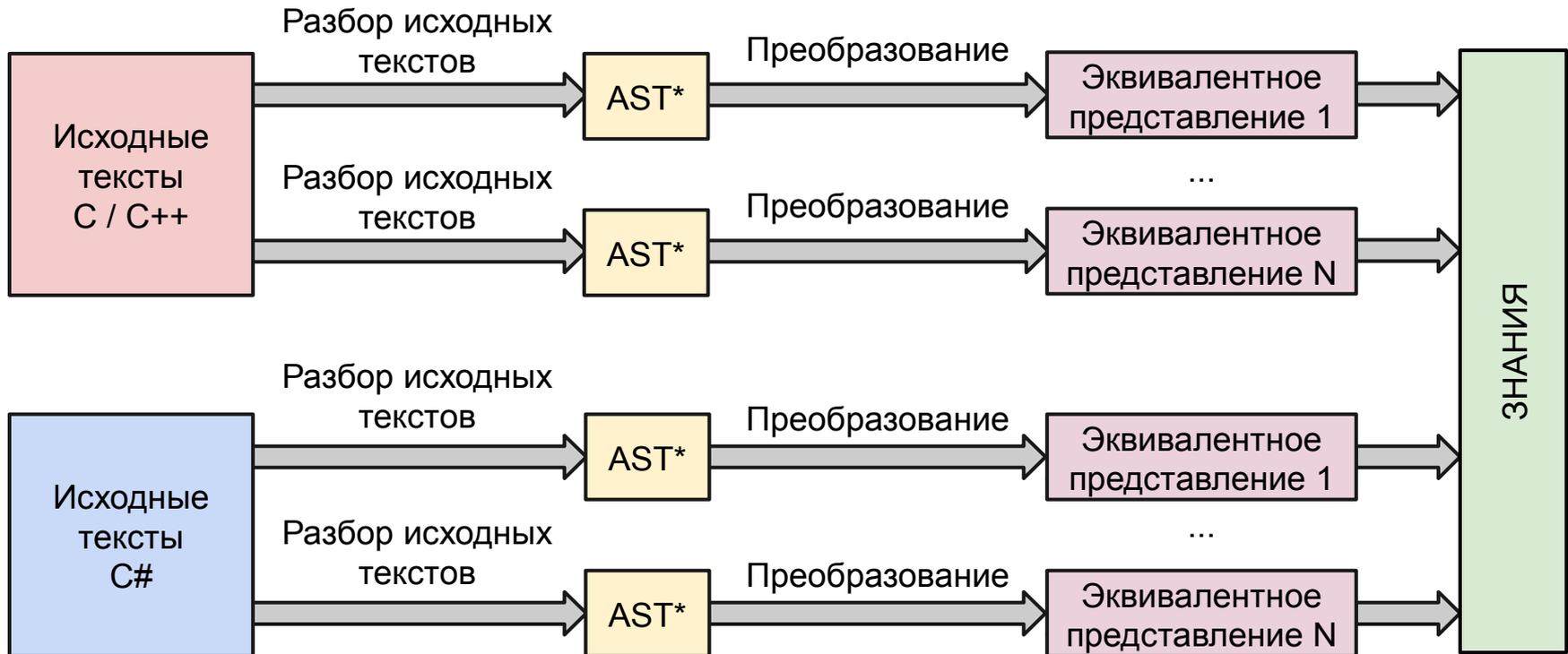


**Построение эквивалентного
представления исходных текстов
программ в форме графа зависимостей
классов, полей, методов, функций и их
перекрестного использования**

