 формат в любой части текста



46 вершин и ~8к ребер!



Google Colab

```
fst = ... # Построение FST конвертации времени

star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()
cdr = pynini.cdrewrite(fst, "", "", star).optimize()
```

Практический пример #3

## Нормализация времени final

3. Последовательно соединить оба FST



Google Colab



### Compose – инструмент для объединения FST

- Объединяет FST так, как будто они выполняются последовательно друг за другом
- Принимает два FST

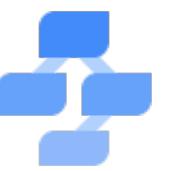
## Практический пример #3

# Нормализация времени final

3. Последовательно соединить оба FST



Google Colab



## Compose – инструмент для объединения FST

- Объединяет FST так, как будто они выполняются последовательно друг за другом
- Принимает два FST

```
normalize_time_cdr = ... # Построение FST конвертации времени  
convert_time_cdr = ... # Построение FST нормализации чисел  
  
fst = pynini.compose(convert_time_cdr, normalize_time_cdr)
```

Практический пример #3

## Нормализация времени final

3. Последовательно соединить оба FST



Google Colab

### Входная строка

В 01:30 AM в парке

```
normalize_time_cdr = ... # Построение FST конвертации времени  
convert_time_cdr = ... # Построение FST нормализации чисел  
  
fst = pynini.compose(convert_time_cdr, normalize_time_cdr)
```

Практический пример #3

## Нормализация времени final

- Последовательно соединить оба FST



Google Colab

### Входная строка

В 01:30 AM в парке

### Выходная строка

В 01:30 в парке

```
normalize_time_cdr = ... # Построение FST конвертации времени  
convert_time_cdr = ... # Построение FST нормализации чисел  
  
fst = pynini.compose(convert_time_cdr, normalize_time_cdr)
```

## Практический пример #3

# Нормализация времени final

3. Последовательно соединить оба FST

В 01:30 AM в парке

**Входная строка**

В 01:30 в парке

**Выходная строка**

В один час тридцать минут в парке

```
normalize_time_cdr = ... # Построение FST конвертации времени  
convert_time_cdr = ... # Построение FST нормализации чисел  
  
fst = pynini.compose(convert_time_cdr, normalize_time_cdr)
```



Google Colab

Практический пример #3

## Нормализация времени final

- Последовательно соединить оба FST



Google Colab

### Входная строка

В 01:30 AM в парке

### Выходная строка

В один час тридцать минут в парке

```
normalize_time_cdr = ... # Построение FST конвертации времени  
convert_time_cdr = ... # Построение FST нормализации чисел  
  
fst = pynini.compose(convert_time_cdr, normalize_time_cdr)
```

## Практический пример #3

# Нормализация времени final

- Последовательно соединить оба FST



Google Colab

## Входная строка

В 01:30 АМ в парке

## Выходная строка

В один час тридцать минут в парке

```
normalize_time_cdr = ... # Построение FST конвертации времени
convert_time_cdr = ... # Построение FST нормализации чисел

fst = pynini.compose(convert_time_cdr, normalize_time_cdr)

print(("В 01:30 АМ в парке" @ fst).string())
# В один час тридцать минут в парке
print(("Агент 02 ждет в 02:00" @ fst).string())
# Агент 02 ждет в два часа
```

# Оптимизация FST

## Проблема

Очень часто FST получается слишком большим, собирается очень долго и весит очень много.

# Оптимизация FST

## Проблема

Очень часто FST получается слишком большим, собирается очень долго и весит очень много.

**Время сборки:** > 10 часов

**Вес:** > 50 GB

# Оптимизация FST



## Топ причин:

1. FST делает больше чем требуется
2. Одна FST выполняет слишком много разных задач
3. Неоптимальное использование скрытых состояний

# Оптимизация FST

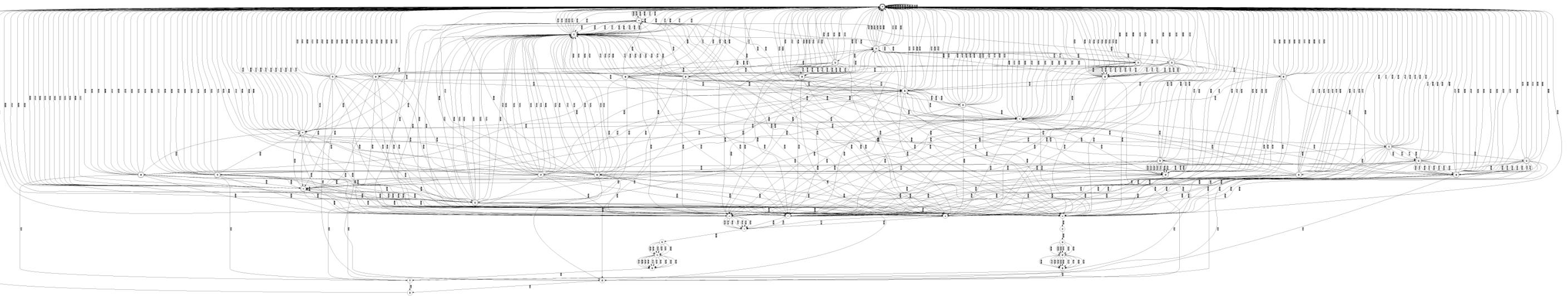


## Топ причин:

1. **FST делает больше чем требуется**
2. Одна FST выполняет слишком много разных задач
3. Неоптимальное использование скрытых состояний

