Grammar v0.3

Changes since last version

- Add production rules for frame slices, similar to Python [1:] and [1:5, :]
- Delete unnecessary production rules for lists (enumeration & itemization), and treat everything related to text as a text box (a table with 1 column).

Abstract

This is the grammar for controlling the order of appearance for 3 main objects: text, image, and table. It works off of Latex syntax with added sequence number header.

Alphabet

The alphabet is the same as the Python alphabet.

Production rules

every line starts with sequence number followed by the object to control

S → <comment> | “<” <seqNum> “>” <objectIdentifier> <newline>

<objectIdentifier> → <imgIdentifier> | <tabIdentifier> | <textIdentifier>

images and svg are treated as a single item

<imgIdentifier> → <imgID>

<imgID> → <stringIdentifier>

can control table as a whole, just an array or cell, or the table ruler

<tabIdentifier> → <tabID> | <arrID> | <ruleID>

<tabID> → <stringIdentifier>

<arrID> → <tabID> <indexChain>

<ruleID> → <topruleID> | <bottomruleID>

<topruleID> → “\toprule{” < arrList> “}”

<bottomruleID> → “\bottomrule{” <arrList> “}”

<arrList> → <arrID> λ | “,” λ | <arrList>

<indexChain> → “[” <index> “]” | “[” <index> “]” “[” <index> “]” | <numpyIndex>

array can have up to 2 dimensions only, because more than that the display would be incomprehensible

text could be a string literal, or an item in a (enumerated) list

<textIdentifier> → <textboxID> | <stringLiteral> | <enumeration> | <itemization>

<textboxID> → <stringIdentifier> λ | “[“ <index> “]”

comments

<comment> → “#” <stringLiteral>

commonly used items

<stringIdentifier> → <char> <stringIdentifierTail>

<stringIdentifierTail> → λ | <char> | <number> <stringItendifierTail>

<stringLiteral> → <char>*

<seqNum> → <number>+ | <number>+ “-” | <number>+ “-” <number>+

<numpyIndex> → “[” <index> “]” | “[” <index> “,” <index> “]” | <numpySlice>

<numpySlice> → “[” λ | <index> “:” λ | <index> “,” λ | <index> “:” λ | <index> “]” |

“[” λ | <index> “:” λ | <index> “]”

terminal symbols

<char> → “a” |…| “z” | “A” |…| “Z” | “_”

<index> → <number>+

<number> → “0” |…| “9”

<newline> → “\n”

Sample code

Images

```
<1-> tensor_img
<2-3> highlighted_tensor
<3> random_graph
```

Textbox

```
# no more string literal, just texbox ID
<1-> textbox0

<2> textbox1[0]
<3-4> textbox2[3]
<4> textbox1[2]
```

Table

```
# the whole table
<3-5> tab1

# one row of the table
<4> tab1[2]

# to select 1 column of the table, we must use list comprehension
col1 = [row[1] for row in tab1]
# or we must select column via numpy or pandas
import numpy as np
tab1np = np.array(tab1)
col1 = tab1np[:, 1] # or [:,:], [2, 3], [1, :] etc.
# then:
<5> col1

# a single cell in the table
<1> tab1[1][2]
<2> tab1[0][0]

# the rule of selected cells
<3> \toprule{tab1[2][2], tab1[2][3]}

# the rule of the entire row
<4> \bottomrule{tab1[2]}
```