

# **Машинное обучение для формирования контекстной подсказки (контент-ассист) при разработке на платформе 1С:Предприятие**

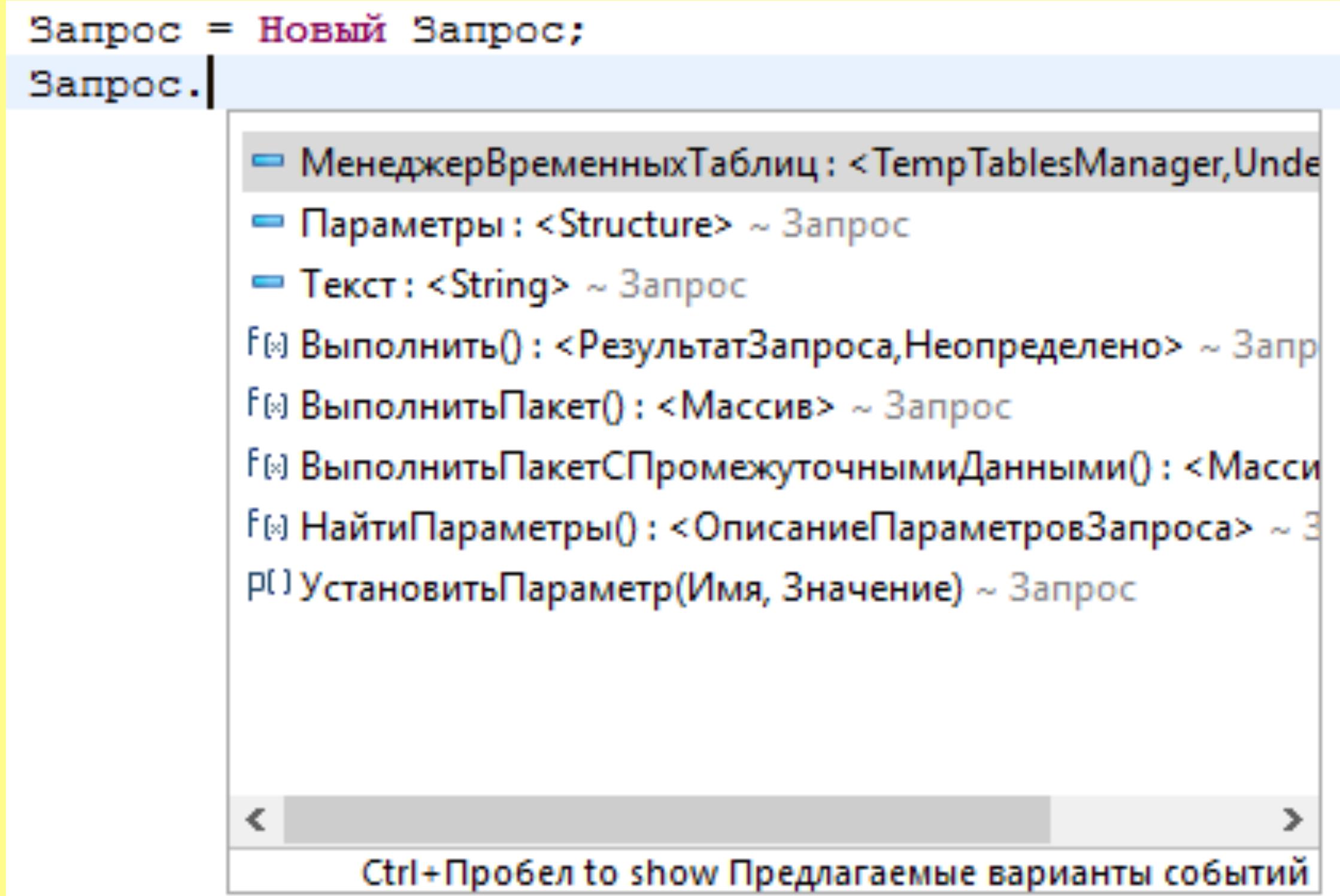
Студент 2 курса магистратуры, М05-912е гр.:  
**Каразеев А. А.**

Научный руководитель:  
**Дзюба М. В.**

# Введение

---

**Контекстная подсказка** — это инструмент, который помогает нам писать и редактировать код программы. С его помощью мы быстрее вводим текст и допускаем меньше ошибок и опечаток.



# Цели и задачи работы

---

- Провести обзор имеющихся решений для построения контекстной подсказки;
- Исследовать предметную область и реализовать механизм переупорядочивания контекстной подсказки с применением машинного обучения для встроенного языка платформы “1С: Предприятие 8”;
- Провести анализ полученных результатов;
- Описать процесс встраивания полученного алгоритма в качестве плагина для среды разработки “1С Enterprise Development Tools” (1С:EDT).

# Существующие решения

---

- Eclipse Code Recommenders
  - Проект архивирован, более не поддерживается;
  - Статистические алгоритмы;
- Eclipse Orion
  - Заточен под веб-разработку;
  - Разработан с учетом особенностей языка JavaScript;
- Jedi
  - Разработан для языка Python;
  - Статический анализатор на основе синтаксического дерева;
  - 120 соавторов, 4.9k звезд на GitHub.

# Существующие решения

---

- Их объединяет следующее:
  - Основаны на знаниях о языке и завязаны на конкретный язык, что делает невозможным их переиспользование с другими языками;
  - А также не учитывают контекст.

# Предполагаемый алгоритм

---

- Алгоритм будет переупорядочивать элементы контекстной подсказки, которые наиболее релевантны в месте вызова подсказки.

# Предполагаемый алгоритм

---

- Алгоритм будет переупорядочивать элементы контекстной подсказки, которые наиболее релевантны в месте вызова подсказки.

Таблица 1 — Пример выдачи контекстной подсказки (слева) и желаемая последовательность элементов (справа).

Например:

```
запрос = Новый Запрос;  
запрос.<|>
```

1. МенеджерВременныхТаблиц	1. Текст
2. Параметры	2. МенеджерВременныхТаблиц
3. Текст	3. УстановитьПараметр()
...	...
8. УстановитьПараметр()	8. Параметры

Под знаком “< | >” понимается место вызова контекстной подсказки.

# Имеющиеся алгоритмы

---

- **Синтаксический анализатор** (не учитывает контекст, может предлагать редко используемые элементы);
- **Статистический подход** - предлагает элементы на основании аналитических данных множества проектов (не учитывает контекст);
- Все они завязаны на конкретный язык и базу знаний о нём, нет возможности переноса опыта на другие языки;

# Имеющиеся алгоритмы

---

- **Синтаксический анализатор** (не учитывает контекст, может предлагать редко используемые элементы);
- **Статистический подход** - предлагает элементы на основании аналитических данных множества проектов (не учитывает контекст);
- Все они завязаны на конкретный язык и базу знаний о нём, нет возможности переноса опыта на другие языки;
- **Машинное обучение** - может предсказывать элементы в зависимости от контекста вызова.

# Машинное обучение

---

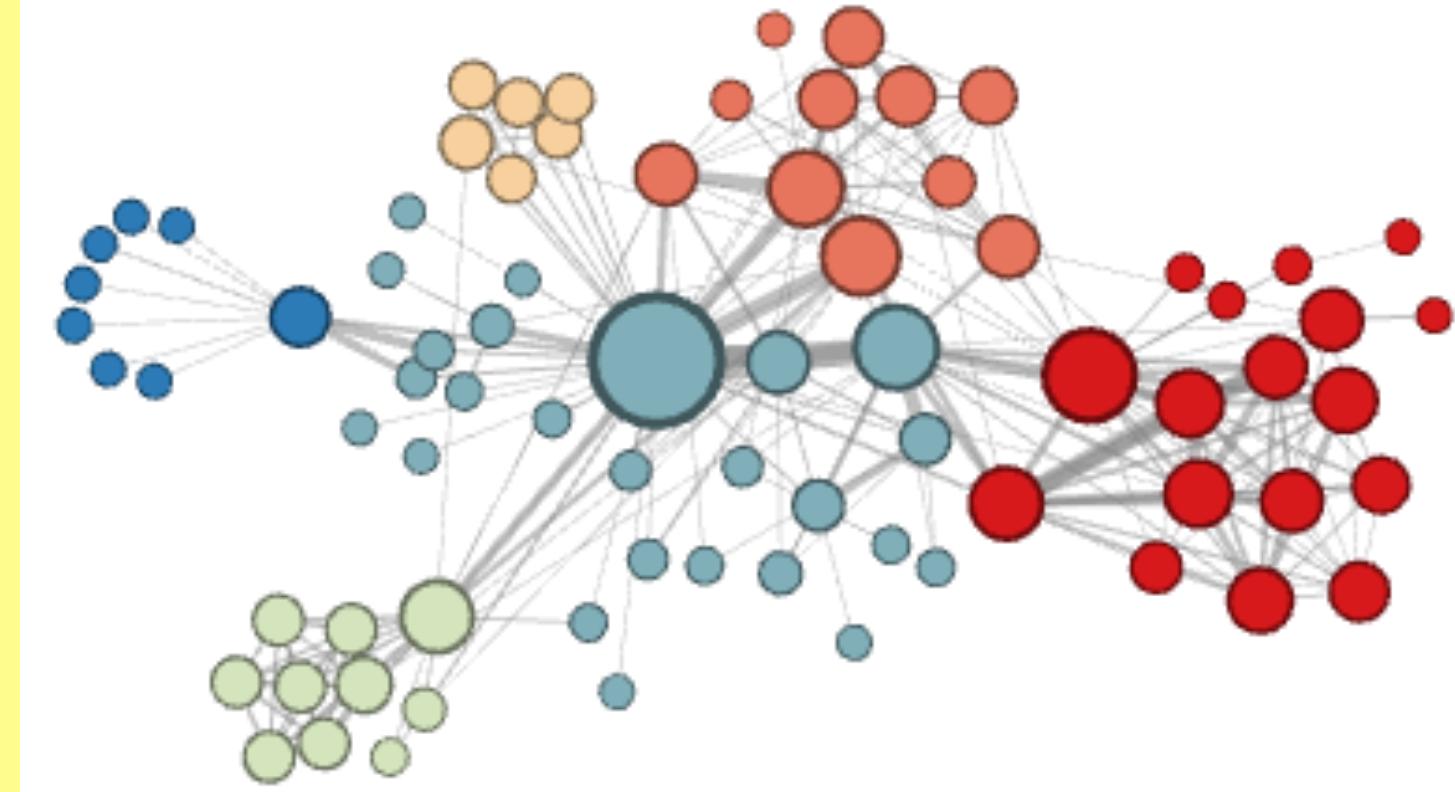
- Проанализировав имеющиеся алгоритмы, было принято решение использовать подход **машинного обучения**;
- **Условные Случайные Поля (CRF)** - сложный процесс обучения, ручной подбор признаков;
- **Graph NNs / Gated Graph NNs** - требуют большой объём вычислительных ресурсов, ручной подбор признаков;
- **RNN / LSTM**

# Машинное обучение

---

- Проанализировав имеющиеся алгоритмы, было принято решение использовать подход **машинного обучения**;
- **Условные Случайные Поля (CRF)** - сложный процесс обучения, ручной подбор признаков;
- **Graph NNs / Gated Graph NNs** - требуют большой объём вычислительных ресурсов, ручной подбор признаков;
- **RNN / LSTM:**
  - word2vec - теряет часть информации об отношениях между переменными, классами и методами;
  - code2seq - не требует ручного подбора признаков, применимо для различных языков программирования, а также может быть использовано для решения других задач и не требует семантического анализа.

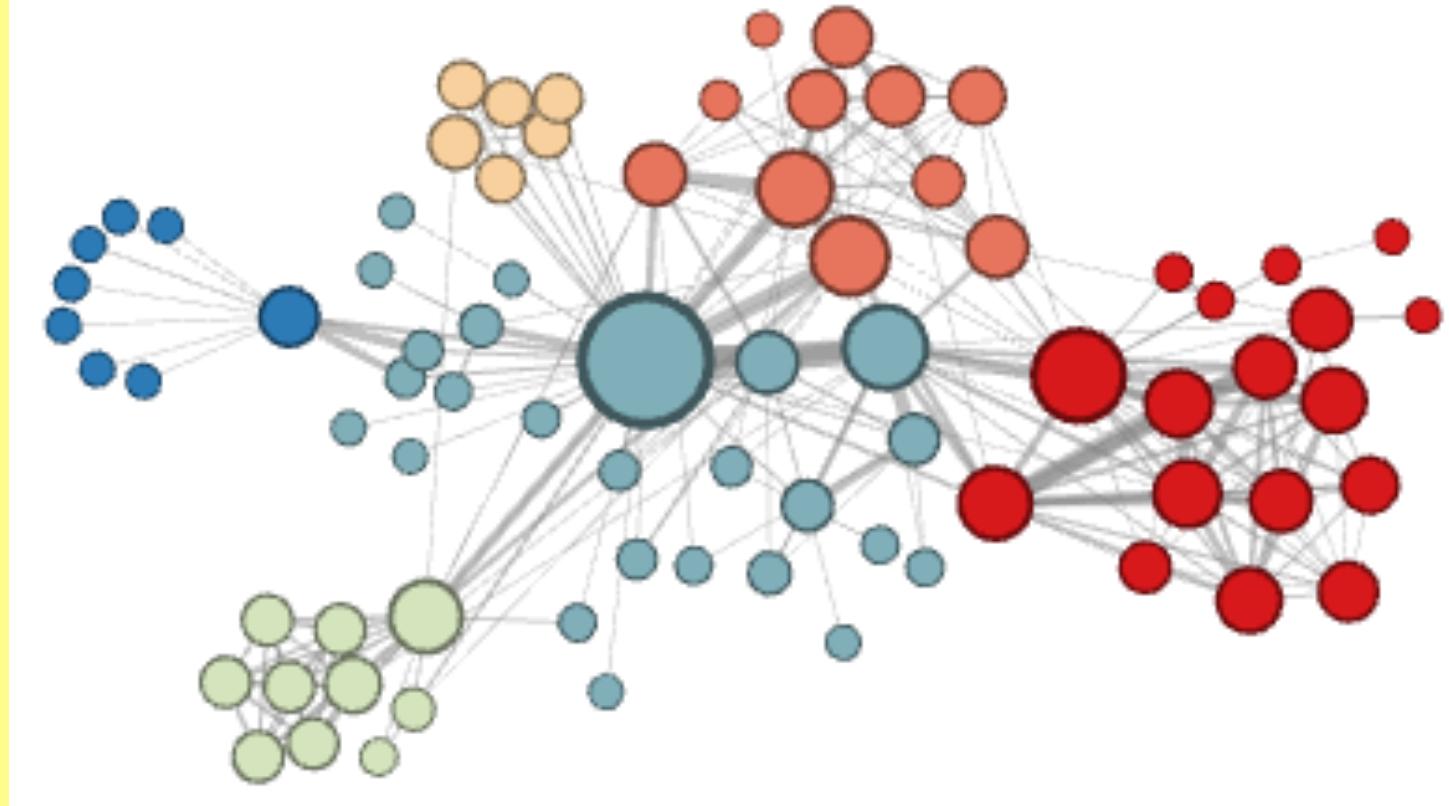
# Архитектура решения - АСД



- Было решено использовать **Абстрактное Синтаксическое Дерево** в качестве описания текста программ;
- АСД наиболее полно характеризует место, из которого была вызвана контекстная подсказка;
- А также хорошо характеризует контекст, в котором был произведён вызов, и “знает” о программе до и после точки вызова;
- Может быть расширено дополнительными параметрами.

# Архитектура решения - АСД

---



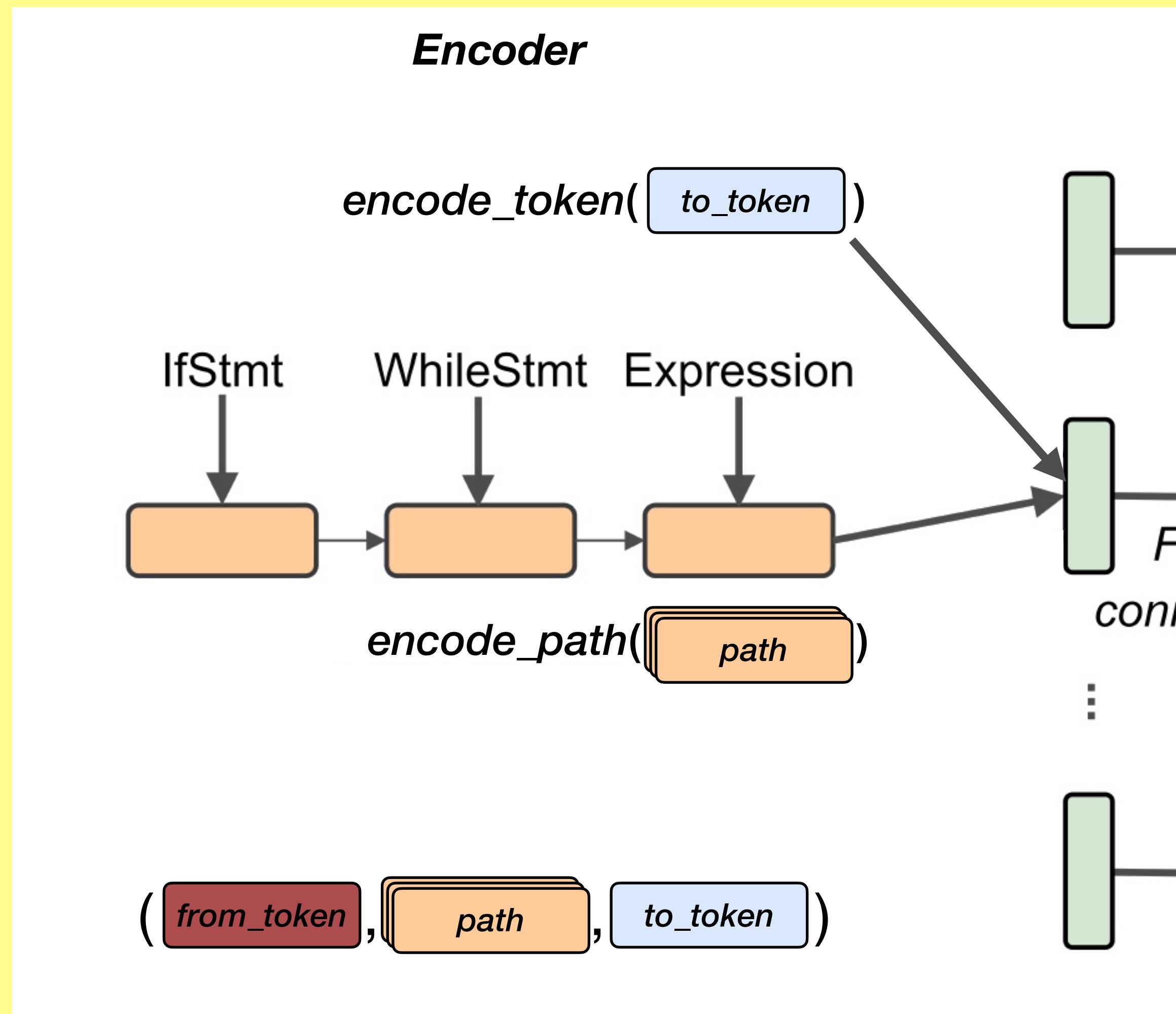
- По тексту программы строится Дерево Разбора, а затем по нему - Абстрактное Синтаксическое Дерево:
  - **Внутренние вершины** сопоставлены с операторами языка программирования (“Пока”, “+”, “=”, “>” и т.п.)
  - **Листья** - операнды (“x”, “42” и т.п.)
- Полученная информация представляет из себя граф, над которым можно производить преобразования и обучать модели.

