

京都産業大学 コンピュータ工学部 青木研究室 第4回中間発表  
2012年12月26日(水) 15:00～18:00 第2実験室棟 3階 65実験室

スパゲッティを題材とした  
ソフトウェアプログラム可食化に関する研究  
(中間発表4)

京都産業大学 コンピュータ工学部 ネットワークメディア学科

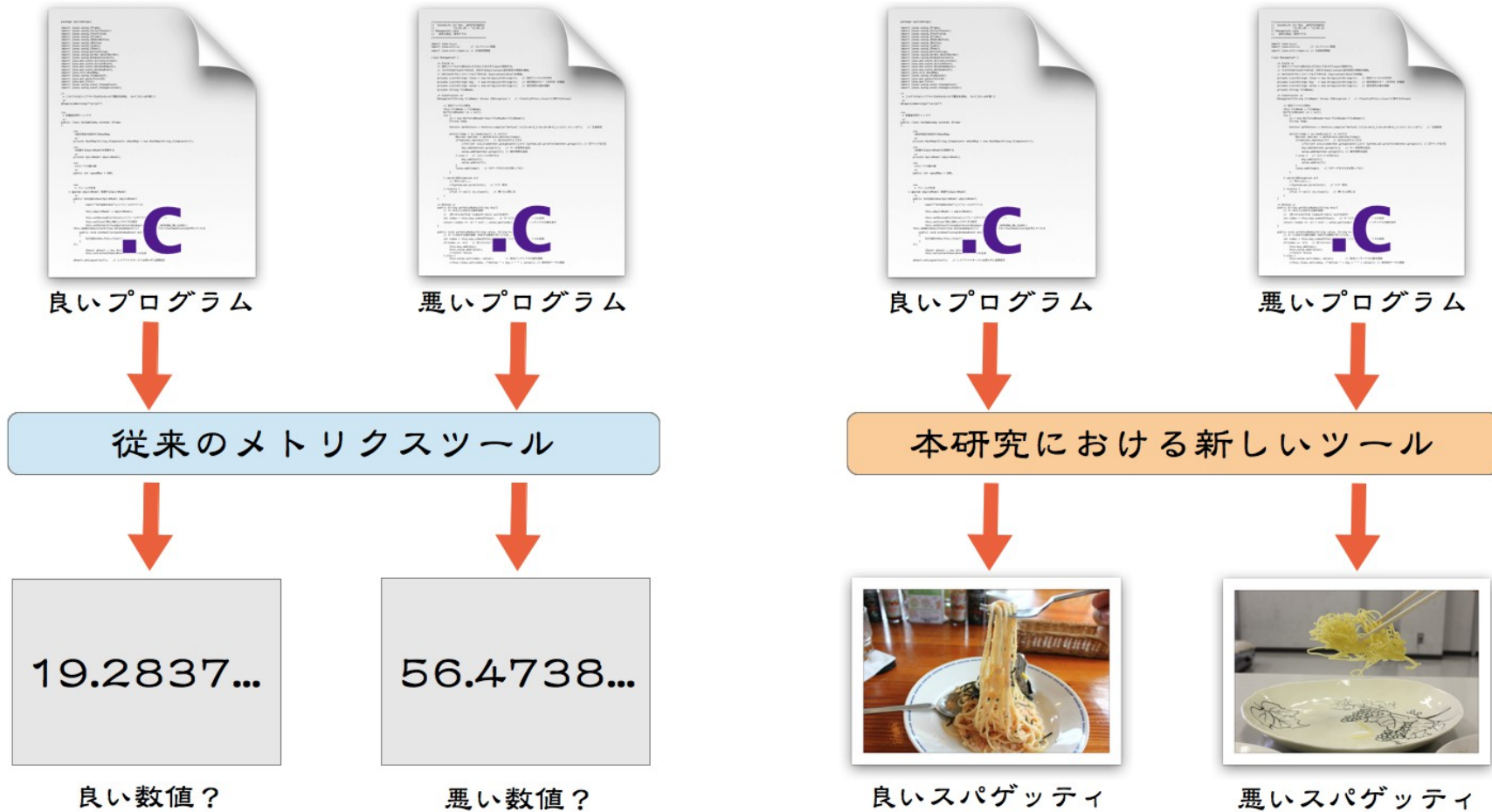
宮崎 雅文

[g0947424@cse.kyoto-su.ac.jp](mailto:g0947424@cse.kyoto-su.ac.jp)

# 目次

1. はじめに（これまでの経緯）
2. 実装物と技術的あれこれ
3. 今後の見通し（実装・実験・論文）
4. さいごに

# 研究の概要

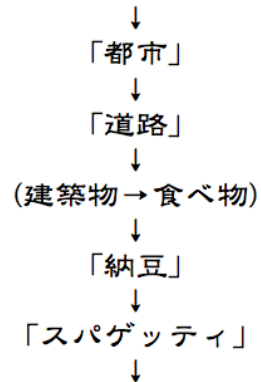


「**可食化**」という新しいアプローチの可能性を探る研究です。

# 第1回：研究テーマ決定

現在のテーマ名の紹介・そこに至った由縁

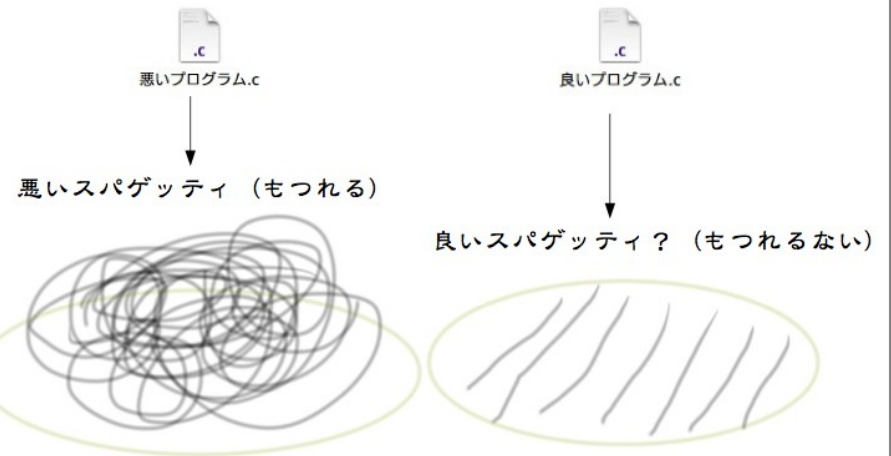
Software System as Cities 【論文】



スパゲッティプログラム？！

2012年06月27日 特別研究 第1回中間発表  
2. 今日に至った経緯：7 of 11

勘違いで青ざめて、工夫で開き直す



2012年06月27日 特別研究 第1回中間発表  
2. 今日に至った経緯：10 of 11

# 第2回：スパゲッティの調査報告

## 「スパゲッティ」と「スパゲッティプログラム」

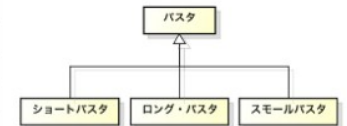


2012年08月22日 特別研究 第2回中間発表  
3. 「スパゲッティ」と「スパゲッティプログラム」：1 of 9

## スパゲッティ — ピンからキリまで

### 【太さ・形状】

太さ	名称	由来 <sup>[2]</sup>
約1.0mm	カペリ・ダンジェロ (Capelli d'angelo)	天使の髪
約1.0-1.2mm	カペリーニ (Capellini)	髪の毛
約1.2-1.4mm	フェデーニ (Fedelini)	糸
約1.4-1.6mm	スパゲッティニ (Spaghettini)	細いスパゲッティ
約1.6-1.9mm	スパゲッティ (Spaghetti)	細いひも
約2.0-2.5mm	ヴェルミチェリ (Vermicelli)	小さい芋虫
約2.0-2.5mm	ブカティーニ (Bucatini)	穴
約5-8mm	ツイーティ (Ziti)	夫婦 (?)
約2mm (板)	タリオリーニ (Tagliolini)	切る
約8mm (板)	タリアテッレ (Tagliatelle)	切る
約10mm (板)	フェットゥッチーネ (Fettuccine)	小さなリボン
約20mm (板)	パッパルデッレ (Pappardelle)	食いしん坊



2012年08月22日 特別研究 第2回中間発表  
3. 「スパゲッティ」と「スパゲッティプログラム」：2 of 9

# 第3回：プログラムの調査報告

## プログラムの古今

### 【昔々】職人技の時代

低級言語（機械語）で、実行効率の良い（gotoだらけの）ものを。  
システム規模拡大で職人技も限界 → **スパゲッティプログラム発生**

### 【近世】構造化プログラミング

エドガー・ダイクストラの基本3構造：「順次」「分岐」「反復」  
構築技法が整備されてきて、さらにシステム規模は膨張。。。

### 【現代】銀の弾丸を求めて

Javaだ、VBだ、ウェブアプリだ、オブジェクト指向だ、LLだ...  
決定的な打開策を探し求め、どこへやら...

2012年10月24日 特別研究 第3回中間発表  
3. プログラムの性質：2 of 4

## プログラミング初学者向けのメトリクス

寺子屋で垣間見た「初学者のプログラム」を活かして計測する。

「**初学者には良いプログラム**」 or 「**初学者らしい悪いプログラム**」

プログラムの指標値	スパゲッティの指標値	関係
関数・メソッド宣言数/行数	作り置き時間	負の相関
変数・関数の名称	スパゲッティ料理の名称	正の相関
インデント・スペース	ゆで時間・太さ	正の相関
コメント記述量	劣化しやすさ	負の相関

2012年10月24日 特別研究 第3回中間発表  
5. スパゲッティへの画像：2 of 6

## 第4回：実装と実験の終了（嘘）

あと数歩分の実装が間に合いませんでした...

つまり、実験などできるはずもなく...

