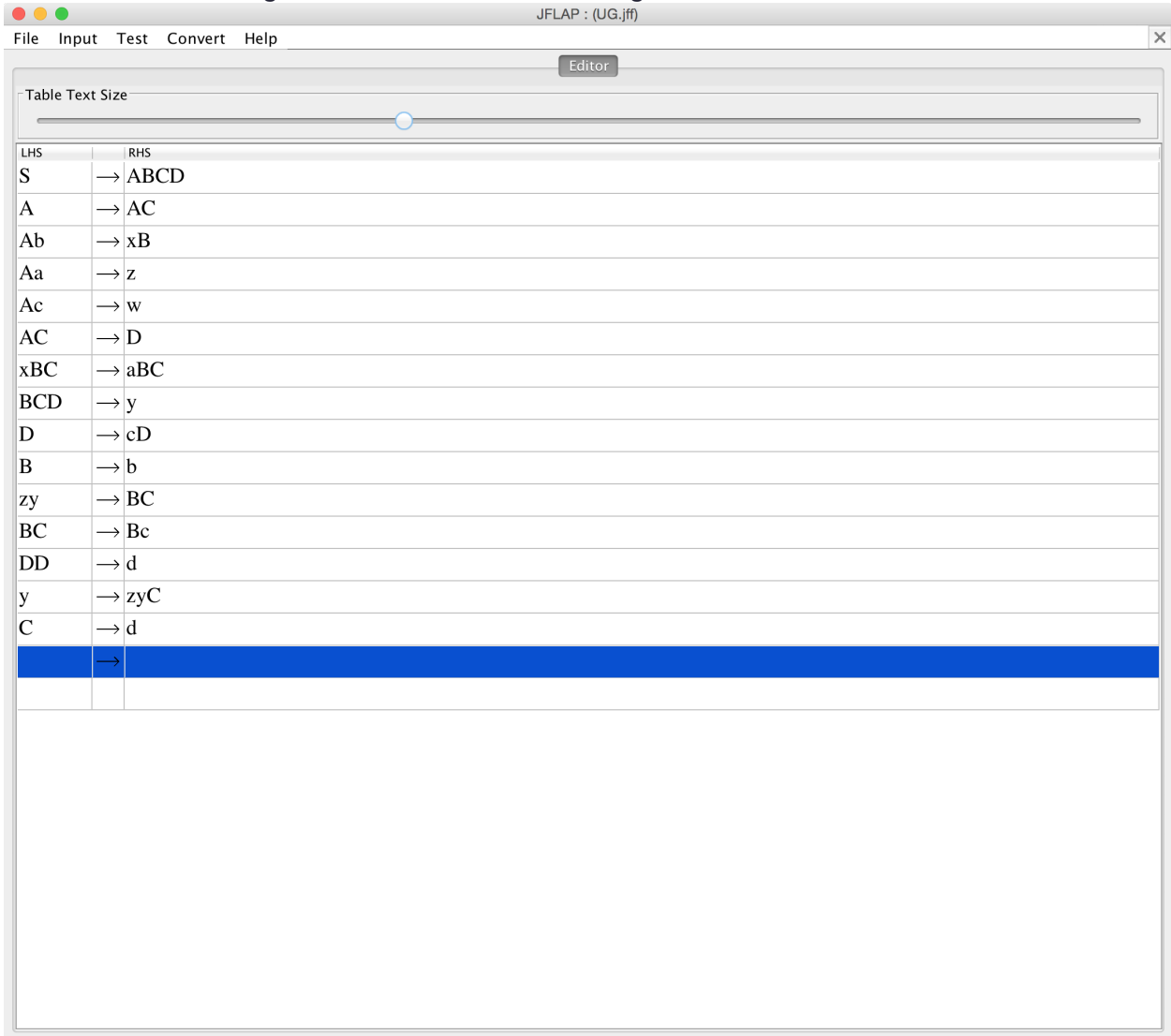


Example: Unrestricted Grammar_{JP}

Consider the following definition of an unrestricted grammar.



The screenshot shows the JFLAP software interface with the following elements:

- Window title: JFLAP : (UG.jff)
- Menu bar: File, Input, Test, Convert, Help
- Editor button: Editor
- Table Text Size: A slider control.
- Grammar table with LHS and RHS columns:

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d
	→

Confirm that this grammar is unrestricted using Test > Test for Grammar Type.

Table Text Size

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d
	→

JFLAP : (UG.jff)

File Input Test Convert Help

Editor

Table Text Size

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d
	→

Grammar Type

This is an Unrestricted Grammar

OK

Consider strings in the language specified by this grammar.