# Построение модели - code2seq

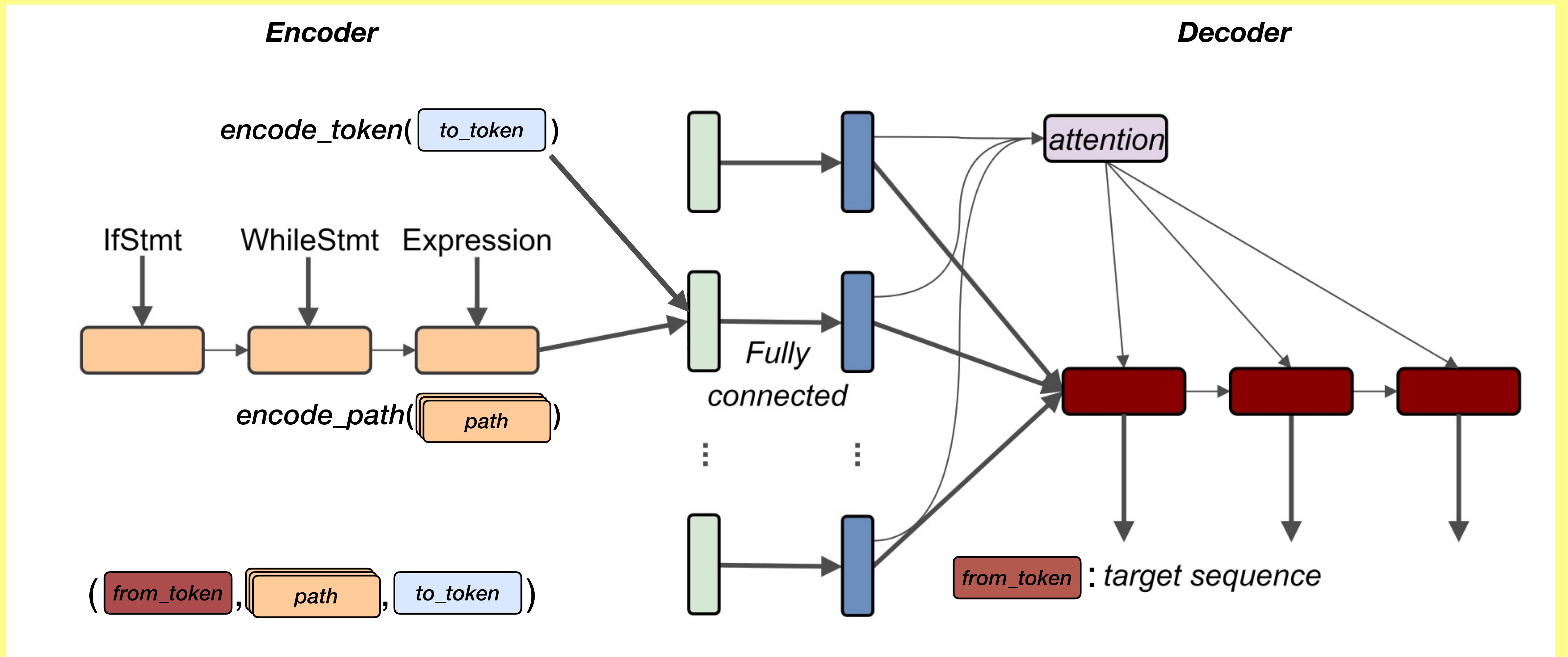
---

- Проанализировав варианты, было решено взять за основу модель code2seq, с помощью которой можно строить векторные представления текста программ;
- Датасет для обучения состоит из набора контекстных цепочек между листьями АСД:
  - (`from_token`, `path`, `to_token`), где
    - `from_token` - начало цепочки (операнд)
    - `path` - последовательность внутренних вершин (операторы)
    - `to_token` - конец цепочки (операнд)

# Построение модели - code2seq



# Построение модели - code2seq



# Обучение модели - code2seq

---

- Предсказывается from\_token на основании path и to\_token;
- Модель представляет из себя Энкодер + Декодер;

$$h_2, \dots, h_l = LSTM(E_{v_2}^{nodes}, \dots, E_{v_l}^{nodes})$$

$$\text{encode\_path}(v_2, \dots, v_l) = [h_l^{\rightarrow}; h_2^{\leftarrow}]$$

$$\text{encode\_token}(w) = \sum_{s \in \text{split}(w)} E_s^{subtokens}$$

$$z = \tanh(W_{in}[\text{encode\_path}(v_2 \dots v_l); \text{encode\_token}(\text{value}(v_l))])$$

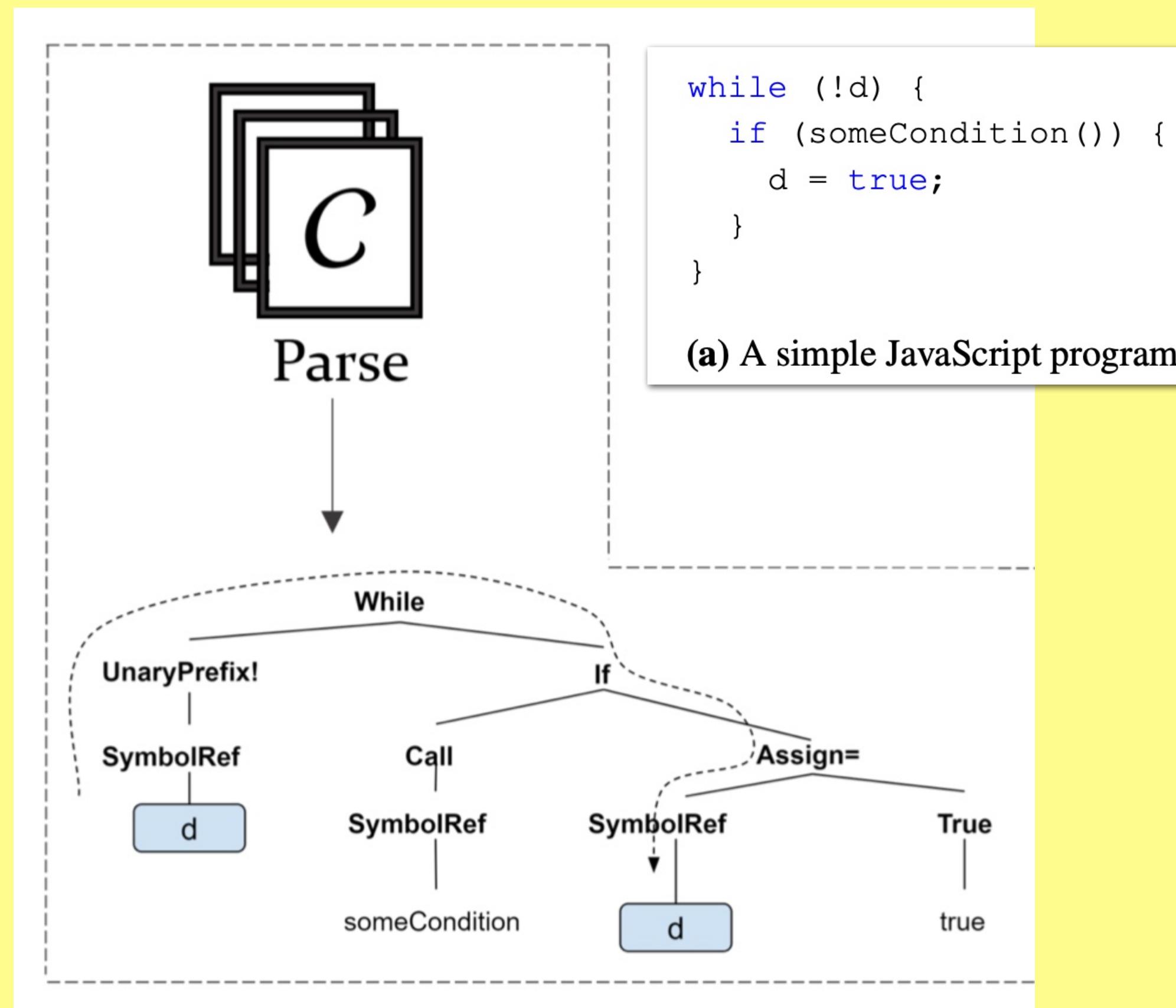
# Пример работы

---

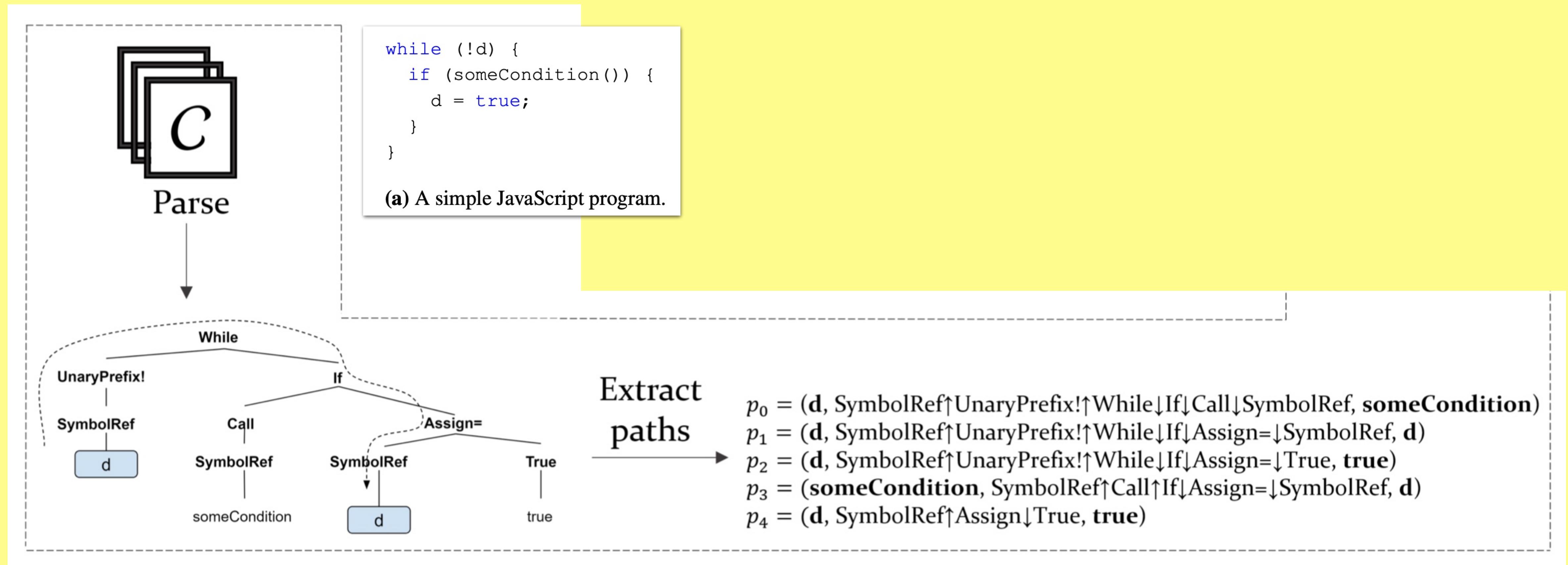
```
while (!d) {  
    if (someCondition()) {  
        d = true;  
    }  
}
```

(a) A simple JavaScript program.