# 実装物と技術的あれこれ



# 作るべき「新しいメトリクスツール」



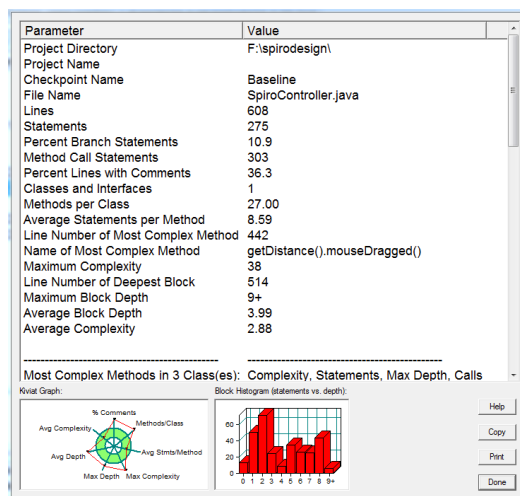
ソースコード



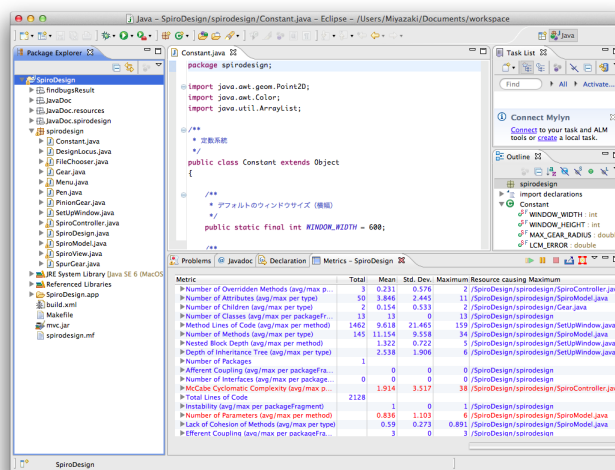
スパゲッティ

# 「冗長も大切」とは言うけれど...

また独立したツール（3つ目）を作るのか？



簡易なメトリクスツール



詳細なメトリクスツール



スパゲッティへのメトリクスツール

Parameter	Value
Project Directory	F:\spirodesign\
Project Name	SpiroController.java
Checkpoint Name	Baseline
File Name	608
Lines	275
Statements	10.9
Percent Branch Statements	303
Method Call Statements	36.3
Percent Lines with Comments	1
Classes and Interfaces	27.00
Methods per Class	8.59
Average Statements per Method	442
Line Number of Most Complex Method	getDistance().mouseDragged()
Name of Most Complex Method	38
Maximum Complexity	514
Line Number of Deepest Block	9+
Maximum Block Depth	3.99
Average Block Depth	2.88
Average Complexity	

Most Complex Methods in 3 Class(es): Complexity, Statements, Max Depth, Calls

簡易なメトリクスツール

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure for SpiroDesign, including packages like spirodesign and classes like Constant.java.
- Code Editor:** Displays the source code for Constant.java, showing imports, class declarations, and a static final integer constant.
- Metrics Window:** A table of metrics for the project.

Metric	Total	Mean	Std. Dev.	Maximum Resource causing Maximum
Number of Overridden Methods (avg/max per type)	3	0.231	0.576	2   SpiroDesign/spirodesign/SpiroController.java
Number of Attributes (avg/max per type)	50	3.846	2.446	11   SpiroDesign/spirodesign/SpiroModel.java
Number of Children (avg/max per type)	2	0.154	0.533	2   SpiroDesign/spirodesign/Clear.java
Number of Classes (avg/max per package)	13	13	0	13   SpiroDesign/spirodesign
Method Lines of Code (avg/max per method)	1462	9.618	21.465	159   SpiroDesign/spirodesign/SetSpWindow.java
Number of Methods (avg/max per type)	145	11.154	9.558	34   SpiroDesign/spirodesign/SpiroModel.java
Nearest Block Depth (avg/max per method)	1	1.322	0.722	5   SpiroDesign/spirodesign/SetSpWindow.java
Depth of Inheritance Tree (avg/max per type)	2	2.538	1.906	6   SpiroDesign/spirodesign/SetSpWindow.java
Number of Packages	1	0	0	0   SpiroDesign/spirodesign
Afferent Coupling (avg/max per package)	0	0	0	0   SpiroDesign/spirodesign
Number of Interfaces (avg/max per package)	0	0	0	0   SpiroDesign/spirodesign
McCabe Cyclomatic Complexity (avg/max per package)	1	1.914	3.517	34   SpiroDesign/spirodesign/SpiroController.java
Total Lines of Code	2128	1	0	1   SpiroDesign/spirodesign
Instability (avg/max per package)	0.836	1.103	0.891	6   SpiroDesign/spirodesign/SpiroModel.java
Number of Parameters (avg/max per method)	0.59	0.273	0.891	3   SpiroDesign/spirodesign/SpiroModel.java
Lack of Cohesion of Methods (avg/max per type)	3	0	0	3   SpiroDesign/spirodesign
Effort Couplings (avg/max per package)	3	0	0	3   SpiroDesign/spirodesign

詳細なメトリクスツール



スパゲッティへのメトリクスツール

特殊な装置

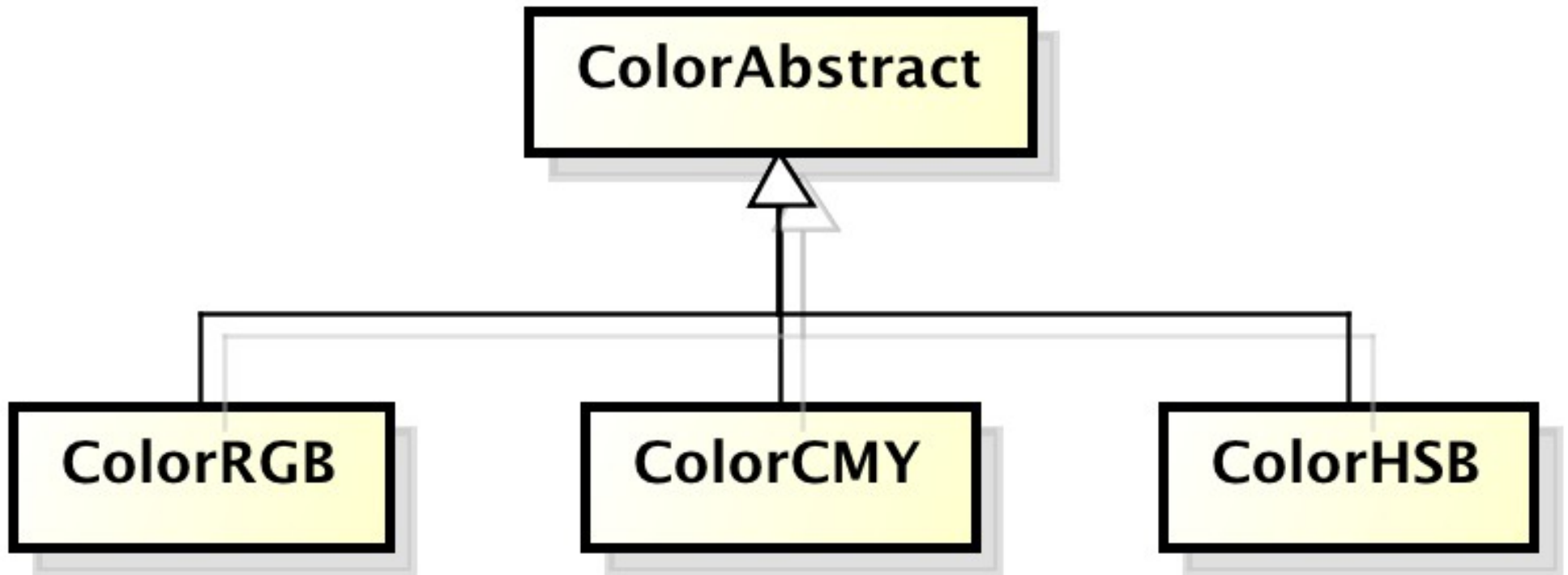
特殊な装置

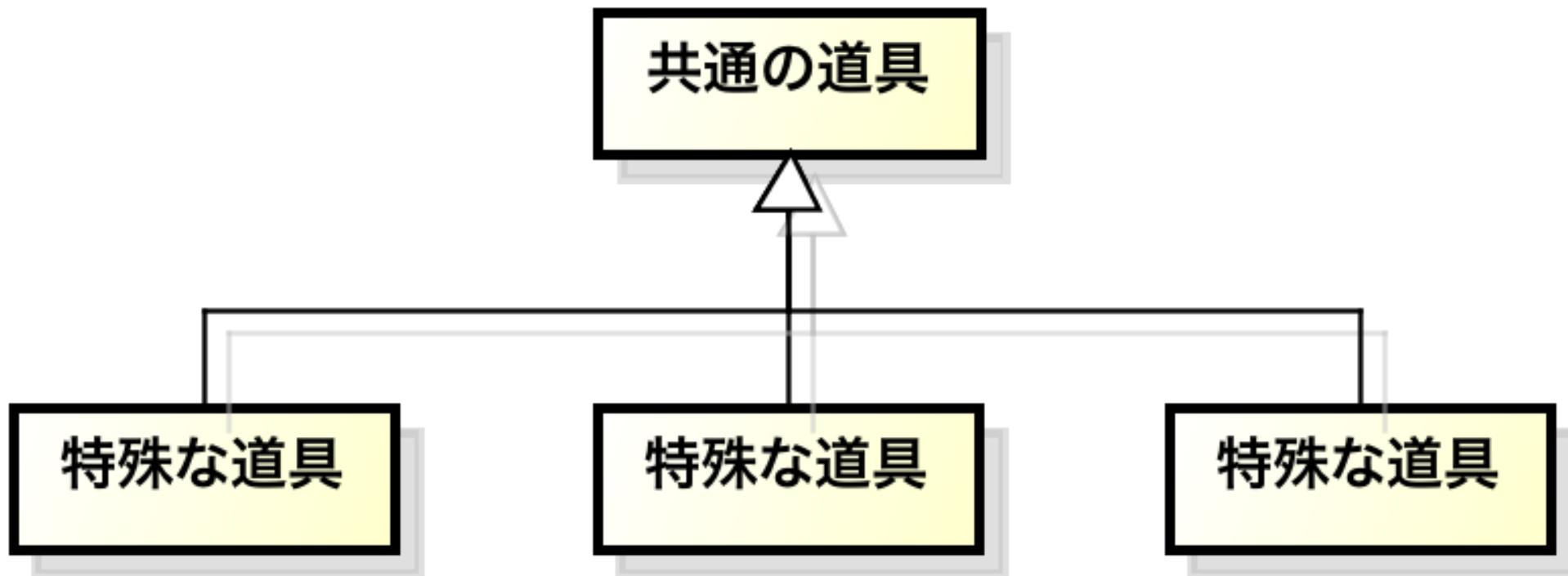
特殊な装置

**ColorRGB**

**ColorCMY**

**ColorHSB**





# 抽象的な「写像ツール」

写像ツール

Parameter	Value
Project Directory	F:\spirodesign\
Project Name	SpiroController.java
Checkpoint Name	Baseline
File Name	SpiroController.java
Lines	608
Statements	275
Percent Branch Statements	10.9
Method Call Statements	303
Percent Lines with Comments	36.3
Classes and Interfaces	1
Methods per Class	27.00
Average Statements per Method	8.59
Line Number of Most Complex Method	442
Name of Most Complex Method	getDistance().mouseDragged()
Maximum Complexity	38
Line Number of Deepest Block	514
Maximum Block Depth	9+
Average Block Depth	3.99
Average Complexity	2.88

Most Complex Methods in 3 Class(es): Complexity, Statements, Max Depth, Calls

Kiviat Graph: Avg Complexity, % Comments, Methods/Class, Avg Stmt/Method, Max Depth, Max Complexity

Block Histogram (statements vs. depth):

```
package spirodesign;

import java.awt.geom.Point2D;
import java.awt.Color;
import java.util.ArrayList;

/**
 * 定数類
 */
public class Constant extends Object
{
    /**
     * サワネットのウィンドウサイズ (横)
     */
    public static final int WINDOW_WIDTH = 680;
}
```