# Оптимизация FST

1. FST делает больше чем требуется



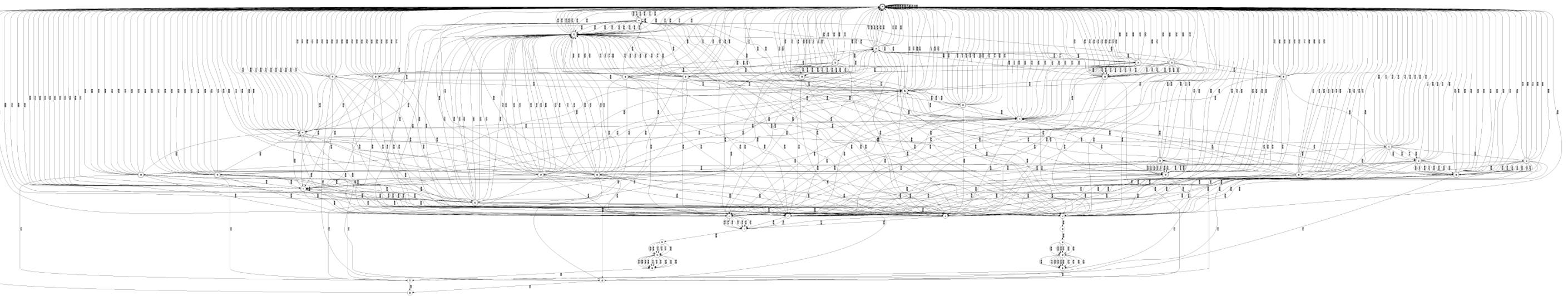
**Входная строка**

11:05:34 PM

**Выходная строка**

# Оптимизация FST

1. FST делает больше чем требуется



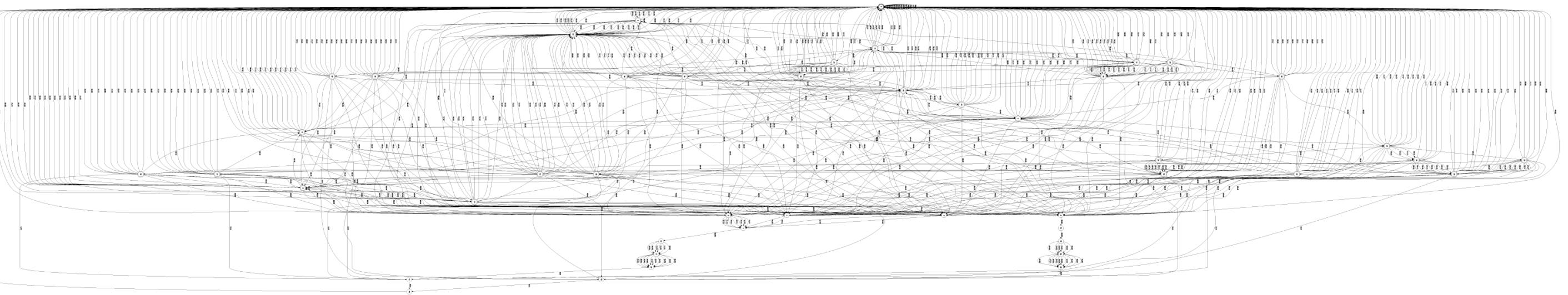
Входная строка

11:05:34 PM

Выходная строка

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

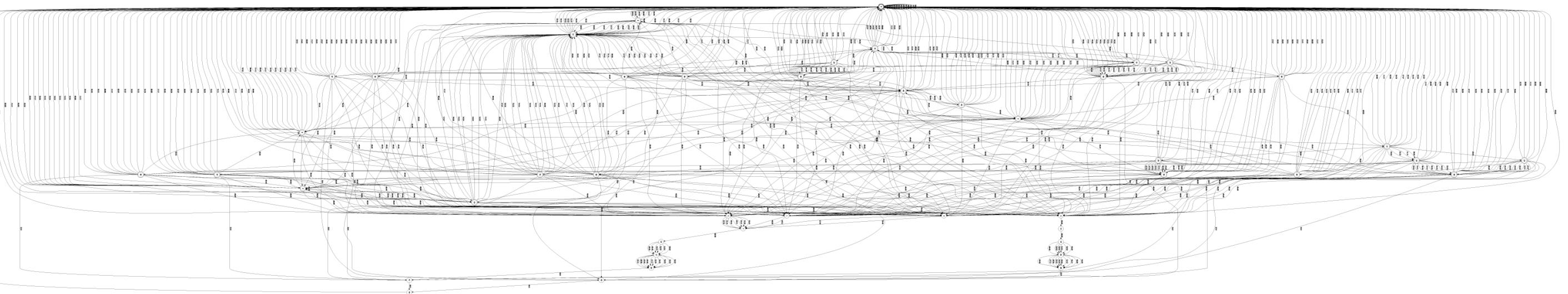
11:05:34 PM

Выходная строка

11

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

11:05:34 PM

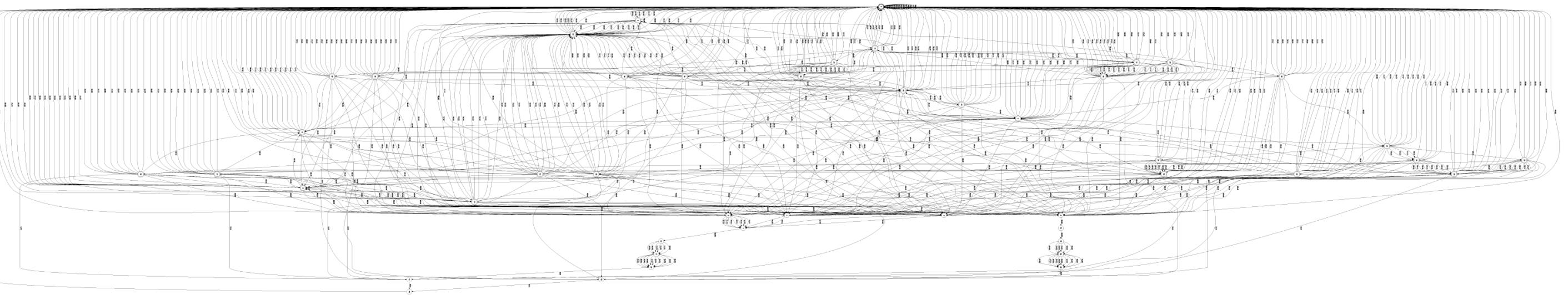
Выходная строка

11

23

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

11:05:34 PM

Выходная строка

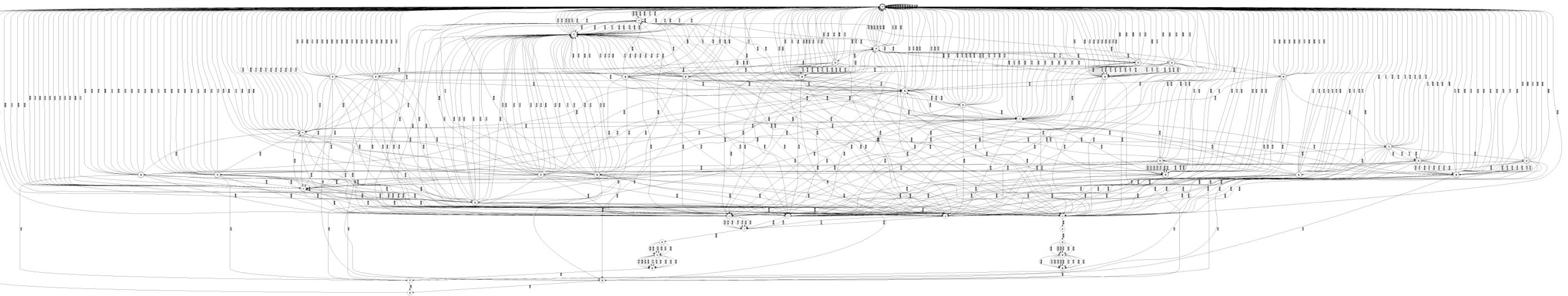
11

23

11

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 PM

**Выходная строка**

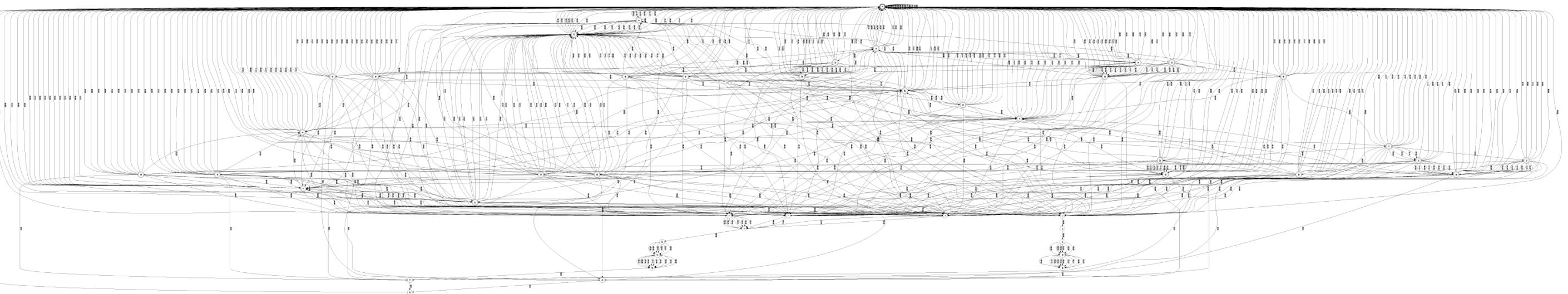
11

23

11

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 PM

**Выходная строка**

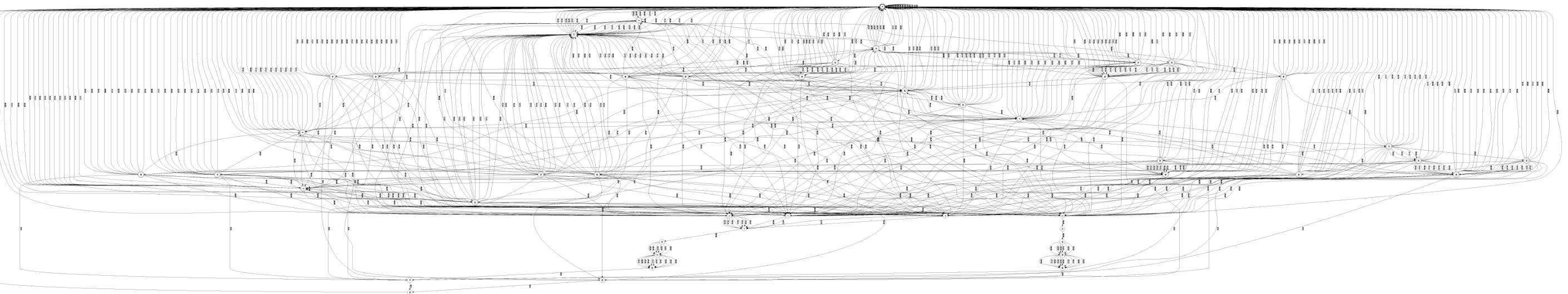
11:

23:

11:

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 PM

**Выходная строка**

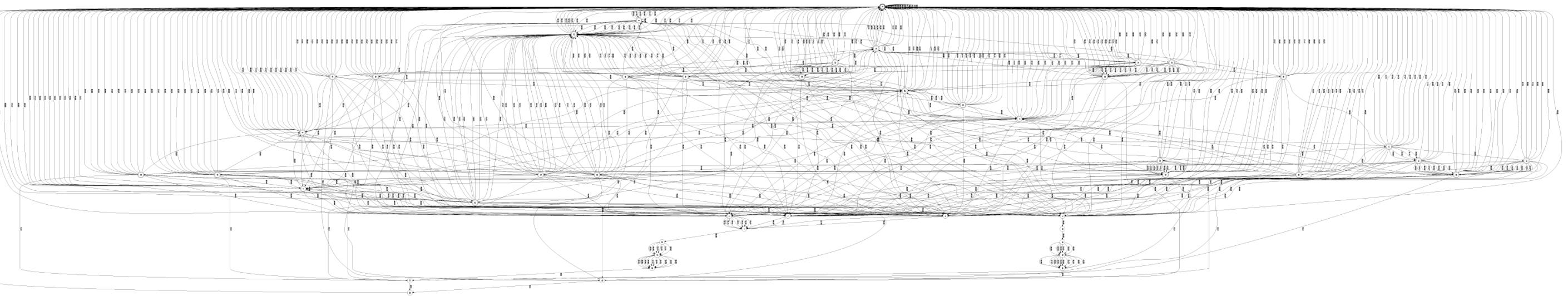
11:

23:

11:

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

11:05:34 PM

Выходная строка

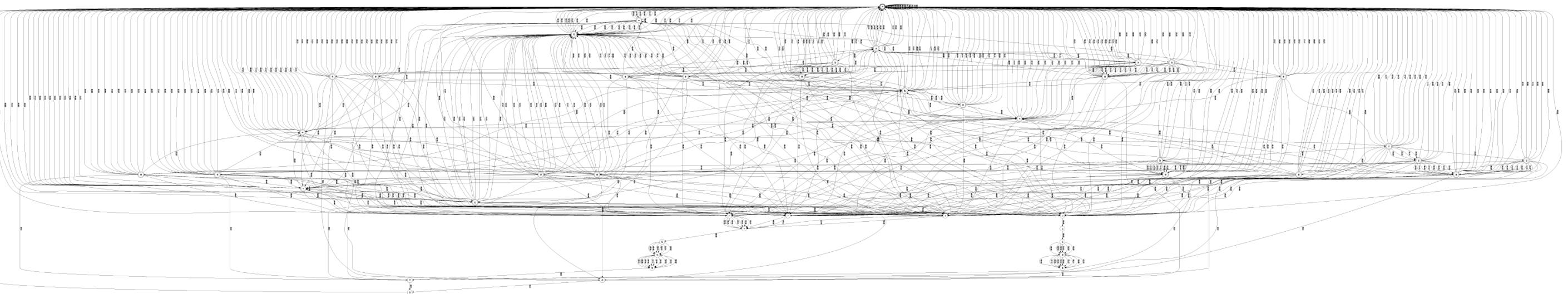
11:05

23:05

11:

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 PM

**Выходная строка**

11:05

11:05

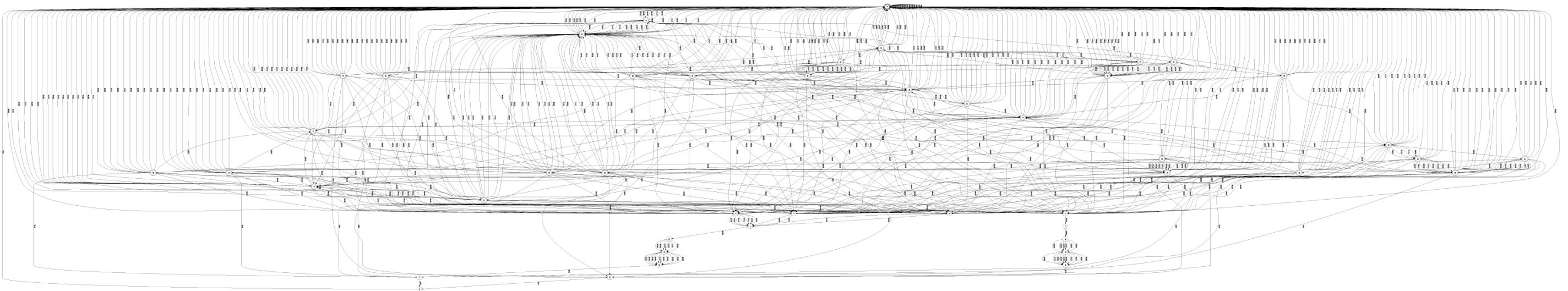
23:05

11:17

11:

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 РМ

**Выходная строка**

11:05

11:05

23:05

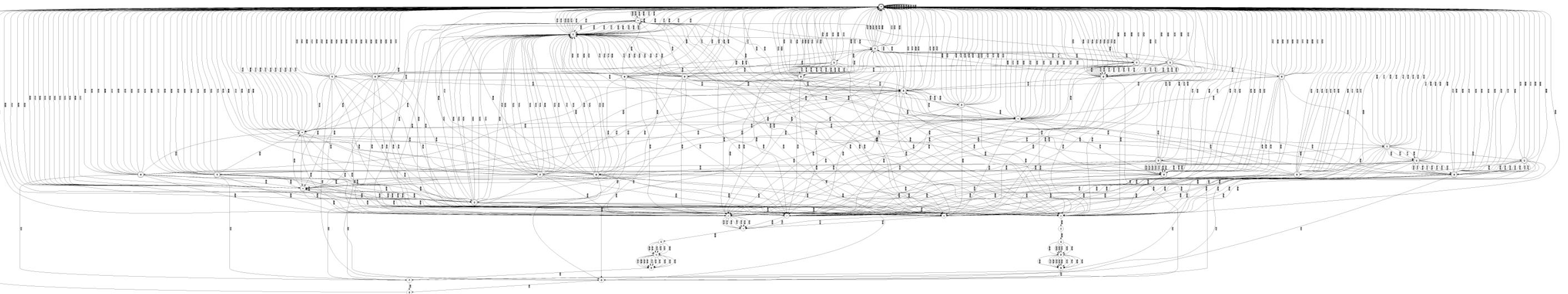
11:17

11:

11:05

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 РМ

**Выходная строка**

11:05

11:05

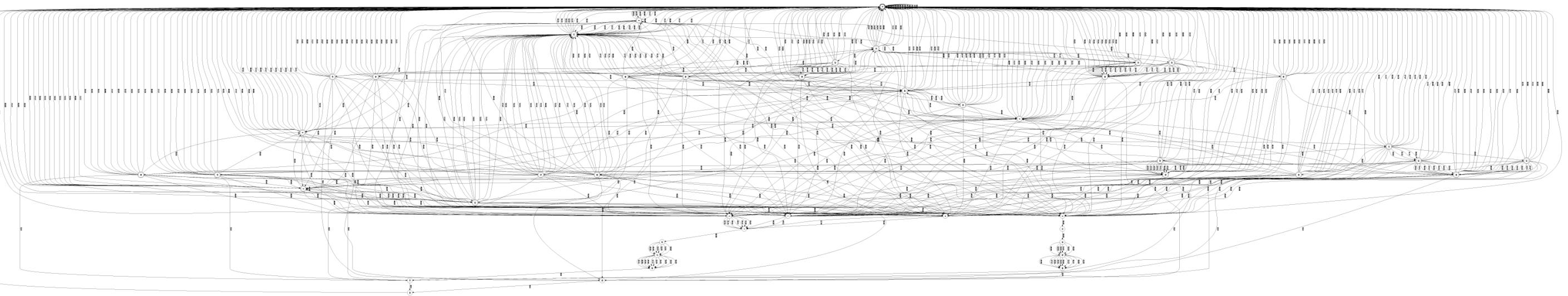
23:05

11:17

11:05

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 PM

**Выходная строка**

11:05

11:05

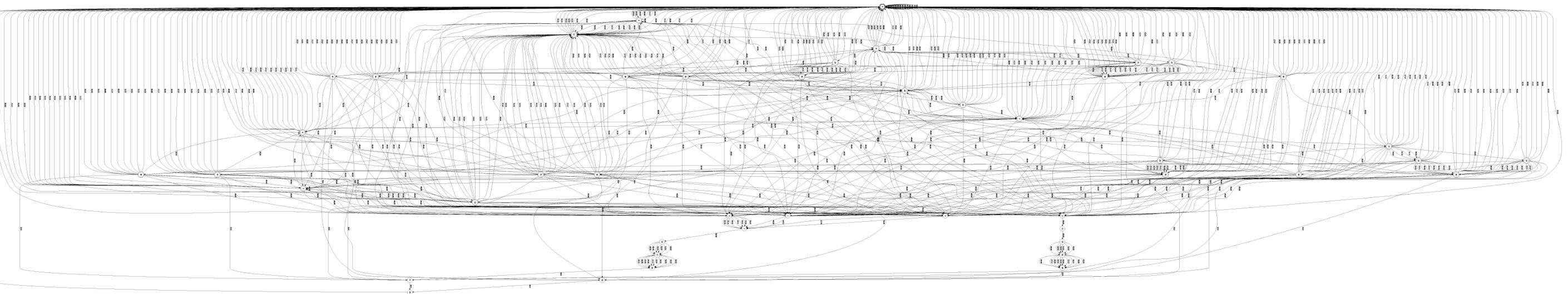
23:05

11:17

11:05

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 PM

**Выходная строка**

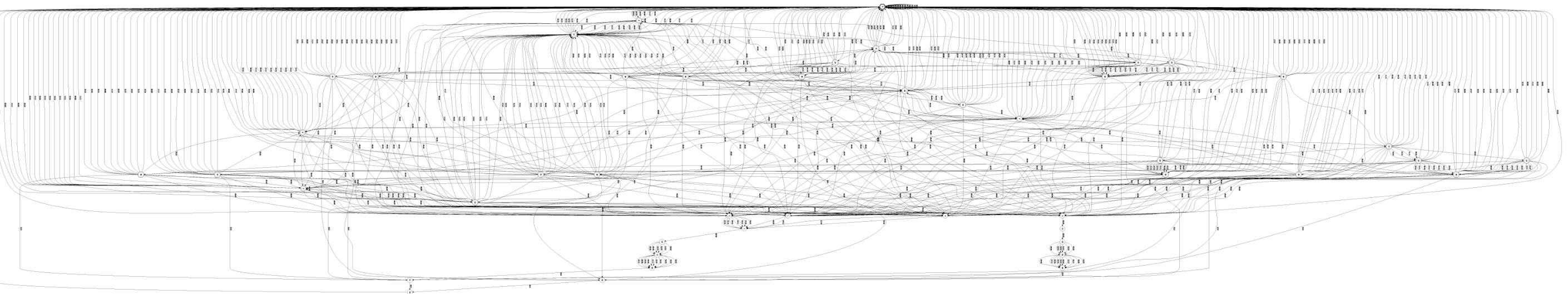
11:05

11:17

11:05

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 РМ

**Выходная строка**

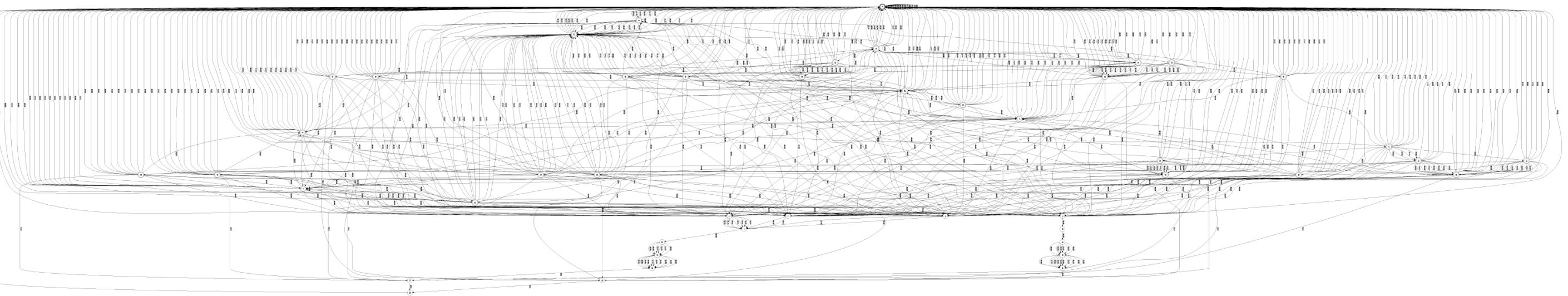
11:05:

11:17:

11:05:

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 РМ

**Выходная строка**

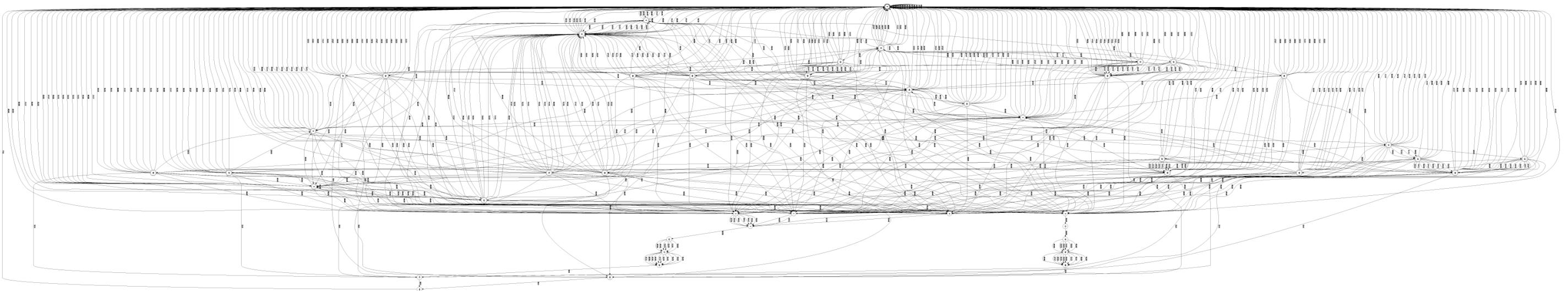
11:05:

11:17:

11:05:

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 РМ

**Выходная строка**

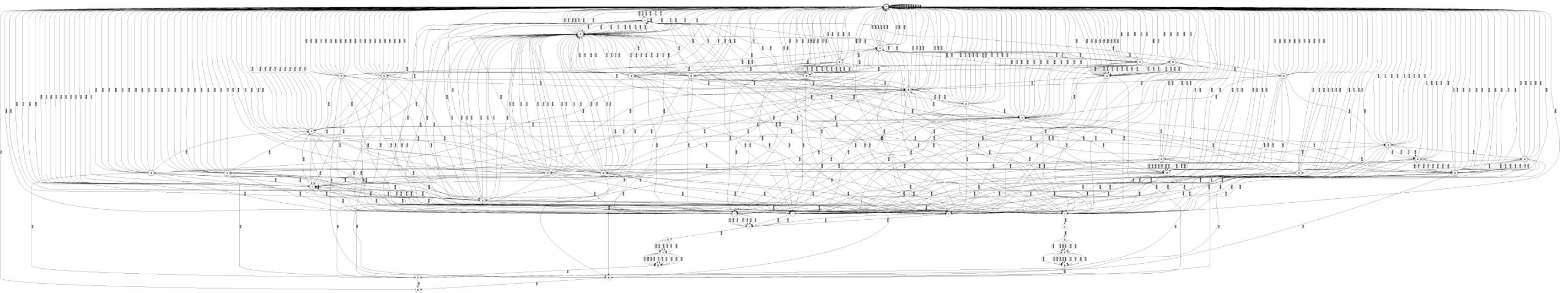
11:05:34

11:17:34

11:05:34

# Оптимизация FST

1. FST делает больше чем требуется



**Входная строка**

11:05:34 **PM**

**Выходная строка**

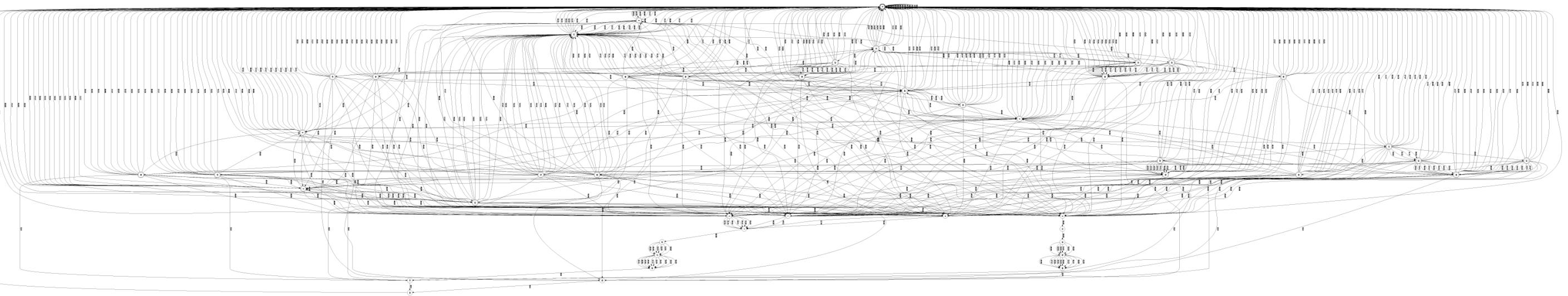
11:05:34

11:17:34

11:05:34

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

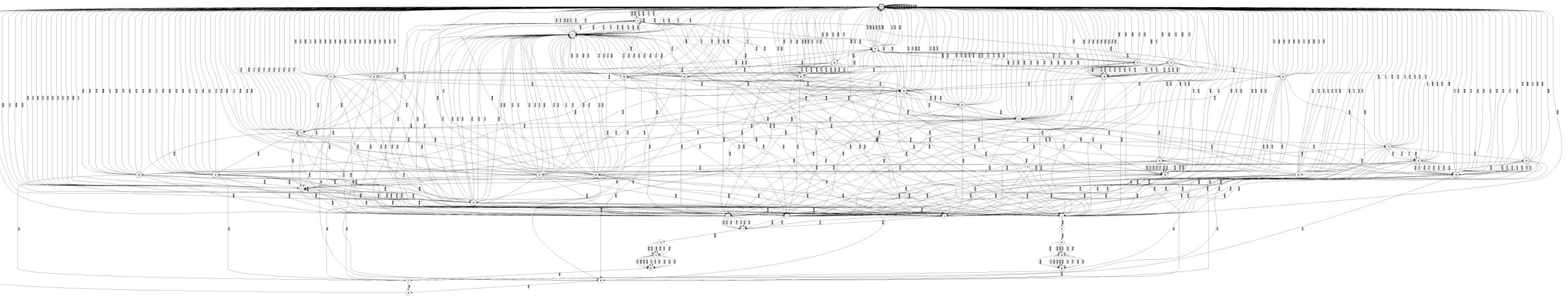
11:05:34 PM

Выходная строка

11:17:34

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

11:05:34 PM

Искомый паттерн

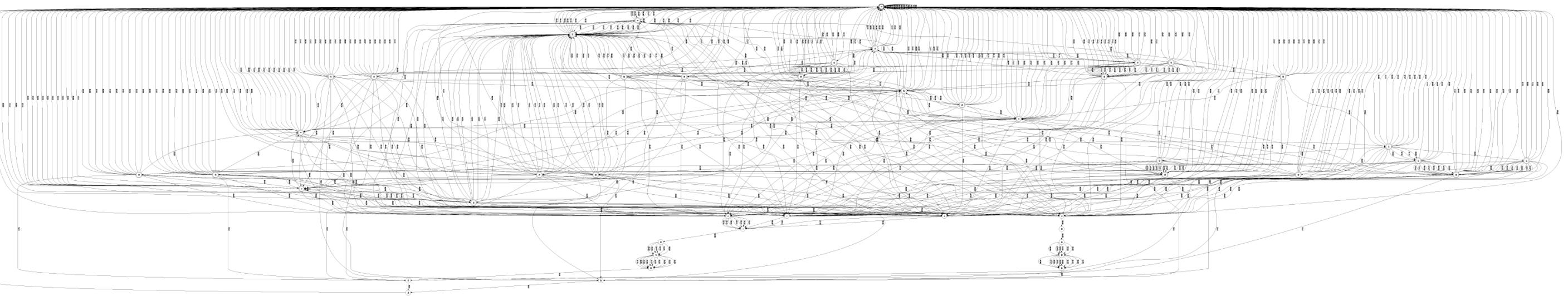
\* \* : \* \* \* M

```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()
cdr = pynini.cdrewrite(fst,
                       "",
                       "",
                       star)
```

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

11:05:34 PM

Искомый паттерн

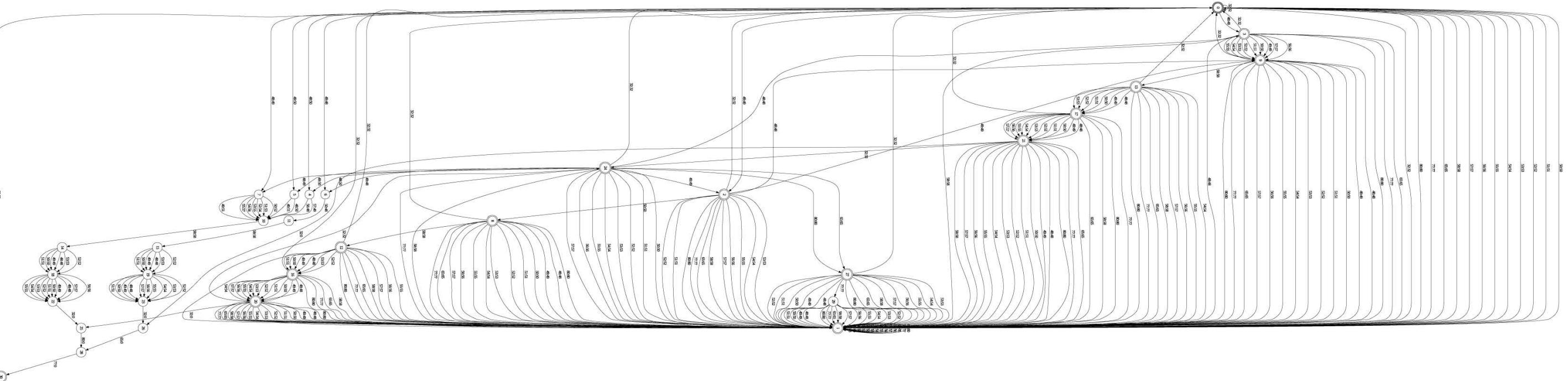
\* \* : \* \* \* M

```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()
cdr = pynini.cdrewrite(fst,
                       "",
                       "",
                       star)
```

