**NMATH**

**Build AST**

Interval Expression

We will use criteria to express the domain and range of a function. To do this job I break our criteria into 2 kinds. The first one is SimpleCriteria and another is CompositeCriteria.

Domain

For convenience, below rules shall be applied in processing interval domain.

* Variables are always stored in variable field of SimpleCriteria.

Interval 1

This is the most simple criteria

Type: **LT** | **LTE** | **GT** | **GTE**

Variable: <<**variable**>>

Value: <<**value**>>

Interval 2

The NMAST for above interval will look like:

Type: **GT\_LT | GTE\_LT | GT\_LTE | GTE\_LTE**

Variable: **x**

Type: **NUMBER | PI\_TYPE | E\_TYPE**

Value: left\_bound

Type: **NUMBER | PI\_TYPE | E\_TYPE**

Value: right\_bound

For example:

Interval 2

Type: **GT\_LT | GTE\_LT | GT\_LTE | GTE\_LTE**

Variable: **x**

Type: **NUMBER | PI\_TYPE | E\_TYPE**

Value: left\_bound

Type: **NUMBER | PI\_TYPE | E\_TYPE**

Value: right\_bound

**Get normal vector of a function f**

Base on derivative of the function f, we can calc normal vector of f at specified point M on f. First we determine the tangent of f at point M by:

**Automatic finding the domain of a function**

For example if we got a function like :

So what is the domain of this function?

* Browse the prefix tree of the input function.
* If a DIV is found:
  + Determine if a variable exists in denominator
  + If a variable exists in denominator and the denominator has the form of then we solve the equation: *Denominator = 0* to find the indefinitive points in the function’s ranges. After that we build the criteria base on these indefinitive points and AND this criteria with existed criteria of the input function.

API Specification

Class NFunction

|  |  |  |
| --- | --- | --- |
| **Return type** | **Prototype** | **Description** |
| void | ***buildCompositeCriteria***(const NMAST \*ast, const char \*vars, int varCount, OutBuiltCriteria \*\*outCriteria) | Convert a NMAST tree into a criteria. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |