Сервис CLI: разбор ввода пользователя и построения цепочки команд

Лексический анализ:

Letter = A | … | Z | a | … | z

Digit = 0 | … | 9

WordCharacter = Letter | Digit | '\' | '.' | ',' | '-'

Word = WordCharacter {WordCharacter}

Character = …

String = '"' Character {Character} '"'

Синтаксический анализ:

Command = Word {Word | String}

LL(1) синтаксический анализ. Пример 1:

Изначальные правила:

1) *Command* = **interface** **range** *Args*

2) *Args* = *Arg* *Args* | **''**

3) *Arg* = **Word** | **String**

Преобразованные правила:

*1) Command* = **interface** *CommandRest*

*2) CommandRest* = **range** *Args*

*3) Args* = ***Word*** *Args* | ***String*** *Args* | **''**

FIRST(*Command*) = {**interface**}

FOLLOW(*Command*) = {**'$'**}

FIRST(*CommandRest*) = {**range**}

FOLLOW(*CommandRest*) = {**'$'**}

FIRST(*Args*) = {**Word**, **String**, **''**}

FOLLOW(*Args*) = {**'$'**}

M[*Command*, **interface**] = 1

M[*CommandRest*, **range**] = 2

M[*Args*, **Word**] = 3(1)

M[*Args*, **String**] = 3(2)

M[*Args*, **'$'**] = 3(3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **interface** | **range** | **Word** | **String** | **'$'** |
| *Command* | 1 |  |  |  |  |
| *CommandRest* |  | 2 |  |  |  |
| *Args* |  |  | 3(1) | 3(2) | 3(3) |

LL(1) синтаксический анализ. Пример 2:

Изначальные правила:

1) *Command* = **interface** *Args* | **interface** **range** *Args*

2) *Args* = *Arg* *Args* | **''**

3) *Arg* = **Word** | **String**

Преобразованные правила:

1) *Command* = **interface** *CommandRest*

2) CommandRest = *Args* | **range** *Args*

3) *Args* = ***Word*** *Args* | ***String*** *Args* | **''**

FIRST(*Command*) = {**interface**}

FOLLOW(*Command*) = {**'$'**}

FIRST(*CommandRest*) = {**Word**, **String**, **''**, **range**}

FOLLOW(*CommandRest*) = {**'$'**}

FIRST(*Args*) = {**Word**, **String**, **''**}

FOLLOW(*Args*) = {**'$'**}

M[*Command*, **interface**] = 1

M[*CommandRest*, **Word**] = 2(1)

M[*CommandRest*, **String**] = 2(1)

M[*CommandRest*, **'$'**] = 2(1)

M[*CommandRest*, **range**] = 2(2)

M[*Args*, **Word**] = 3(1)

M[*Args*, **String**] = 3(2)

M[*Args*, **'$'**] = 3(3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **interface** | **range** | **Word** | **String** | **'$'** |
| *Command* | 1 |  |  |  |  |
| *CommandRest* |  | 2(2) | 2(1) | 2(1) | 2(1) |
| *Args* |  |  | 3(1) | 3(2) | 3(3) |

LL(1) синтаксический анализ. Пример 3:

Изначальные правила:

1) *Command1* = **interface** *Args*

2) *Command2* = **interface** **range** *Args*

3) *Args* = *Arg* *Args* | **''**

4) *Arg* = **Word** | **String**

Преобразованные правила:

1) *Command1* = **interface** *CommandRest1*

2) *CommandRest1* = *Args*

3) *Command2* = **interface** *CommandRest2*

4) *CommandRest2* = **range** *CommandRest3*

5) *CommandRest3* = *Args*

6) *Args* = ***Word*** *Args* | ***String*** *Args* | **''**

FIRST(*Command1*) = {**interface**}

FOLLOW(*Command1*) = {**'$'**}

FIRST(*CommandRest1*) = {**Word**, **String**, **''**}

FOLLOW(*CommandRest1*) = {**'$'**}

FIRST(*Command2*) = {**interface**}

FOLLOW(*Command2*) = {**'$'**}

FIRST(*CommandRest2*) = {**range**}

FOLLOW(*CommandRest2*) = {**'$'**}

FIRST(*CommandRest3*) = {**Word**, **String**, **''**}

FOLLOW(*CommandRest3*) = {**'$'**}

FIRST(*Args*) = {**Word**, **String**, **''**}

FOLLOW(*Args*) = {**'$'**}

M[*Command1*, **interface**] = 1

M[*CommandRest1*, **Word**] = 2

M[*CommandRest1*, **String**] = 2

M[*CommandRest1*, **'$'**] = 2

M[*Command2*, **interface**] = 3

M[*CommandRest2*, **range**] = 4

M[*CommandRest3*, **Word**] = 5

M[*CommandRest3*, **String**] = 5

M[*CommandRest3*, **'$'**] = 5

M[*Args*, **Word**] = 6(1)

