Syntax Diagram Generator

March 11, 2012

Short description

Syntax Diagram Generator is an application written in Python which convert definition of diagram (in structured form) to SVG image(s).

Getting started

1 Installation

Before start using Syntax Diagram Generator, user should assure that specific libraries and packages have been installed in a operation system:

- setuptools (http://pypi.python.org/pypi/setuptools)
- pysvg (http://pypi.python.org/pypi/pysvg/0.2.1)
- CairoSVG (http://cairosvg.org/)
- python-tk (http://wiki.python.org/moin/TkInter)

After satisfy these assumptions, installation of Syntax Diagram Generator is executing the command:

```
python setup.py install
```

2 Running an application

To get a SVG diagram, a appriopriate Python file should be prepared:

Listing 1: Example of input Python file

```
# -*- coding: utf-8 -*-
import sys
sys.path.append('.')
from sdgen.svg import *

#definition of diagram
#examples will appear in the next chapter
data = {
        ...
}

#a second parameter tells about directory
#to write all diagrams
result = as_svg(data, 'directory')
#printing resulting main image
#first index - number of image
#second index - 0 for name, 1 for data
print result[0][1].encode('utf-8')
```

Then generating a image (or images) is executing a command:

```
python examples/inputfile.py directory > outputimage.svg
```

All images will be deployed into the specified directory and the main image into `outputimage.svg` file.

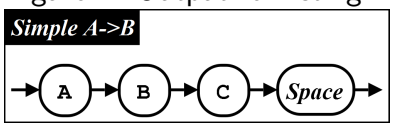
Examples

1 Terminal example

Listing 2: Terminal example

```
{
    "view": "Group",
    "name": "Simple A->B",
    "children": [
        {"view": "Terminal", "value": "A"},
        {"view": "Terminal", "value": "B"},
        {"view": "Terminal", "value": "C"},
        {"view": "Terminal", "value": " "}
     ]
}
```

Figure 1: Output for listing 2

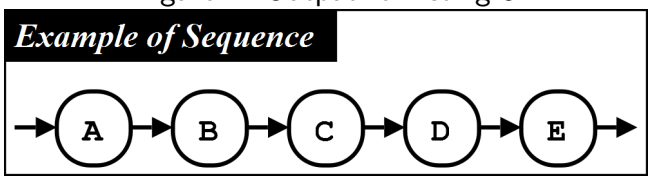


2 Sequence example

Listing 3: Sequence example

```
{
    "view": "Group",
    "name": "Example of Sequence",
    "children": [
        {"view": "Terminal", "value": "A"},
        {
                'children': [
                        {"view": "Terminal", "value": "B"},
                        {"view": "Terminal", "value": "C"},
                        {"view": "Terminal", "value": "D"}
                ],
                "name": "Sequence BCD",
                "view": "Sequence"
        },
        {"view": "Terminal", "value": "E"}
     ]
}
```

Figure 2: Output for listing 3

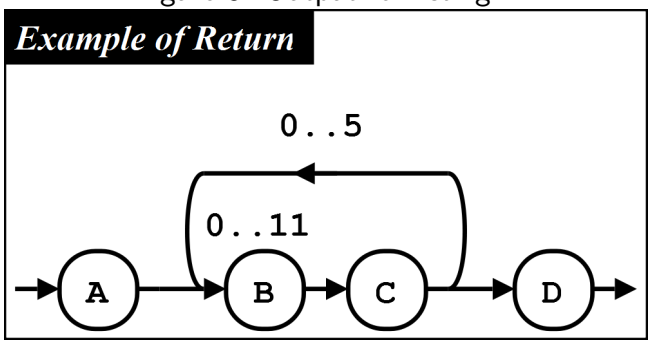


3 Return example

Listing 4: Return example

```
{
    "view": "Group",
    "name": "Example of Return",
    "children": [
        {"view": "Terminal", "value": "A"},
        {
             "children" : [
                        {
                                'children': [
                                        {
                                                'children': [
                                                        {"view": "Terminal", "value": "B"}
                                                ],
                                                "name": "Quantity Above B",
                                                "view": "QuantityAbove",
                                                "value": "0..11"
                                        },
                                        {"view": "Terminal", "value": "C"}
                                ],
                                "name": "Return BC",
                                "view": "Return"
                        }
             ],
             "name": "Quantity above Return",
             "view": "QuantityAbove",
             "value": "0..5",
        },
        {"view": "Terminal", "value": "D"}
    ]
}
```

Figure 3: Output for listing 4



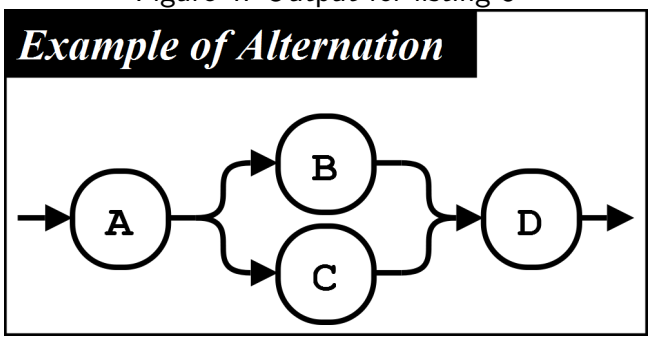
4 Alternation example

Listing 5: Alternation example

```
{
    "view": "Group",
    "name": "Example of Alternation",
    "children": [
        {"view": "Terminal", "value": "A"},
        {
                'children': [
                        {"view": "Terminal", "value": "B"},
                        {"view": "Terminal", "value": "C"}
                ],
                "name": "Alternation BC",
                "view": "Alternation"
        },
        {"view": "Terminal", "value": "D"}
```

```
    ]
}
```

