初学者の学習支援のための PythonとRuby間の翻訳

2011SE286 渡辺将匡
2014SE003 赤羽里帆
2014SE020 廣瀬隼大

目次

- 研究の背景
- 研究の目的
- 学習時間の削減方法
- 翻訳ソフトウェアを使うメリット
- 翻訳する言語
- アプローチ
- 関連研究
- 検証と評価
- まとめ

研究の背景

インターネット

第四次産業

人工知能

SNS

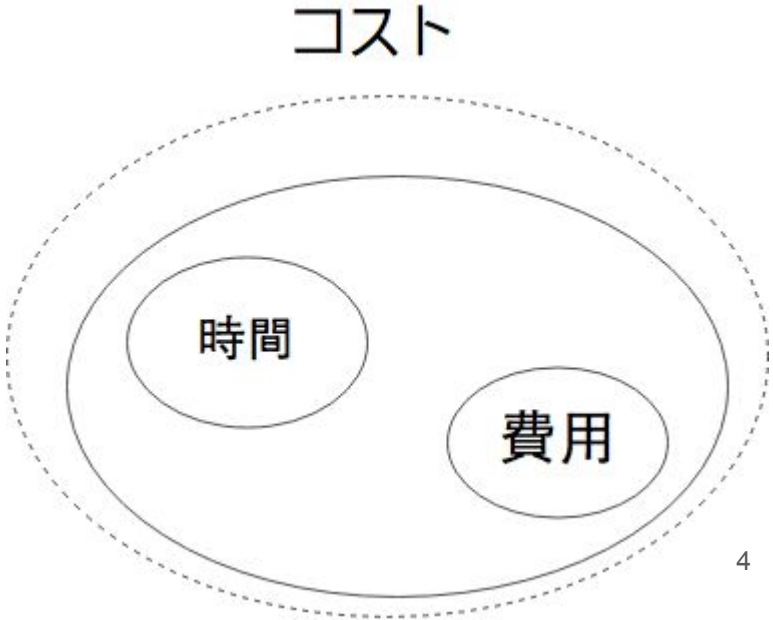
義務教育化政策

プログラミング言語

研究の背景

・言語を習得するにかかる大きなコスト

・学習時間の削減を行うことによりコスト削減



研究の目的

初学者の学習支援をする

翻訳ソフトウェアの作成

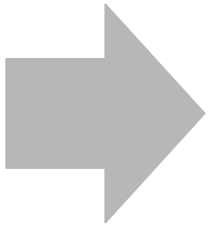
学習時間の削減方法

二つのソースコードを比べながら学習

⇨それぞれの類似点や異なる点を発見できる

⇨理解が深まる

⇨効果的な学習が期待できる



学習時間の削減

翻訳ソフトウェアを使うメリット

・比較できるソースコードが入手可能

⇨探す必要がない

・ソースコードを瞬時に入手可能

⇨時間が削減できる

翻訳する言語

Python

Ruby

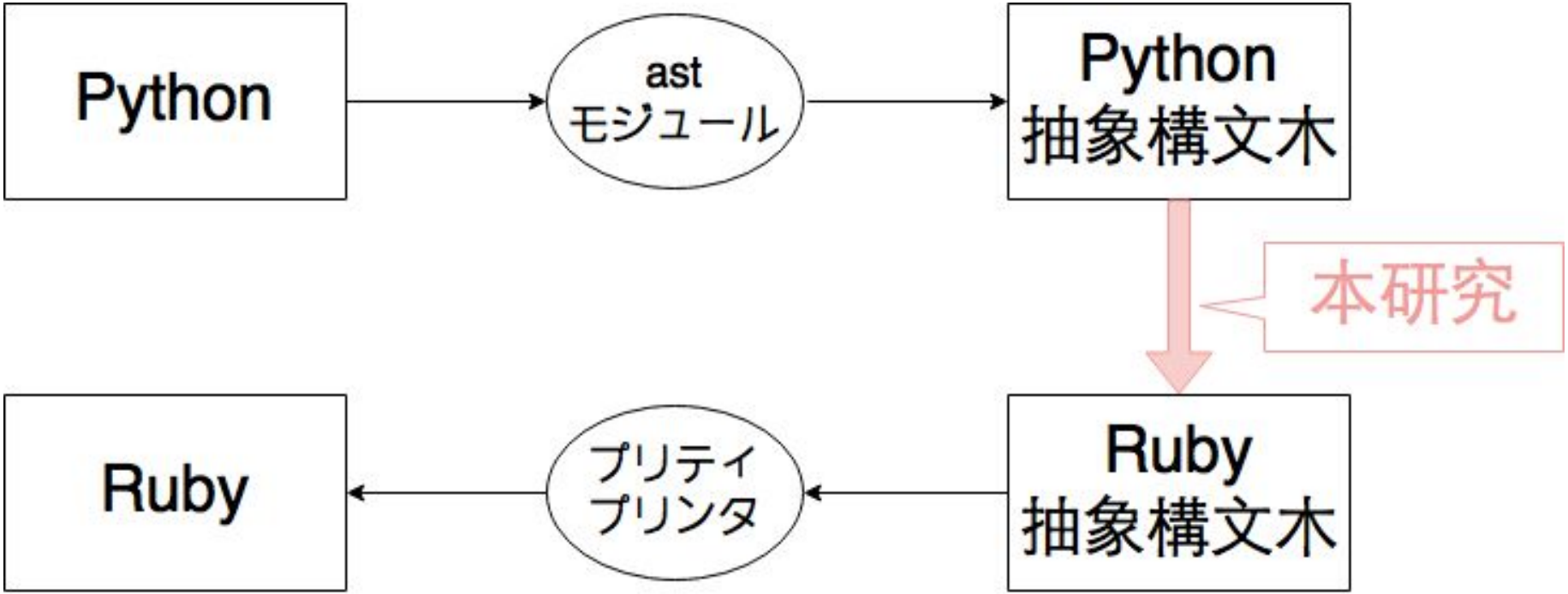
・Pythonは現在最も人気な言語 [1]

・軽量スクリプト言語の中で最もメジャーな二言語

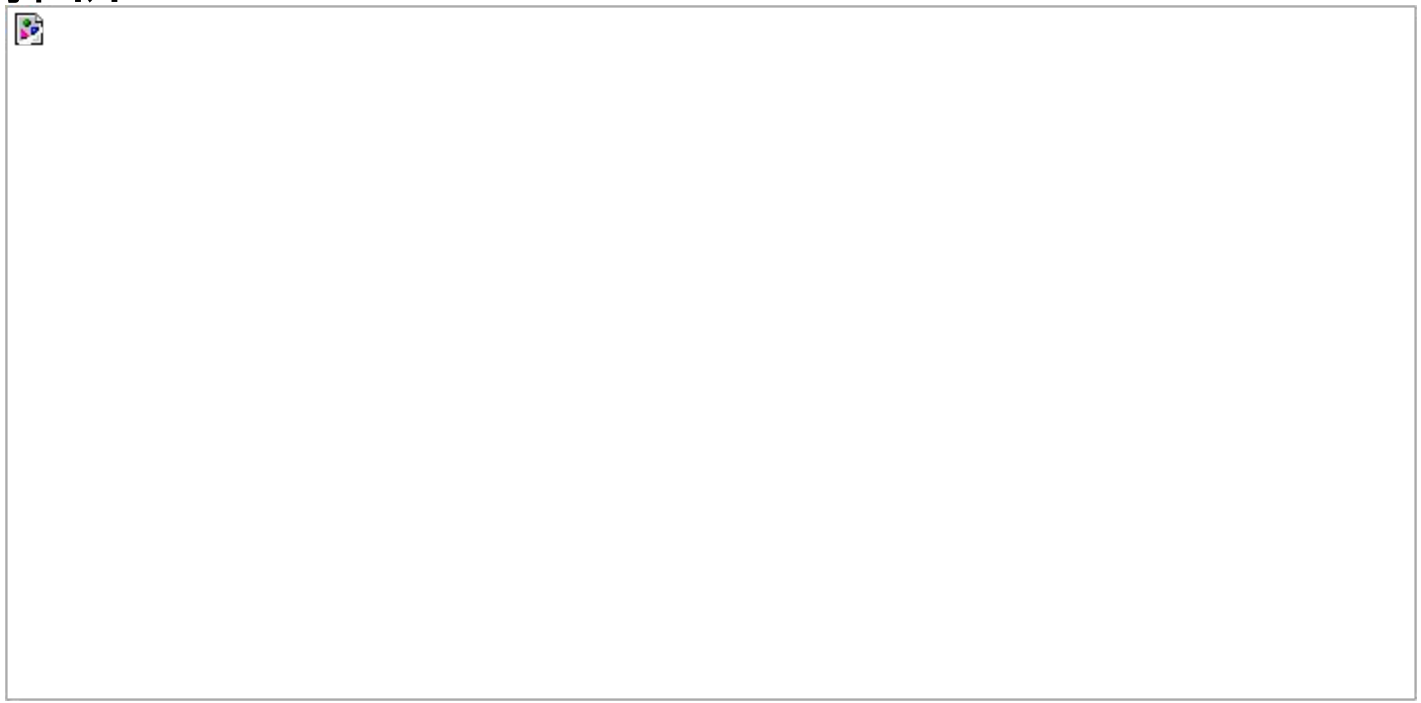
・ソースコードがweb上に多く存在

・PythonとRubyは類似点が多い [2]

アプローチ



字句解析

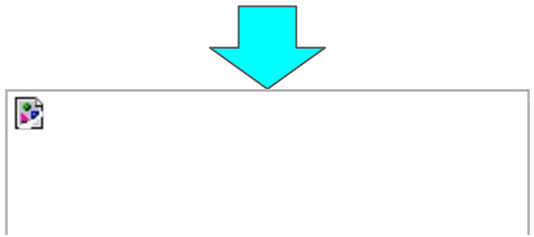


構文

規則的な字句の並び

例 if文 → if 条件式 : の順に書かれる

if　　条件式　　:



BNF記法

構文を定義するメタ言語

例

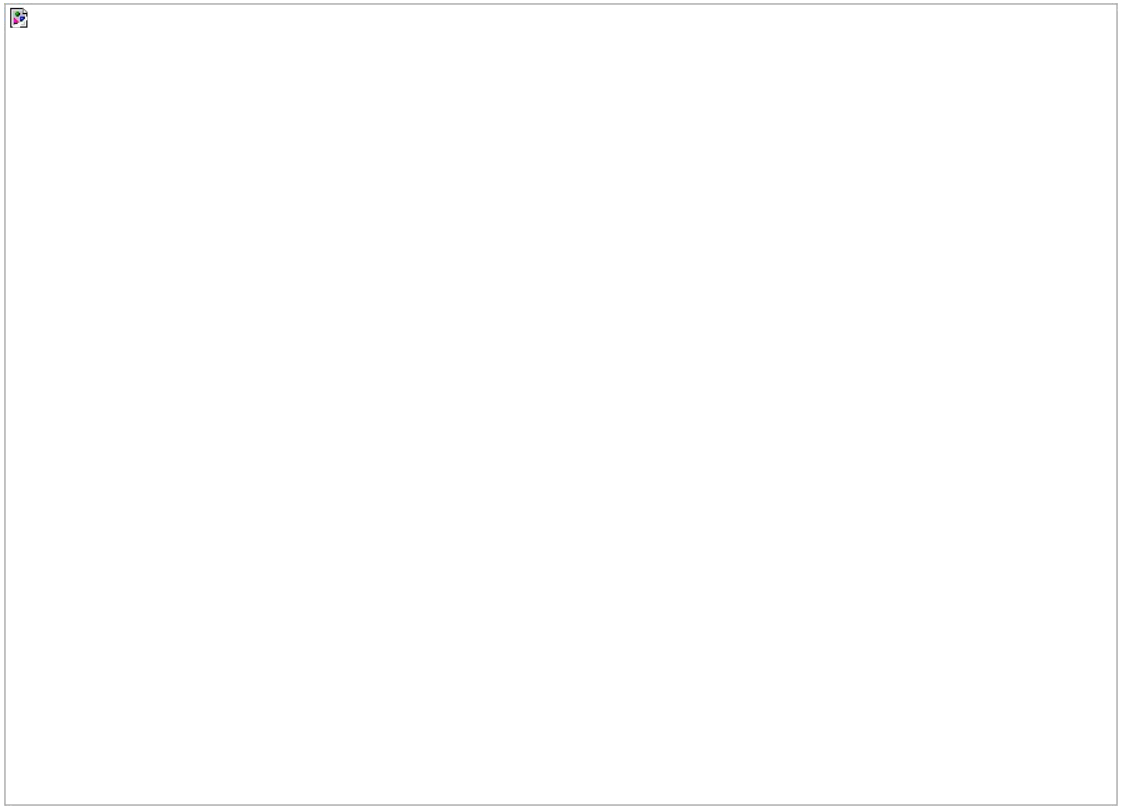
<NUMBER> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

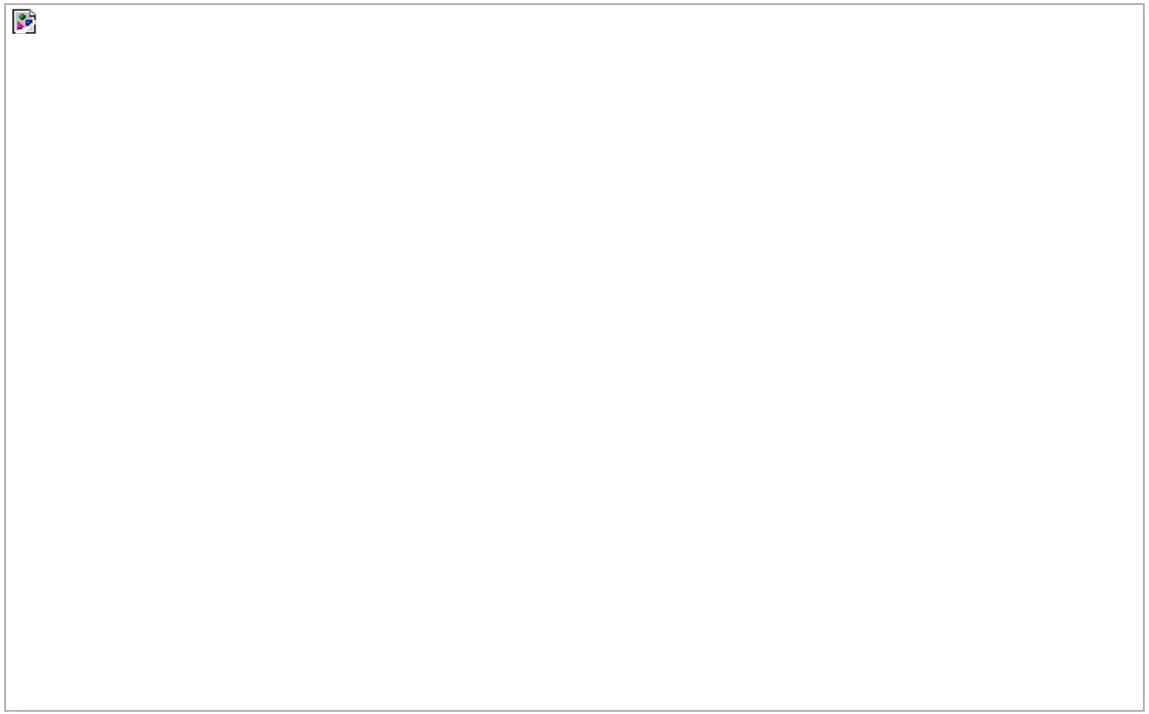
<EXCEPTZERO> ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

<BIGNUMBER>::=<NUMBER> |<NUMBER> <BIGNUMBER>

<TENMORE>::= <EXCEPTZERO> <BIGNUMBER>

構文解析





プログラムの実行上で意味の無いものを取り除いた解析木

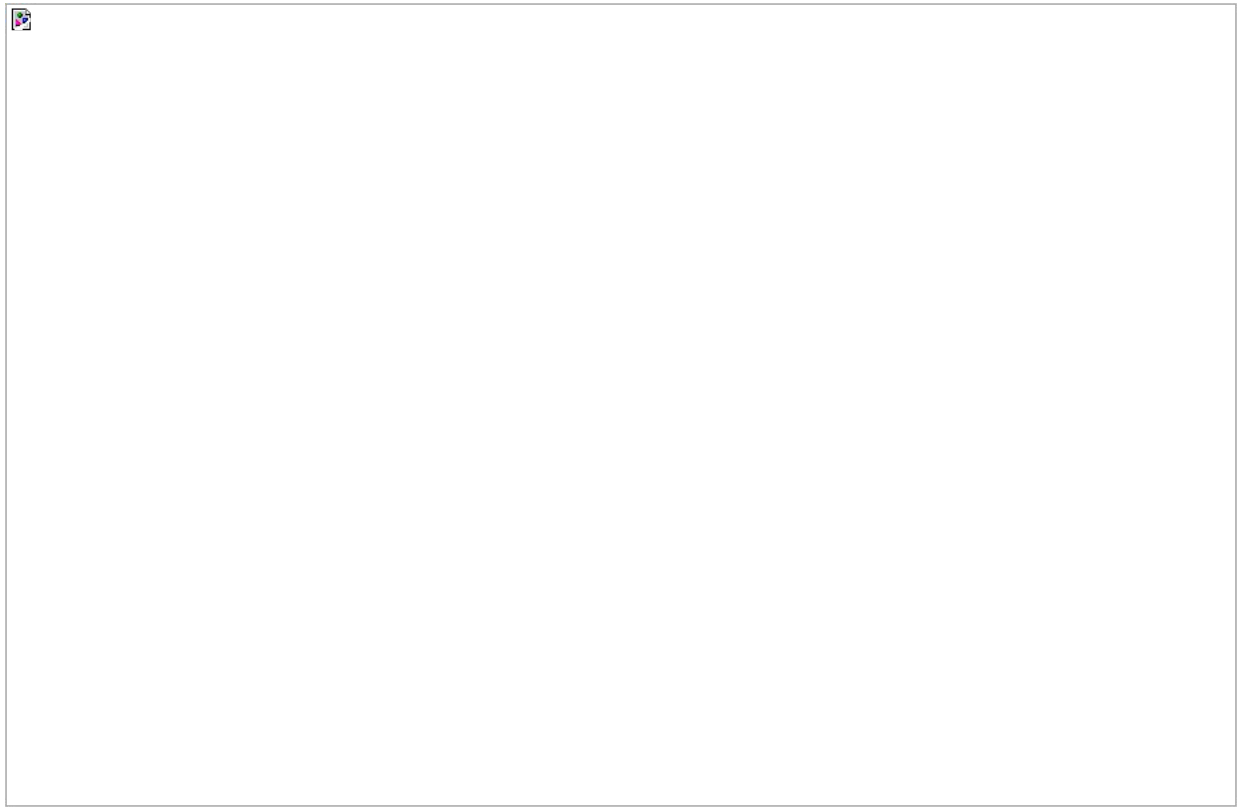
→抽象構文木

現在のソフトウェアの役割

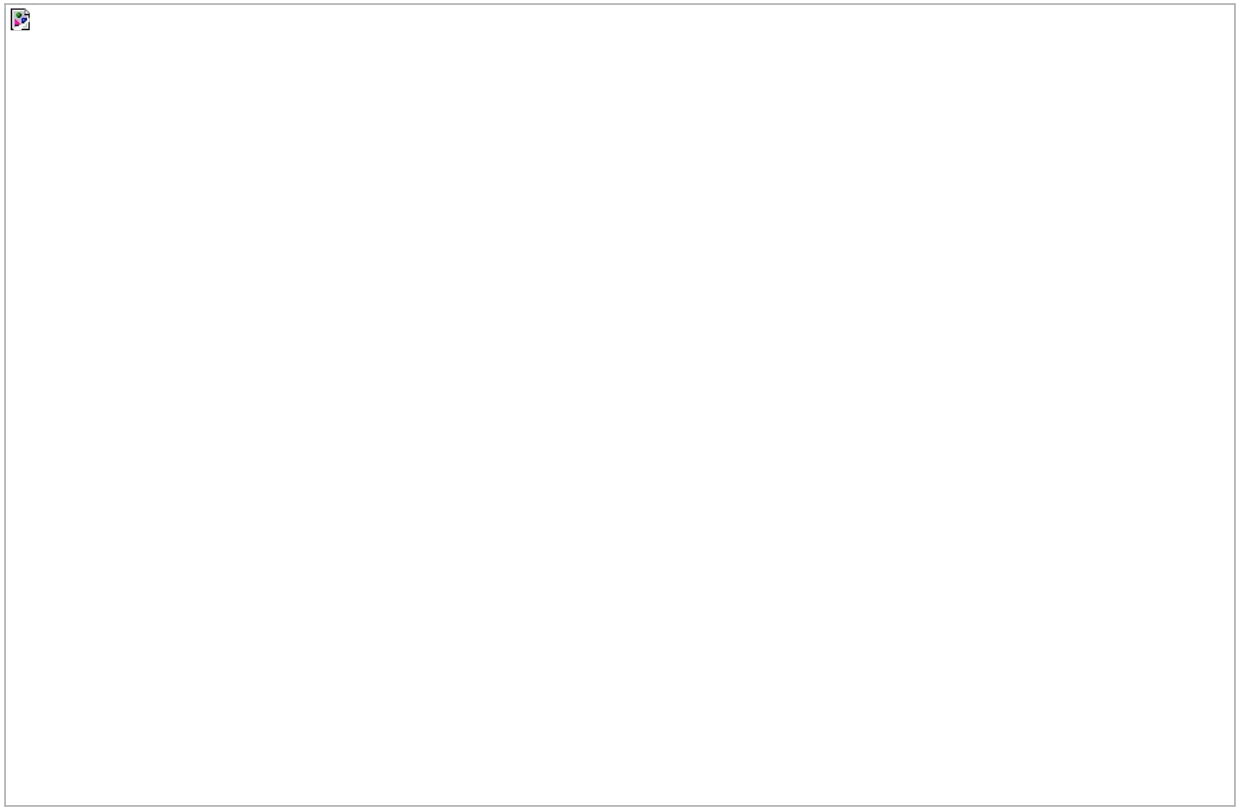
関連ソフトウェアとの比較

	対象言語	方法	使用言語
青山学院大学[2]	PythonをRubyに翻訳する.	外部のソフトウェアを利用して構文解析し, 得られた中間語を翻訳する.	不明
コンパイラ[3]	高級プログラミング言語を低級プログラミング言語に翻訳する.	字句解析を行った後,構文解析を行い, それによって得られた中間語から目的コードを生成し, 目的コードを出力する.	複数
mtsystems[4]	C言語をJavaに翻訳する.	他のソフトウェアで中間語を生成し, その中間語をJavaのソースコードに翻訳	不明
本稿の翻訳ソフトウェア	PythonをRubyに翻訳する.	標準ライブラリを使用し,それによって生成される抽象構文木を翻訳する.	Python

翻訳例

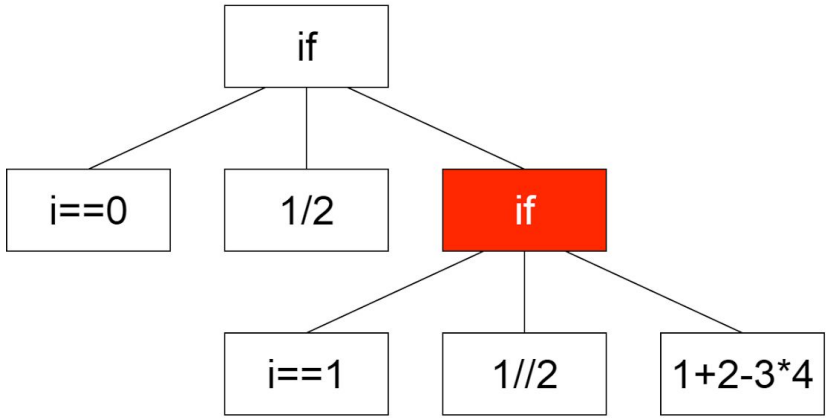


if構文の違い

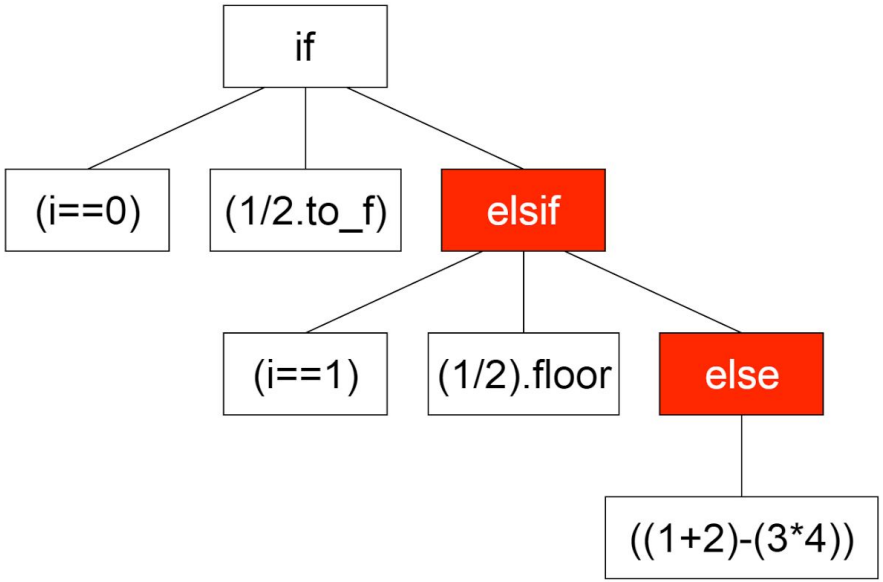


if解析木(簡易版)の違い

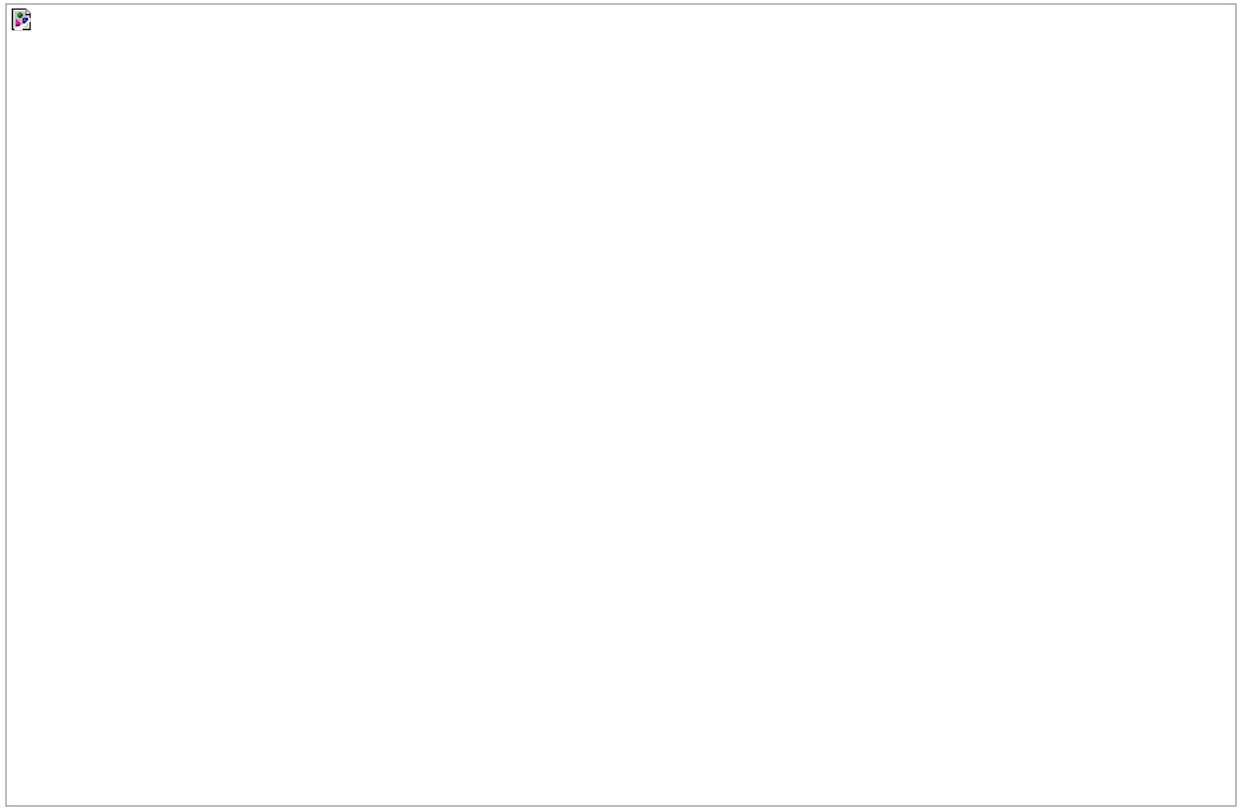
Python



Ruby

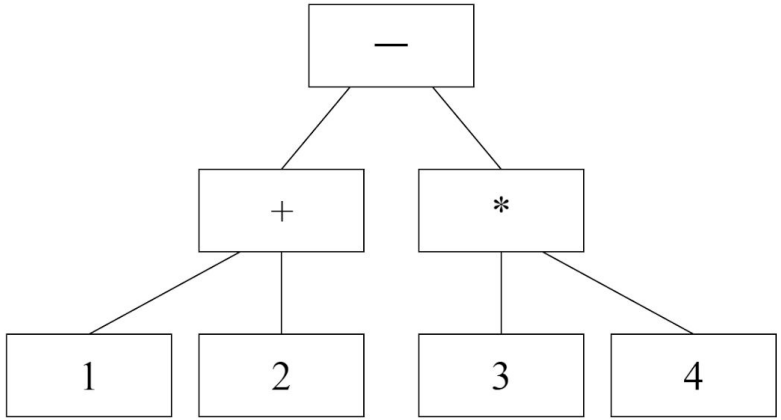


四則演算の違い