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

11:05:34 PM

Искомый паттерн

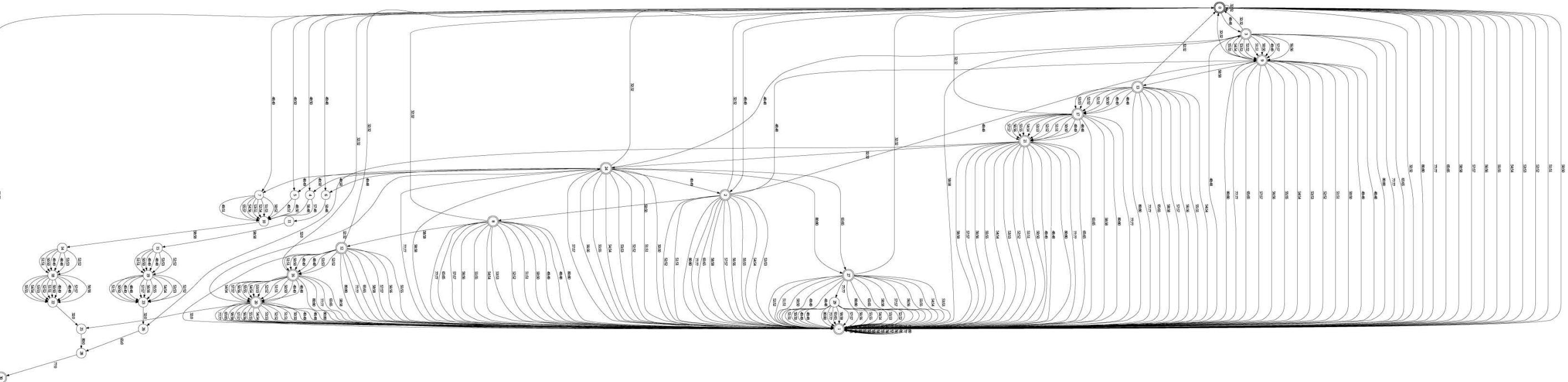
\* \* : \* \* \* M → \_ \* \* : \* \* \* M \_

```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()
cdr = pynini.cdrewrite(fst,
                       pynini.union(*" ", "[BOS]"),
                       pynini.union(*" ", "[EOS]"),
                       star)
```

# Оптимизация FST

1. FST делает больше чем требуется



Входная строка

В 05:34 PM завтра

Искомый паттерн

\* \* : \* \* \* M → \_ \* \* : \* \* \* M \_

```
fst = ... # Построение FST конвертации времени
```

```
star = pynini.union(*[f"[{i}]" for i in range(1, 256)]).closure()
cdr = pynini.cdrewrite(fst,
                       pynini.union(*" ", "[BOS]"),
                       pynini.union(*" ", "[EOS]"),
                       star)
```

Практический пример #4

# Структурированная нормализация

## Задача

Создать универсальный FST для нормализации

Например:

В 11:11 PM придет:11 гостей  $\Rightarrow$  В двадцать три часа  
одиннадцать минут придет: одиннадцать гостей  
На 12 МБ  $\Rightarrow$  На двенадцать мегабайт  
На 12 МБ  $\Rightarrow$  На двенадцати мегабайтах

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В **11:11** РМ придет:**11** гостей

### Искомый паттерн

\* \*

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

### Искомый паттерн

:\*\*

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

#### 1. Поиск и выделение токенов

В 11:11 PM придет:11 гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

#### 1. Поиск и выделение токенов

В **11:11 PM** придет:11 гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

#### 1. Поиск и выделение токенов

В <11:11 PM> придет:11 гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

#### 1. Поиск и выделение токенов

В <11:11 PM> придет:11 гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

#### 1. Поиск и выделение токенов

В <11:11 PM> придет:**11** гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

#### 1. Поиск и выделение токенов

В <11:11 PM> придет:<11> гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <11:11 PM> придет:<11> гостей

### 2. Анализ и трансформации

В <23:11> придет:<11> гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 РМ придет:11 гостей

### 1. Поиск и выделение токенов

В <11:11 РМ> придет:<11> гостей

### 2. Анализ и трансформации

В <23:11> придет:<11> гостей

### 3. Перевод в текст

В <два...> придет:<одинадцать> гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <11:11 PM> придет:<11> гостей

### 2. Анализ и трансформации

В <23:11> придет:<11> гостей

### 3. Перевод в текст

В <два...> придет:<одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <time|11:11 PM> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <time12|11:11 PM> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time24|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
```

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <time12|11:11 PM> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time24|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
fst = create_tokens @ convert_time @ norm_int @ norm_time @ token_rm
print(("В 11:11 PM придет:11 гостей" @ fst).string())
```

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <time12|11:11 PM> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time24|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
fst = create_tokens @ convert_time @ norm_int @ norm_time @ token_rm
print(("В 11:11 PM придет:11 гостей" @ fst).string())
```

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <time12|11:11 PM> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time24|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
fst = create_tokens @ convert_time @ norm_int @ norm_time @ token_rm
print(("В 11:11 PM придет:11 гостей" @ fst).string())
```

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <time12|11:11 PM> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time24|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
fst = create_tokens @ convert_time @ norm_int @ norm_time @ token_rm
print(("В 11:11 PM придет:11 гостей" @ fst).string())
```

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

В 11:11 РМ придет:11 гостей

### 1. Поиск и выделение токенов

В <time12|11:11 РМ> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time24|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
fst = create_tokens @ convert_time @ norm_int @ norm_time @ token_rm
print(("В 11:11 РМ придет:11 гостей" @ fst).string())
```

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

В 11:11 PM придет:11 гостей

### 1. Поиск и выделение токенов

В <time12|11:11 PM> придет:<int|11> гостей

### 2. Анализ и трансформации

В <time24|23:11> придет:<int|11> гостей

### 3. Перевод в текст

В <text|два...> придет:<text|одинадцать> гостей

### 4. Удаление токенов

В двадцать три часа... придет:одинадцать гостей

```
create_tokens = time12_cdr @ time24_cdr @ int_cdr
fst = create_tokens @ convert_time @ norm_int @ norm_time @ token_rm
print(("В 11:11 PM придет:11 гостей" @ fst).string())
```