M[*Args*, **String**] = 6(2)

M[*Args*, **'$'**] = 6(3)

Общая таблица синтаксического анализа:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **interface** | **range** | **Word** | **String** | **'$'** |
| *Command1* | 1 |  |  |  |  |
| *CommandRest1* |  |  | 2 | 2 | 2 |
| *Command2* | 3 |  |  |  |  |
| *CommandRest2* |  | 4 |  |  |  |
| *CommandRest3* |  |  | 5 | 5 | 5 |
| *Args* |  |  | 6(1) | 6(2) | 6(3) |

Объединим *Command1* и *Command2*:

1) Правила 1 и 3 объединяем в одно

2) Правило 2 и 4 объединяем так, чтобы правые части правил стали вариантами

В итоге получаем следующий набор правил:

1) *Command* = **interface** *CommandRest*

2) *CommandRest* = *Args* | **range** *CommandRest2*

3) *CommandRest2* = *Args*

4) *Args* = ***Word*** *Args* | ***String*** *Args* | **''**

Для таблицы синтаксического анализа это приведет к объединению строк 1 и 3, 2 и 4:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **interface** | **range** | **Word** | **String** | **'$'** |
| *Command* | 1 |  |  |  |  |
| *CommandRest* |  | 2(2) | 2(1) | 2(1) | 2(1) |
| *CommandRest2* |  |  | 3 | 3 | 3 |
| *Args* |  |  | 4(1) | 4(2) | 4(3) |

LL(1) синтаксический анализ. Пример 4:

Изначальные правила:

1) *Command1* = **ping** *Args*

2) *Command2* = **interface** *Args*

3) *Command3* = **interface** **range** *Args*

3) *Args* = *Arg* *Args* | **''**

4) *Arg* = **Word** | **String**

Преобразованные правила:

1) *Command1* = **ping** *CommandRest1*

2) *CommandRest1* = *Args*

3) *Command2* = **interface** *CommandRest2*

4) *CommandRest2* = *Args*

5) *Command3* = **interface** *CommandRest3*

6) *CommandRest3* = **range** *CommandRest4*

7) *CommandRest4* = *Args*

8) *Args* = ***Word*** *Args* | ***String*** *Args* | **''**

FIRST(*Command1*) = {**ping**}

FOLLOW(*Command1*) = {**'$'**}

FIRST(*CommandRest1*) = {**Word**, **String**, **''**}

FOLLOW(*CommandRest1*) = {**'$'**}

FIRST(*Command2*) = {**interface**}

FOLLOW(*Command2*) = {**'$'**}

FIRST(*CommandRest2*) = {**Word**, **String**, **''**}

FOLLOW(*CommandRest2*) = {**'$'**}

FIRST(*Command3*) = {**interface**}

FOLLOW(*Command3*) = {**'$'**}

FIRST(*CommandRest3*) = {**range**}

FOLLOW(*CommandRest3*) = {**'$'**}

FIRST(*CommandRest4*) = {**Word**, **String**, **''**}

FOLLOW(*CommandRest4*) = {**'$'**}

FIRST(*Args*) = {**Word**, **String**, **''**}

FOLLOW(*Args*) = {**'$'**}

M[*Command1*, **ping**] = 1

M[*CommandRest1*, **Word**] = 2

M[*CommandRest1*, **String**] = 2

M[*CommandRest1*, **'$'**] = 2

M[*Command2*, **interface**] = 3

M[*CommandRest2*, **Word**] = 4

M[*CommandRest2*, **String**] = 4

M[*CommandRest2*, **'$'**] = 4

M[*Command3*, **interface**] = 5

M[*CommandRest3*, **range**] = 6

M[*CommandRest4*, **Word**] = 7

M[*CommandRest4*, **String**] = 7

M[*CommandRest4*, **'$'**] = 7

M[*Args*, **Word**] = 8(1)

M[*Args*, **String**] = 8(2)