- What is the alphabet of the language of this grammar?
- What are the set of non-terminals (variables) used in specifying this grammar?
- What strings are in the language?
- Is the empty string an element of this language?

Predict whether the string "abcd" is in the language.

- What features of the grammar are you using in making your prediction?

Try using Input > Brute Force Parse to determine if the string "abcd" is in the language.

- What issues do you run into using this approach?

Try using Input > User Control Parse to explore potential derivations of "abcd".

- What features of the grammar are you using in narrowing down to a derivation of that string?
- Is "abcd" actually in the language?

Editor

Table Text Size

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d
	→

JFLAP : (UG.jff)

File Input Test Convert Help

Build LL(1) Parse Table
Build SLR(1) Parse Table
Brute Force Parse
Multiple Brute Force Parse
User Control Parse

Editor

Tab

LHS

S	CYK Parse
	Multiple CYK Parse
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d
	→

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC

Input a string to begin.

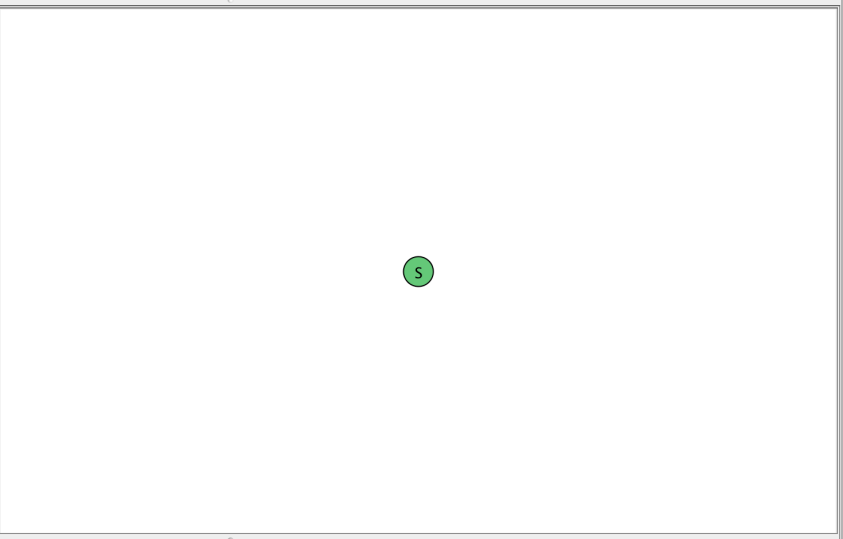
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d



S

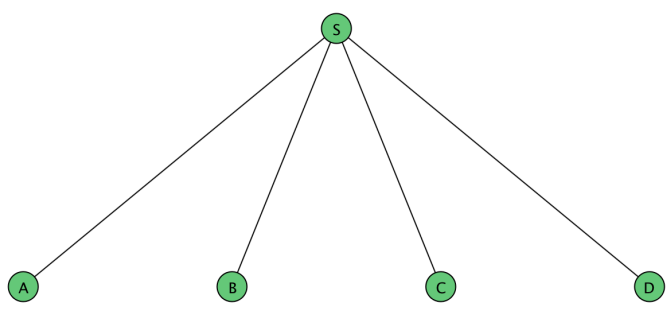
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC



A B C D

Derived current Strings using S→ABCD production

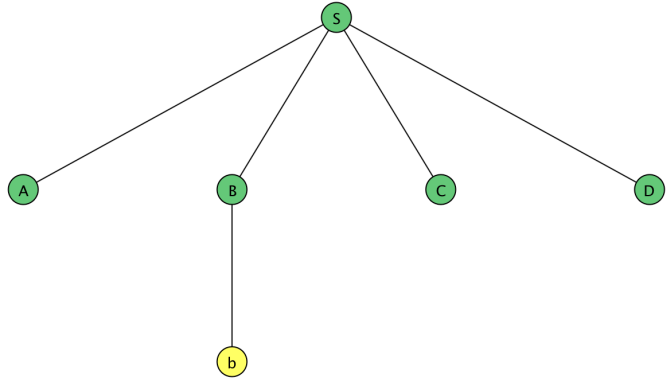
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC



A b C D

Derived current Strings using B→b production

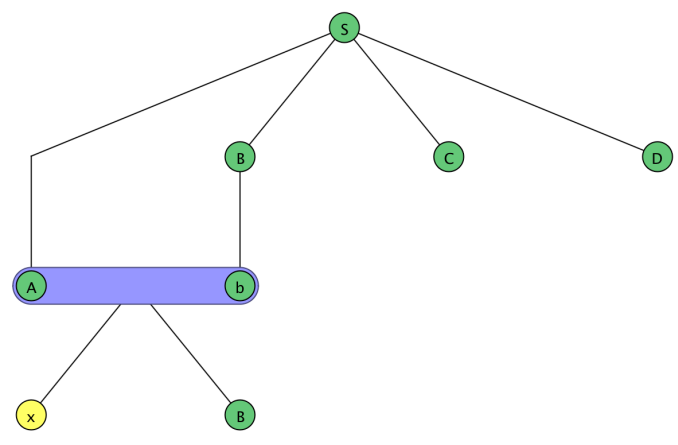
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC



x B C D

Derived current Strings using Ab→xB production

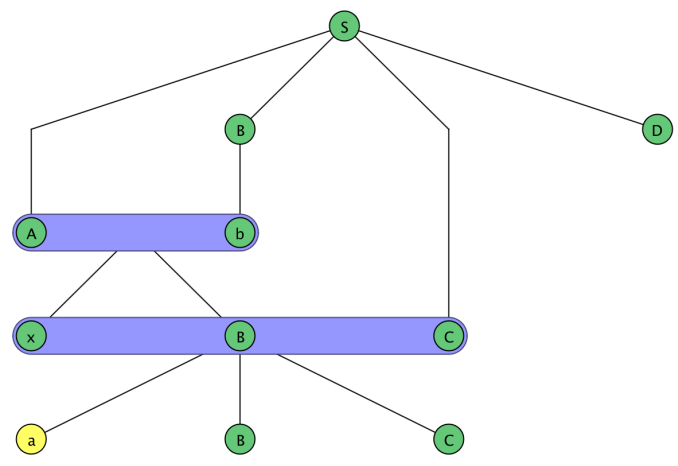
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC



a B C D

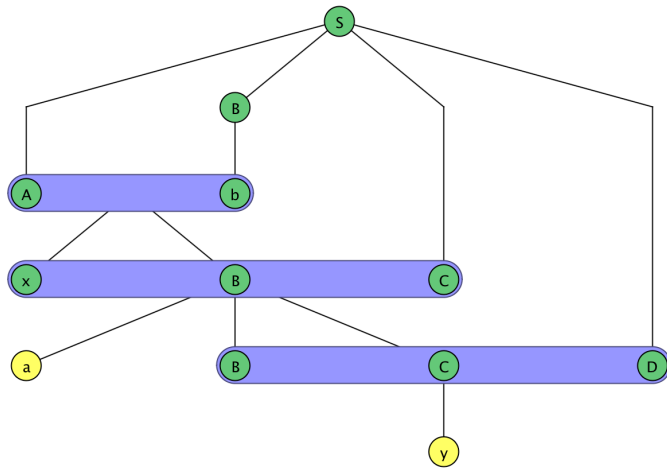
Derived current Strings using xBC→aBC production

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
S	→ ABCD
A	→ AC
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC



ay

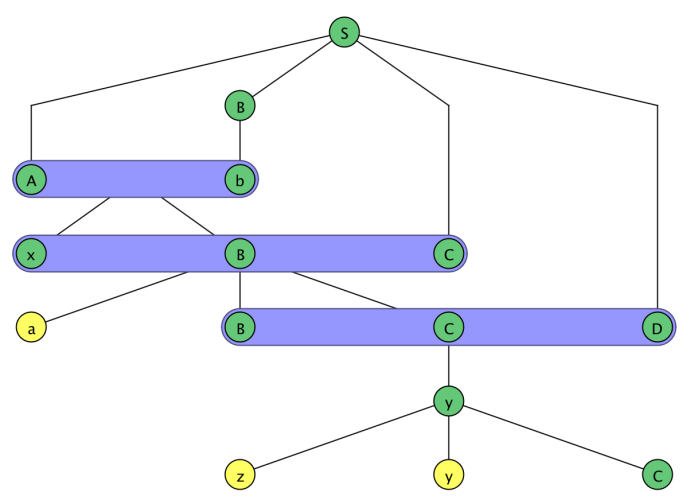
Derived current Strings using BCD→y production