Figure 4: Output for listing 5

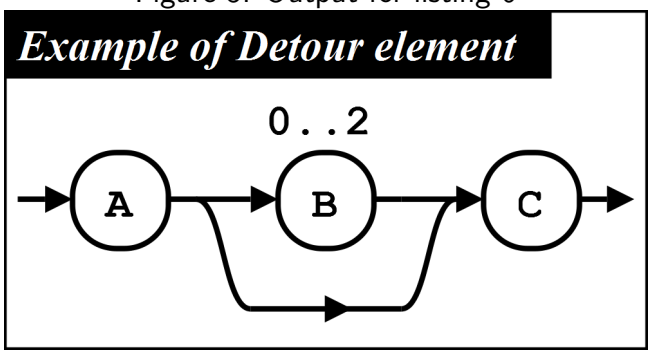


5 Detour example

Listing 6: Detour example

```
{
    "view": "Group",
    "name": "Example of Detour element",
    "children": [
        {"view": "Terminal", "value": "A"},
        {
                "view": "Detour",
                "children": [
                        {
                                "children": [
                                        {"view": "Terminal", "value": "B"}
                                ],
                                "name": "Quantity Above B",
                                "view": "QuantityAbove",
                                "value": "0..2"
                        },
                ]
        },
        {"view": "Terminal", "value": "C"}
    ]
}
```

Figure 5: Output for listing 6



6 Inverse terminal example

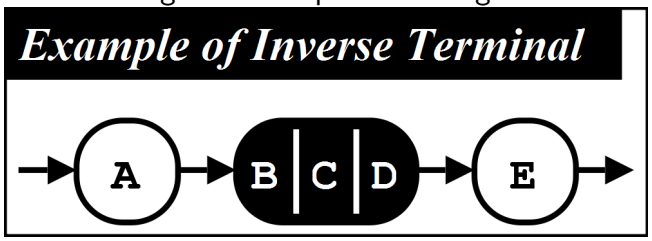
Listing 7: Inverse terminal example

```
# -*- coding: utf-8 -*-
import sys
sys.path.append('.')
from sdgen.svg import *

data = {
    "view": "Group",
    "name": "Example of Inverse Terminal",
    "children": [
        {"view": "Terminal", "value": "A"},
        {
                'children': [
                        {"view": "Terminal", "value": "B"},
                        {"view": "Terminal", "value": "C"},
                        {"view": "Terminal", "value": "D"}
                ],
                "name": "Inv Terminal BCD",
                "view": "InvTerminal"
        },
        {"view": "Terminal", "value": "E"}
     ]
}

result = as_svg(data, sys.argv[1])
print result[0][1].encode('utf-8')
```

Figure 6: Output for listing 7



7 Nonterminal example

Listing 8: Nonterminal example

```
{
    "view": "Group",
    "name": "Example of NonTerminal",
    "children": [
        {"view": "Terminal", "value": "A"},
        {
                'children': [
                        {
                                'children': [
                                        {"view": "Terminal", "value": "B"},
                                        {"view": "Terminal", "value": "C"}
                                ],
                                "name": "Non Terminal C",
                                "view": "NonTerminal"
                        }
                ],
                "name": "Quantity Above Non Terminal",
                "view": "QuantityAbove",
                "value": "0..n"
        },
        {"view": "Terminal", "value": "D"}
     ]
}
```

Figure 7: Output for listing 8

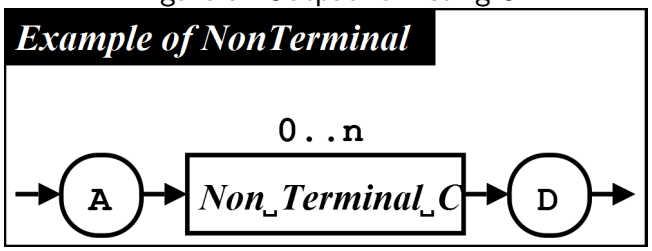
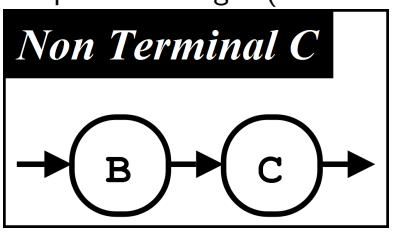


Figure 8: Output for listing 8 (internal subimage)



8 Nested groups example

Listing 9: Nested groups example

```
{
    "view": "Group",
    "name": "Example of nested groups",
    "children": [
        {"view": "Terminal", "value": "A"},
        {
                'children': [
                        {"view": "Terminal", "value": "C1"},
                        {
                                "view": "Detour",
                                'children': [
                                        {
                                        'children': [
                                                {"view": "Terminal", "value": "C"},
                                                {"view": "Terminal", "value": "D"},
                                                {"view": "Terminal", "value": "E"}
                                        ],
                                        "name": "Inv Terminal CD",
                                        "view": "InvTerminal"
                                        }
                                ],
                        },
                        {"view": "Terminal", "value": "C2"},
                ],
                "name": "Internal group",
                "view": "Group"
        },
        {"view": "Terminal", "value": "B"}
    ]
}
```

Figure 9: Output for listing 9