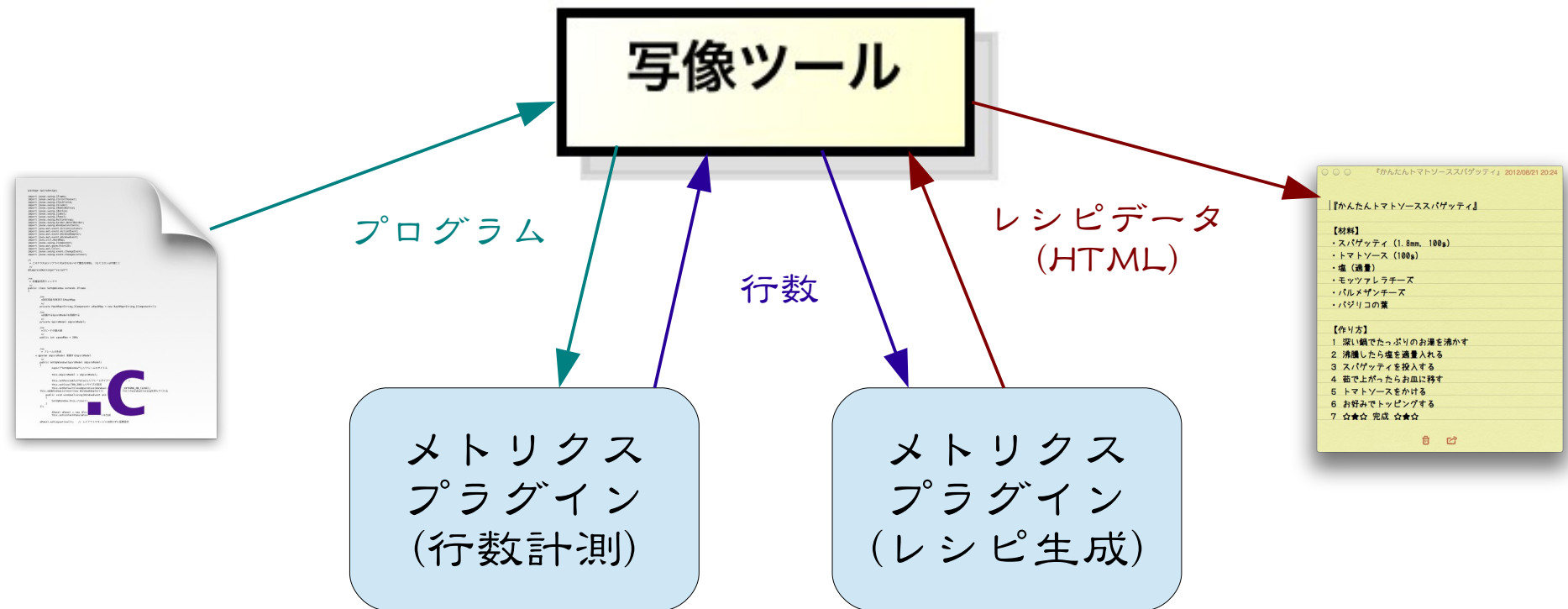
Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum
Number of Overridden Methods (avg/max per package)	3	0.231	0.376	2	/SpiroDesign/spirodesign/SpiroController.java
Number of Attributes (avg/max per type)	50	3.846	2.445	11	/SpiroDesign/spirodesign/SpiroModel.java
Number of Children (avg/max per type)	2	0.154	0.333	2	/SpiroDesign/spirodesign/Car.java
Number of Classes (avg/max per package)	13	1.3	0	13	/SpiroDesign/spirodesign
Method Lines of Code (avg/max per method)	1462	9.618	21.465	159	/SpiroDesign/spirodesign/SpiroWindow.java
Number of Methods (avg/max per type)	145	11.154	9.358	34	/SpiroDesign/spirodesign/SpiroModel.java
Nested Block Depth (avg/max per method)	1	1.322	0.722	5	/SpiroDesign/spirodesign/SpiroWindow.java
Depth of Inheritance Tree (avg/max per type)	1	2.538	1.906	6	/SpiroDesign/spirodesign/SpiroWindow.java
Number of Packages	1	0	0	0	/SpiroDesign/spirodesign
Affluent Coupling (avg/max per package)	0	0	0	0	/SpiroDesign/spirodesign
Number of Interfaces (avg/max per package)	0	0	0	0	/SpiroDesign/spirodesign
McCabe Cyclomatic Complexity (avg/max per package)	1,914	3.317	18	18	/SpiroDesign/spirodesign/SpiroController.java
Total Lines of Code	2128	1	0	1	/SpiroDesign/spirodesign
Instability (avg/max per package/fragment)	0.836	1.103	6	6	/SpiroDesign/spirodesign/SpiroModel.java
Number of Parameters (avg/max per method)	0.59	0.273	0.831	3	/SpiroDesign/spirodesign/SpiroModel.java
Lack of Cohesion of Methods (avg/max per type)	3	0	0	3	/SpiroDesign/spirodesign
Effort Couplings (avg/max per package/fragment)	3	0	0	3	/SpiroDesign/spirodesign



利点1：保守性・再利用性が高まる。



# もう一つの大切な利点

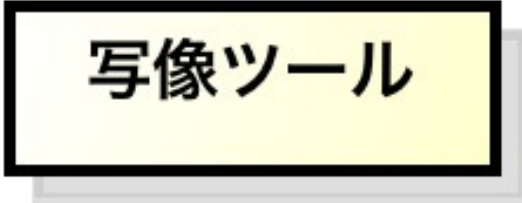


利点2：あるメトリクス結果を他のメトリクスに利用できる。  
メトリクスの融合が簡単に実現できる。

## 写像ツール

# SynapseScript NEW!!

プラグイン同士をつなげるためのプログラミング言語



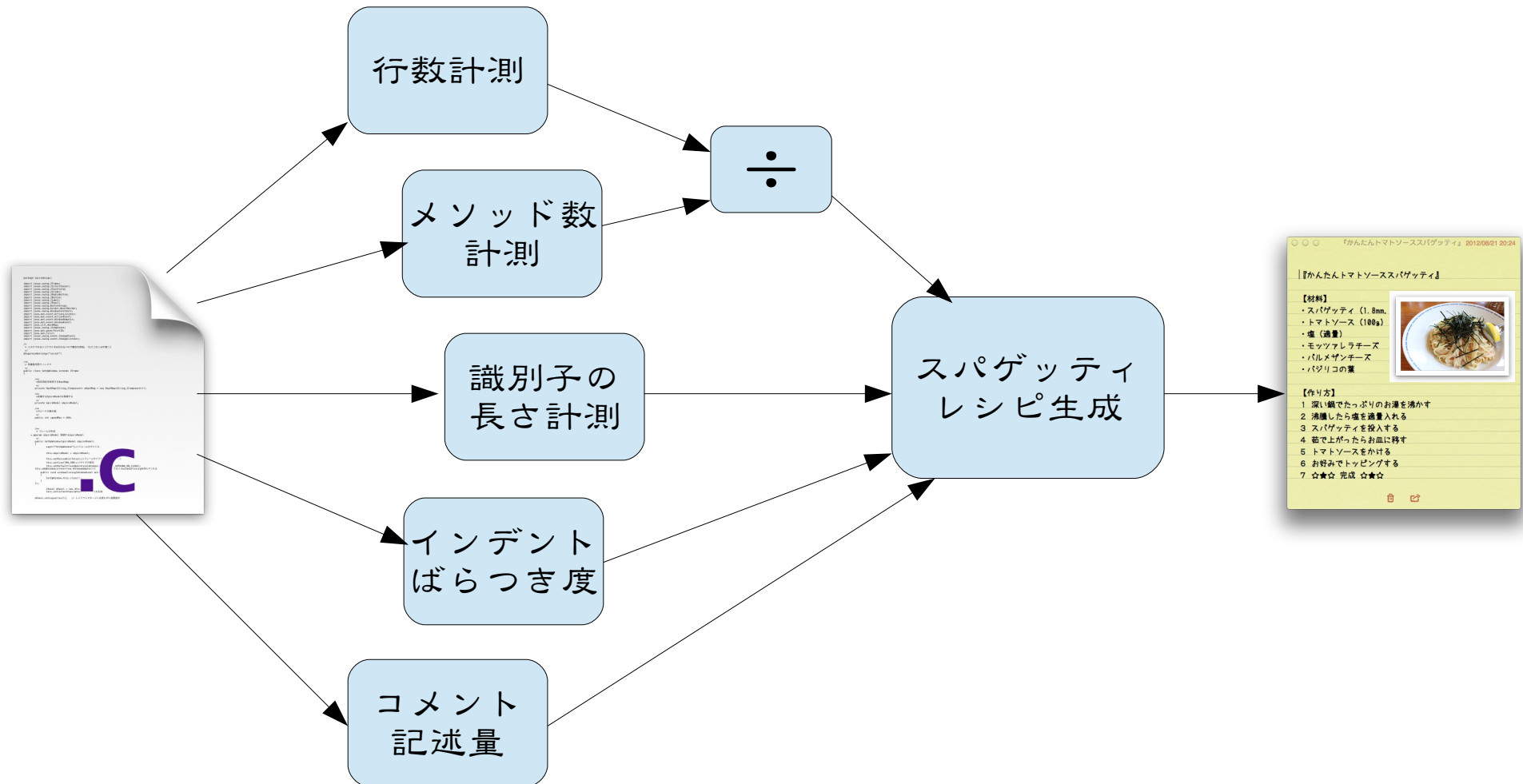
写像ツール

# プログラム→スパゲッティ

前回の中間発表(3)で紹介した指標群。

プログラムの指標値	スパゲッティの指標値	関係
関数・メソッド宣言数/行数	作り置き時間	負の相関
変数・関数の名称	スパゲッティ料理の名称	正の相関
インデント・スペース	ゆで時間・太さ	正の相関
コメント記述量	劣化しやすさ	負の相関

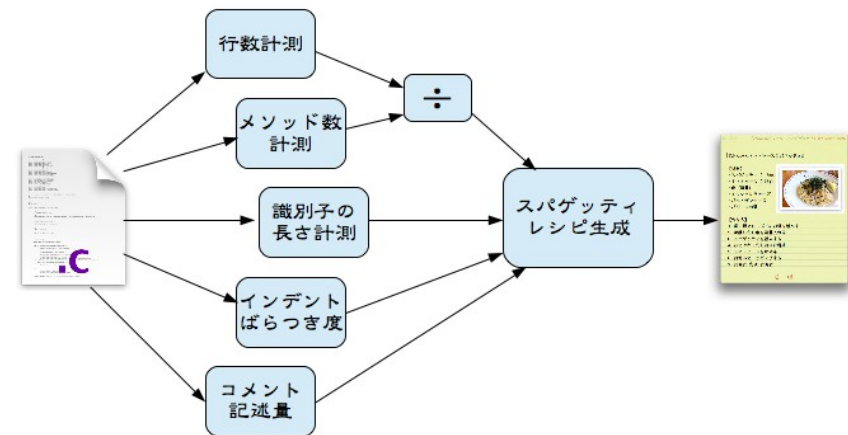
# スパゲッティへの写像



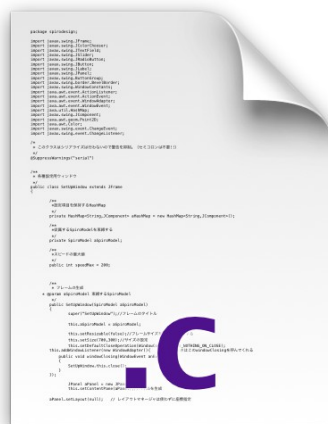
# スパゲッティへの写像

```
sourceFile      = File("/path/to/program.c", "read");  
linesOfCode    = Metrics("/path/to/metrics/LOC");  
numberOfMethods = Metrics("/path/to/metrics/NOM");  
averageLength  = Metrics("/path/to/metrics/AverageLength");  
indentVariance = Metrics("/path/to/metrics/VOI");  
commentRatio   = Metrics("/path/to/metrics/CommentRatio");  
division       = Metrics("/path/to/defaults/Division");  
codeSpaghetti  = Metrics("/path/to/metrics/CodeSpaghetti");  
htmlFile       = File("/path/to/recipe.html", "write");
```