M[*Args*, **'$'**] = 8(3)

Общая таблица синтаксического анализа:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **ping** | **interface** | **range** | **Word** | **String** | **'$'** |
| *Command1* | 1 |  |  |  |  |  |
| *CommandRest1* |  |  |  | 2 | 2 | 2 |
| *Command2* |  | 3 |  |  |  |  |
| *CommandRest2* |  |  |  | 4 | 4 | 4 |
| *Command3* |  | 5 |  |  |  |  |
| *CommandRest3* |  |  | 6 |  |  |  |
| *CommandRest4* |  |  |  | 7 | 7 | 7 |
| *Args* |  |  |  | 8(1) | 8(2) | 8(3) |

Объединим *Command1* и *Command2*:

1) Правила 1, 3, 5 объединяем в одно

2) Правила 4 и 6 объединяем так, чтобы правые части правил стали вариантами

В итоге получаем следующий набор правил:

1) *Command* = **ping** CommandRest1 | **interface** *CommandRest2*

2) *CommandRest1* = *Args*

3) *CommandRest2* = *Args* | **range** *CommandRest3*

4) *CommandRest3* = *Args*

5) *Args* = ***Word*** *Args* | ***String*** *Args* | **''**

Для таблицы синтаксического анализа это приведет к объединению строк 1 и 3, 2 и 4:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **ping** | **interface** | **range** | **Word** | **String** | **'$'** |
| *Command* | 1(1) | 1(2) |  |  |  |  |
| *CommandRest1* |  |  |  | 2 | 2 | 2 |
| *CommandRest2* |  |  | 3(2) | 3(1) | 3(1) | 3(1) |
| *CommandRest3* |  |  |  | 4 | 4 | 4 |
| *Args* |  |  |  | 5(1) | 5(2) | 5(3) |

Правила построения таблицы синтаксического анализа для набора команд:

1. Создаем общее правило для всех команд для описания аргументов следующего вида:

1) *Args* = ***Word*** *Args* | ***String*** *Args* | **''**

Для этого правила имеем следующие множества FIRST и FOLLOW:

FIRST(*Args*) = {**Word**, **String**, **''**}

FOLLOW(*Args*) = {**'$'**}

Таблица синтаксического анализа будет иметь следующий вид:

M[*Args*, **Word**] = 1(1) — правило 1, вариант 1

M[*Args*, **String**] = 1(2) — правило 1, вариант 2

M[*Args*, **'$'**] = 1(3) — правило 1, вариант 3

В виде таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Word** | **String** | **'$'** |
| *Args* | 1(1) | 1(2) | 1(3) |

2. Для каждой команды из N слов (**word1** **word2** … **wordN**) создаем набор правил следующего вида:

0) *Command* = **word1** *CommandRest1*

1) *CommandRest1* = **word2** *CommandRest2*

…

N-1) *CommandRestN-1* = **wordN** *CommandRestN*

N) *CommandRestN* = *Args*

Для этого набора правил имеем следующие множества FIRST и FOLLOW:

FIRST(*Command*) = {**word1**}

FOLLOW(*Command*) = {**'$'**}

FIRST(*CommandRest1*) = {**word2**}

FOLLOW(*CommandRest1*) = {**'$'**}

…

FIRST(*CommandRestN-1*) = {**wordN**}

FOLLOW(*CommandRestN-1*) = {**'$'**}

FIRST(*CommandRestN*) = {**Word**, **String**, **''**}

FOLLOW(*CommandRestN*) = {**'$'**}

Таблица синтаксического анализа будет иметь следующий вид:

M[*Command*, **word1**] = 0 — правило 0

M[*CommandRest1*, **word2**] = 1 — правило 1

…

M[*CommandRestN-1*, **wordN**] = N-1 — правило N-1

M[*CommandRestN*, **Word**] = N — правило N

M[*CommandRestN*, **String**] = N — правило N

M[*CommandRestN*, **'$'**] = N — правило N

В виде таблицы:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **word1** | **word2** | ... | **wordN** | **Word** | **String** | **'$'** |
| *Command* | 0 |  |  |  |  |  |  |
| *CommandRest1* |  | 1 |  |  |  |  |  |
| ... |  |  |  |  |  |  |  |
| *CommandRestN-1* |  |  |  | N-1 |  |  |  |
| *CommandRestN* |  |  |  |  | N | N | N |

3. Объединяем все наборы правил для всех команд.