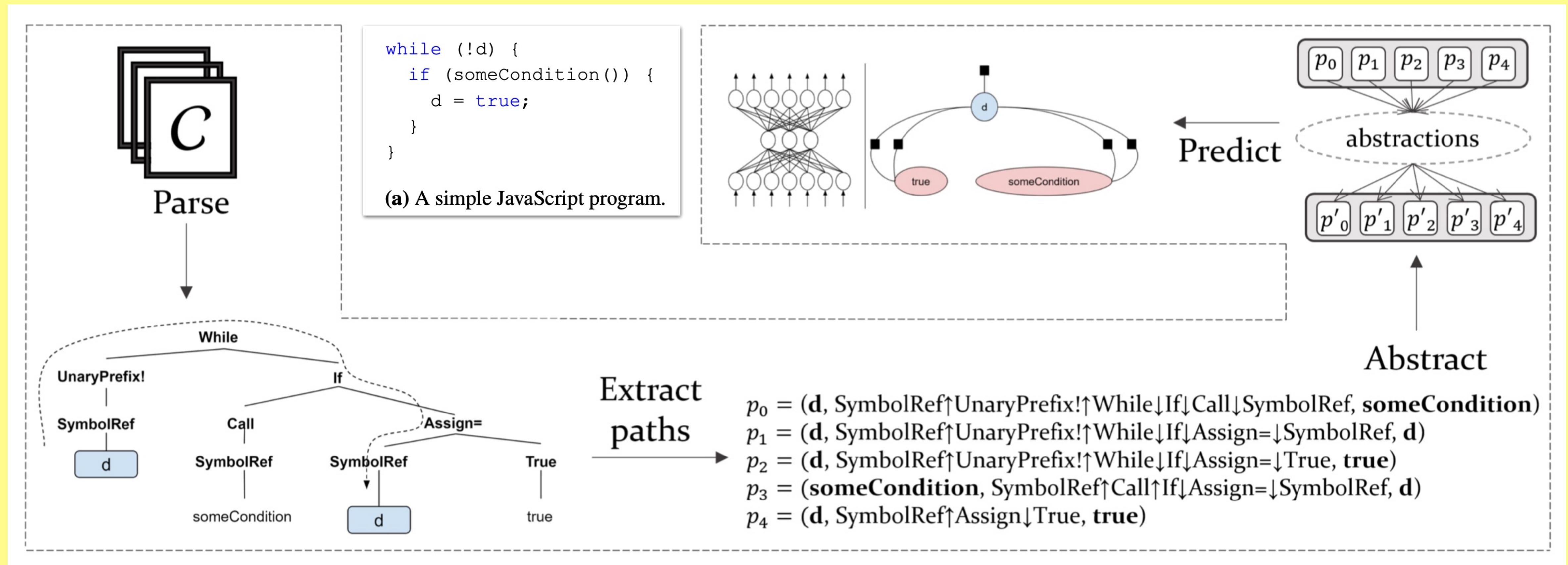
# Пример работы



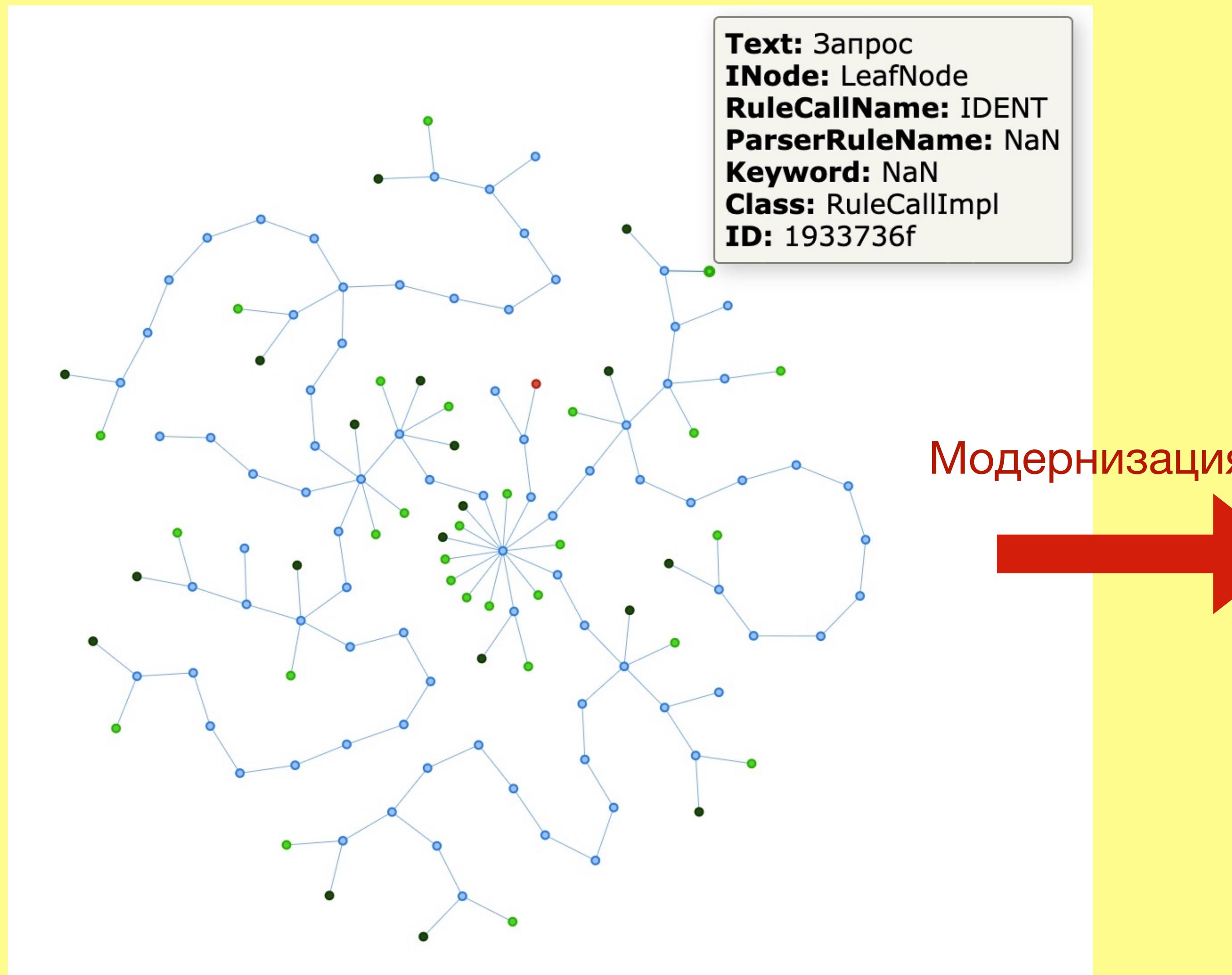
# Пример работы



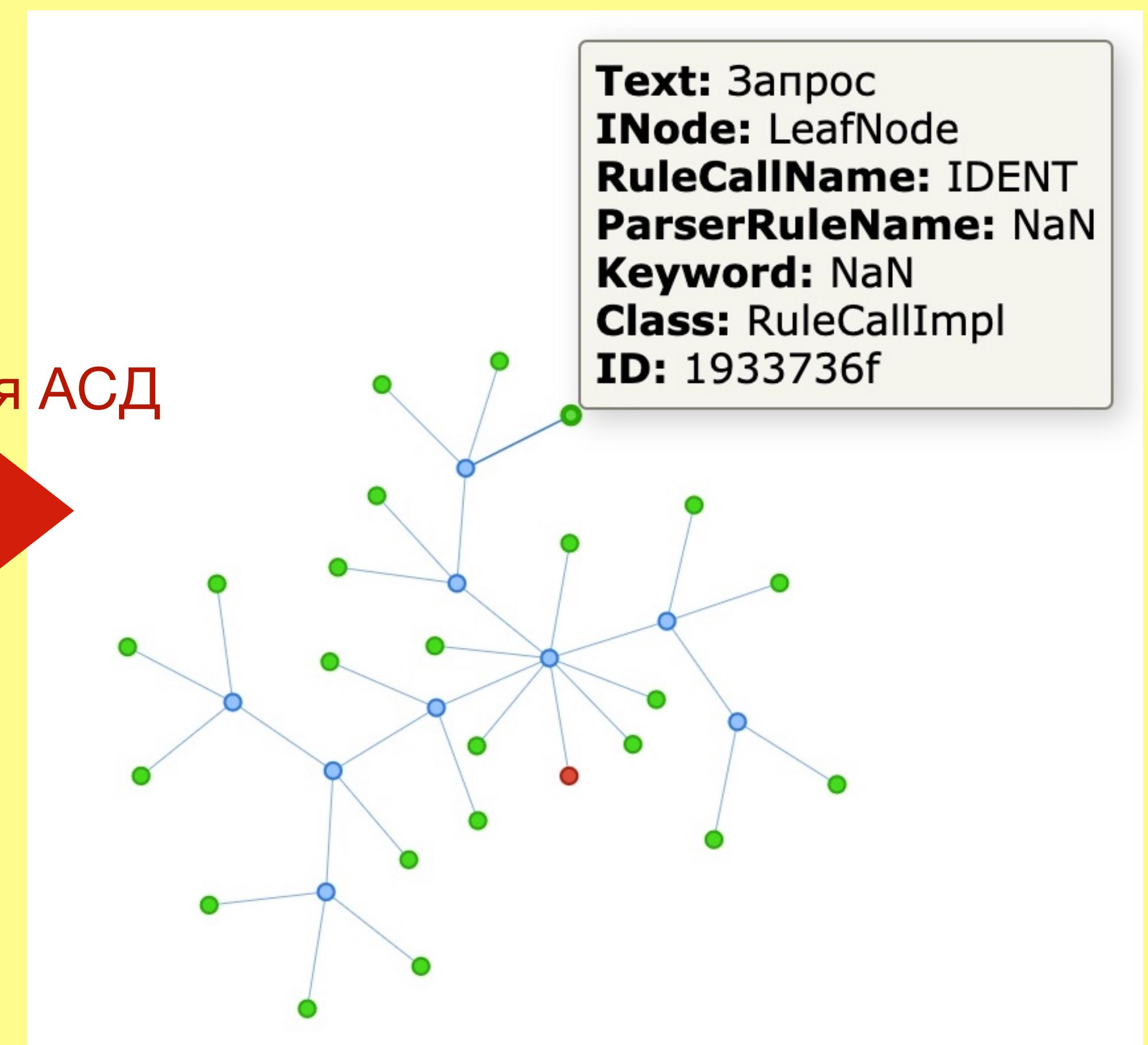
# Пример работы



# Подготовка данных - АСД



Модернизация АСД



```
Document > Document > Object module
Documents → Document → Object module
Процедура имяПроцедуры(Параметр) Экспорт
Запрос = Новый Запрос;

Если Параметр = Истина Тогда
    Запрос.Текст = "текст запроса";
КонецЕсли;

КонецПроцедуры
```

# Встраивание в “1С:EDT”

---

- Плагин встраивается в процесс build проекта через точку расширения Eclipse и обрабатывает каждый модуль встроенного языка;
- Собирает информацию о контекстах (Деревья Разбора);
- Происходит модернизация деревьев и на выходе имеем АСД, по которым уже строятся контекстные цепочки;
- Цепочки подаются на вход модели, которая предсказывает релевантность элементов контекстной подсказки.

# Результаты

---

- Полученные результаты были сделаны только на основании АСД;
- Это не конечный результат, АСД можно расширить признаками: учитывать типизацию переменных, типы модулей (обработчик, класс, метод и т. п.);
- Поставленные задачи были достигнуты и предложены способы улучшения алгоритма.

```

34     ЭтоЗначение = 1;
35     НеИзменять = 0;
36     Элементы.Параметр.Доступность = Ложь;
37     КонецЕсли;
38
39 // Установить связь параметров выбора
40 Если СвязьПоВладельцу = "Параметр" Тогда
41     ЭтоПараметр = 1;
42     ЭтоЗначение = 0;
43     НеИзменять = 0;
44     Элементы.ЭтоЗначение.Доступность = Ложь;
45     Элементы.Значение.Доступность = Ложь;
46     Элементы.Параметр.Доступность = Истина;
47 ИначЕсли ЗначениеЗаполнено(СвязьПоВладельцу) Тогда
48     МассивСвязейПараметровВыбора = Новый Массив;
49     // НоваяСвязь = Новый СвязьПараметровВыбора("Отбор.Владелец", "Св
50     // МассивСвязейПараметровВыбора.Добавить(НоваяСвязь);
51     МассивСвязейПараметровВыбора.|  

52     Элементы.Значение.СвязиПараметровВыбора.Добавить() ~ Массив
53 КонецЕсли;
54
55 Элементы.Значение.ОграничениеТипа
56
57 КонецПроцедуры
58
59 #КонецОбласти
60
61
62
63

```

Space to show Event proposals

from\_token → to\_token:  
Добавить → МассивСвязейПараметровВыбора

Топ-5:

1. Вставить
2. Добавить
3. ЭтотОбъект
4. Параметры
5. Элементы

```

73 КонецЕсли;
74
75 // Удаление из табличной части подсистем, помеченных на удаление.
76 МассивУдаляемыхСтрок = Новый Массив;
77 Для Каждого СтрокаРазмещения Из Размещение Цикл
78     Если СтрокаРазмещения.Подсистема.ПометкаУдаления = Истина Тог
79         // МассивУдаляемыхСтрок.Добавить(СтрокаРазмещения);
80         МассивУдаляемыхСтрок.|  

81 КонецЕсли;
82 КонецЦикла;
83 Для Каждого СтрокаРазмещения
84     Размещение.Удалить(Строка)
85 КонецЦикла;
86
87 // Заполнение реквизитов "Настройки"
88 ПроиндексироватьНастройки();
89 КонецПроцедуры
90
91 #КонецОбласти
92

```

Space to show Event proposals

from\_token → to\_token:  
Добавить → МассивУдаляемыхСтрок

Топ-5:

1. Вставить
2. ЭтотОбъект
3. Текст
4. Добавить
5. СлужебныеПроцедурыИФункции

125

124 &amp;НаСервере

125 **Процедура УстановитьУсловноеОформление()**

126

```

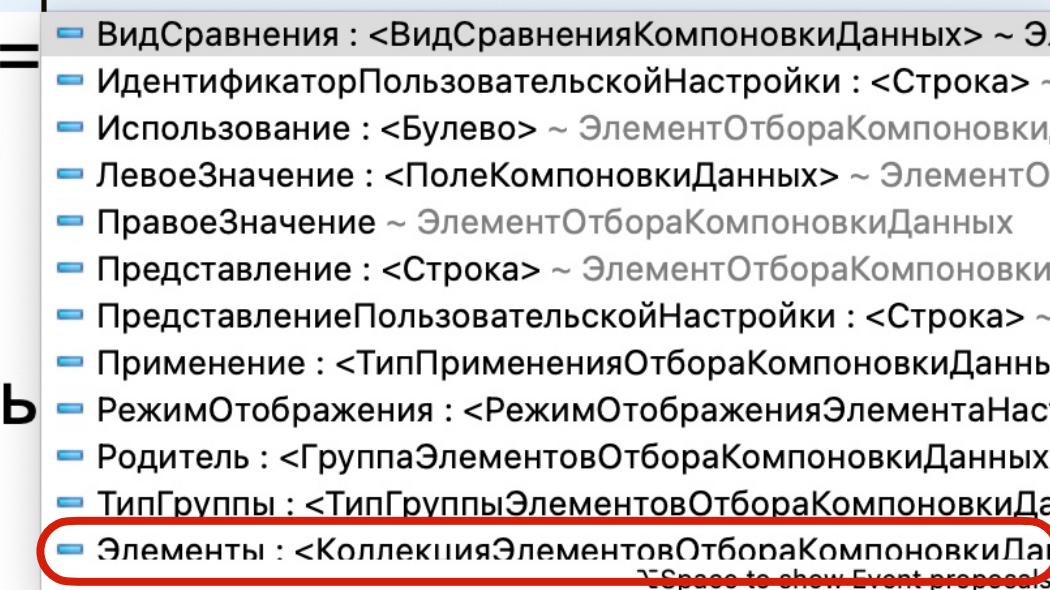
127     Список.УсловноеОформление.Элементы.Очистить();
128     Элемент = Список.УсловноеОформление.Элементы.Добавить();
129
130     ГруппаОтбора = Элемент.Отбор.Элементы.Добавить(Тип("Группа"));
131     ГруппаОтбора.ТипГруппы = ТипГруппыЭлементовОтбораКомпоновки;
132     ОтборЭлемента = ГруппаОтбора.Элементы.Добавить(Тип("Элемент"));
133     ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных();
134     ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных();
135 // ОтборЭлемента = ГруппаОтбора.Элементы.Добавить(Тип("Элемент"));
136     ОтборЭлемента = ГруппаОтбора.Элементы.Добавить(Тип("Элемент"));
137     ОтборЭлемента.ЛевоеЗначение = ОтборЭлемента.ЛевоеЗначение;
138     ОтборЭлемента.ВидСравнения = ОтборЭлемента.ВидСравнения;
139     ОтборЭлемента.ПравоеЗначение = ОтборЭлемента.ПравоеЗначение;
140
141     Элемент.Оформление.Установить();
142
143 КонецПроцедуры

```

from\_token → to\_token:  
 Элементы → ГруппаОтбора

Топ-5:

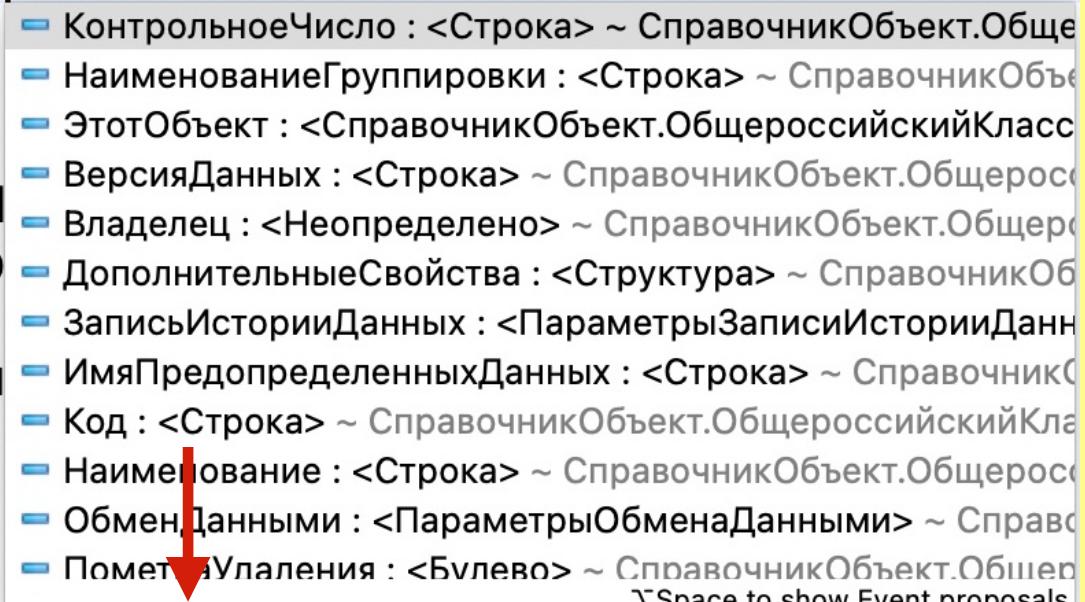
1. Структура
2. ЭтотОбъект
3. Текст
4. Элементы
5. ОписаниеОповещения



```

139
140     КорневойЭлемент = Справочники.ОбщероссийскийКлассификаторОсновныхФондов.*;
141     Если КорневойЭлемент.Пустая() Тогда
142         ЭлементОбъект = Справочники.ОбщероссийскийКлассификаторОсновныхФондо
143         ЭлементОбъект.Код = "OK 013-94";
144         ЭлементОбъект.Наименование = НСтр("ги='Утвержден Постановлением Госс
145         ЭлементОбъект.НаименованиеГруппировки =
146             НСтр("ги='Утвержден Постановлением Госстандарта РФ от 26 декабря
147
148         ОбновлениеИнформационнойБазы.ЗаписатьОбъект(ЭлементОбъект);
149
150 // КорневойЭлемент = ЭлементОбъект.Ссылка;
151     КорневойЭлемент = ЭлементОбъект.| КонецЕсли;
152
153
154     Пока Выборка.Следующий() Цикл
155         ЭлементОбъект = Выборка.Ссылка.П
156         ЭлементОбъект.Родитель = Корнево
157
158         ОбновлениеИнформационнойБазы.Зап
159     КонецЦикла;
160
161 КонецПроцедуры
162

```



from\_token → to\_token:  
 Ссылка → ЭлементОбъект

Топ-5:

1. Элементы
2. Параметры
3. НСтр
4. Ссылка
5. ОбщегоНазначенияКлиентСервер

```

10
11  Если ЗначениеЗаполнено(ЗначениеШтри
12    Запрос = Новый Запрос;
13    Запрос.УстановитьПараметр("Знач
14      // Запрос.Текст ="ВЫБРАТЬ ПЕРВЫ
15      // |   ШтрихкодыУпаковокТоваро
16      // |   ИЗ
17      // |   Справочник.ШтрихкодыУпа
18      // |   ГДЕ
19      // |   ШтрихкодыУпаковокТоваро
20    Запрос.

```

Результат

Если НЕ

Тек

Общ

КонецЕс

КонецЕсли;

from\_token → to\_token:

Текст → Запрос

Топ-5:

1. Структура
2. Текст
3. ЭтотОбъект
4. Вставить
5. Запрос

Space to show Event proposals

- МенеджерВременныхТаблиц : <МенеджерВременныхТабли...
- Параметры : <Структура> ~ Запрос
- Текст : <Строка> ~ Запрос
- ТребуемаяАктуальностьДанных : <ТребуемаяАктуальност...
- ТребуемоеВремяАктуальностиДанных : <Дата,Стандартна...
- F[...] Выполнить() : <Неопределено,РезультатЗапроса> ~ Запро...
- F[...] ВыполнитьПакет() : <Массив> ~ Запрос
- F[...] ВыполнитьПакетСПромежуточнымиДанными() : <Массив> ~ Запро...
- F[...] НайтиПараметры() : <ОписаниеПараметровЗапроса> ~ Запро...
- P[...] УстановитьПараметр(Имя, Значение) ~ Запрос

```

147
148 &НаКлиенте
149⊕ Процедура СписокПередНачаломДобавления(Элемент, Отказ, Копирование, Родитель, Группа, П
150
151   Отказ = Истина;
152
153   ПараметрыФормы = Новый Структура;
154   Если Копирование Тогда
155     // Если Не Элементы.Список.ТекущиеДанные.Свойство("Ссылка") Тогда
156     Если Не Элементы.Список.ТекущиеДанные.| Возврат:
157       КонецЕсли;
158       ПараметрыФормы.Вставить("ЗначениеКопир
159       КонецЕсли;
160       ПараметрыФормы.Вставить("ЗначенияЗаполнени
161       ПараметрыФормы.ЗначенияЗаполнения.Вставить
162
163       ОткрытьФорму("Справочник.БанковскиеСчета.Ф
164
165   КонецПроцедуры
166
167
168

```

Space to show Event proposals

- Основной : <Булево> ~ ДанныеФормыСтруктура
- ПометкаУдаления : <Булево> ~ ДанныеФормыСтруктура
- Свойство(Ключ) : <Булево> ~ ДанныеФормыСтруктура
- Свойство(Ключ, НайденноеЗначение) : <Булево> ~ ДанныеФормыСтруктура
- Банк : <СправочникСсылка.Банки> ~ ДанныеФормыСтруктура
- ВалютаДенежныхСредств : <СправочникСсылка.Валюты> ~ ДанныеФормыСтруктура
- ВидСчета : <Строка> ~ ДанныеФормыСтруктура
- Владелец : <СправочникСсылка.Контрагенты,Справочник...
- ДатаЗакрытия : <Дата> ~ ДанныеФормыСтруктура
- Код : <Строка> ~ ДанныеФормыСтруктура
- Наименование : <Строка> ~ ДанныеФормыСтруктура
- НомерСчета : <Строка> ~ ДанныеФормыСтруктура