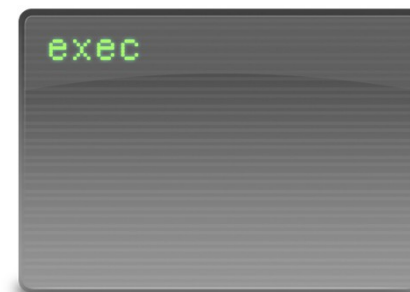
```
sourceFile  
  to: linesOfCode  
  to: division.dividend  
  to: codeSpaghetti.linesPerMethod;  
sourceFile  
  to: numberOfMethods  
  to: division.divisor;  
sourceFile  
  to: indentVariance  
  to: codeSpaghetti.indentVariance;  
sourceFile  
  to: commentRatio  
  to: codeSpaghetti.commentRatio;  
codeSpaghetti  
  to: htmlFile;
```



# ビルドツール「make」の転用



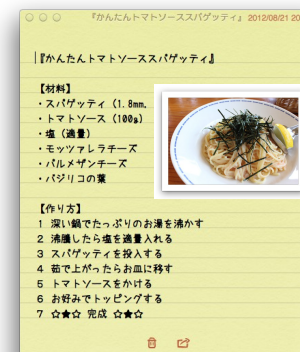
make  
( gcc )



実行可能プログラム



make  
( perl, php, awk, ... )



レシピ

利点：プラグイン単体でも利用できる。  
他のツールからも利用できる。

デモ

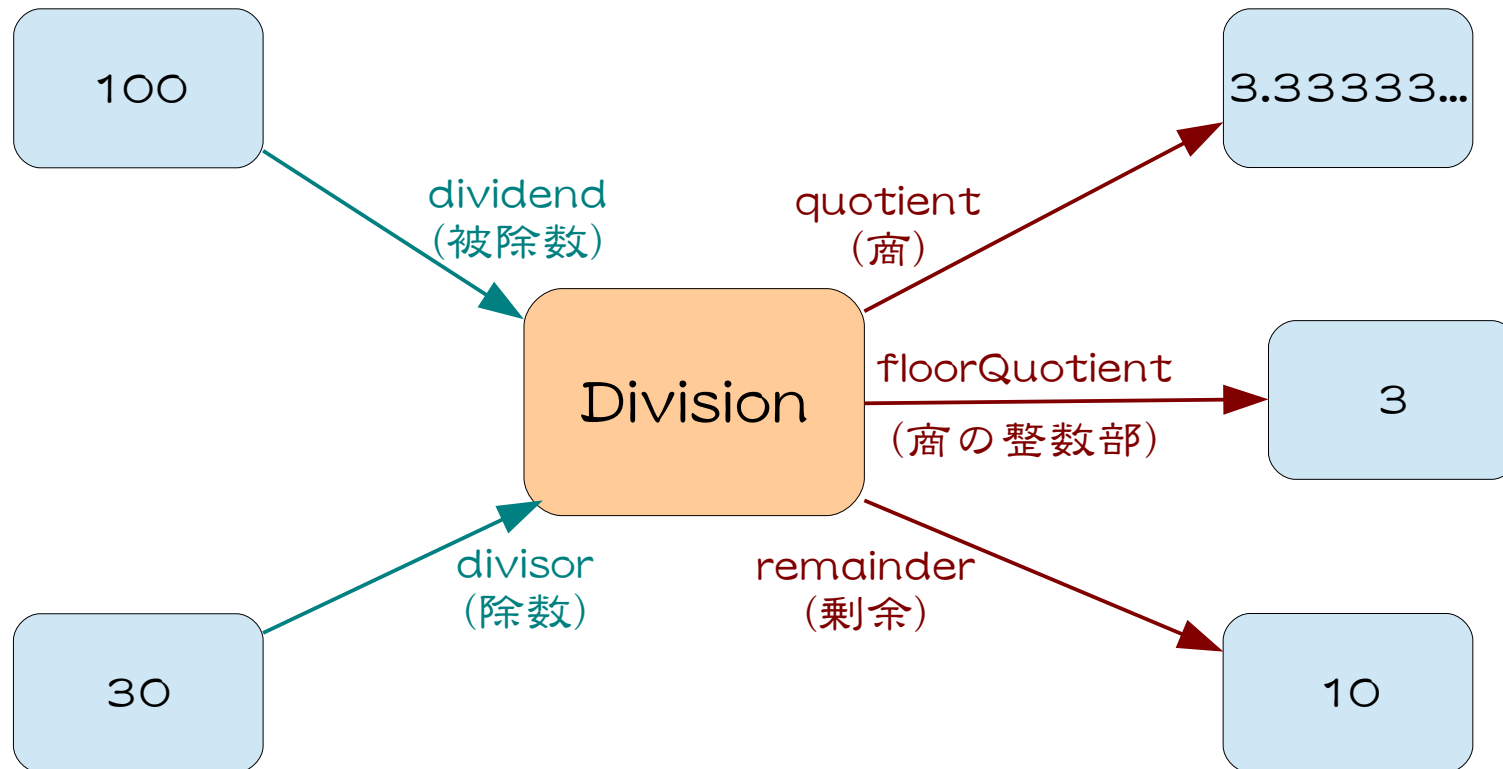
# SynapseScriptで割り算

(スパゲッティメトリクスは未完成です、ごめんなさい)

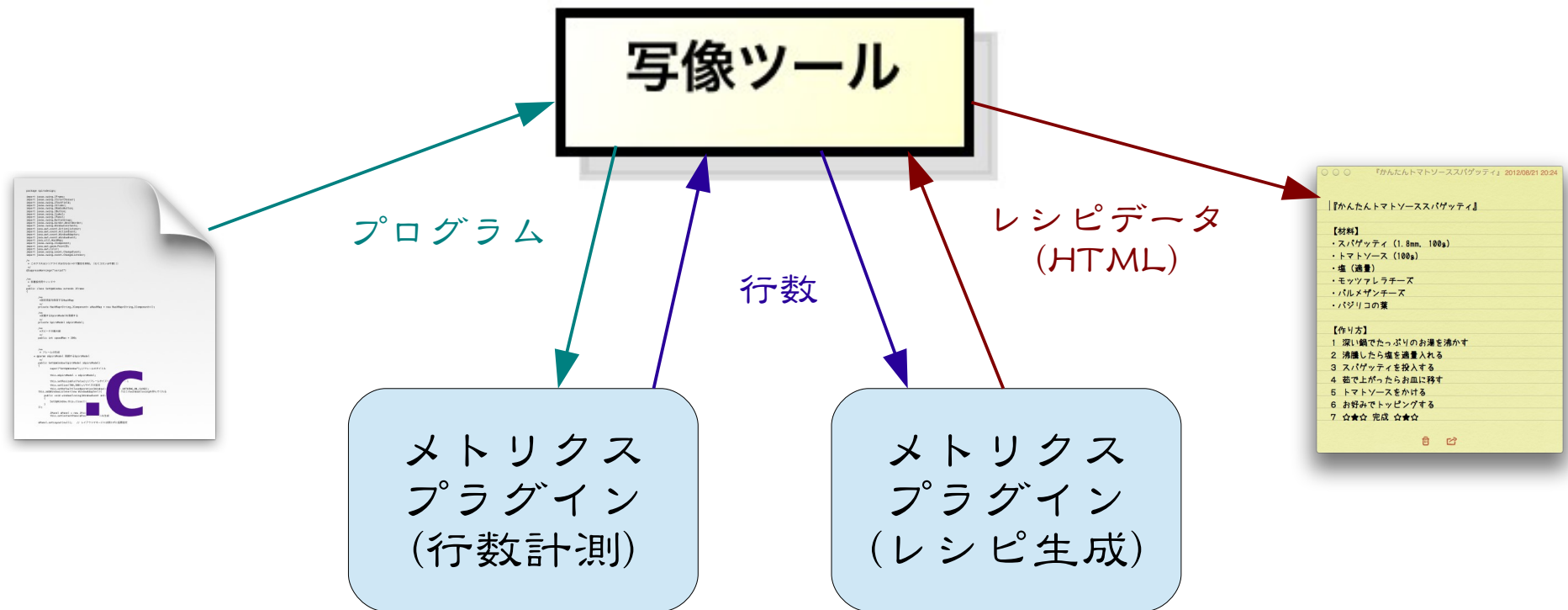


# 割り算プラグイン「Division」

2入力3出力のプラグイン。「100÷30」の模式図。







# メトリクスプラグイン NEW!!

特定の要求に応えるための小さな道具。入力したら出力される。

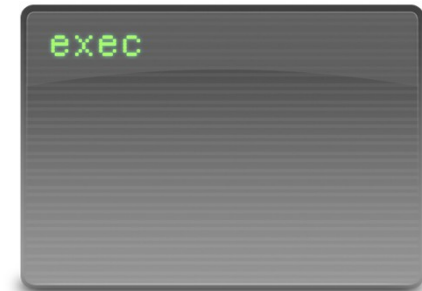
メトリクス  
プラグイン  
(行数計測)

メトリクス  
プラグイン  
(レシピ生成)

# ビルドツール「make」の転用



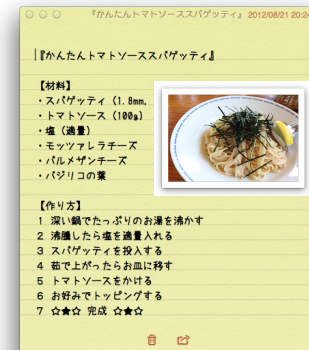
make  
( gcc )



実行可能プログラム



make  
( perl, php, awk, ... )



レシピ

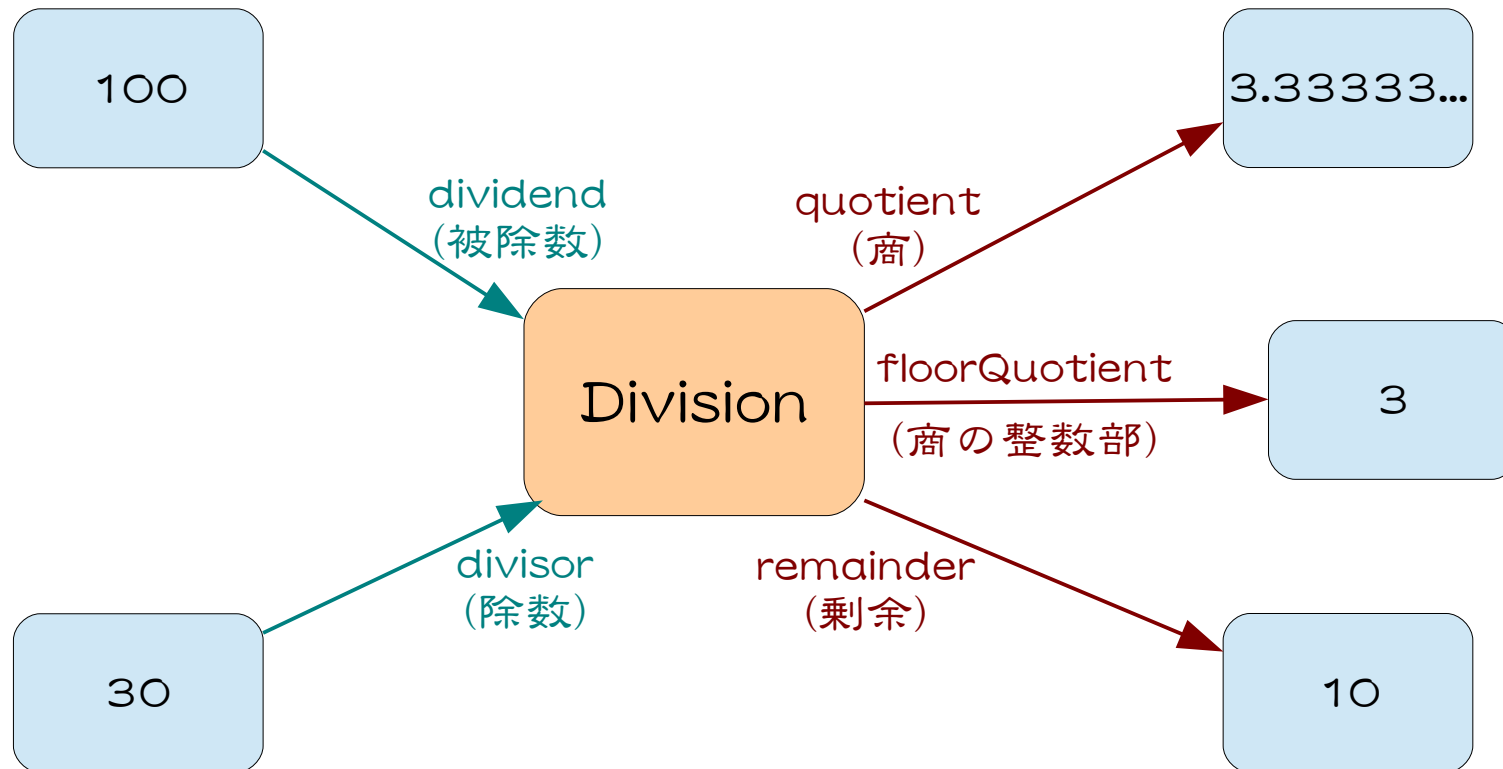
利点：プラグイン単体でも利用できる。  
他のツールからも利用できる。

デモ

メトリクスプラグインで割り算

# 割り算プラグイン「Division」(再)

2入力3出力のプラグイン。「100÷30」の模式図。







もうひとつ...

(せっかく作ったので紹介したい...)

SynapseScriptからDivisionできた。



SmalltalkからDivisionできた。



Smalltalkからmakeコマンドを叩けた。

どうやって？

# UnixProcess >> cshBullet:

csh (シェルの一種) にコマンドを送ることができるメッセージ

The screenshot shows the Smalltalk IDE interface. The title bar reads "UnixProcess class>>cshBullet:". The menu bar includes "Browser", "Edit", "Find", "View", "Package", "Class", "Protocol", "Method", "Tools", and "Help". The toolbar contains various icons for navigation and editing. The "Find:" field is empty. The interface is divided into several panes:

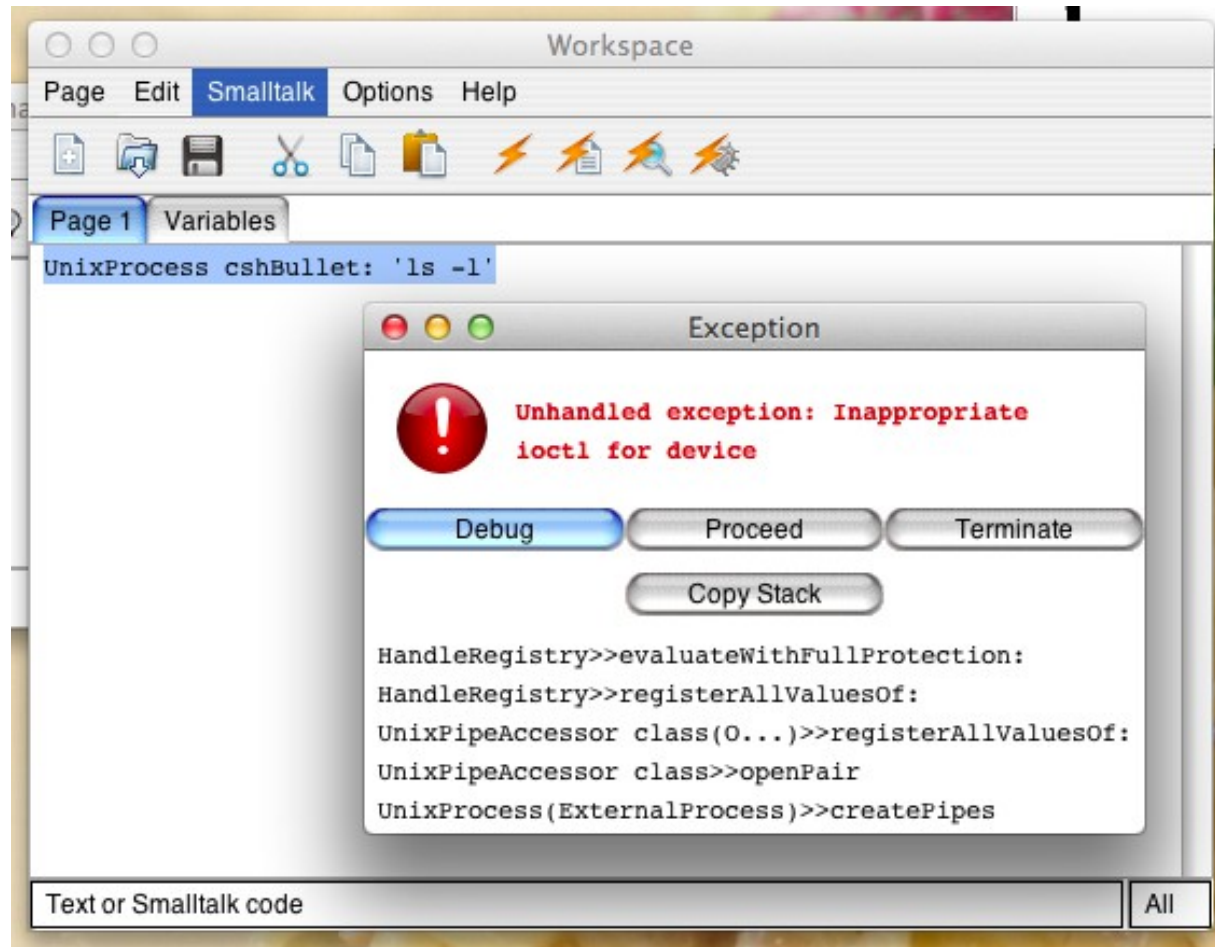
- Package/Class Browser:** Shows a tree view of packages and classes. The "UnixProcess" class is selected.
- Class/Shared Variable/Instance Variable:** The "Class" tab is active, showing a list of methods: "class initialization", "defaulting", "examples", "instance creation", "private", and "process reaping".
- Instance Variable:** The "cshBullet:" message is selected, showing its arguments: "defaultClass", "directoryListing", "forkJob:arguments:", "forkJob:arguments:descr", "forkJob:arguments:envir", "forkJob:descriptors:", "install", and "pipeConnectionFor:argum".
- Source:** Shows the source code for the "cshBullet:" message. The code is as follows:

```
cshBullet: string
    "Execute a csh command, and ignore the reply.
    Not appropriate if the command would return a long reply (>= 4096 bytes)
    since the process would never exit."
    "UnixProcess cshBullet: 'rm useless.file'"

    | cmd proc connection|
    connection := self pipeConnectionFor: 'csh'
        arguments: #('-ft')
        setProcessDescriptor: [:ipd | proc := pd].
    cmd := connection writeStream.
    [cmd nextPutAll: string; cr; commit.
    proc wait]
    ensure: [cmd close]
```
- Method:** "#cshBullet: (examples)"
- Package:** "OS-ExternalProcess"

# Smalltalkとシェルの壁

動かない orz



壁穴を見つける、それ即ちプログラマ。

Smalltalkからシェルは叩けない

けど

Cからシェルは叩けるはず

それに

SmalltalkからCも叩けるらしい (DLLCC)

つまり

Smalltalkからシェルは叩ける (え

デモ

Smalltalkからシェルコマンド



もうひとつだけ...

(せっかくの好機なので紹介したい...)



