Exercise: Context Free Grammar with Brackets$_{\text{JP}}$

Simple Sentence

Consider the following context-free grammar that defines a very small subset of valid English sentences. Note that the terminal symbols have the following English interpretations: **a** = “a”, **t** = “the”, **d** = “dog”, **r** = “ran”.

V = { *S, Article, Noun, Subject, Verb* }

T = { a, d, r, t }

S = *S*

S → *Subject Verb*

Subject → *Article Noun*

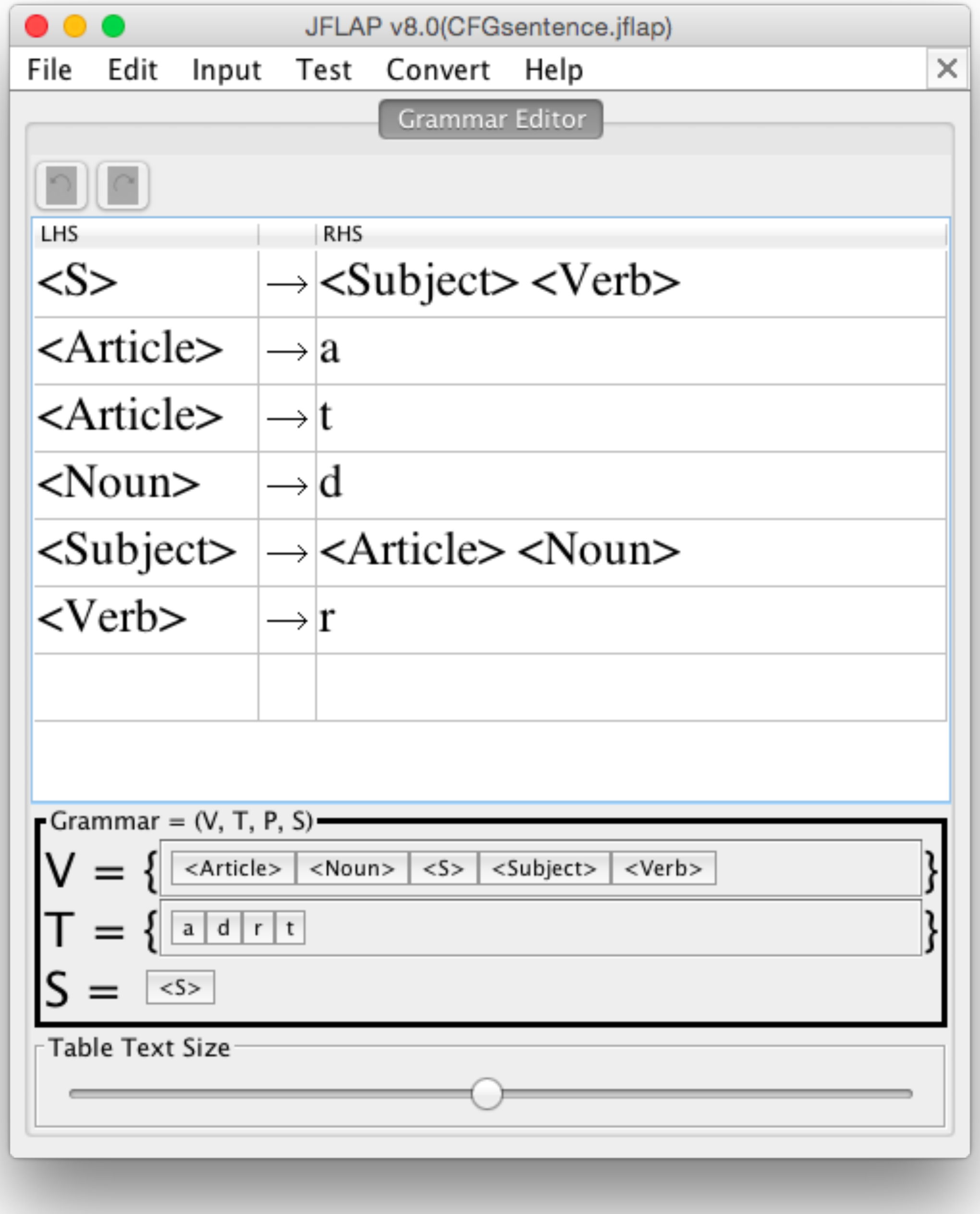
Article → a | t

Noun → d

Verb → r

Enter the context-free grammar in JFLAP using multi-character non-terminal symbols.

Here is an example of the result.



LHS		RHS
<S>	→	<Subject> <Verb>
<Article>	→	a
<Article>	→	t
<Noun>	→	d
<Subject>	→	<Article> <Noun>
<Verb>	→	r

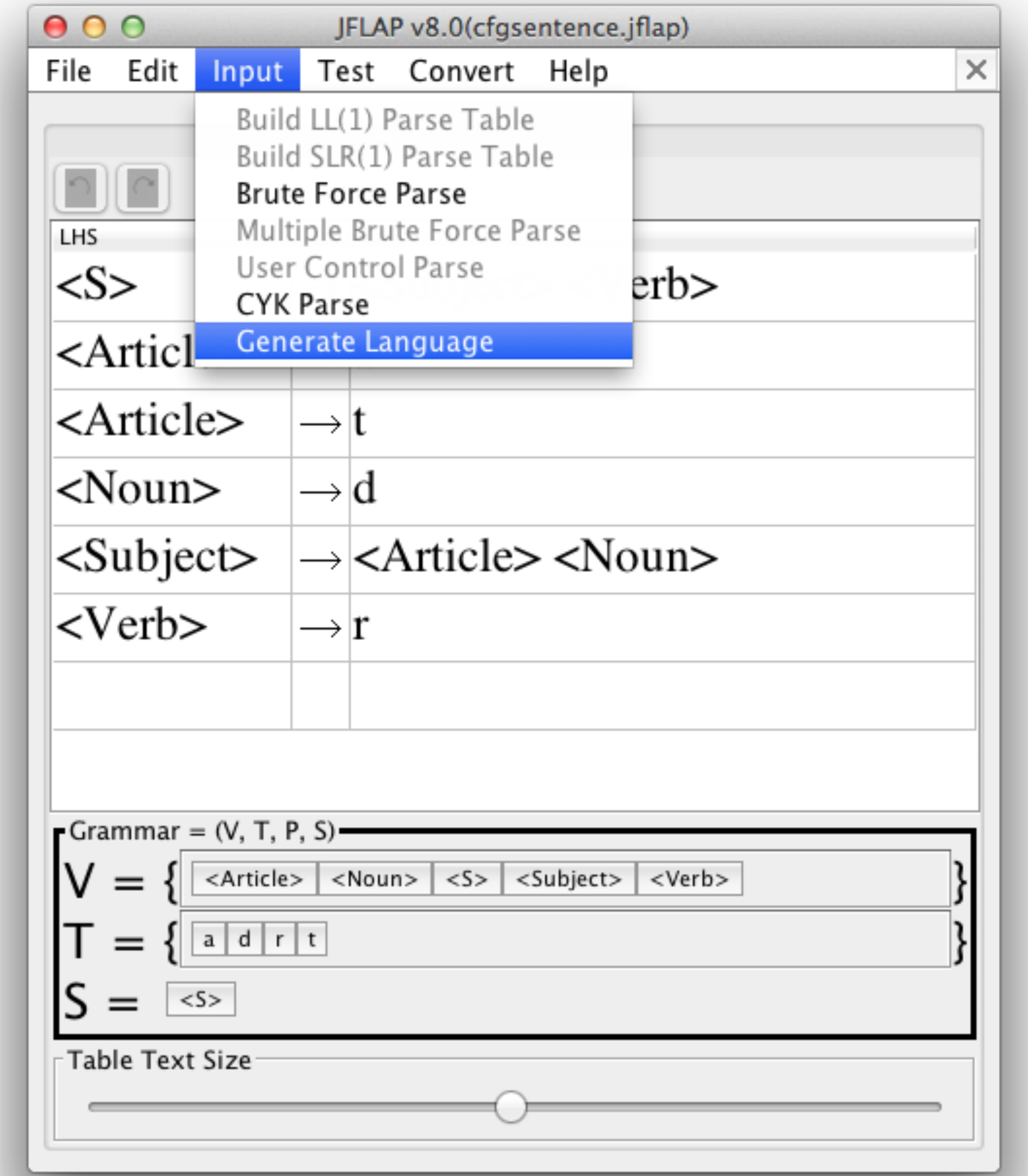
Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

Now use the Generate Language feature of JFLAP to produce sentences in the language. In this example we have entered **5** as the number of strings to generate.



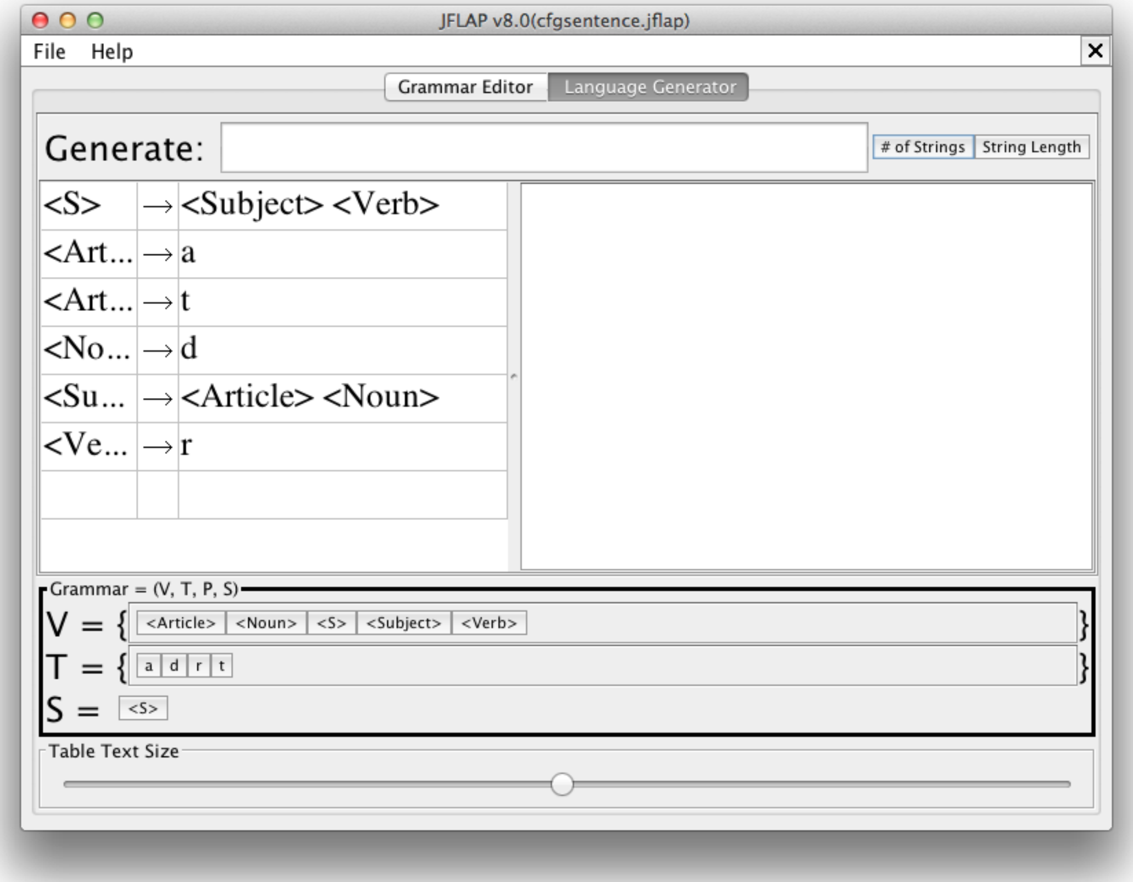
LHS		
<S>		erb>
<Articl		
<Article>	→	t
<Noun>	→	d
<Subject>	→	<Article> <Noun>
<Verb>	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



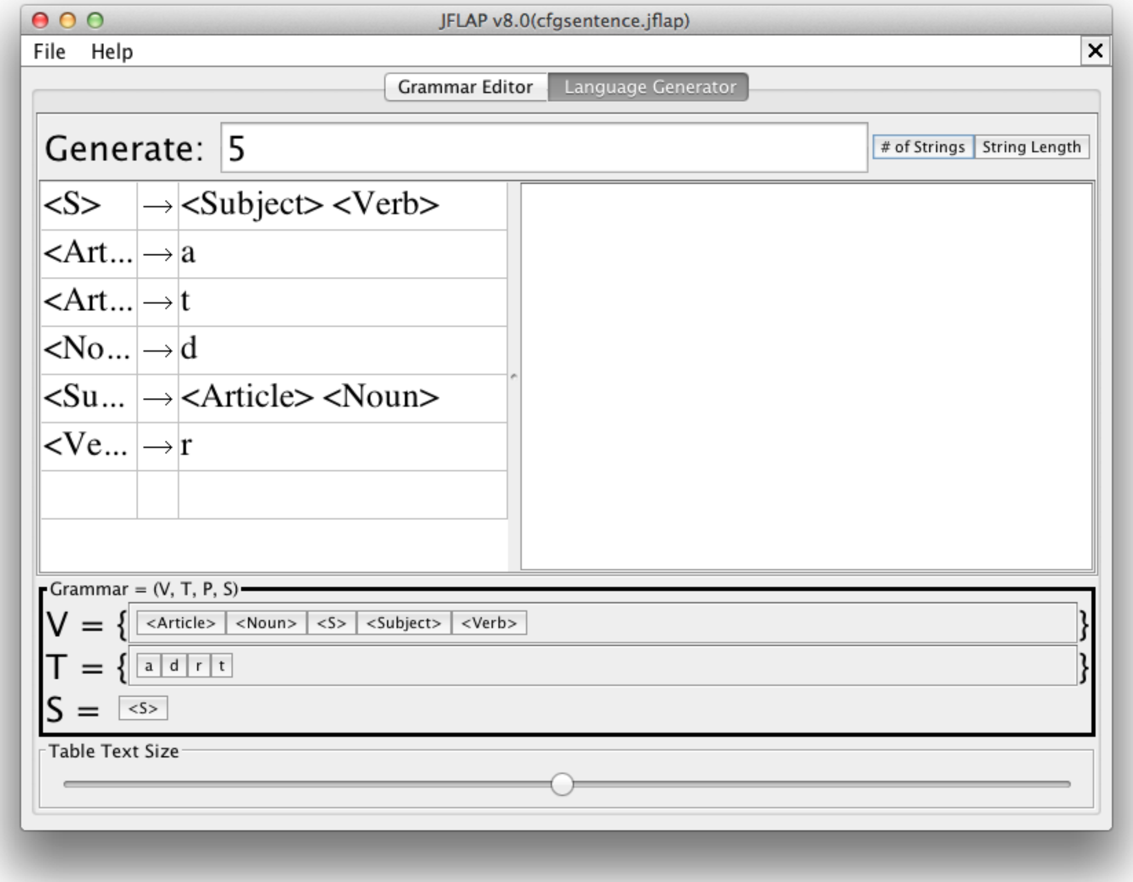
<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Generate: 5

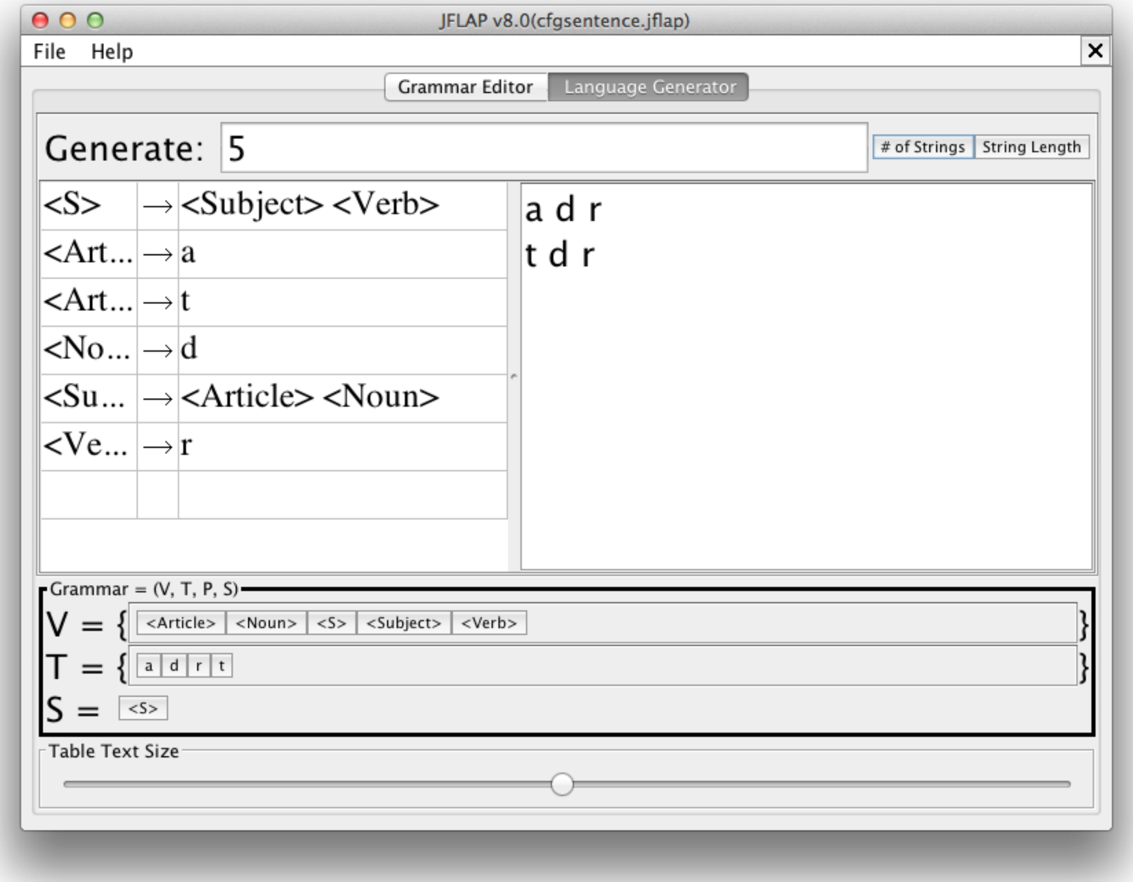
<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Generate: 5

<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

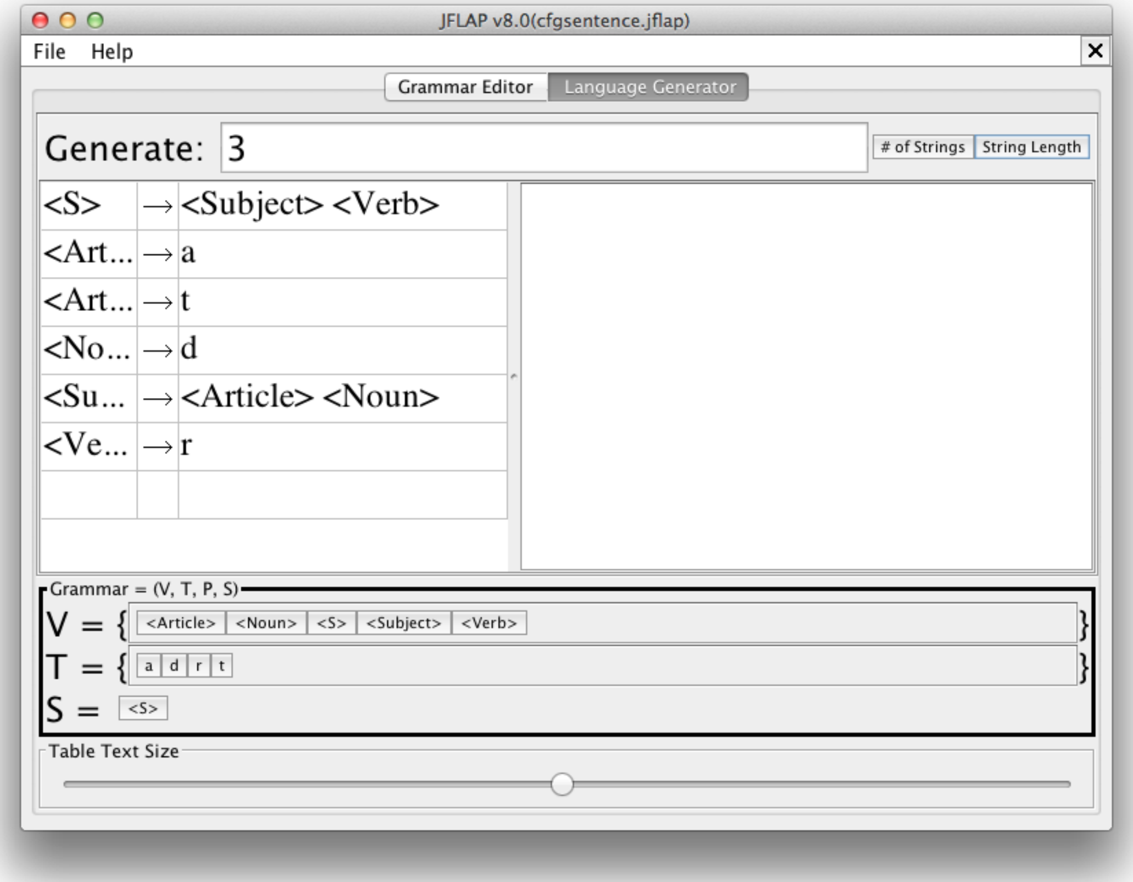
```
a d r
t d r
```

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Generate: 3

<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

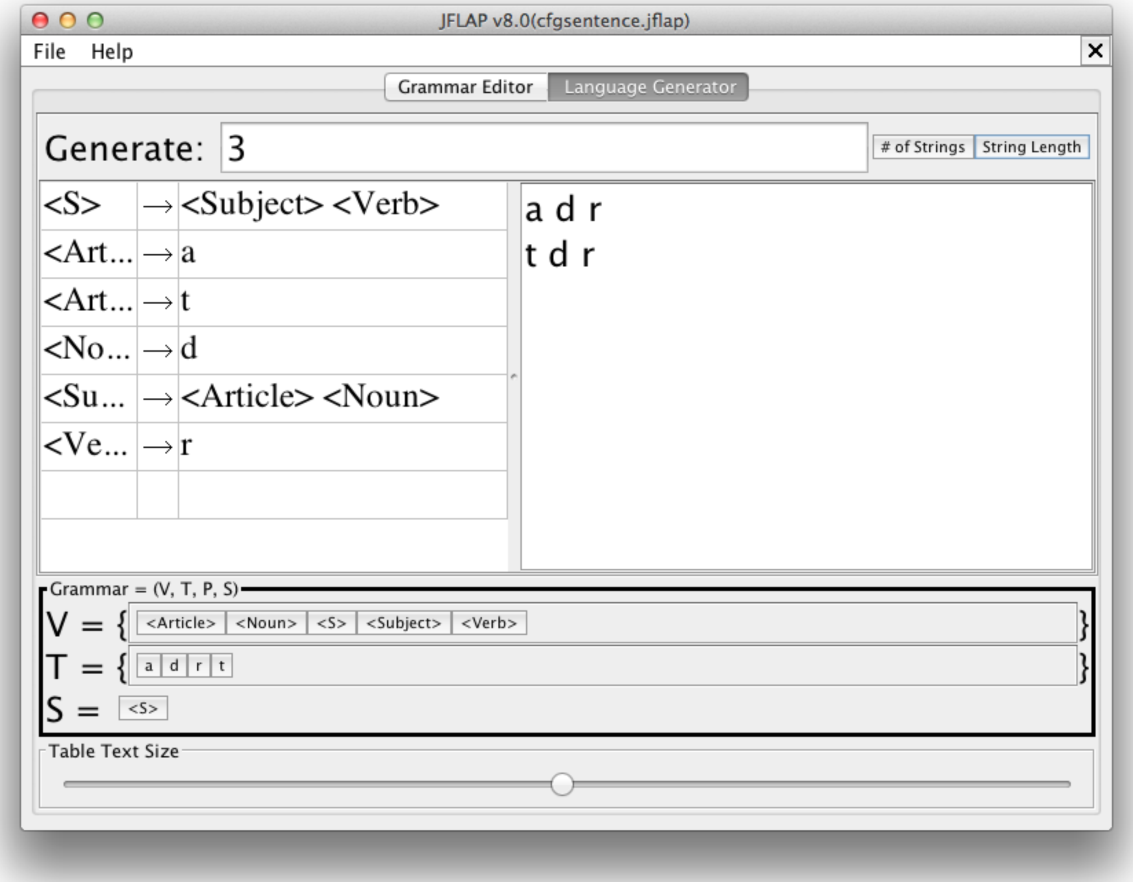
Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

Notice that only 2 strings have been generated, because that is the cardinality of this language. We would get the same result had we asked for 3 strings.



Generate: 3

<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

```
a d r
t d r
```

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

What set of strings does this grammar produce?

ANSWER: { adr, tdr }

Using the interpretation of terminal symbols, what English sentences do these strings represent?

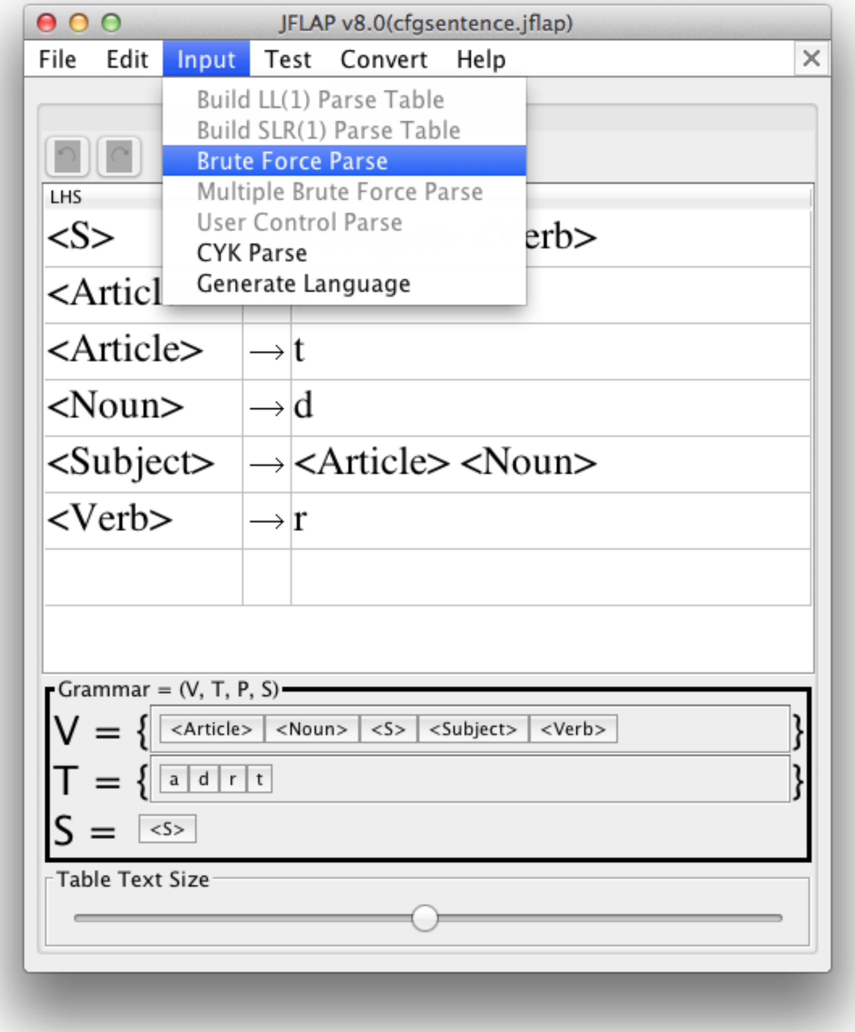
ANSWER: A dog ran. The dog ran.

Sample Solution (see `CFGsentence.jflap`)

Brute Force Parser

Use the brute force parser to explore the derivation of string “atr”.

The following sequence of images explores the derivation of string “atr” using the *Brute Force Parse* feature.



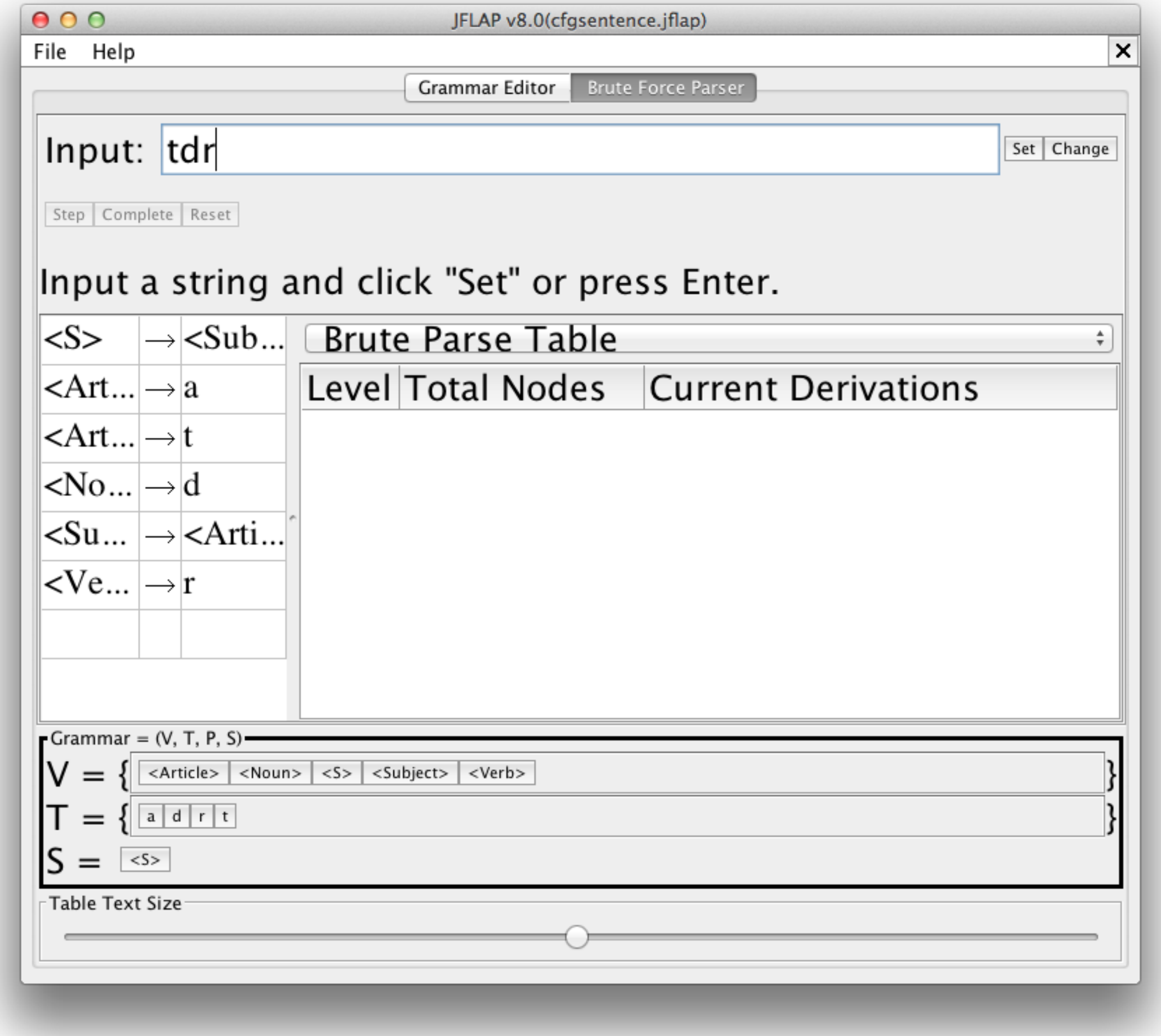
LHS		
<S>		erb>
<Articl		
<Article>	→	t
<Noun>	→	d
<Subject>	→	<Article> <Noun>
<Verb>	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Input a string and click "Set" or press Enter.

<S>	→	<Sub...
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Arti...
<Ve...	→	r

Level	Total Nodes	Current Derivations

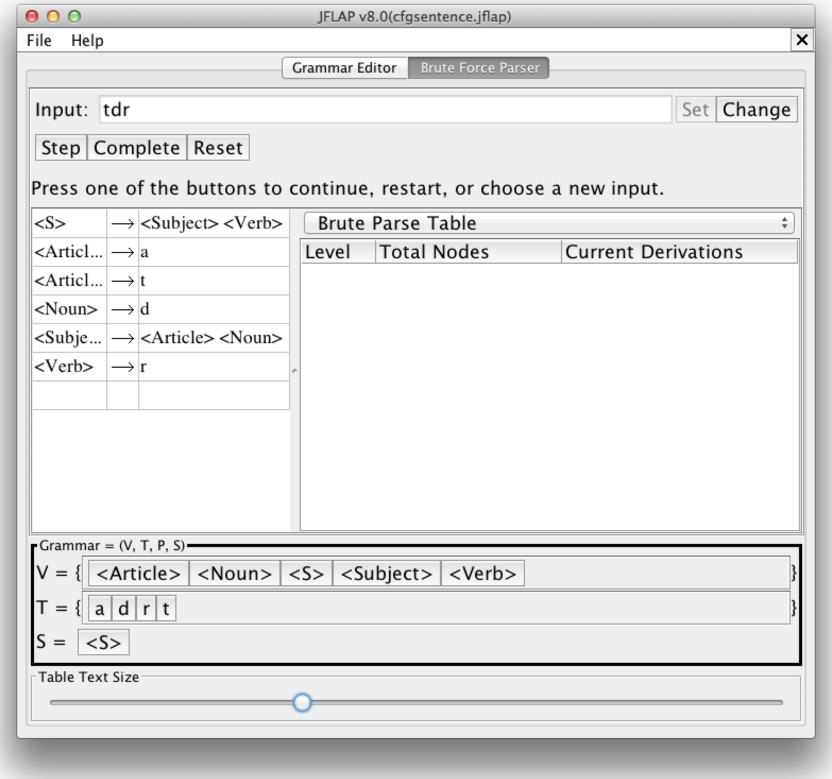
Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

To begin, provide the input string “tdr” and select *Set*. Then step through the derivation.



Input: tdr

Press one of the buttons to continue, restart, or choose a new input.

<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

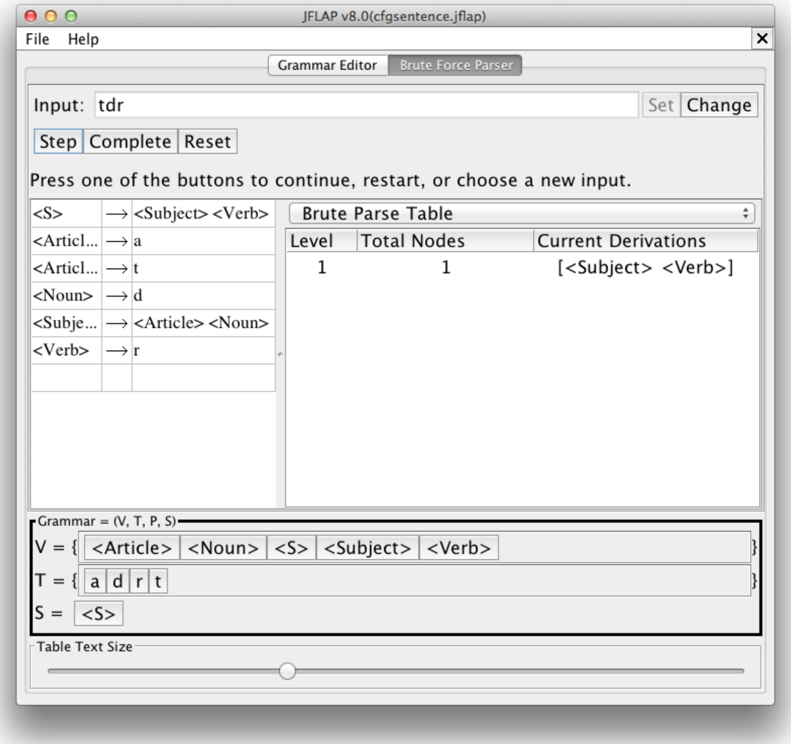
Level	Total Nodes	Current Derivations

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Press one of the buttons to continue, restart, or choose a new input.

<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

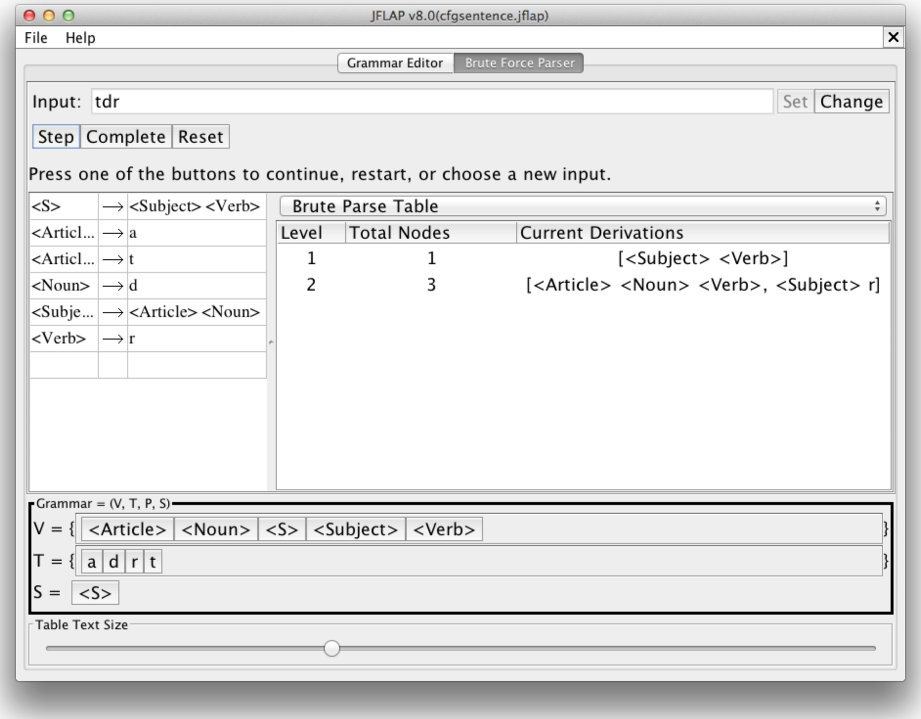
Level	Total Nodes	Current Derivations
1	1	[<Subject> <Verb>]

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Press one of the buttons to continue, restart, or choose a new input.

<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

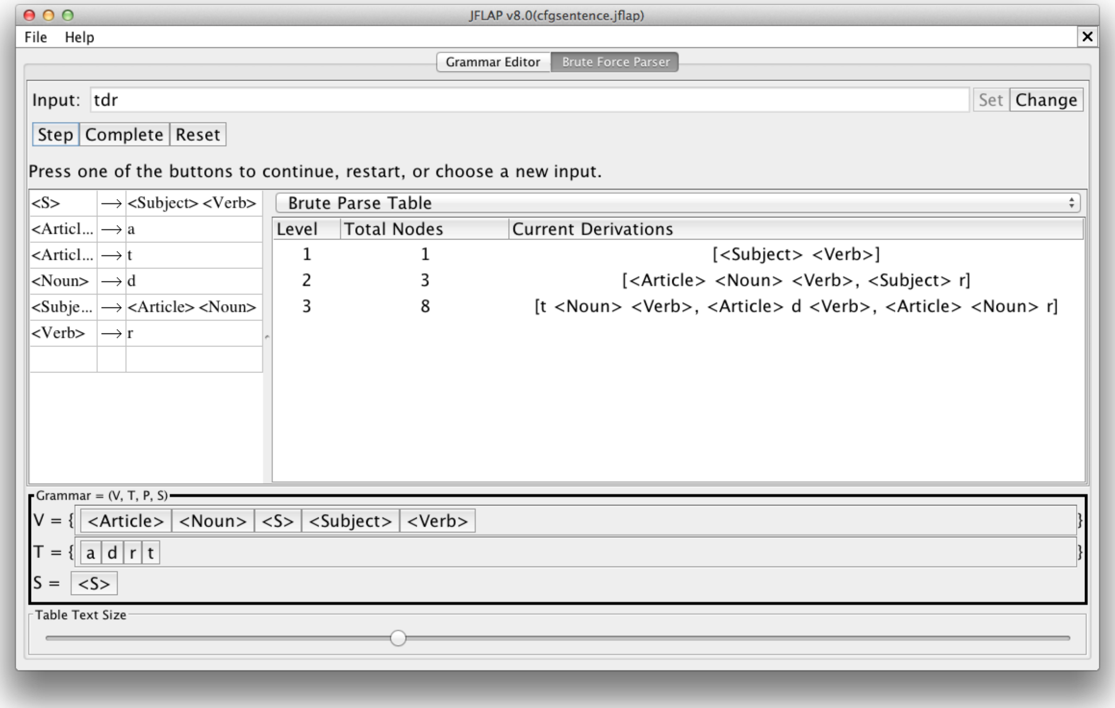
Level	Total Nodes	Current Derivations
1	1	[<Subject> <Verb>]
2	3	[<Article> <Noun> <Verb>, <Subject> r]

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Press one of the buttons to continue, restart, or choose a new input.

<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

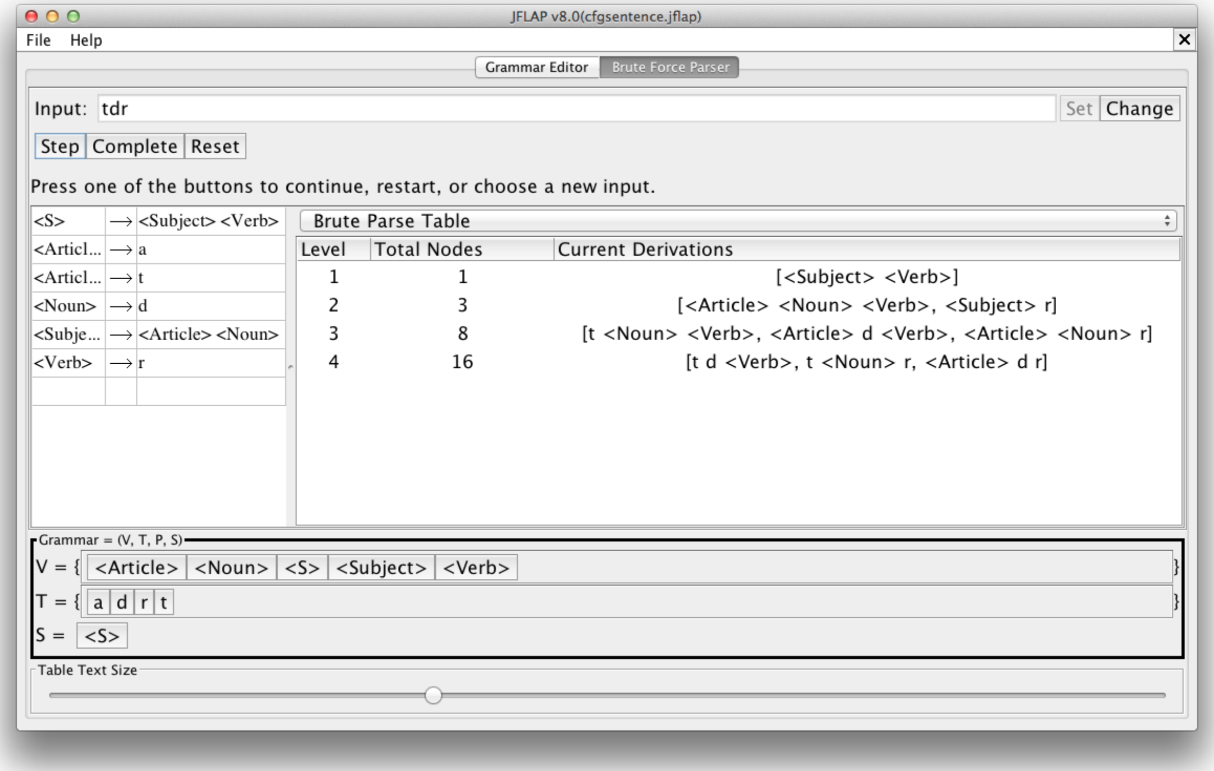
Level	Total Nodes	Current Derivations
1	1	[<Subject> <Verb>]
2	3	[<Article> <Noun> <Verb>, <Subject> r]
3	8	[t <Noun> <Verb>, <Article> d <Verb>, <Article> <Noun> r]

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Press one of the buttons to continue, restart, or choose a new input.

<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

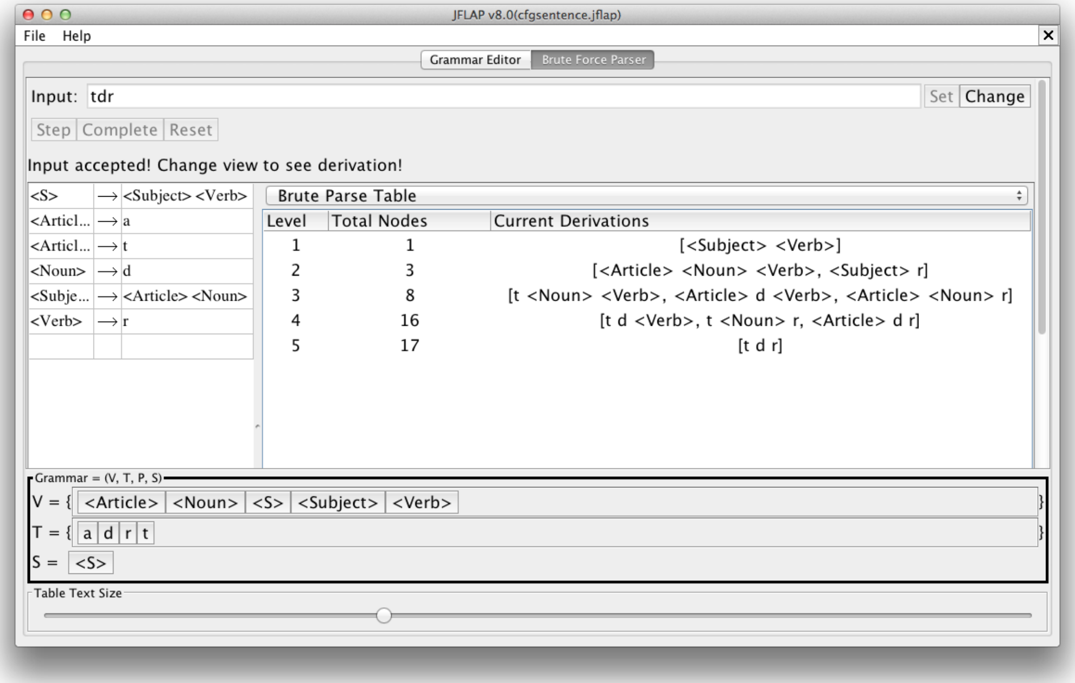
Level	Total Nodes	Current Derivations
1	1	[<Subject> <Verb>]
2	3	[<Article> <Noun> <Verb>, <Subject> r]
3	8	[t <Noun> <Verb>, <Article> d <Verb>, <Article> <Noun> r]
4	16	[t d <Verb>, t <Noun> r, <Article> d r]

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Input accepted! Change view to see derivation!

<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

Level	Total Nodes	Current Derivations
1	1	[<Subject> <Verb>]
2	3	[<Article> <Noun> <Verb>, <Subject> r]
3	8	[t <Noun> <Verb>, <Article> d <Verb>, <Article> <Noun> r]
4	16	[t d <Verb>, t <Noun> r, <Article> d r]
5	17	[t d r]

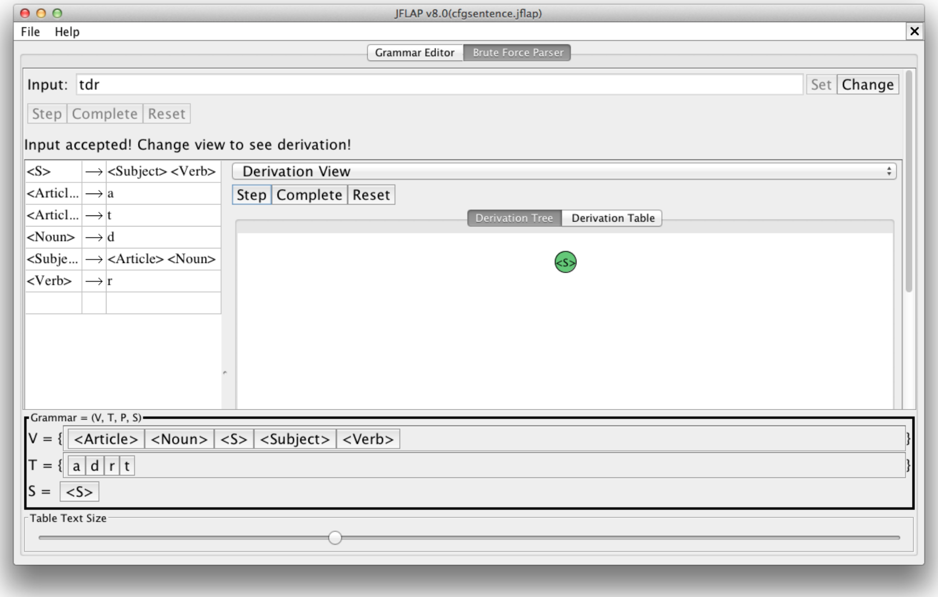
Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

The derivation is complete at this point. Dismiss the Brute Force Parse tab and choose *Derivation View* to explore the derivation tree.



Input: tdr

Input accepted! Change view to see derivation!

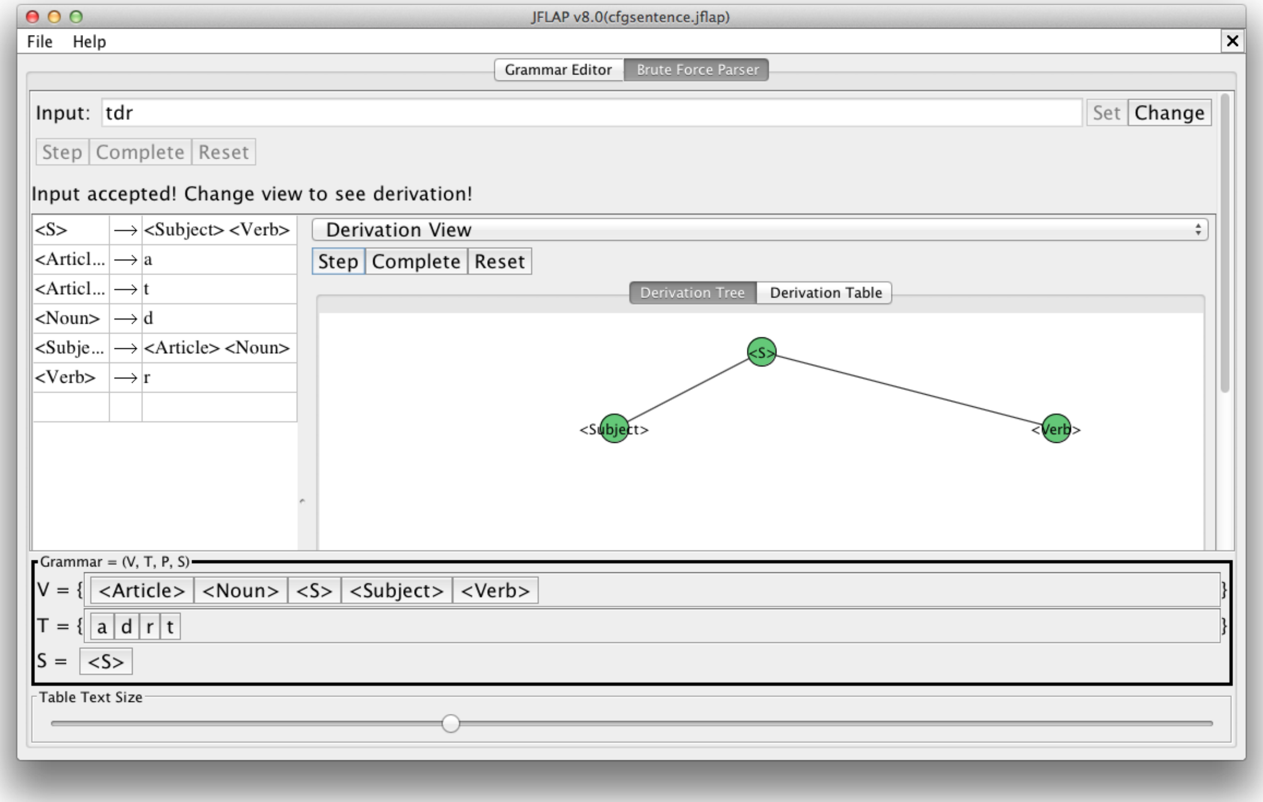
<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Input accepted! Change view to see derivation!

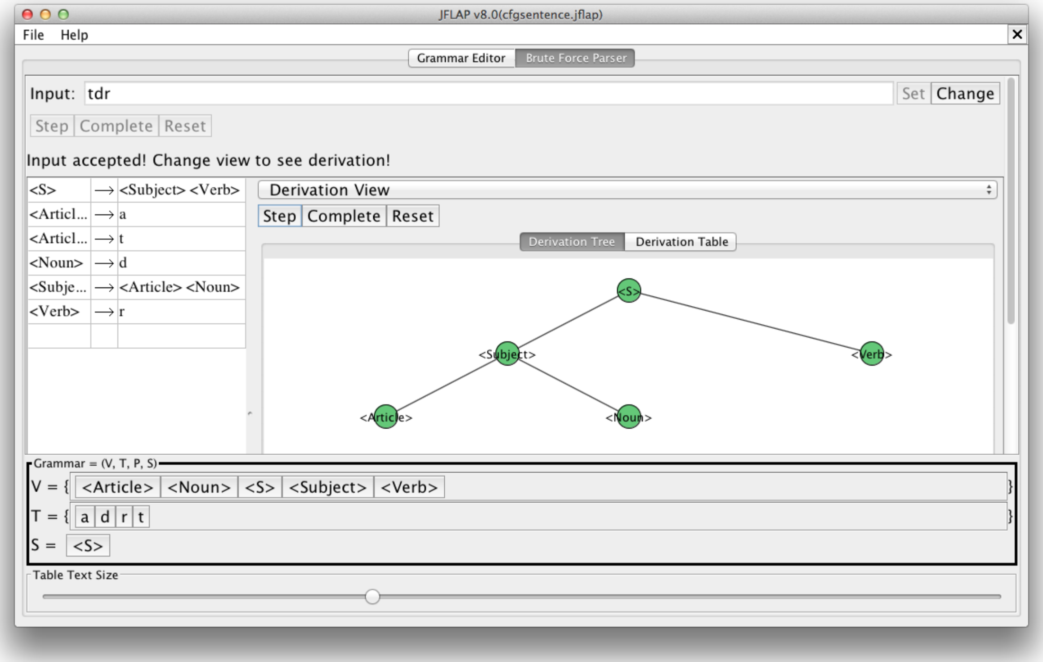
<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Input accepted! Change view to see derivation!

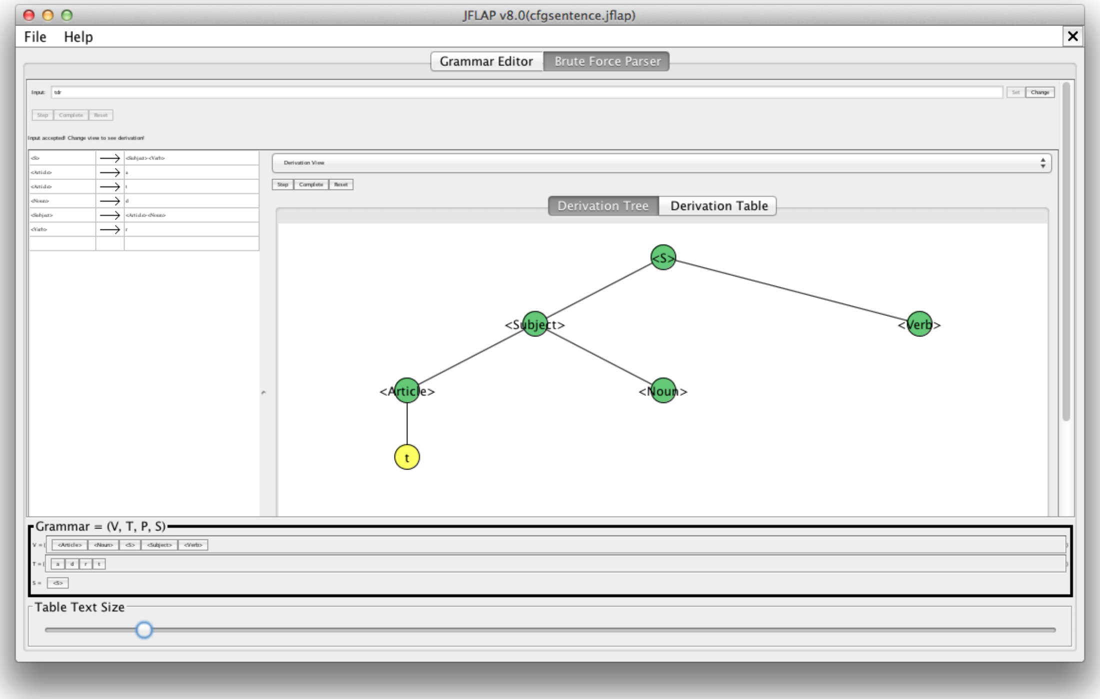
<S>	⟶	<Subject> <Verb>
<Articl...	⟶	a
<Articl...	⟶	t
<Noun>	⟶	d
<Subje...	⟶	<Article> <Noun>
<Verb>	⟶	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Input: tdr

Input accepted! Change view to see derivation!

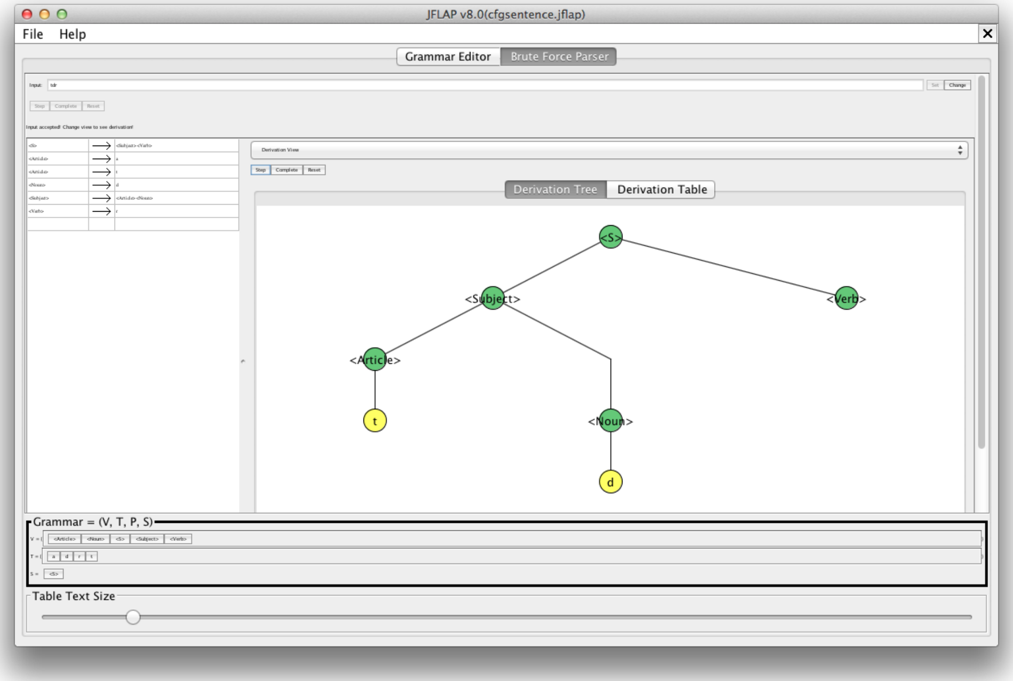
<S>	⟶	<Subject> <Verb>
<Article>	⟶	a
<Article>	⟶	t
<Noun>	⟶	d
<Subject>	⟶	<Article> <Noun>
<Verb>	⟶	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



JFLAP v8.0(cfgsentence.jflap)

Input: tdr

Input accepted! Change view to see derivation!

<S>	→	<Subject><Verb>
<Article>	→	a
<Article>	→	t
<Noun>	→	d
<Subject>	→	<Article><Noun>
<Verb>	→	r

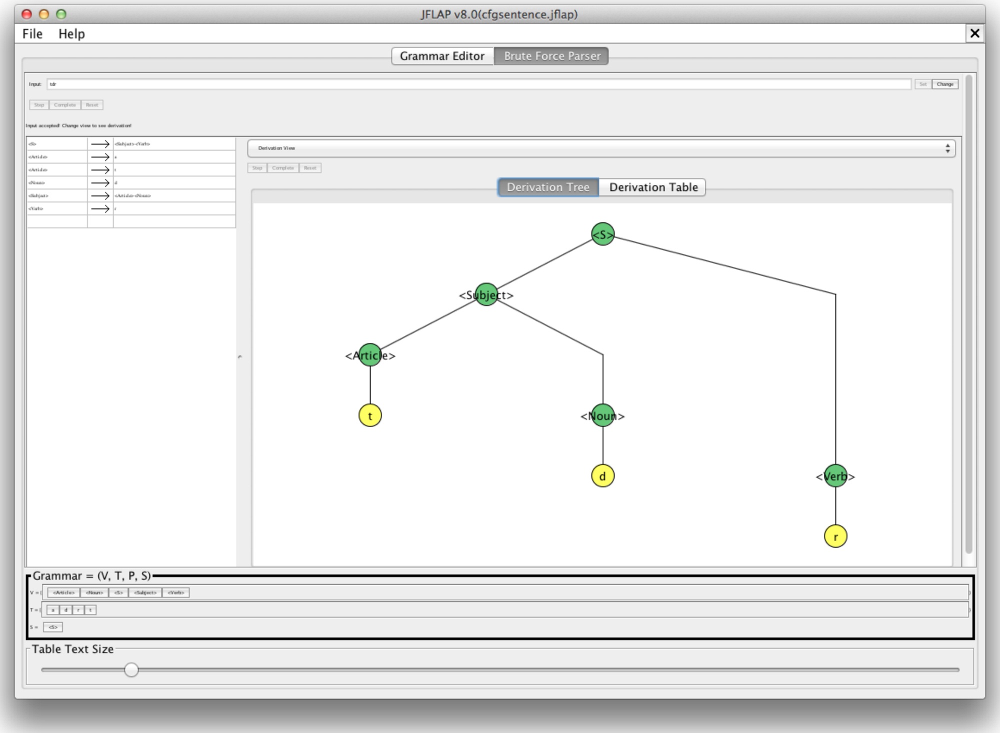
Derivation View

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb>

T = { a d r t

S = <S>



JFLAP v8.0(cfgsentence.jflap)

Input: tdr

Input accepted! Change view to see derivation!

<S>	→	<Subject><Verb>
<Article>	→	a
<Article>	→	t
<Noun>	→	d
<Subject>	→	<Article><Noun>
<Verb>	→	r

Derivation View

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb>

T = { a d r t

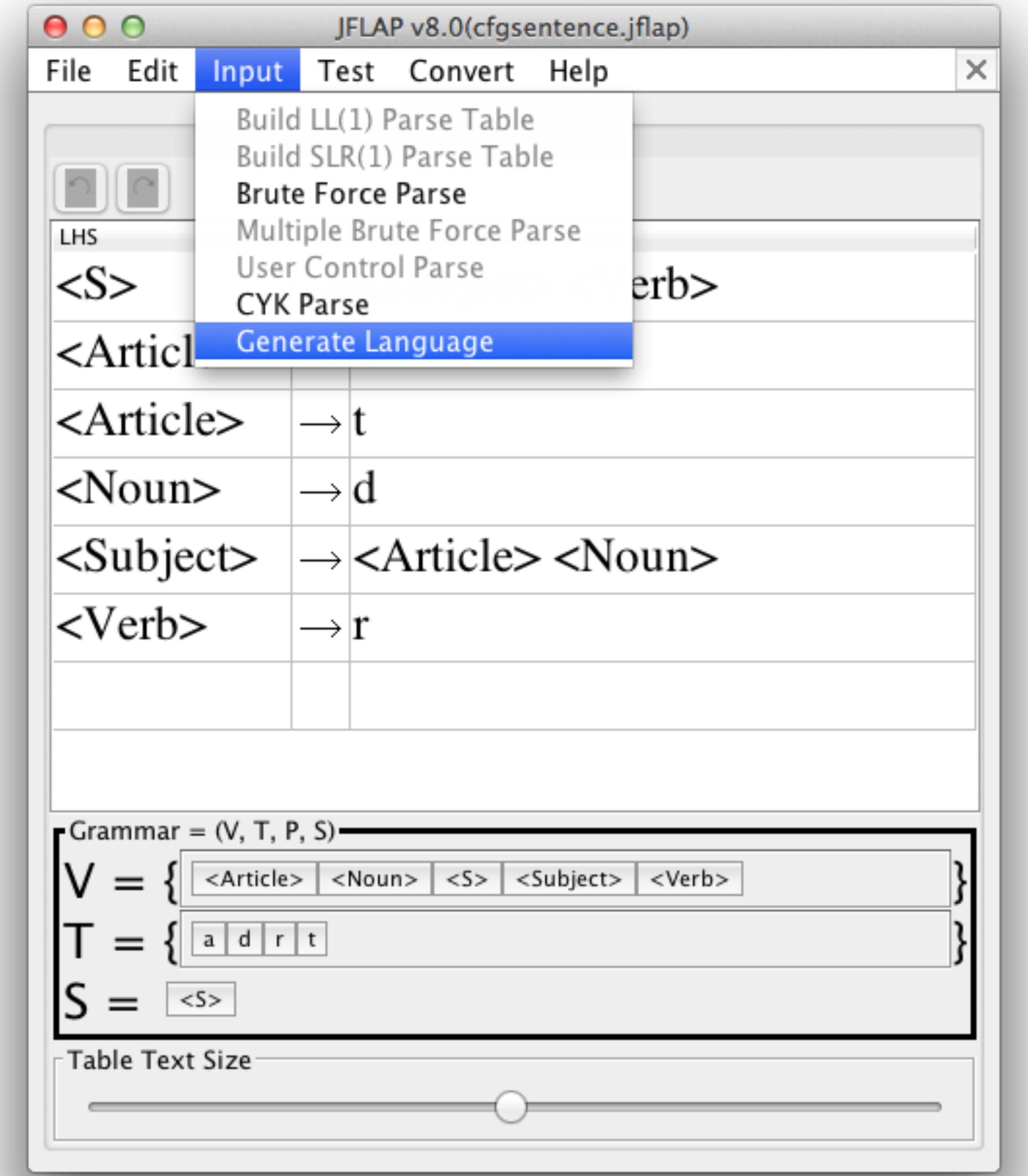
S = <S>

Generate Language

Use the generate language feature of FLAP to generate strings in the language.

Sample Solution

Dismiss the Brute Force Parse tab and choose *Generate Language* to have JFLAP generate strings in the language



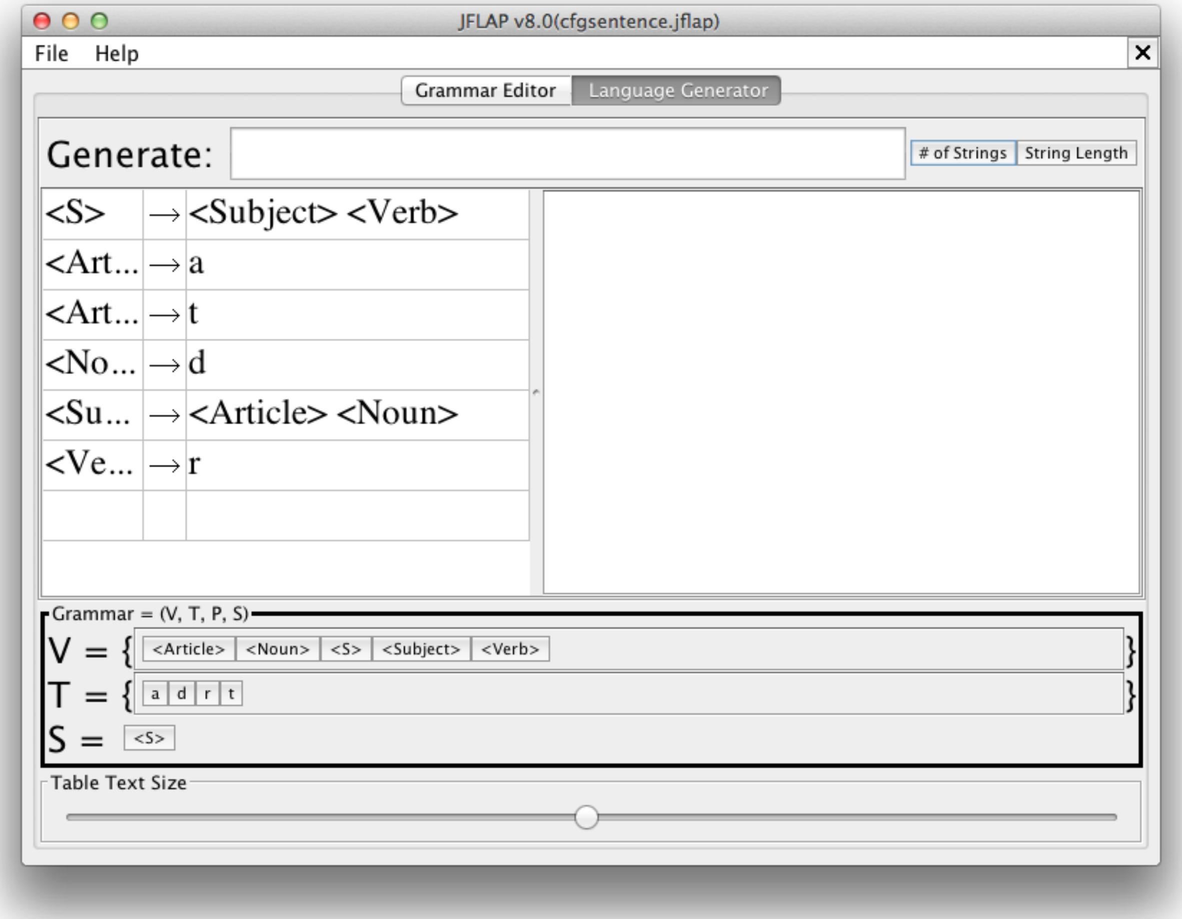
LHS		
<S>		erb>
<Articl		
<Article>	→	t
<Noun>	→	d
<Subject>	→	<Article> <Noun>
<Verb>	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

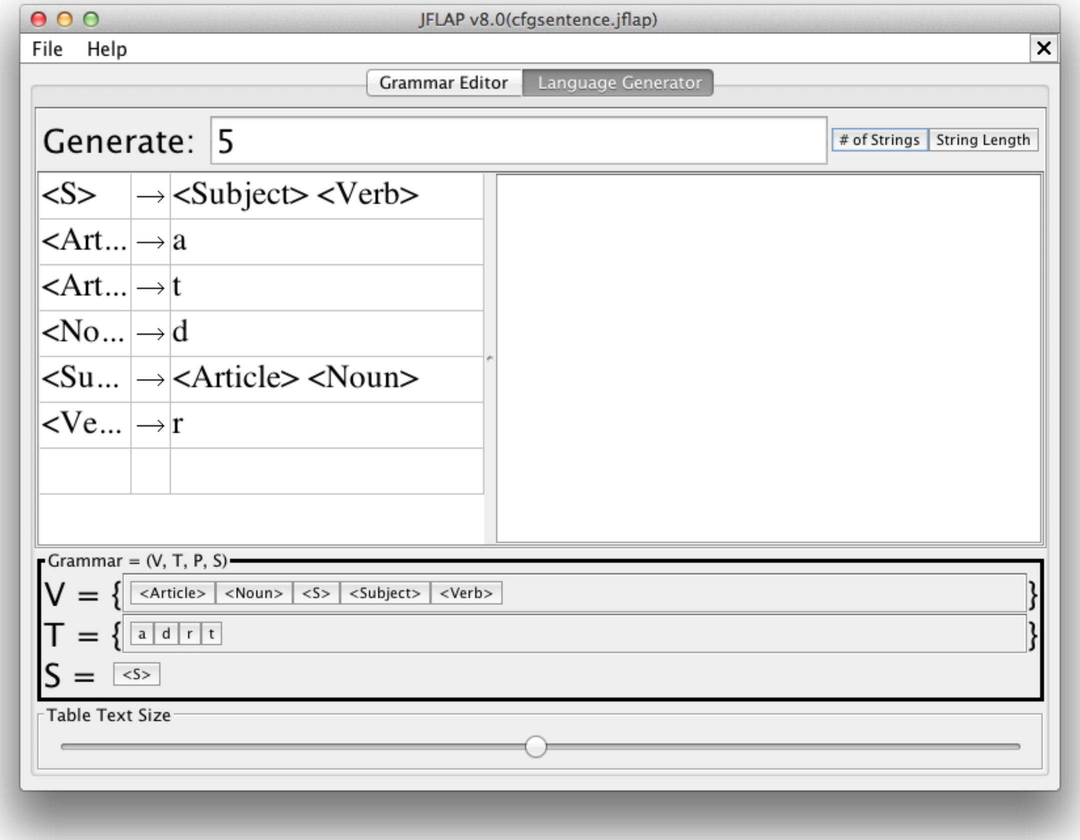
Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

Enter the number 5 as the maximum number of strings to generate and select *# of Strings.* Review the strings produced.



Generate: 5

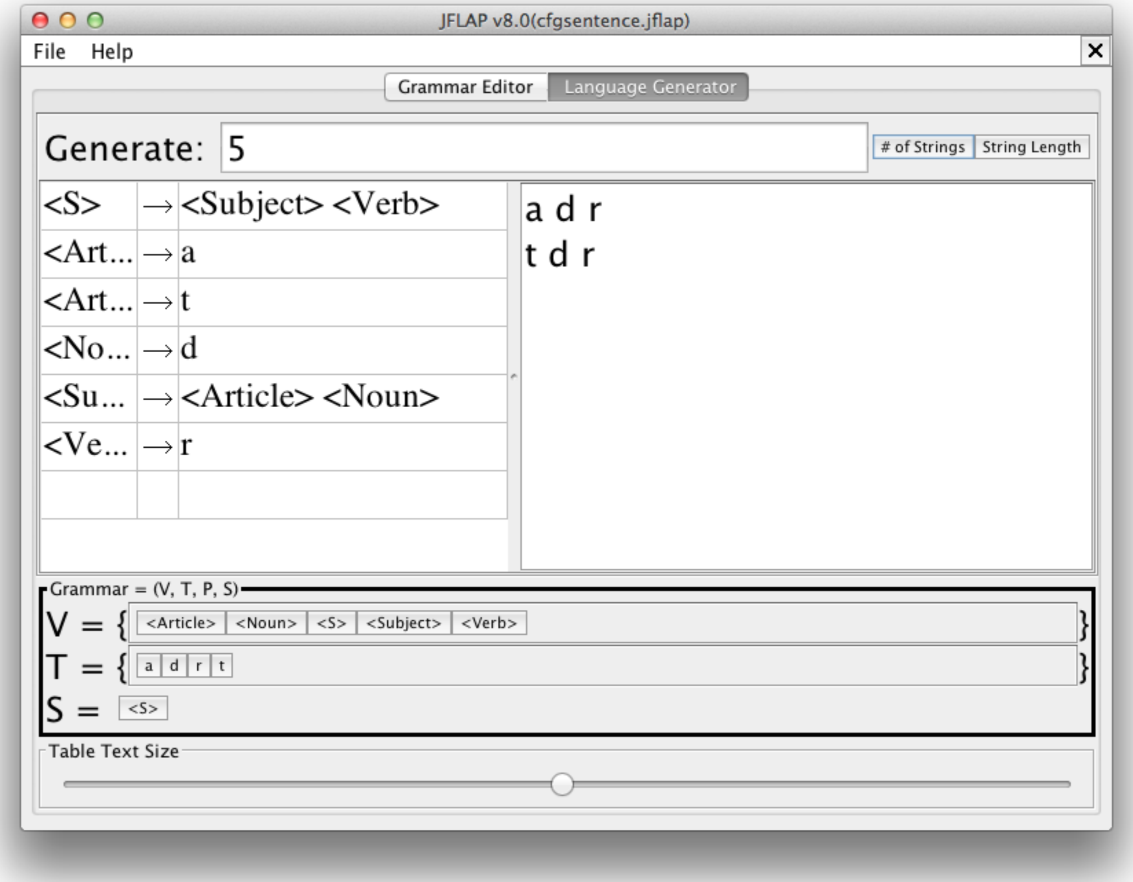
<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Generate: 5