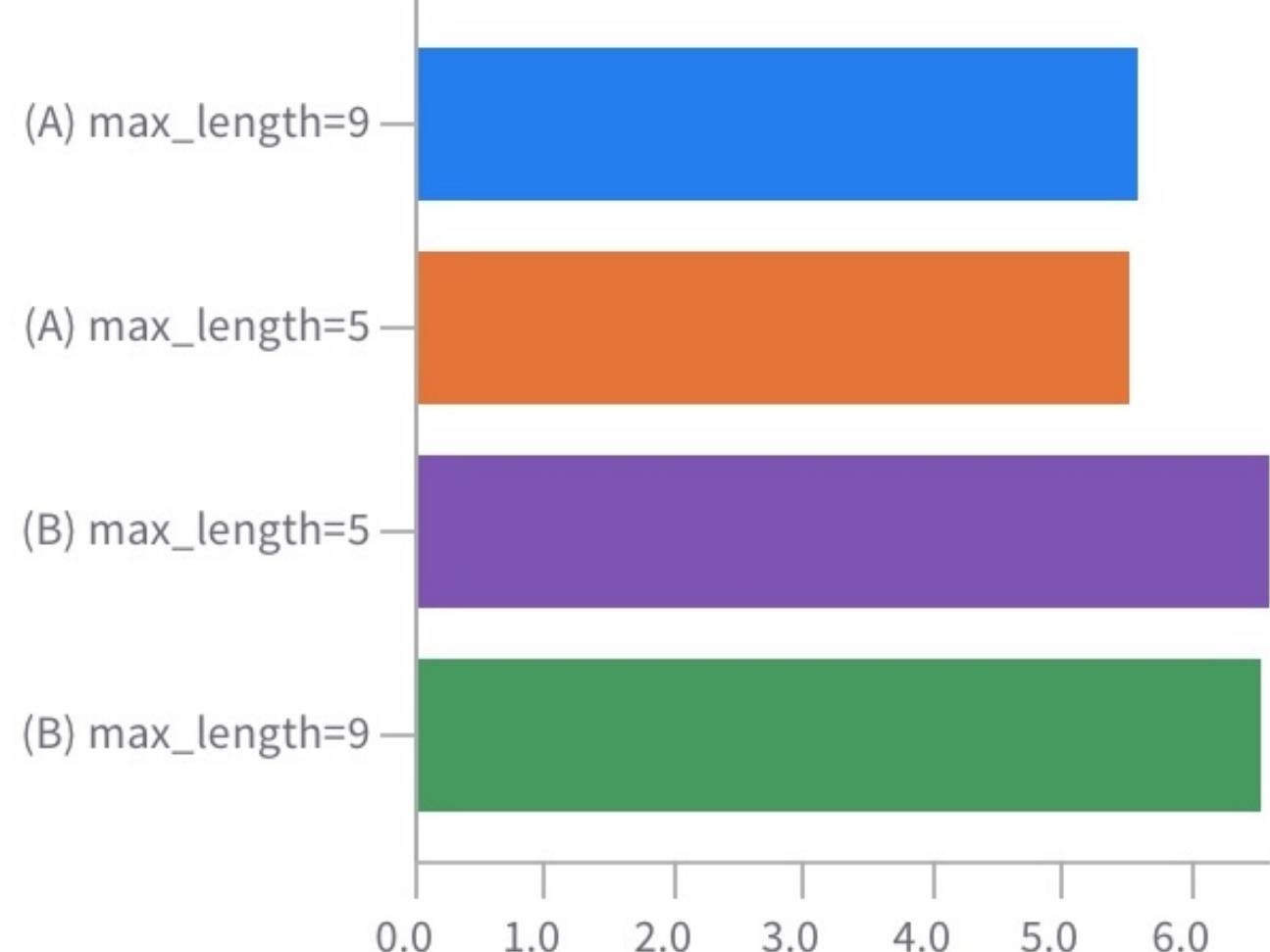
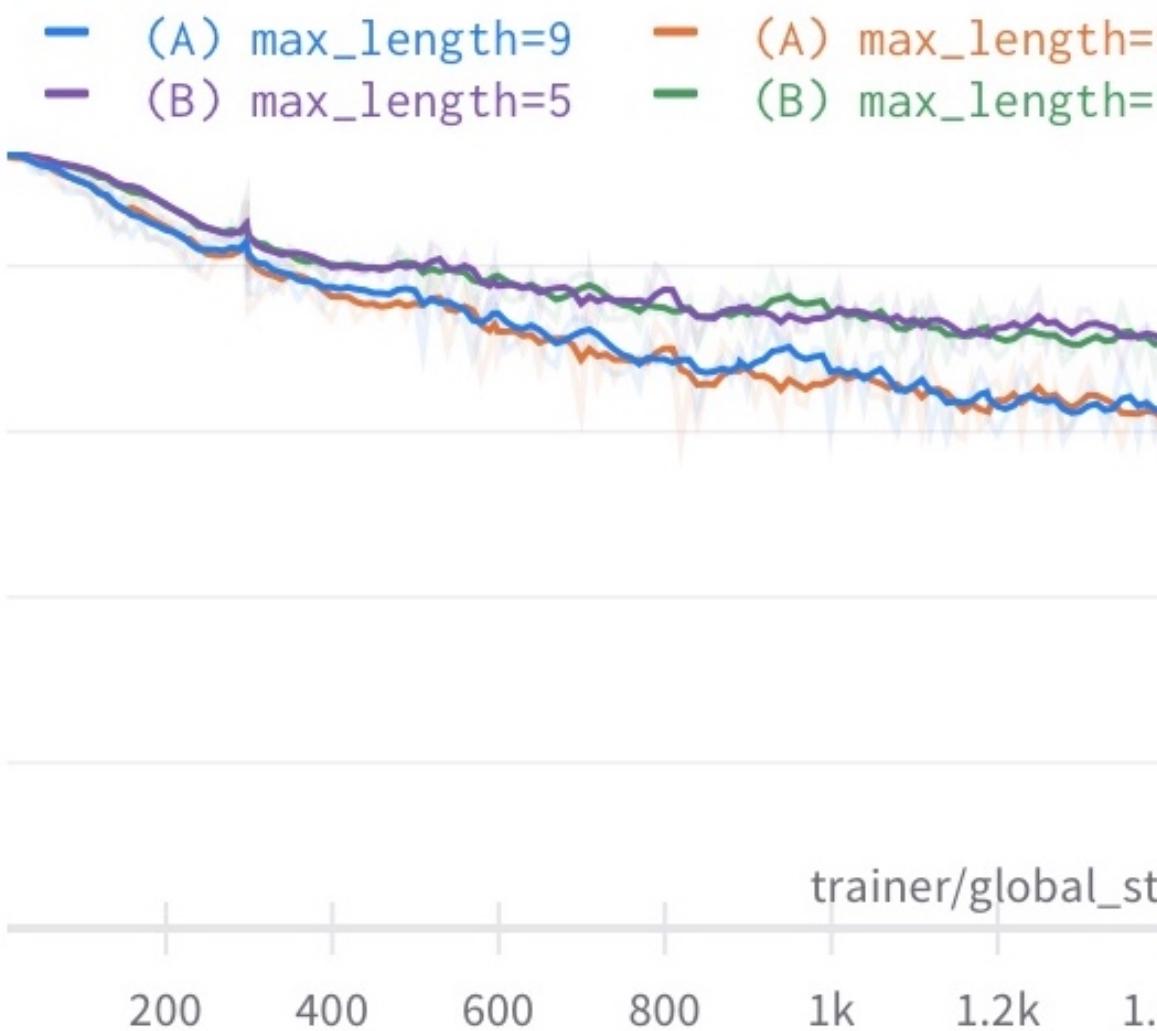
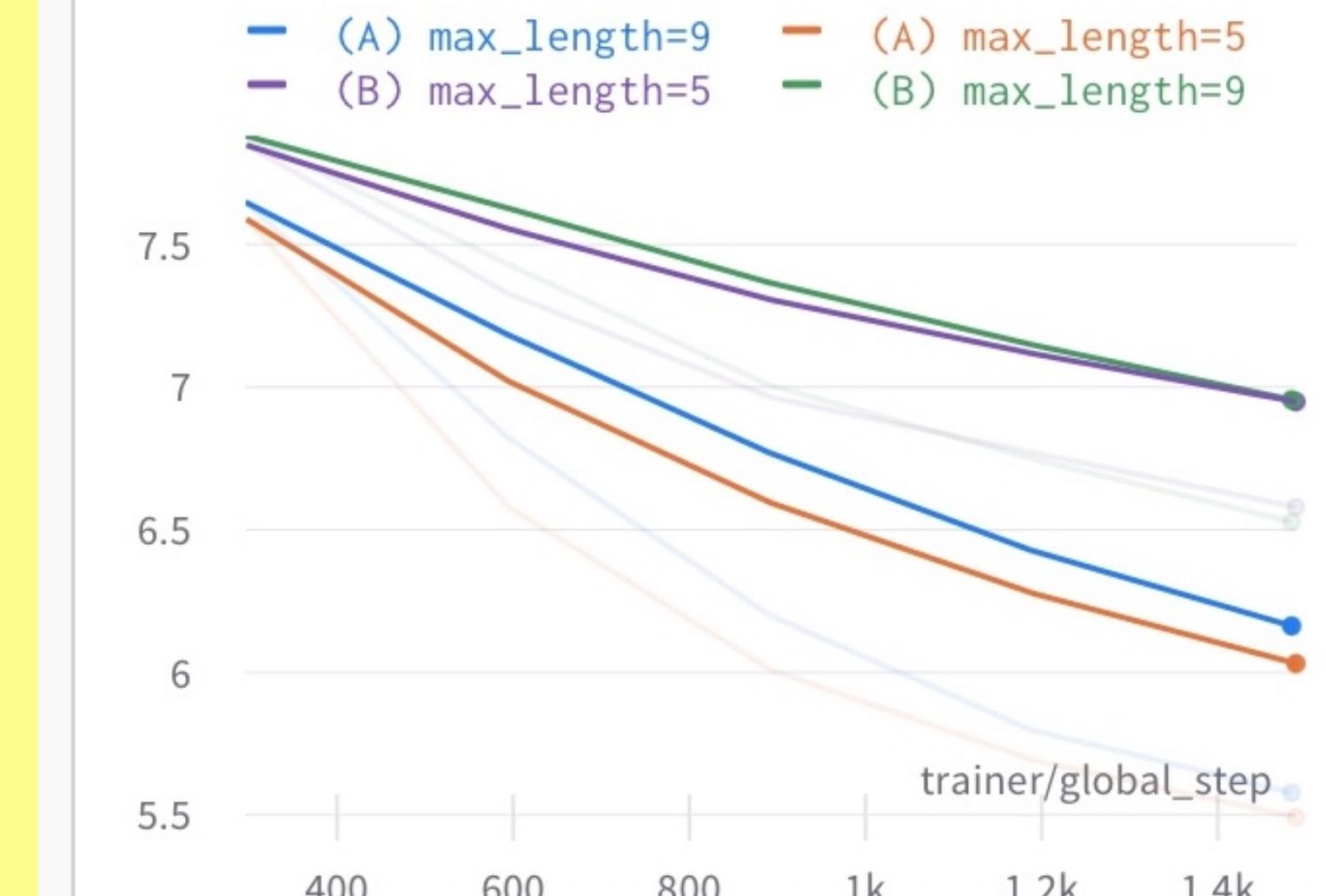
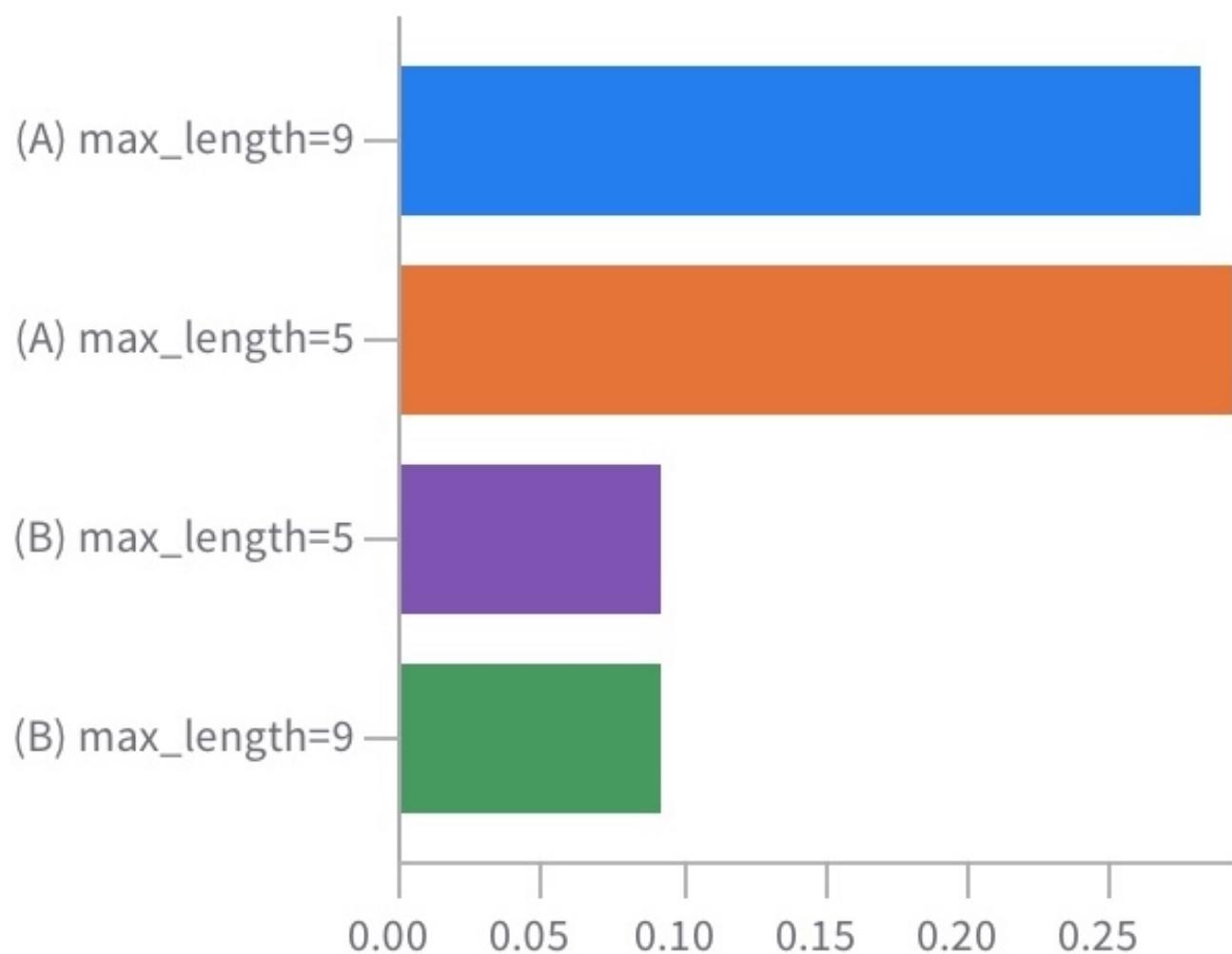
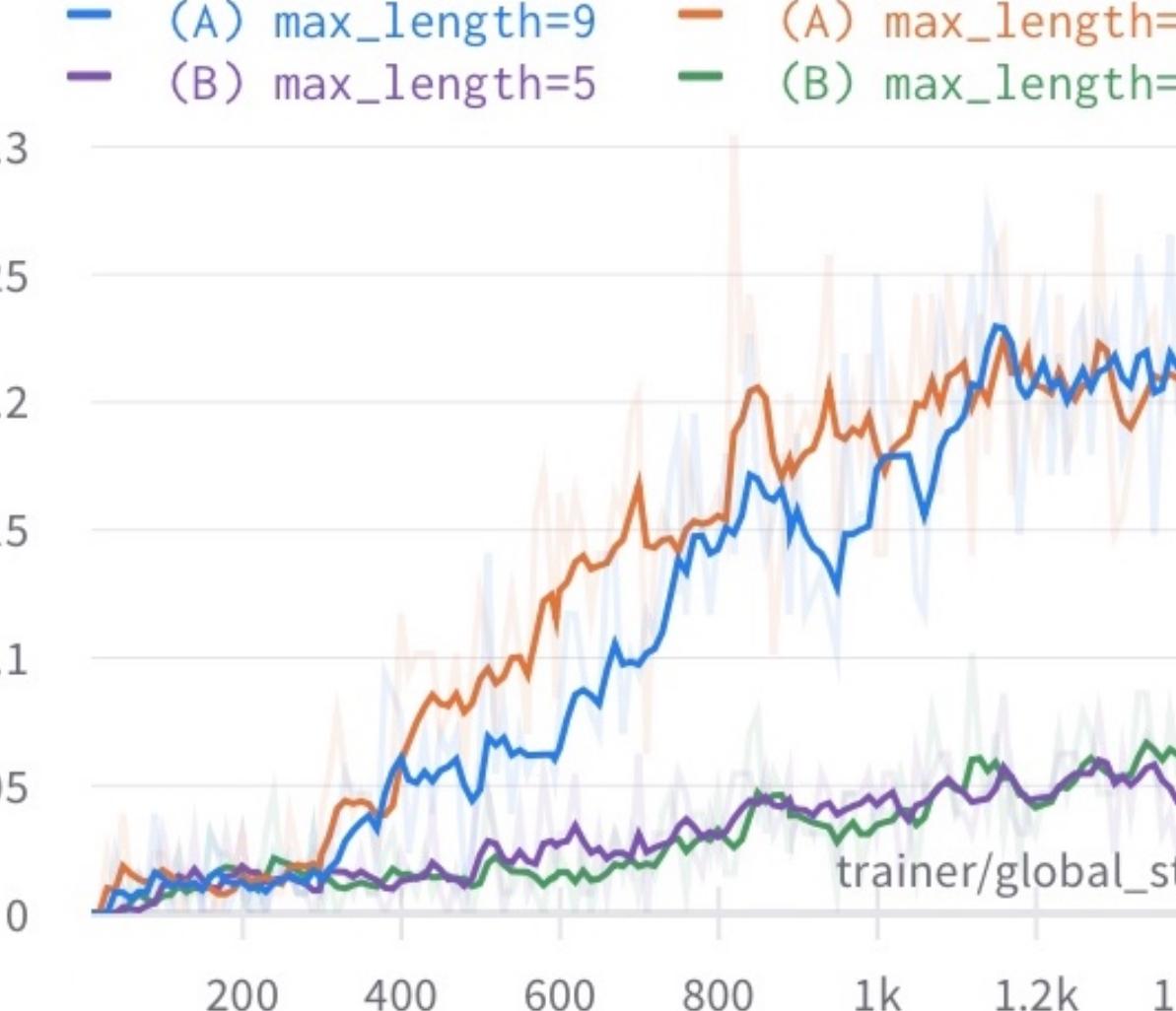
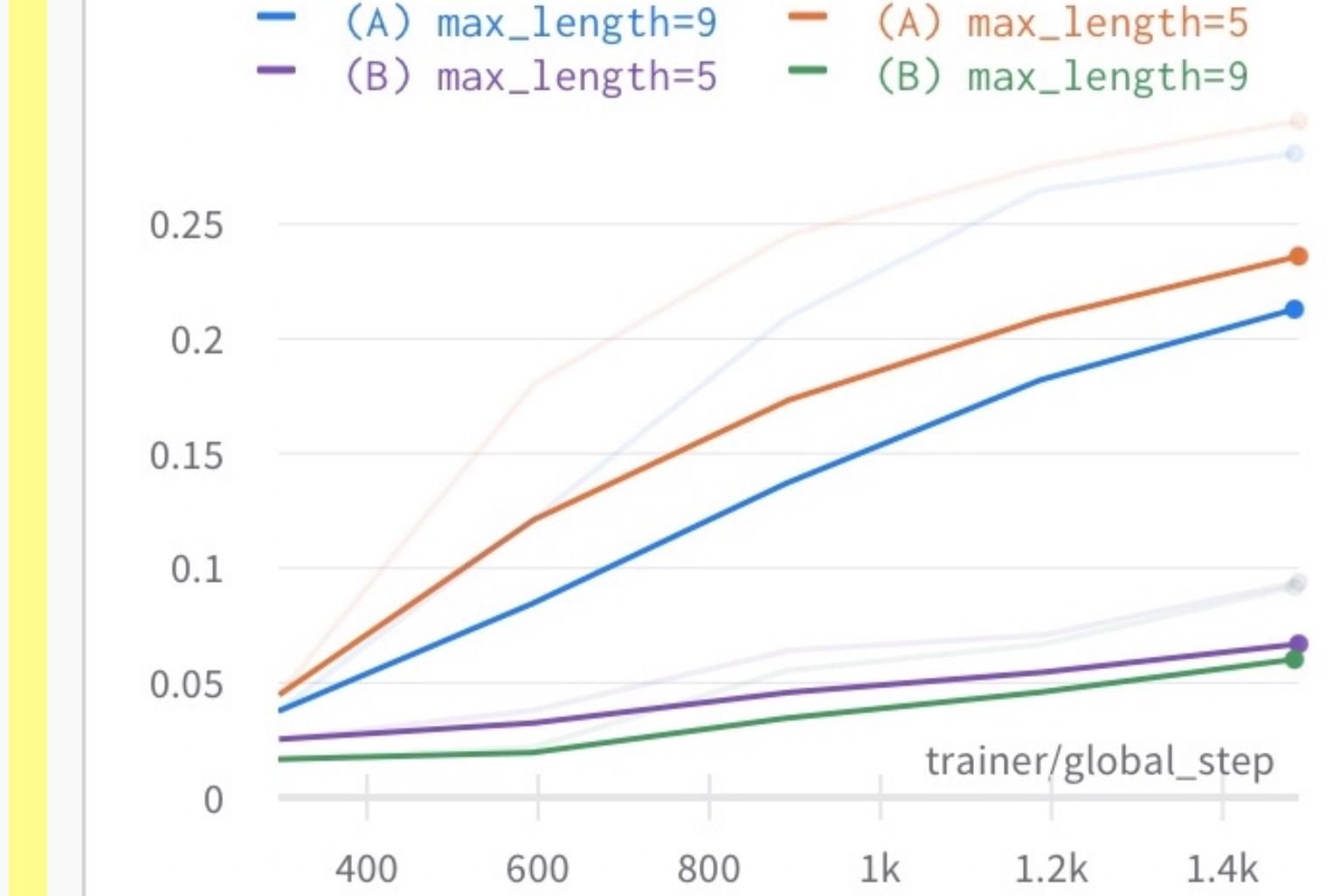
Space to show Event proposals

from\_token → to\_token:

Свойство → ТекущиеДанные

Топ-5:

1. Элементы
2. Параметры
3. Вставить
4. Свойство
5. НСтр

**test/loss****train/loss****val/loss****test/precision****train/precision****val/precision**

# Использованные технологии

---