# 言語処理系

字句解析

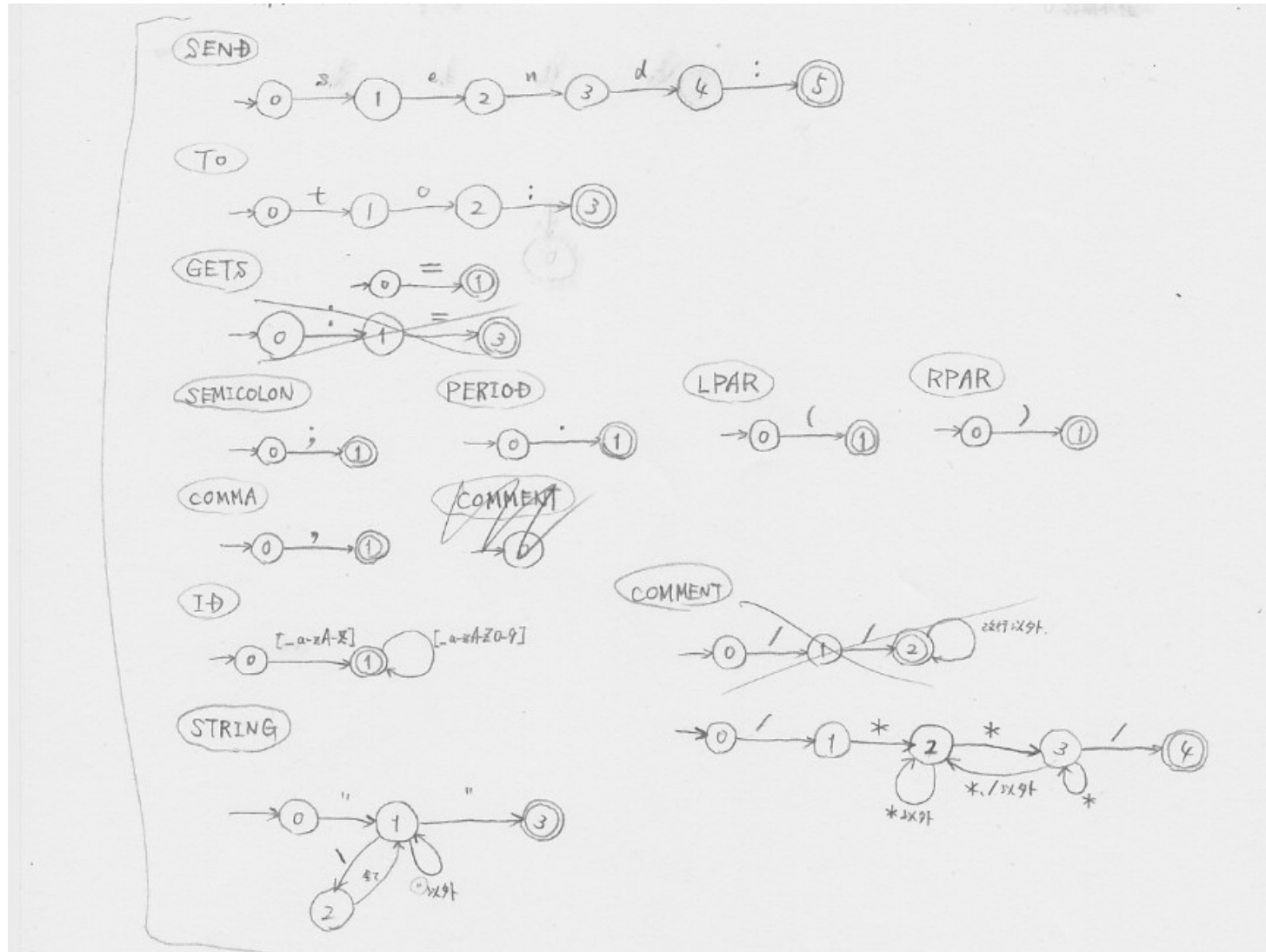
構文解析

意味解析

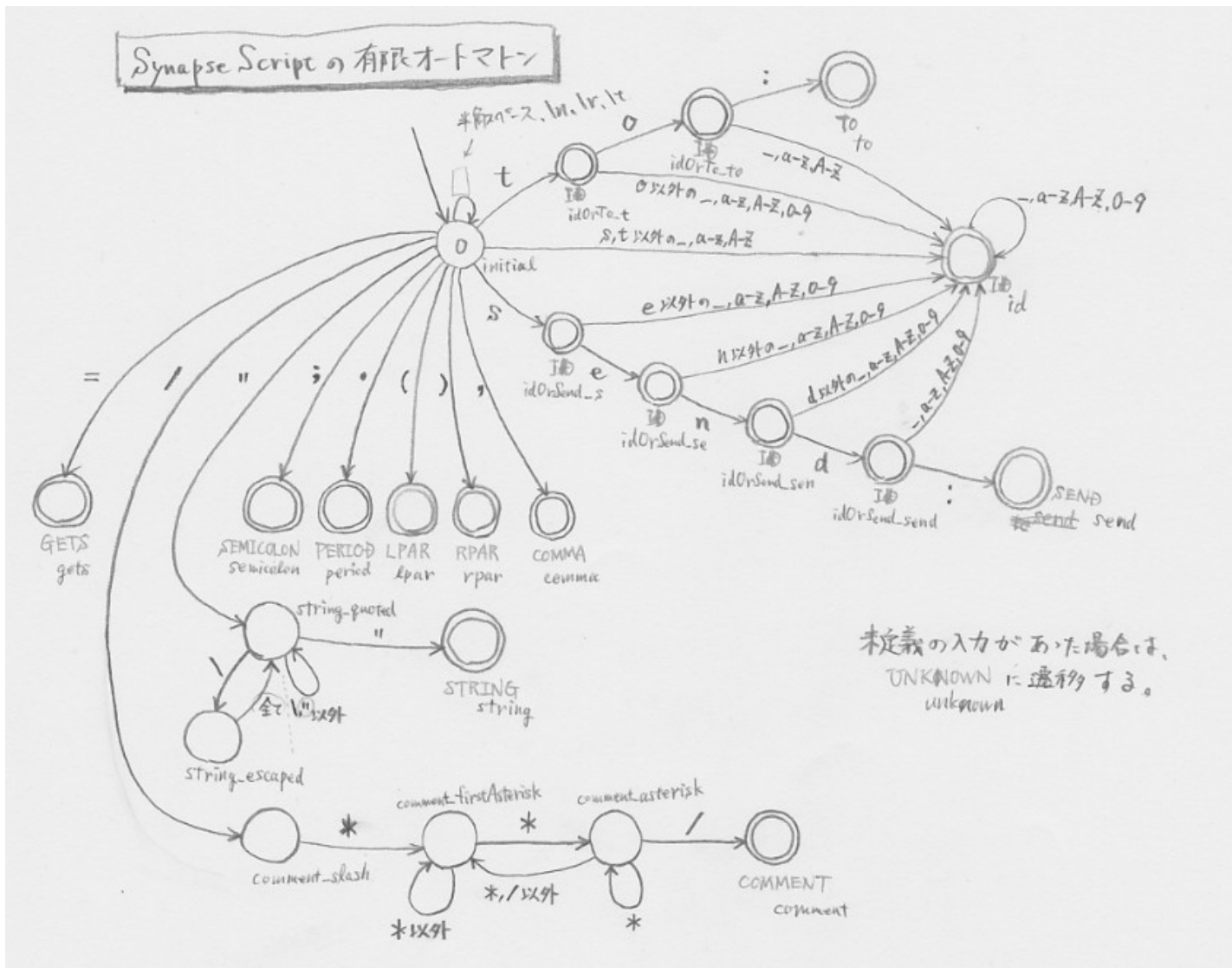
# 正規表現

```
01 : %{
02 : #include "defs.h"
03 : %}
04 : %option nounput
05 : %%
06 : "send:"           { return(SEND); }
07 : "to:"            { return(TO); }
08 : ":@"            { return(GETS); }
09 : ";"              { return(SEMICOLON); }
10 : "\."            { return(PERIOD); }
11 : "("              { return(LPAR); }
12 : ")"              { return(RPAR); }
13 : ","              { return(COMMA); }
14 : [_a-zA-Z][_a-zA-Z0-9]* { return(ID); }
15 : "\""             { parseString(); return(STRING); }
16 : "\\r\\n"|"\\r"|"\\n"|" "|"\\t" { }
17 : "//".*          { return(COMMENT); }
18 : .                { return(UNKNOWN); }
19 : %%
20 : int yywrap(void) {
21 :     return(1);
22 : }
23 : int nextChar() {
24 :     int aCharacter = input();
```

# 単体のオートマトンたち

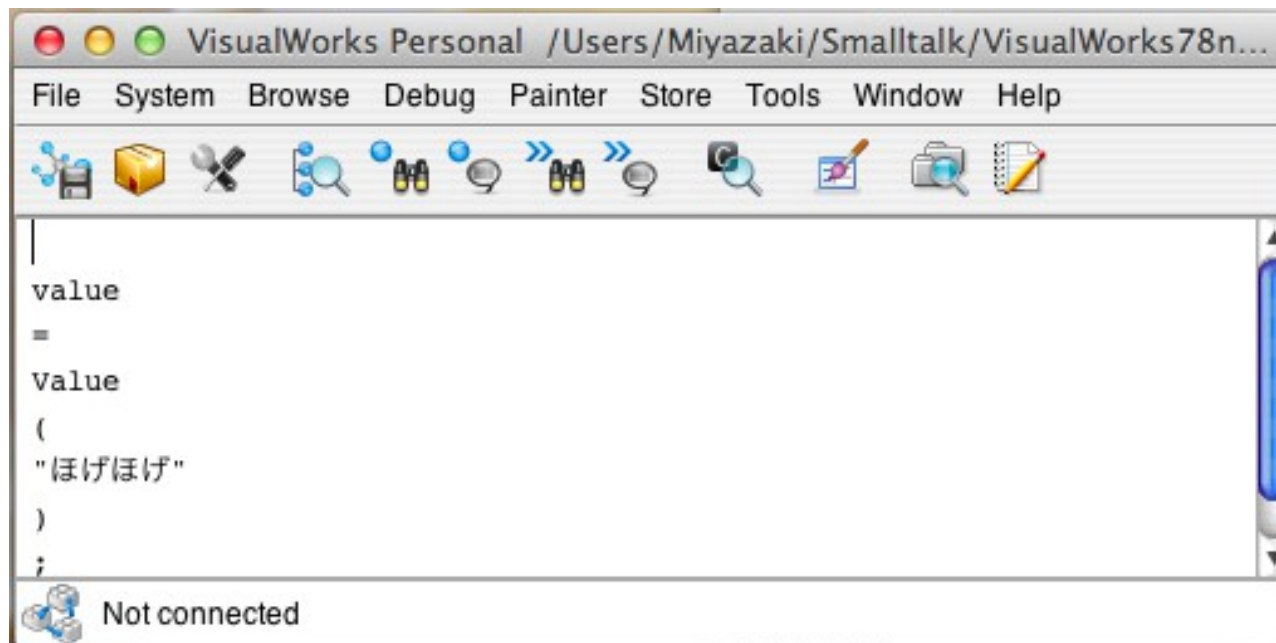


# 結合したオートマトン



# 字句解析

```
| aScanner |  
aScanner := KSU.SynapseScanner scan: 'value = Value("ほげほげ");'.  
aScanner tokenCollection do:  
    [:token |  
        Transcript  
            cr;  
            show: token tokenString]
```



# BNF

```
01 : %token    SEND TO GETS SEMICOLON PERIOD LPAR RPAR COMMA ID NUMBER STRING COMMENT UNKNOWN
02 :
03 : Program
04 :   : Statements
05 : Statements
06 :   : Statement
07 :   | Statements Statement
08 : Statement
09 :   : Assignment SEMICOLON
10 :   | Synapse SEMICOLON
11 :   | Comment
12 : Assignment
13 :   : VariableName GETS Constructor
14 : Synapse
15 :   : Variable Mappings
16 : Mappings
17 :   : Mapping
18 :   | Mappings Mapping
19 : Mapping
20 :   : TO Variable
21 :   | SEND MemberName TO Variable
22 : Variable
23 :   : VariableName
24 :   | VariableName PERIOD MemberName
25 : VariableName
26 :   : ID
27 : MemberName
28 :   : ID
29 : Constructor
30 :   : ConstructorName LPAR RPAR
31 :   | ConstructorName LPAR Arguments RPAR
32 : ConstructorName
33 :   : ID
34 : Arguments
35 :   : Argument
36 :   | Arguments COMMA Argument
37 : Argument
38 :   : STRING
39 : Comment
40 :   : COMMENT
41 :
```

# 構文のオートマトン

SynapseScript の文法に対する 閉包群 (正規集合) 2012.11.23 (Thu)