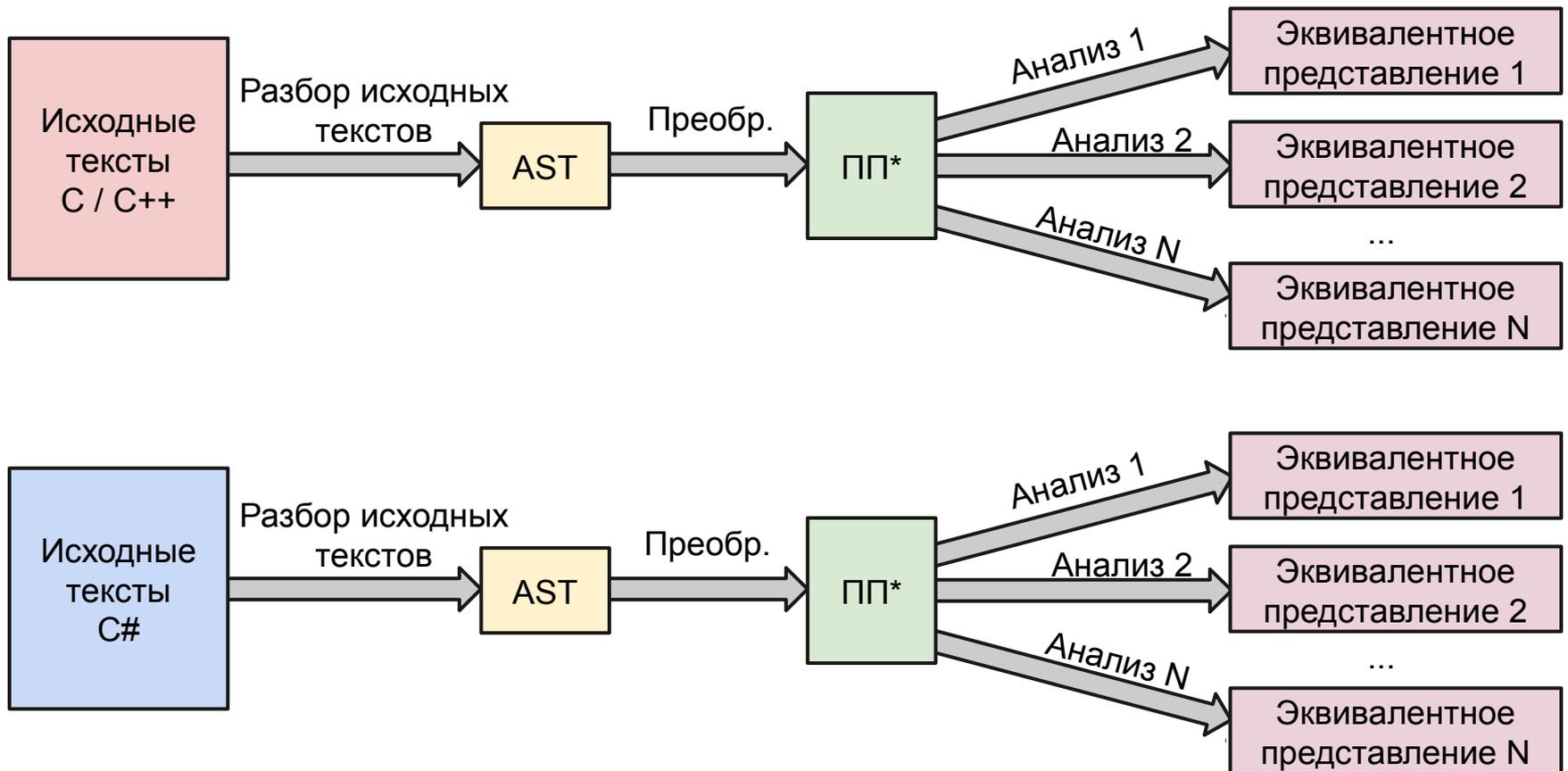
**Пустыгин А.Н., Ковалевский А.А., Белоусов И.С.
Челябинский Государственный Университет**