Практический пример #4

## Структурированная нормализация



Google Colab

Входная строка

На 12 МБ

Практический пример #4

## Структурированная нормализация



### Входная строка

На 12 МБ

#### 1. Поиск и выделение токенов

<word|На> <int|12> <word|МБ>

Google Colab

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

На 12 МБ

#### 1. Поиск и выделение токенов

```
<word|На> <int|12> <word|MБ>
```

#### 2. Анализ и трансформации

```
<prep|На> <num|12> <short|MБ>
```

Практический пример #4

## Структурированная нормализация



Google Colab

### Входная строка

На 12 МБ

#### 1. Поиск и выделение токенов

<word|На> <int|12> <word|МБ>

#### 2. Анализ и трансформации

<prep|На> <num|12> <short|noun|МБ>

Практический пример #4

## Структурированная нормализация



### Входная строка

На 12 МБ

#### 1. Поиск и выделение токенов

```
<word|На> <int|12> <word|МБ>
```

#### 2. Анализ и трансформации

```
<prep|На> <num|type:2|12> <short|noun|МБ>
```

Google Colab

Практический пример #4

## Структурированная нормализация



### Входная строка

На 12 МБ

#### 1. Поиск и выделение токенов

```
<word|На> <int|12> <word|МБ>
```

#### 2. Анализ и трансформации

```
<prep|case:посл|На> <num|type:2|12>  
<short|noun|МБ>
```

Google Colab

## Практический пример #4

# Структурированная нормализация



Google Colab

## Входная строка

На 12 МБ

### 1. Поиск и выделение токенов

```
<word|На> <int|12> <word|МБ>
```

### 2. Анализ и трансформации

```
<prep|case:предп|На> <num|type:2|case:предп|12>
<short|noun|case:предп|numr: плюс|МБ>
```

Практический пример #4

## Структурированная нормализация

### 3. Согласование

```
<prep|case:16cf|На> <num|type:2|case:16cf|12>  
<short|noun|case:16cf|numr: sing|MБ>
```



Google Colab

Практический пример #4

## Структурированная нормализация

### 3. Согласование

```
<prep|case:accs|На> <num|type:2|case:nomn|12>
<short|noun|case:nomn|numr: plur|МБ>
```

**Правила русского языка:**

```
<prep|case:accs> <num|case:accs>
<prep|case:loct> <num|case:loct>
```



Google Colab

Практический пример #4

## Структурированная нормализация



### 3. Согласование

```
<prep|case:accs|На> <num|type:2|case:accs|12>  
<short|noun|case:accs|на> numr: sing |МБ>
```

**Правила русского языка:**

```
<prep|case:accs> <num|case:accs>  
<prep|case:loct> <num|case:loct>
```

Практический пример #4

## Структурированная нормализация



### 3. Согласование

```
<prep|case:accs|На> <num|type:2|case:accs|12>  
<short|noun|case:gent|numr:plur|МБ>
```

**Правила русского языка:**

```
<prep|case:accs> <num|type:2>  
    <noun|case:gent|numr:plur>  
<prep|case:loct> <num|type:2>  
    <noun|case:loct|numr:plur>
```

Практический пример #4

## Структурированная нормализация



Google Colab

### 3. Согласование

```
<prep|case:accs|На> <num|type:2|case:accs|12>  
<short|noun|case:gent|loct|numr: plur |МБ>
```

#### Правила русского языка:

```
<prep|case:accs> <num|type:2>  
    <noun|case:gent|numr:plur>  
<prep|case:loct> <num|type:2>  
    <noun|case:loct|numr:plur>
```

Практический пример #4

## Структурированная нормализация



Google Colab

### 3. Согласование

```
<prep|case:16cf|На> <num|type:2|case:16cf|12>  
<short|noun|case:gent|numr: plur|МБ>
```

### 4. Перевод в текст

```
<prep|case:16cf|На> <num|type:2|case:16cf|12>  
<short|noun|case:gent|numr: plur|МБ>
```

Практический пример #4

## Структурированная нормализация



Google Colab

### 3. Согласование

```
<prep|case:16cf|На> <num|type:2|case:16cf|12>  
<short|noun|case:gent|numr: plur |МБ>
```

### 4. Перевод в текст

```
<prep|case:16cf|На>  
<num|type:2|case:16cf|двеnadцать>  
<short|noun|case:gent|numr: plur |мегабайт>
```

Практический пример #4

## Структурированная нормализация



### 3. Согласование

```
<prep|case:16cf|На> <num|type:2|case:16cf|12>  
<short|noun|case:gent|numr: plur|МБ>
```

### 4. Перевод в текст

```
<prep|case:16cf|На>  
<num|type:2|case:16cf|двенадцать>  
<short|noun|case:gent|numr: plur|мегабайт>
```

### 5. Удаление токенов

На ~~двенадцать~~ мегабайтах

## Практический пример #4

### Структурированная нормализация



Google Colab

#### 3. Согласование

```
<prep|case:16cf|На> <num|type:2|case:16cf|12>  
<short|noun|case:gent|numr: plur|МБ>
```

#### 4. Перевод в текст

```
<prep|case:16cf|На>  
<num|type:2|case:16cf|двенадцать>  
<short|noun|case:gent|numr: plur|мегабайт>
```

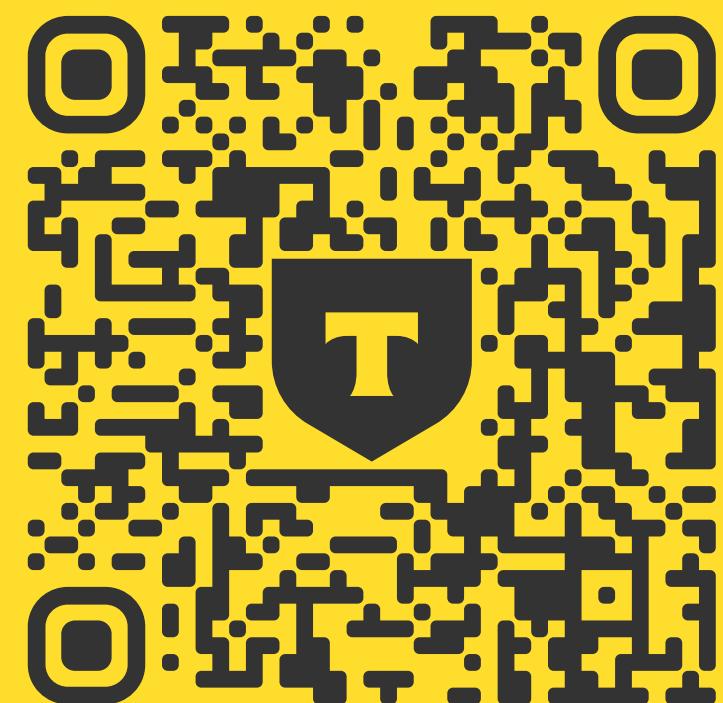
#### 5. Удаление токенов

На двенадцать мегабайт

На двенадцати мегабайтах

# Выводы

- FST – продвинутая технология
  - Сложные преобразования текста
  - Можно добавлять правила русского языка
  - Может быть использована не только для нормализации
- Комбинированный метод нормализации – наиболее оптимальный



Github



**Спасибо!**