**I<sub>0</sub>**

- Program → Statements
- Statements → Statement
- Statements → Statements Statement
- Statement → Assignment SEMICOLON
- Statement → Synapse SEMICOLON
- Statement → Comment
- Assignment → VariableName GETS Constructor
- Synapse → Variable Mappings
- Variable → VariableName
- Variable → VariableName PERIOD MemberName
- VariableName → ID
- Comment → COMMENT

**I<sub>1</sub>**

- Program → Statements
- Statements → Statements Statement
- Statement → Assignment SEMICOLON
- Statement → Synapse SEMICOLON
- Statement → Comment
- Assignment → VariableName GETS Constructor
- Synapse → Variable Mapping
- Variable → VariableName
- Variable → VariableName PERIOD MemberName
- VariableName → ID
- Comment → COMMENT

**I<sub>2</sub>**

- Statements → Statement

**I<sub>3</sub>**

- Statement → Assignment SEMICOLON

**I<sub>4</sub>**

- Statement → Synapse SEMICOLON

**I<sub>5</sub>**

- Statement → Comment

**I<sub>6</sub>**

- Assignment → VariableName GETS Constructor
- Variable → VariableName
- Variable → VariableName PERIOD MemberName

**I<sub>7</sub>**

- Synapse → Variable Mappings
- Mappings → Mappings Mapping
- Mapping → TO Variable
- Mapping → SEND MemberName TO Variable

**I<sub>8</sub>**

- VariableName → ID

**I<sub>9</sub>**

- Comment → COMMENT

**I<sub>10</sub>**

- Statements → Statements Statement

**I<sub>11</sub>**

- Statement → Assignment SEMICOLON

**I<sub>12</sub>**

- Statement → Synapse SEMICOLON

**I<sub>13</sub>**

- Statement → Comment

**I<sub>14</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>15</sub>**

- Synapse → Variable Mapping

**I<sub>16</sub>**

- Variable → VariableName

**I<sub>17</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>18</sub>**

- VariableName → ID

**I<sub>19</sub>**

- Comment → COMMENT

**I<sub>20</sub>**

- Statements → Mapping

**I<sub>21</sub>**

- Mapping → TO Variable
- Variable → VariableName
- Variable → VariableName PERIOD MemberName
- VariableName → ID

**I<sub>22</sub>**

- Mapping → SEND MemberName TO Variable
- MemberName → ID

**I<sub>23</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>24</sub>**

- Constructor → ConstructorName LPAR RPARG
- Constructor → ConstructorName LPARG RPARG

**I<sub>25</sub>**

- ConstructorName → ID

**I<sub>26</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>27</sub>**

- MemberName → ID

**I<sub>28</sub>**

- Mappings → Mappings Mapping

**I<sub>29</sub>**

- Mapping → TO Variable
- Variable → VariableName
- Variable → VariableName PERIOD MemberName
- VariableName → ID

**I<sub>30</sub>**

- Mapping → SEND MemberName TO Variable
- MemberName → ID

**I<sub>31</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>32</sub>**

- Synapse → Variable Mapping

**I<sub>33</sub>**

- Variable → VariableName

**I<sub>34</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>35</sub>**

- VariableName → ID

**I<sub>36</sub>**

- Comment → COMMENT

**I<sub>37</sub>**

- Statements → Statements Statement

**I<sub>38</sub>**

- Statement → Assignment SEMICOLON

**I<sub>39</sub>**

- Statement → Synapse SEMICOLON

**I<sub>40</sub>**

- Statement → Comment

**I<sub>41</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>42</sub>**

- Synapse → Variable Mapping

**I<sub>43</sub>**

- Variable → VariableName

**I<sub>44</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>45</sub>**

- VariableName → ID

**I<sub>46</sub>**

- Comment → COMMENT

**I<sub>47</sub>**

- Statements → Statements Statement

**I<sub>48</sub>**

- Statement → Assignment SEMICOLON

**I<sub>49</sub>**

- Statement → Synapse SEMICOLON

**I<sub>50</sub>**

- Statement → Comment

**I<sub>51</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>52</sub>**

- Synapse → Variable Mapping

**I<sub>53</sub>**

- Variable → VariableName

**I<sub>54</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>55</sub>**

- VariableName → ID

**I<sub>56</sub>**

- Comment → COMMENT

**I<sub>57</sub>**

- Statements → Statements Statement

**I<sub>58</sub>**

- Statement → Assignment SEMICOLON

**I<sub>59</sub>**

- Statement → Synapse SEMICOLON

**I<sub>60</sub>**

- Statement → Comment

**I<sub>61</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>62</sub>**

- Synapse → Variable Mapping

**I<sub>63</sub>**

- Variable → VariableName

**I<sub>64</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>65</sub>**

- VariableName → ID

**I<sub>66</sub>**

- Comment → COMMENT

**I<sub>67</sub>**

- Statements → Statements Statement

**I<sub>68</sub>**

- Statement → Assignment SEMICOLON

**I<sub>69</sub>**

- Statement → Synapse SEMICOLON

**I<sub>70</sub>**

- Statement → Comment

**I<sub>71</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>72</sub>**

- Synapse → Variable Mapping

**I<sub>73</sub>**

- Variable → VariableName

**I<sub>74</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>75</sub>**

- VariableName → ID

**I<sub>76</sub>**

- Comment → COMMENT

**I<sub>77</sub>**

- Statements → Statements Statement

**I<sub>78</sub>**

- Statement → Assignment SEMICOLON

**I<sub>79</sub>**

- Statement → Synapse SEMICOLON

**I<sub>80</sub>**

- Statement → Comment

**I<sub>81</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>82</sub>**

- Synapse → Variable Mapping

**I<sub>83</sub>**

- Variable → VariableName

**I<sub>84</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>85</sub>**

- VariableName → ID

**I<sub>86</sub>**

- Comment → COMMENT

**I<sub>87</sub>**

- Statements → Statements Statement

**I<sub>88</sub>**

- Statement → Assignment SEMICOLON

**I<sub>89</sub>**

- Statement → Synapse SEMICOLON

**I<sub>90</sub>**

- Statement → Comment

**I<sub>91</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>92</sub>**

- Synapse → Variable Mapping

**I<sub>93</sub>**

- Variable → VariableName

**I<sub>94</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>95</sub>**

- VariableName → ID

**I<sub>96</sub>**

- Comment → COMMENT

**I<sub>97</sub>**

- Statements → Statements Statement

**I<sub>98</sub>**

- Statement → Assignment SEMICOLON

**I<sub>99</sub>**

- Statement → Synapse SEMICOLON

**I<sub>100</sub>**

- Statement → Comment

**I<sub>101</sub>**

- Statements → Mapping

**I<sub>102</sub>**

- Mapping → TO Variable
- Variable → VariableName
- Variable → VariableName PERIOD MemberName
- VariableName → ID

**I<sub>103</sub>**

- Mapping → SEND MemberName TO Variable
- MemberName → ID

**I<sub>104</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>105</sub>**

- Constructor → ConstructorName LPAR RPARG
- Constructor → ConstructorName LPARG RPARG

**I<sub>106</sub>**

- ConstructorName → ID

**I<sub>107</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>108</sub>**

- MemberName → ID

**I<sub>109</sub>**

- Mappings → Mappings Mapping

**I<sub>110</sub>**

- Mapping → TO Variable
- Variable → VariableName
- Variable → VariableName PERIOD MemberName
- VariableName → ID

**I<sub>111</sub>**

- Mapping → SEND MemberName TO Variable
- MemberName → ID

**I<sub>112</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>113</sub>**

- Synapse → Variable Mapping

**I<sub>114</sub>**

- Variable → VariableName

**I<sub>115</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>116</sub>**

- VariableName → ID

**I<sub>117</sub>**

- Comment → COMMENT

**I<sub>118</sub>**

- Statements → Statements Statement

**I<sub>119</sub>**

- Statement → Assignment SEMICOLON

**I<sub>120</sub>**

- Statement → Synapse SEMICOLON

**I<sub>121</sub>**

- Statement → Comment

**I<sub>122</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>123</sub>**

- Synapse → Variable Mapping

**I<sub>124</sub>**

- Variable → VariableName

**I<sub>125</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>126</sub>**

- VariableName → ID

**I<sub>127</sub>**

- Comment → COMMENT

**I<sub>128</sub>**

- Statements → Statements Statement

**I<sub>129</sub>**

- Statement → Assignment SEMICOLON

**I<sub>130</sub>**

- Statement → Synapse SEMICOLON

**I<sub>131</sub>**

- Statement → Comment

**I<sub>132</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>133</sub>**

- Synapse → Variable Mapping

**I<sub>134</sub>**

- Variable → VariableName

**I<sub>135</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>136</sub>**

- VariableName → ID

**I<sub>137</sub>**

- Comment → COMMENT

**I<sub>138</sub>**

- Statements → Statements Statement

**I<sub>139</sub>**

- Statement → Assignment SEMICOLON

**I<sub>140</sub>**

- Statement → Synapse SEMICOLON

**I<sub>141</sub>**

- Statement → Comment

**I<sub>142</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>143</sub>**

- Synapse → Variable Mapping

**I<sub>144</sub>**

- Variable → VariableName

**I<sub>145</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>146</sub>**

- VariableName → ID

**I<sub>147</sub>**

- Comment → COMMENT

**I<sub>148</sub>**

- Statements → Statements Statement

**I<sub>149</sub>**

- Statement → Assignment SEMICOLON

**I<sub>150</sub>**

- Statement → Synapse SEMICOLON

**I<sub>151</sub>**

- Statement → Comment

**I<sub>152</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>153</sub>**

- Synapse → Variable Mapping

**I<sub>154</sub>**

- Variable → VariableName

**I<sub>155</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>156</sub>**

- VariableName → ID

**I<sub>157</sub>**

- Comment → COMMENT

**I<sub>158</sub>**

- Statements → Statements Statement

**I<sub>159</sub>**

- Statement → Assignment SEMICOLON

**I<sub>160</sub>**

- Statement → Synapse SEMICOLON

**I<sub>161</sub>**

- Statement → Comment

**I<sub>162</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>163</sub>**

- Synapse → Variable Mapping

**I<sub>164</sub>**

- Variable → VariableName

**I<sub>165</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>166</sub>**

- VariableName → ID

**I<sub>167</sub>**

- Comment → COMMENT

**I<sub>168</sub>**

- Statements → Statements Statement

**I<sub>169</sub>**

- Statement → Assignment SEMICOLON

**I<sub>170</sub>**

- Statement → Synapse SEMICOLON

**I<sub>171</sub>**

- Statement → Comment

**I<sub>172</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>173</sub>**

- Synapse → Variable Mapping

**I<sub>174</sub>**

- Variable → VariableName

**I<sub>175</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>176</sub>**

- VariableName → ID

**I<sub>177</sub>**

- Comment → COMMENT

**I<sub>178</sub>**

- Statements → Statements Statement

**I<sub>179</sub>**

- Statement → Assignment SEMICOLON

**I<sub>180</sub>**

- Statement → Synapse SEMICOLON

**I<sub>181</sub>**

- Statement → Comment

**I<sub>182</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>183</sub>**

- Synapse → Variable Mapping

**I<sub>184</sub>**

- Variable → VariableName

**I<sub>185</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>186</sub>**

- VariableName → ID

**I<sub>187</sub>**

- Comment → COMMENT

**I<sub>188</sub>**

- Statements → Statements Statement

**I<sub>189</sub>**

- Statement → Assignment SEMICOLON

**I<sub>190</sub>**

- Statement → Synapse SEMICOLON

**I<sub>191</sub>**

- Statement → Comment

**I<sub>192</sub>**

- Assignment → VariableName GETS Constructor

**I<sub>193</sub>**

- Synapse → Variable Mapping

**I<sub>194</sub>**

- Variable → VariableName

**I<sub>195</sub>**

- Variable → VariableName PERIOD MemberName

**I<sub>196</sub>**

- VariableName → ID

**I<sub>197</sub>**

- Comment → COMMENT

**I<sub>198</sub>**

- Statements → Statements Statement

**I<sub>199</sub>**

- Statement → Assignment SEMICOLON

**I<sub>200</sub>**

- Statement → Synapse SEMICOLON

**I<sub>201</sub>**

- Statement → Comment

# SLR(1) 構文解析表

SynapseScript の SLR(1) 構文解析表 (構文解析動作関数の部)