Проблема статического анализа МНОГОЯЗЫЧНЫХ ПРОЕКТОВ



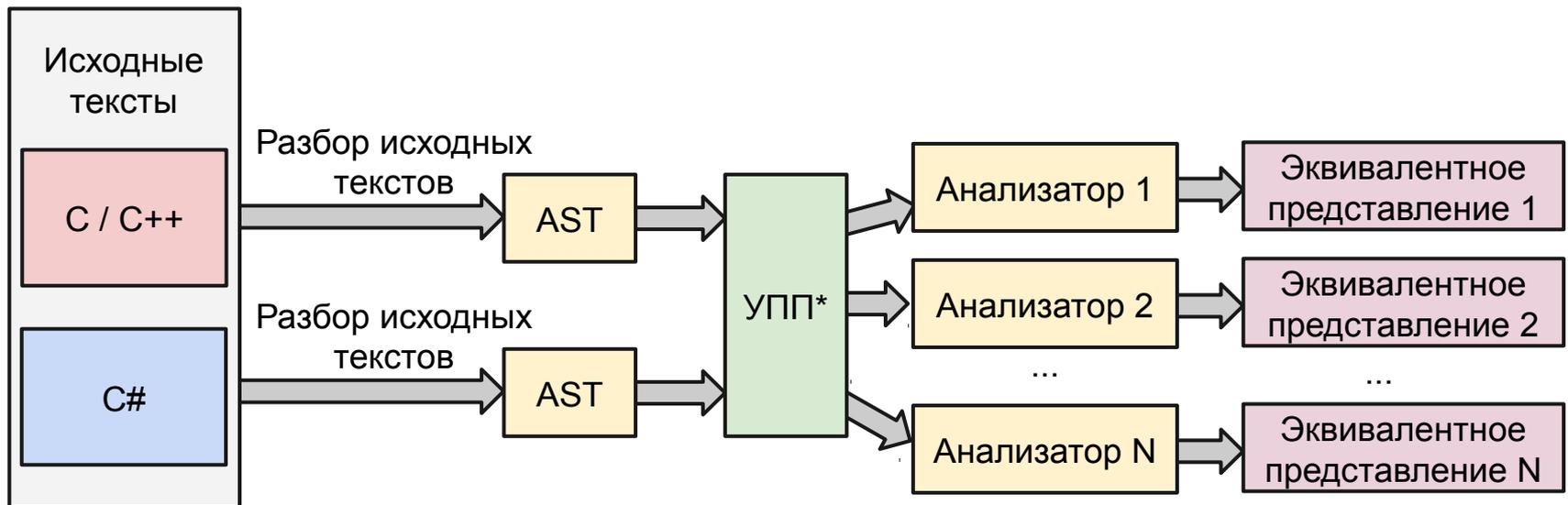
AST* - набор данных, являющихся результатом синтаксического разбора

Плохое решение



ПП - промежуточное представление

Универсальное промежуточное представление исходного текста



УПП - Универсальное промежуточное представление

Для решения поставленной задачи следует разработать

- Формат эквивалентного представления в форме графа зависимостей классов, полей, методов, функций и их перекрестного использования
- Прототип генератора сериализации представления в текстовый формат

Состав эквивалентного представления

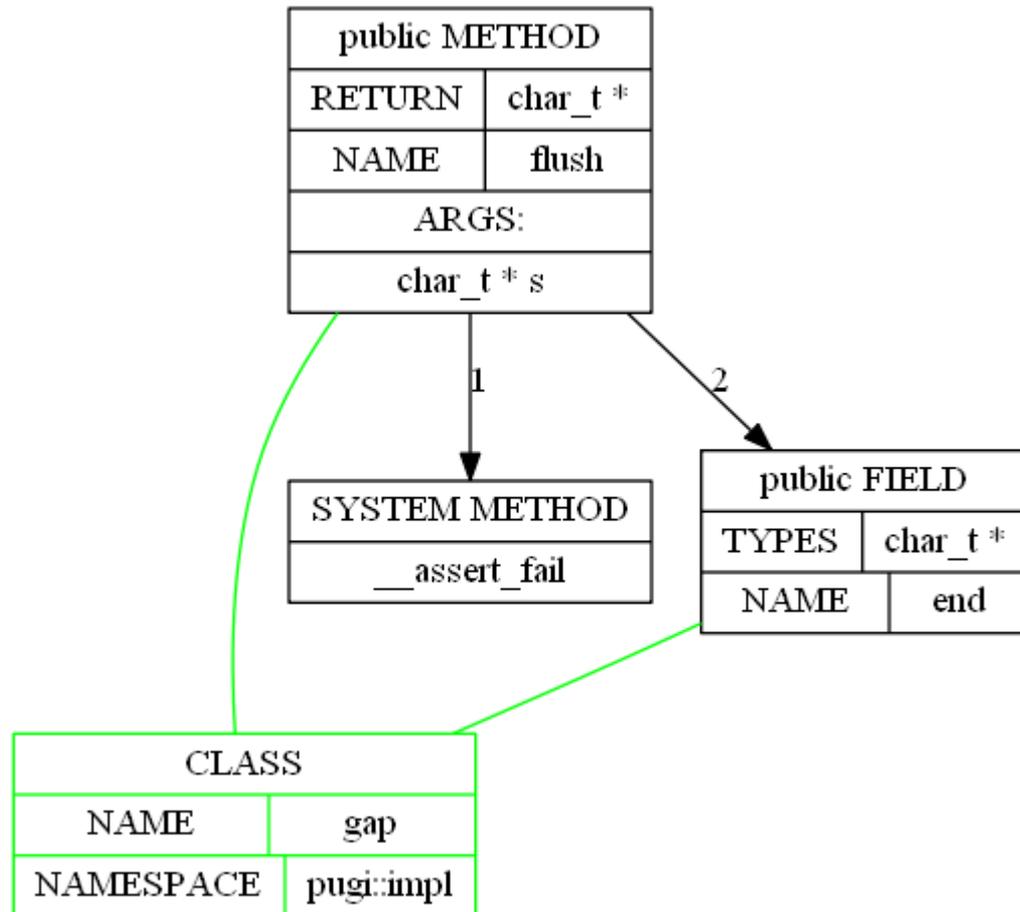
Эквивалентное представление содержит:

- Узлы классов, полей, методов, функций
- Связи отношений наследования классов
- Связи принадлежности методов и членов родительскому классу
- Связи перекрестного использования полей, методов и функций

Формат эквивалентного представления

- Текстовый формат эквивалентного представления основан на нотации языка описания графов DOT, позволяющий использовать инструменты для визуализации графов (Graphviz, dot)

Визуализация фрагмента графа зависимостей, записанного в формате DOT



Описание фрагмента эквивалентного представления

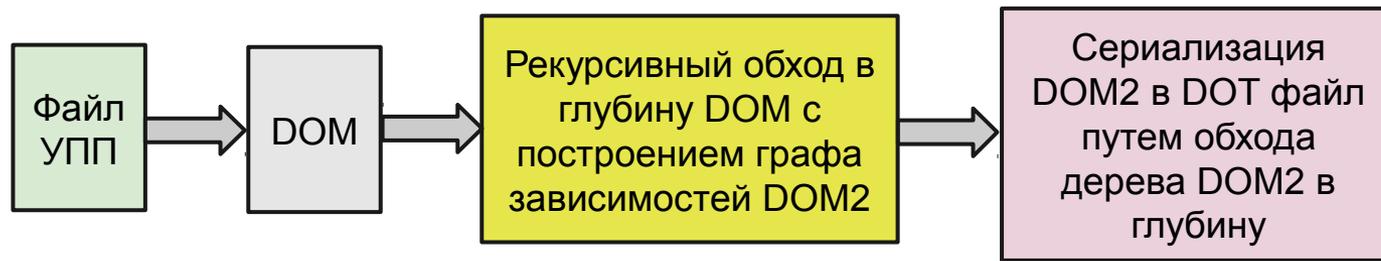
В классе «gar» из пространства имен «pugi::impl» определен метод «flush» с аргументом «s» типа «char_t*» и типом возвращаемого значения «char_t*», который содержит системный вызов «__assert_fail» без аргументов и дважды использует публичное поле «end» объекта класса «gar»

Фрагмент эквивалентного представления в формате DOT

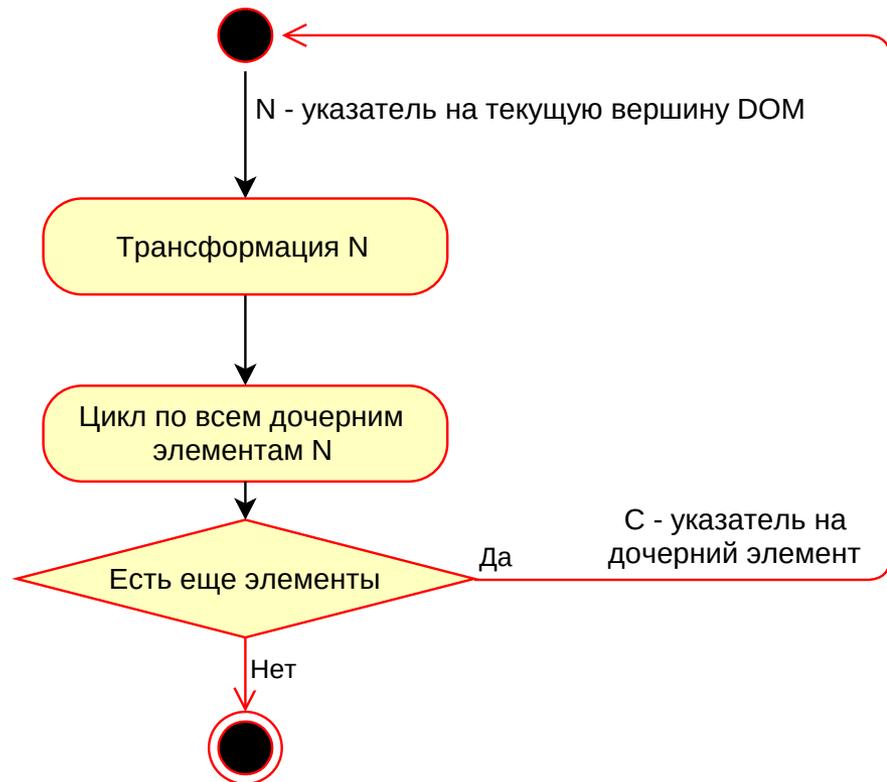
```
digraph G{
#блок класса
"a09452d0CD" [shape=record,color="green",label="{CLASS|{NAME|gap}|{NAMESPACE|pugi::impl}}"];
#блок метода flush
"a0945970MD" [shape=record,label="{public METHOD|{RETURN|char_t * }|{NAME|flush}| ARGES:|
char_t * s}"];
{ "a0945970MD" -> "a09452d0CD" [dir="none", color="green"]; }
}
edge [color="black",fontcolor="black"];
"a0945970MD" -> "a08e8490MD" [label=1]
}
#блок поля end
"a0945480FD" [shape=record,label="{public FIELD|{TYPES|char_t * }|{NAME|end}}"];
{ "a0945480FD" -> "a09452d0CD" [dir="none", color="green"]; }
}
edge [color="black",fontcolor="black"];
"a0945970MD" -> "a0945480FD" [label=2]
}
#блок системного вызова
"a08e8490MD" [shape=record,label="{SYSTEM METHOD|__assert_fail}"];
}
```

Трансформация УПП в форме графа зависимостей

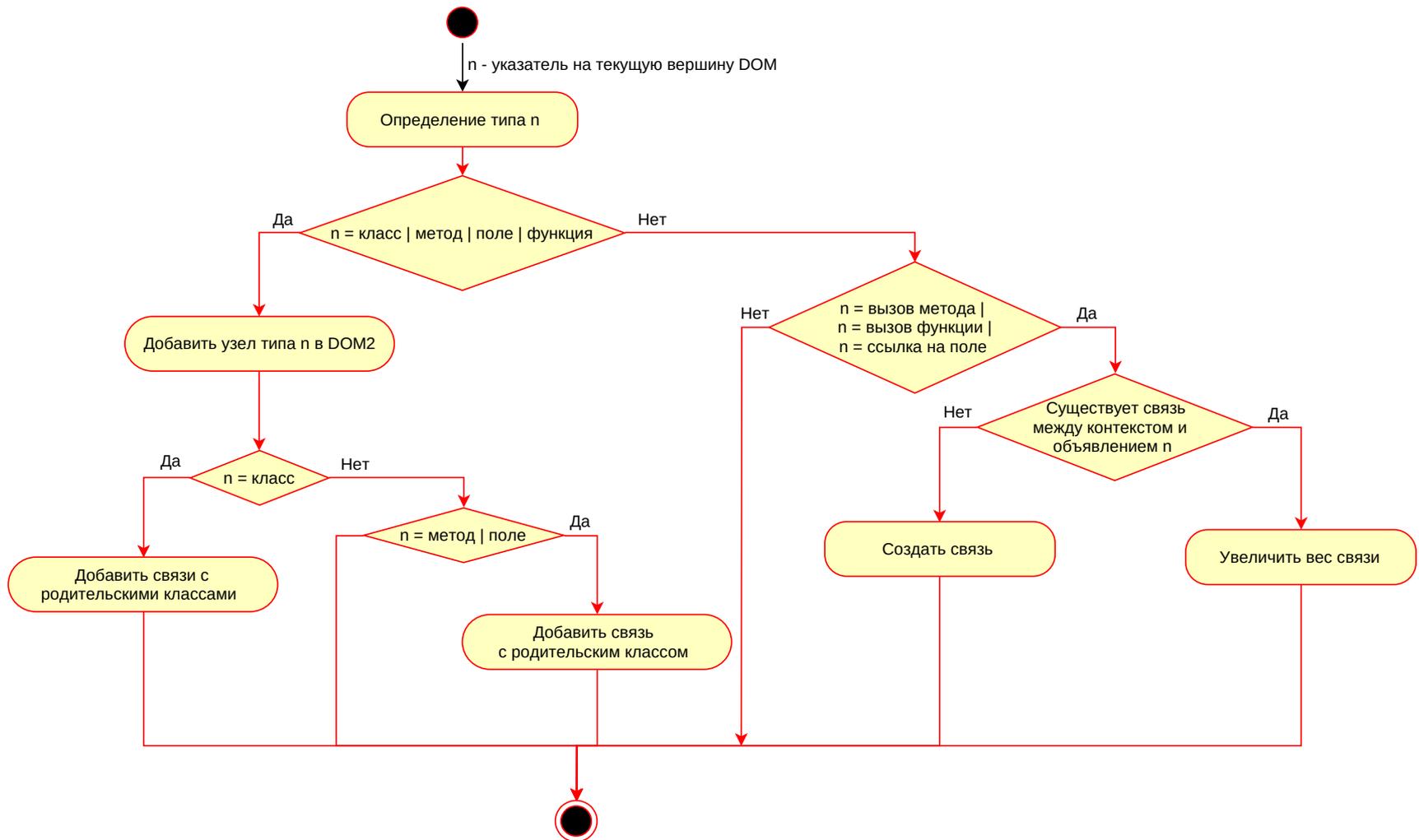
Алгоритм генерации графа зависимостей двухпроходный. Файл УПП загружается в память, преобразуясь в DOM, которая рекурсивно обходится в глубину и информация о каждом узле из контекста графа зависимостей сохраняется в индексном наборе данных DOM2. После обхода происходит генерация графа зависимостей в файл на языке описания графов DOT путем обхода дерева DOM2 в глубину.



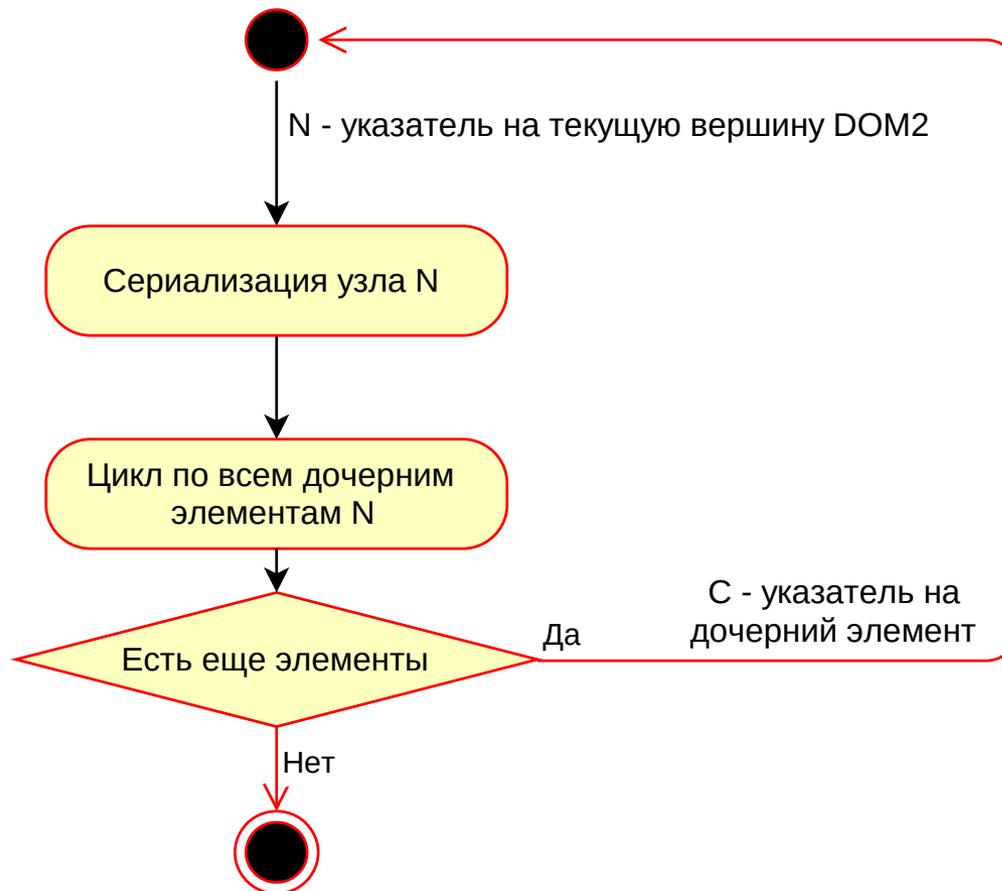
Алгоритм построением графа зависимостей DOM2