<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

```
a d r
t d r
```

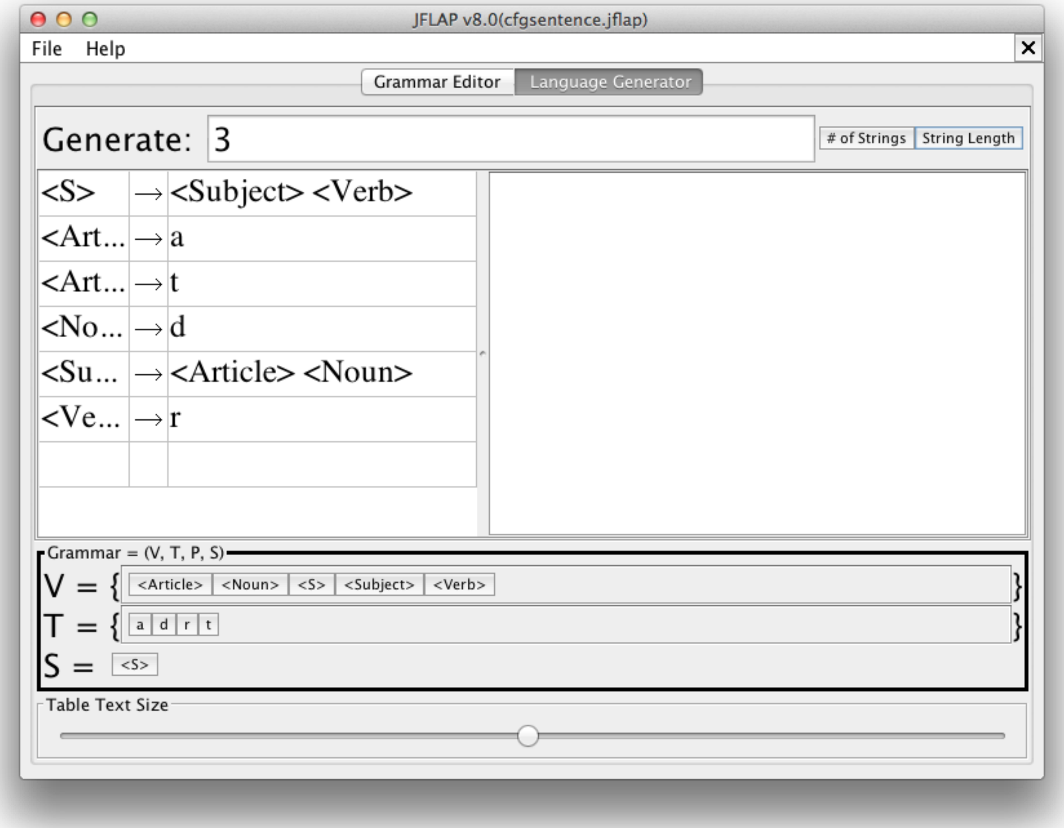
Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

Enter the number 3 as the length of produced strings and select *String Length.* Review the strings produced.



Generate: 3

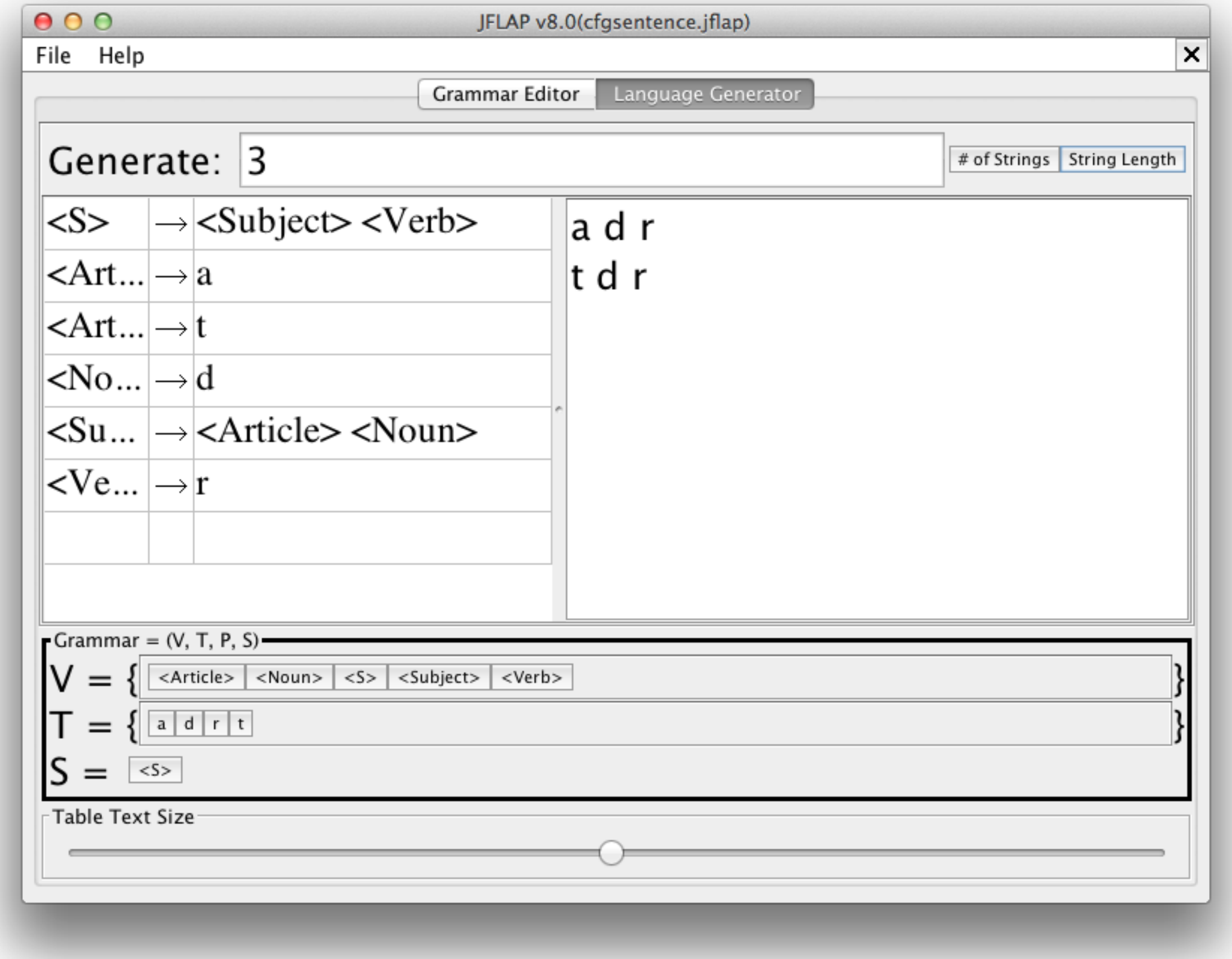
<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>



Generate: 3

<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

```
a d r
t d r
```

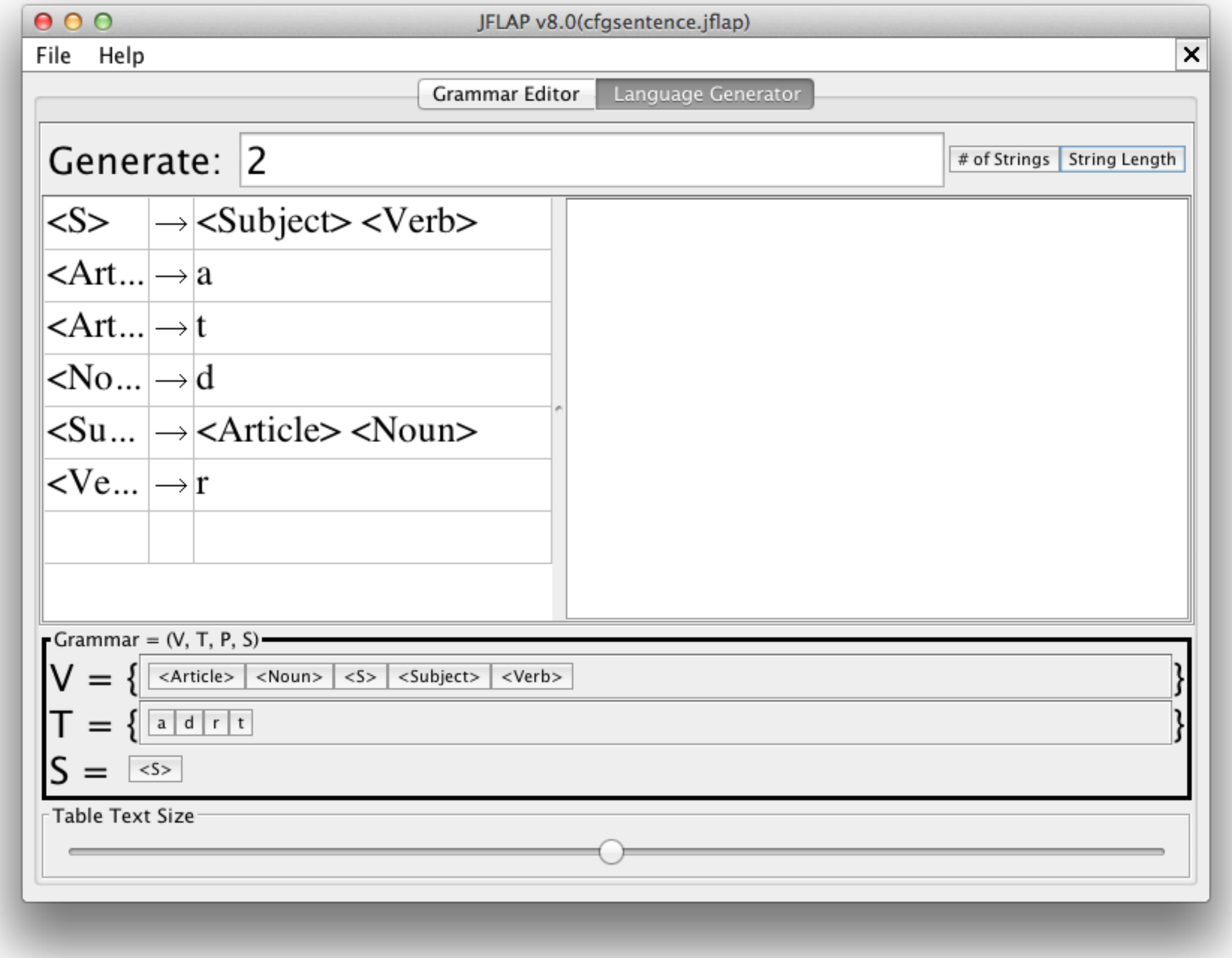
Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>

Enter the number 2 as the length of produced strings and select *String Length.* Why is the set of strings produced now empty?



Generate: 2

<S>	→	<Subject> <Verb>
<Art...	→	a
<Art...	→	t
<No...	→	d
<Su...	→	<Article> <Noun>
<Ve...	→	r

Grammar = (V, T, P, S)

V = { <Article> <Noun> <S> <Subject> <Verb> }

T = { a d r t }

S = <S>