2012.11.28 (Wed)

状態	SEND	TO	GETS	SUCCESS	PERIOD	LEAF	RPAR	COMP	ID	STRINGS	COMMENT	ACCEPTANCE
0									8	9		5
1									8	9	(1-1)	
2									(2-1)	(2-1)	(2-1)	
3				18								
4				19								
5									(5-1)	(5-1)	(5-1)	
6	(6-2)	(6-2)	20	(6-2)	21							
7	25	29										
8	(8-1)	(8-1)	(8-1)	(8-1)	(8-1)				(9-1)	(9-1)	(9-1)	
9									(10-1)	(10-1)	(10-1)	
10									(18-1)	(18-1)	(18-1)	
18									(19-1)	(19-1)	(19-1)	
19									28			
20									30			
21												
22	25	29			(22-1)							
23	(23-1)	(23-1)			(23-1)							
24								8				
25								30				
26					(26-1)							
27						39						
28					(28-1)							
29	(29-1)	(29-1)			(29-1)							
30	(30-1)	(30-1)			(30-1)							
31	(31-1)	(31-1)			(31-1)							
34	(34-1)	(34-1)			(34-1)							
35	(35-1)	(35-1)			(35-1)	21						
37		41										
39						42			45			
41								8				
42					(42-1)							
43						48	49					
44						(44-1)	(44-1)					
45						(45-1)	(45-1)					
46	(46-1)	(46-1)			(46-1)							
48					(48-1)							
49									45			
50						(50-1)	(50-1)					

(○-○)  
↑  
↓  
送受信  
(80)

2012.11.28 (Wed) 完成  
SLR(1) 構文解析表 (SLR(1))  
構文解析動作関数の部

SynapseScript の SLR(1) 構文解析表 (先行関数の部)

2012.11.29 (Thu)

状態	Program	Structure	Access	Assign	Storage	Mapping	Mapping	Variable	Variable	Variable	Constant	Constant	Argument	Argument	Command
0	1	2	3	4				7	6						5
1		10	3	4				7	6						5
2															
3															
4															
5															
6															
7							22	23							
8															
9															
10															
18															
19															
20											26	27			
21											29				
22								31							
23															
24								34	35						
25											37				
26															
27															
28															
29															
30															
31															
34															
35															
37															
39													43	44	
41											46	35			
42															
43															
44															
45															
46															
48															
49															
50															

2012.11.29 (Thu) 完成



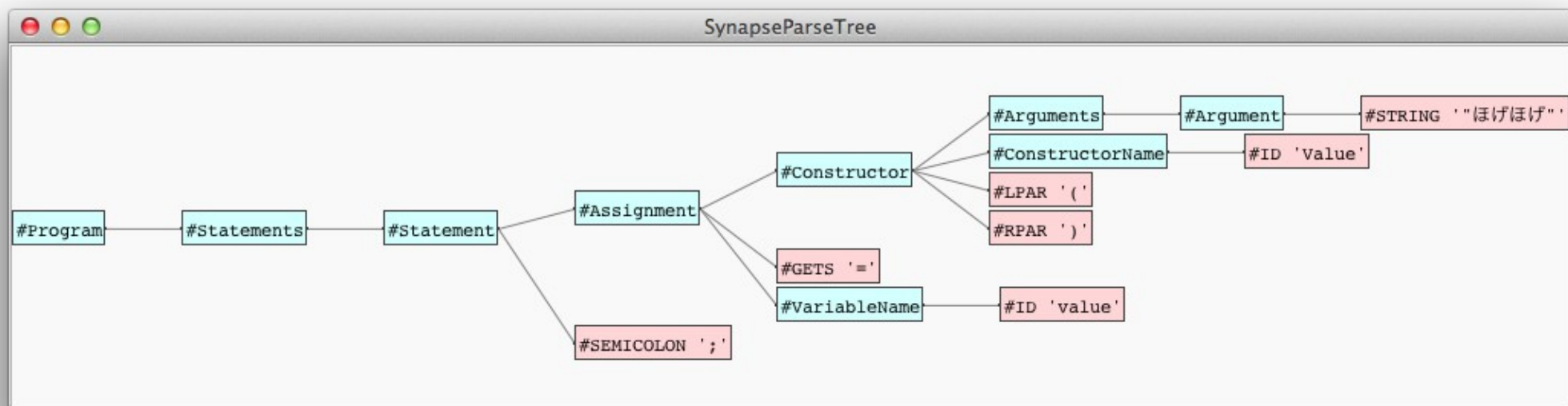
# 構文解析時のスタックの動き

```
transcript = Transcript("hoge");
```

```
#ID #GETS #ID #LPAR #STRING #RPAR #SEMICOLON
```

```
0  
12 還元 (0 ID 8  
    還元 (0 VariableName 6  
        還元 (0 VariableName 6 GETS 20  
            還元 (0 VariableName 6 GETS 20 ID 28  
11 還元 (0 VariableName 6 GETS 20 ConstructorName 27  
    還元 (0 VariableName 6 GETS 20 ConstructorName 27 LPAR 39  
    還元 (0 VariableName 6 GETS 20 ConstructorName 27 LPAR 39 STRING 45  
10 還元 (0 VariableName 6 GETS 20 ConstructorName 27 LPAR 39 Argument 44  
9 還元 (0 VariableName 6 GETS 20 ConstructorName 27 LPAR 39 Arguments 43  
    還元 (0 VariableName 6 GETS 20 ConstructorName 27 LPAR 39 Arguments 43 RPAR 48  
47 還元 (0 VariableName 6 GETS 20 Constructor 26  
33 還元 (0 Assignment 3  
    還元 (0 Assignment 3 SEMICOLON 18  
23 還元 (0 Statement 2  
13 還元 (0 Statements 1  
    還元 (0 Statements 1 ID 8  
10 還元 (0 Statements 1 VariableName 6  
9 還元 (0 Statements 1 Variable 7
```

# 構文解析 + 意味解析



# 言語処理のススメ

終

# 今回の成果物

- メトリクスの新しい仕組みの構築
- SynapseScript (言語設計と処理系実装)
- メトリクスプラグイン
- ShellInterpreter (Smalltalk → シェル)

# 今後の見通し

# 課題

- レシピ生成プラグイン
- 調整
- 実験
- 論文

# レシピ生成プラグイン

## Cプログラムの解析に苦戦 (typedef-nameの問題)

→これを解決しないとスパゲッティへの写像はできない。

既に用意されている基本型は、  
字句解析で型名として判別できる  
が、typedefによる新しい型名定義  
では、ユーザが自由に名前を付け  
るため、単純な字句解析では型名  
であることを判別できない。

```
typedef int hoge;  
  
int main ( ) {  
    int firstValue;  
    hoge secondValue;  
    foo thirdValue;  
    ...  
}
```

# レシピ生成プラグイン

typedef-nameの問題は昨日中に解決

→あとはプロプロセッサ向けの制御行をどうにかすれば解析できる。

解析できれば、指標値が求まる。求まればレシピを生成できる。

```
#include <stdio.h>
#include "defs.h"

#define SIZE 10

#ifdef HOGE
#endif
```

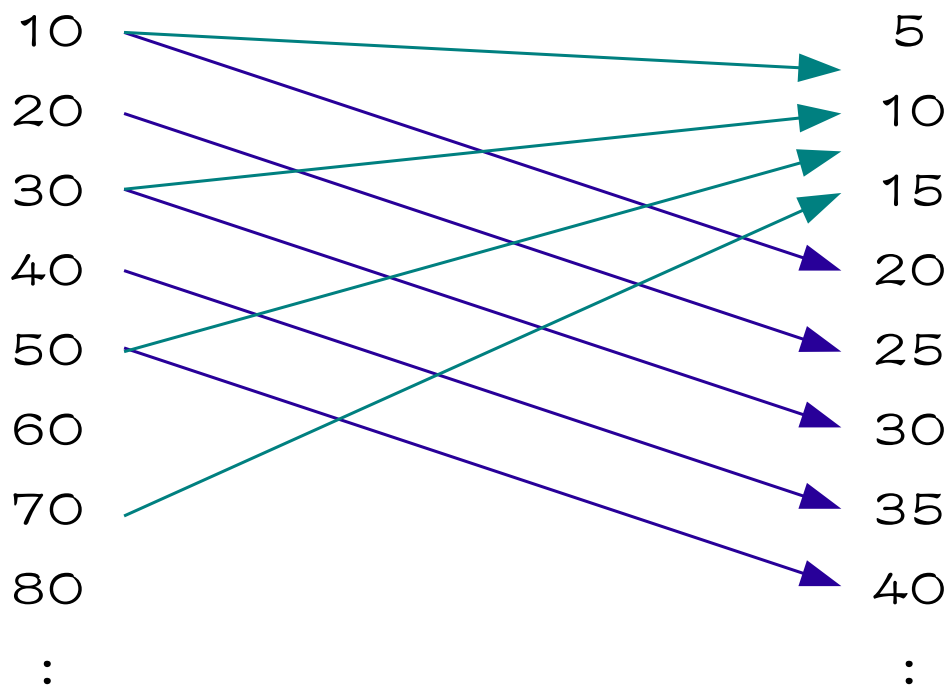


# 調整

指標値の写像関係を調整しなければならない。ちょっと泥臭い。

行数/メソッド数 [個]

作り置き時間 [分]



# 実験

被験者の方にスパゲッティを食べていただき... (予定)

詳しい内容は近日決定。

# 論文

- スタッフは作成済み  
→ ウェブサイトに公開中。
- 現時点で半分以上書けるはず  
→ 冬休みに。。
- 実験が終わり次第残りを執筆  
→ 早く実験せねば。。。

# さいごに

# 実験について

毎度「宜しくお願いします」と言い続けて、

されど実験を執り行うことはありませんでした。

しかし、来月こそは、来月こそは実験を行います。

パスタを食べていただくことになりましょう。

宜しくお願い申し上げます。