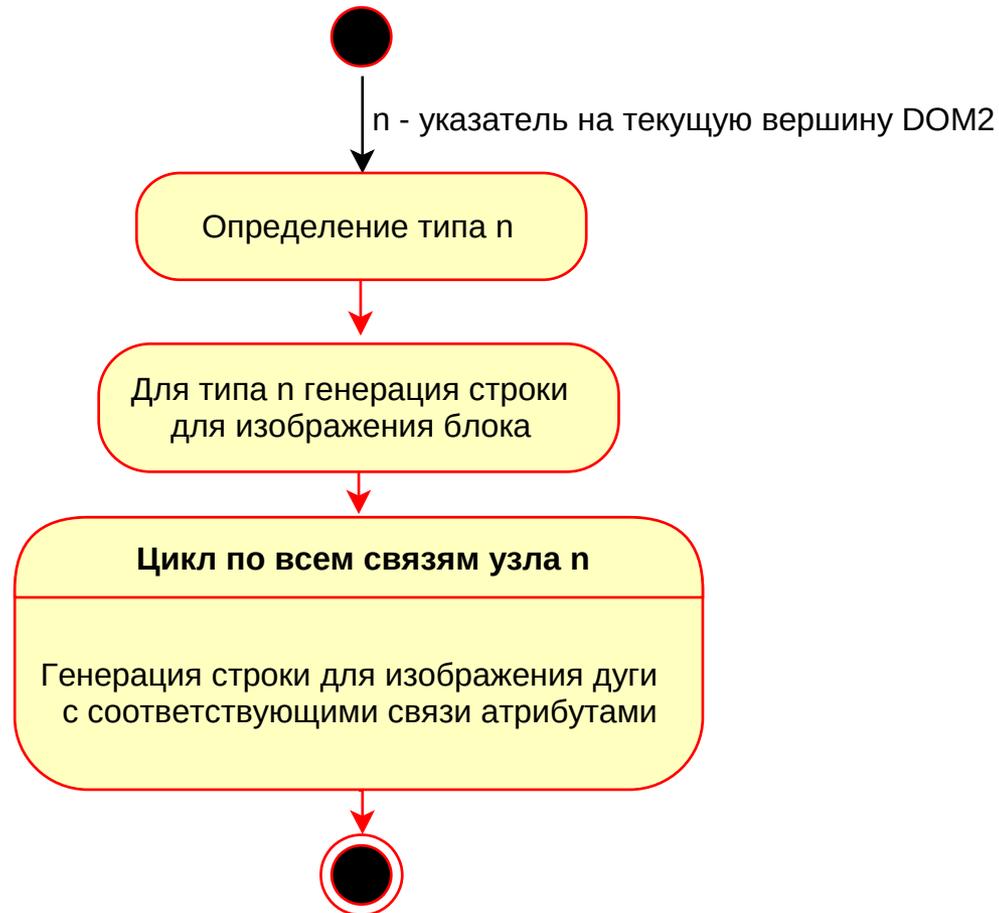
Алгоритм блока трансформации при построении DOM2



Алгоритм сериализации DOM2



Алгоритм сериализации узла DOM2



Результаты

- Предложено эквивалентное представление в форме графа зависимостей классов, полей, методов, функций и их перекрестного использования.
- Разработан прототип генератора представления в формате языка описания графов DOT