LDX User Technical Guide

LDX (Language for Data eXploration) is a specification language that extends Tregex, a query language for tree-structured data. It allows you to partially specify structural properties of a tree, as well as the nodes' labels. The language is especially useful for specifying the order of notebook's query operations and their type and parameters. This guide is divided into two main sections: nodes relationships and nodes operations.

1 Nodes Relationships

Nodes relationships in LDX define the structural properties between different nodes. Here are some of the most common relationships:

Children Relationship:

Expression: `A CHILDREN <B,C>`

Description: B and C are the only children of A in the specified order.

Siblings Relationship:

Expression: `A SIBLINGS {B,C}`

Description: B and C are siblings of A.

Descendants Relationship:

Expression: `A DESCENDANTS {B,C}`

Description: B and C are descendants of A.

Unordered Relationship:

Expression: `A DESCENDANTS {B,C, *}`

Description: B and C are descendants of A, not necessarily in any specific order.

Relationship With Additional Unnamed Nodes:

Expression: `A DESCENDANTS {B,C, *}`

Description: B and C are descendants of A, in no particular order, with potentially more unnamed descendants.

2 Nodes Operations

Operations in LDX are used to define actions performed on the nodes using the LIKE operator. Nodes operations can be categorized into two types: simple operations and special operations.

2.1 Simple Operations

Filter Operation:

Expression: `A LIKE [F,category,eq,comedy]`

Description: A is an equality filter on 'category', where the filter term is the string 'comedy'.

Group-By Operation:

Expression: `A LIKE [G,category,CNT,show_id]`

Description: A is a group-by operation that shows the numbers of shows for each category.

2.2 Special Operations

Regular Filter:

Expression: `A LIKE [F,category,ne,.*]`

Description: A is a non-equality filter on 'category', where the filter term is some term.

Regular Group-By:

Expression: `A LIKE [G,.*,AVERAGE,.*]`

Description: A is a group-by operation on some column, employing average on some column.

Filter Using Continuity Variable:

Expression:
```
A LIKE [F,(?<col>.*),.*]
B LIKE [G,(?<col>.*),.*]
```

Description: This expression uses a continuity variable which is like '.*' but also stores the value. In this example, A is a filter operation on some column, and B is a group-by operation on the same column.

Group-By Using Continuity Variable:

Expression:
```
A LIKE [G,.*,(?<func>.*),(?<col>.*)]
B LIKE [G,.*,(?<func>.*),(?<col>.*)]
```

Description: A and B are group-by operations with the same aggregation function and column.

Advanced Example:

Expression:
```
A LIKE [F,(?<col>.*delay.*),ge,(?<term>[0-9]{3,4})]
B LIKE [F,(?<col>.*),le,(?<term>.*)]
```

Description: This is a more complex example that uses regex and continuity variables. A is a filter operation on some column that includes the word 'delay' and is greater than or equal to some value between 100 to 1000. B is the same action but with less than or equal condition.