Проект: Представление данных в S-выражениях на Ruby

Исполнители: Витченко Владислав (20222), Иванишкин Дмитрий (20222)

Описание задачи

Формат данных

Условие: Определите формат представления данных с древовидной структурой в виде S-выражений. В качестве семантической основы рекомендуется рассмотреть форматы XML и/или JSON. Допускается использование как классических S-выражений, так и расширенных, используемых в языке Clojure.

Подход к решение:

В качестве семантической основы выбран формат XML(SXML).

Пример трансформации данных из XML формата - в формат S-выражений проекта

<fruits>

<fruit weight="5" fresh="true">

<name>apple</name>

<color>green</color>

</fruit>

<fruit amount="20">

<name>orange</name>

<color>orange</color>

</fruit>

</fruits>

(fruits

(fruit (@ (fresh true) (weight 5))

(name “apple”)

(color “green”)

)

(fruit (@ (amount 20))

(name “orange”)

(color “orange”)

)

)

* Атрибуты узла записываются с использованием синтаксиса (@ (attrName arrValue) …)
* Пара ключ значение (key value)

Язык запросов

Условие: Разработайте язык навигации и простых запросов для заданного формата. Должны поддерживаться: относительный или абсолютный путь к узлу дерева, путь с условиями на свойства промежуточных узлов, путь с переменной вложенностью. В качестве семантической основы рекомендуется рассмотреть язык XPath для XML. Реализуйте функции поиска и модификации заданного формата посредством этого языка.

Подход к решению:

2.1) относительный или абсолютный путь к узлу дерева

* name - относительный поиск элемента(элементов)
* fruit/name - относительный поиск элемента(элементов)
* /fruits/fruit/name - абсолютный элемент

2.2) с условиями на свойства промежуточных узлов

* Условия на параметры узла задаются в квадратных скобках (/fruits/fruit[fresh=true])
* Условия на значение узла задаются в квадратных скобках без значения параметра (/fruits/fruit/name[“apple”])
* Можно использовать несколько условий одновременно (/fruits/fruit[weight=5][fresh=true])

2.3) Путь с переменной вложенностью.

* /fruits/\*/name
* в качестве звездочки можно использовать любую последовательность в дереве выражения

Представление схемы

Условие:

Разработайте представление схемы (по аналогии с XML Schema). Реализуйте функции проверки документа по схеме.

Подход к решению:

Схема проверки документа может быть примерно такой

(element (& (name fruits) (type sequence))

(element (& (name fruit) (type sequence))

(element (& (name name) (type string)))

(element (& (name color) (type string)))

)

(element (& (name fruit) (type sequence))

(element (& (name name) (type string)))

(element (& (name color) (type string)))

)

)

Выделяется несколько типов – атомарные(Boolean, integer, string). И составной element – который может содержать в себе другие составные типы или атома. Sequence – набор element’ов. Директива type – проверяет тип. Директива name – проверяет имя узла.

Программная реализация

В качестве структуры хранения документа предлагается использовать ассоциативный массив/hash(Ruby).

Для алгоритма поиска будет использован алгоритм поиска в ширину.

Ссылки:

* <http://msdn.microsoft.com/ru-ru/library/vstudio/ms256086(v=vs.100).aspx>
* <https://igor.io/2012/12/06/sexpr.html>
* <http://en.wikipedia.org/wiki/SXML>
* <https://en.wikipedia.org/wiki/XPath>