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d



a z y C

Derived current Strings using y→zyC production

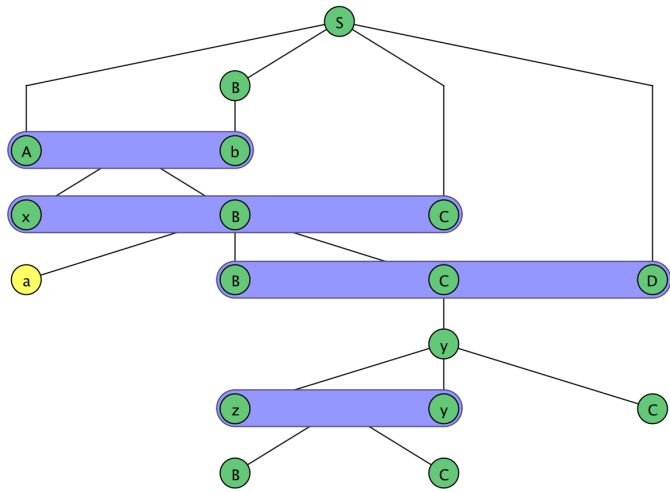
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d



a B C C

Derived current Strings using zy→BC production

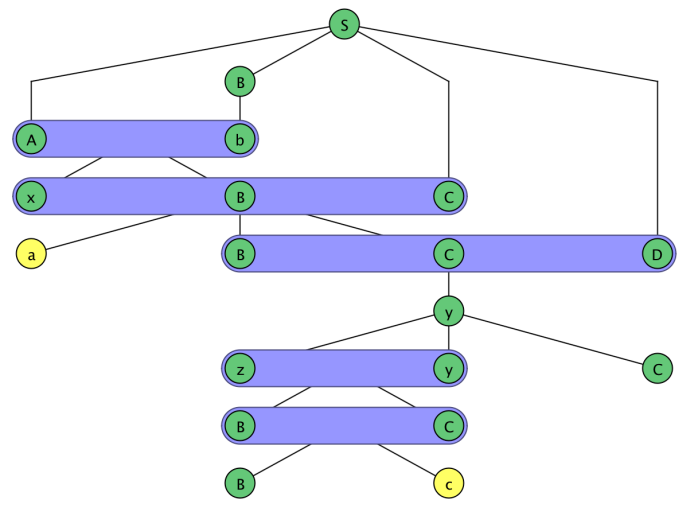
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d



a B c C

Derived current Strings using BC→Bc production

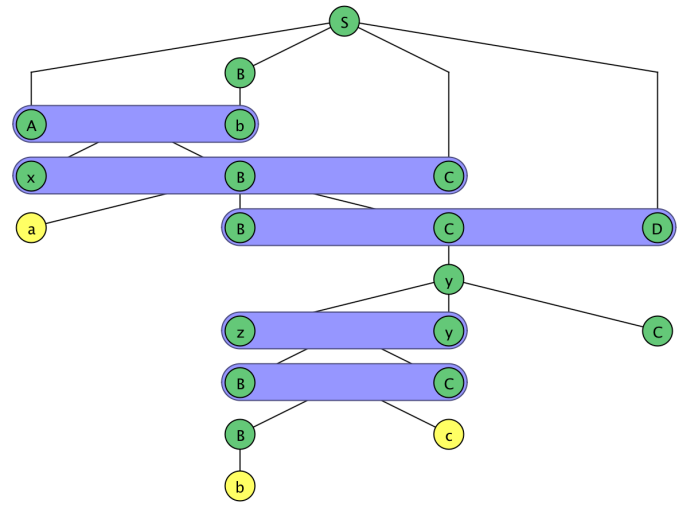
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd

LHS	RHS
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d



a b c C

Derived current Strings using B→b production

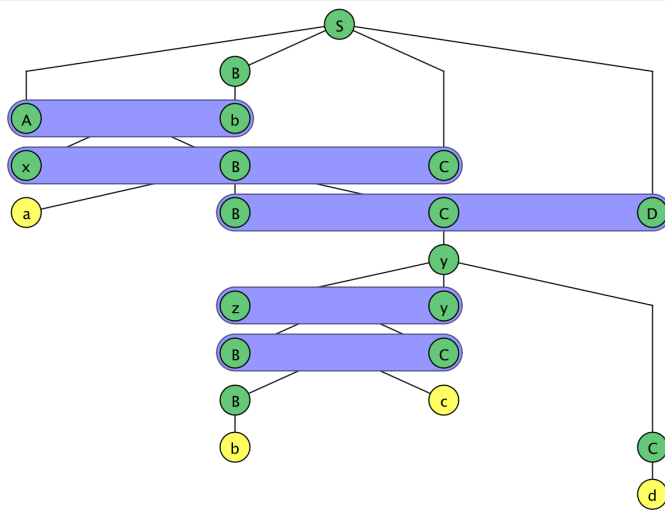
Editor User Control Parser

Table Text Size

Start Previous Step Noninverted Tree

Input abcd
String Accepted!

LHS	RHS
Ab	→ xB
Aa	→ z
Ac	→ w
AC	→ D
xBC	→ aBC
BCD	→ y
D	→ cD
B	→ b
zy	→ BC
BC	→ Bc
DD	→ d
y	→ zyC
C	→ d



a b c d

Derived current Strings using C→d production