Syntax Diagram Generator Documentation

Release 2.0

Bartosz Alchimowicz

March 17, 2012

CONTENTS

Contents:

CHAPTER

ONE

SHORT DESCRIPTION

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

CHAPTER

TWO

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
```

CHAPTER

THREE

RUNNING AN APPLICATION

To get diagrams, an appriopriate Python file should be prepared:

```
# -*- 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, argv[1])
#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.

CHAPTER

FOUR

EXAMPLES

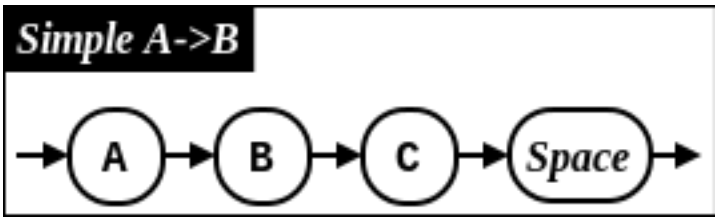
Below there are some examples of using Syntax Diagram Generator to generate images.

4.1 Terminal example

```
import sys
sys.path.append('.')
from sdgen.svg import *

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

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



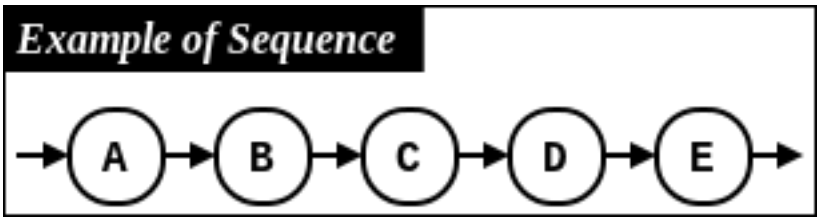
4.2 Sequence example

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

data = {
    "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"}
     ]
}

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



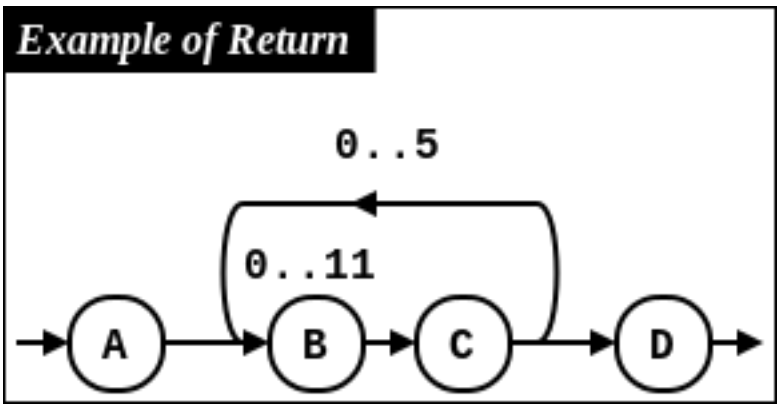
4.3 Return example

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

data = {
    "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"}
    ]
}

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

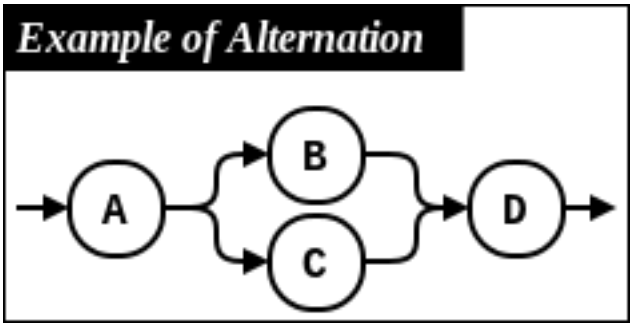


4.4 Alternation example

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

data = {
    "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"}
    ]
}

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

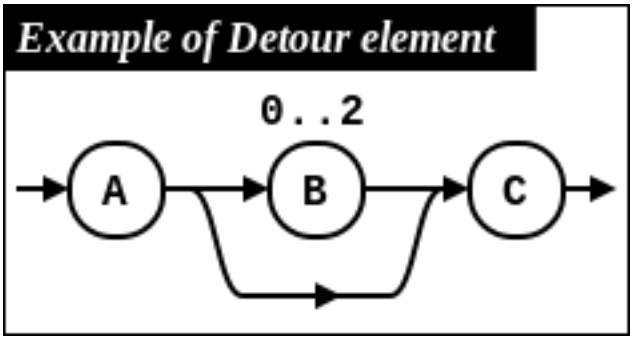


4.5 Detour example

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

data = {
    "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"}
     ]
}

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

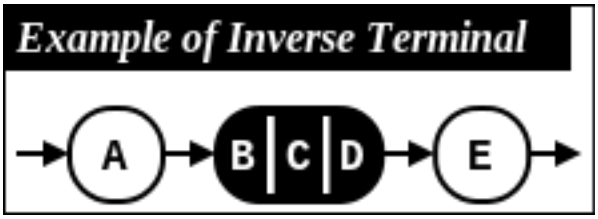


4.6 Inverse Terminal example

```
# -*- 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')
```



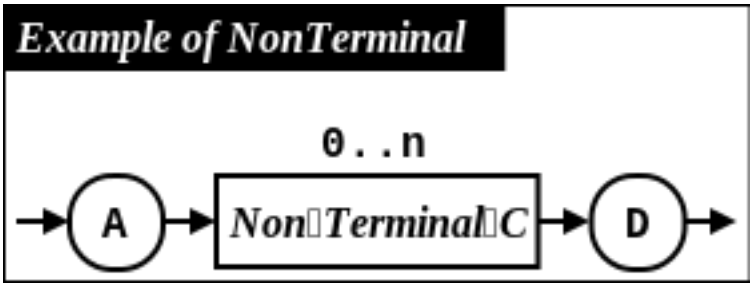
4.7 Nonterminal example

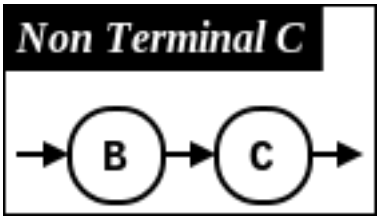
```
# -*- coding: utf-8 -*-
import sys
sys.path.append('.')
from sdgen.svg import *

data = {
    "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"}
        ]
}

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





4.8 Nested groups example

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

data = {
    "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"}
    ]
}

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