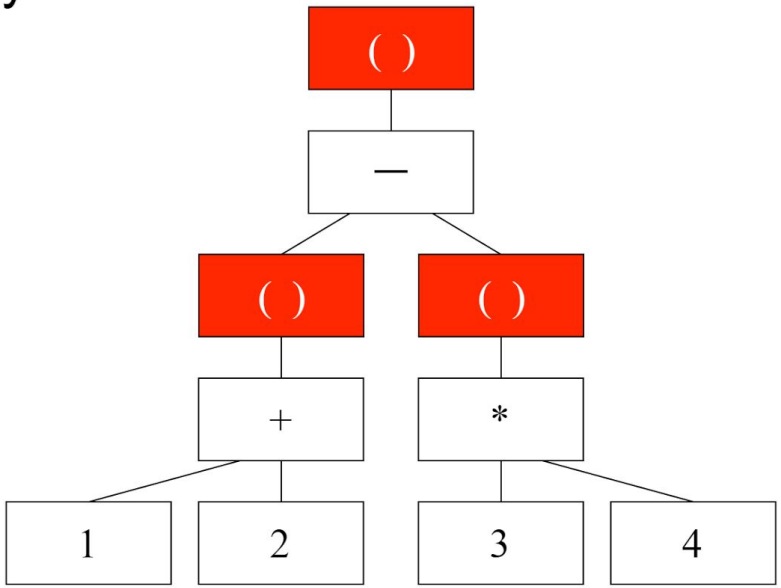


((1+2)-(3*4))の解析木(簡易版)の違い

Python



Ruby



再帰降下法による翻訳規則の制定(簡易版) if構文

$$\mathcal{T}\left[\!\left[\begin{array}{l}\texttt{if } c_1\texttt{:}\\ \quad s_1\\ \texttt{elif } c_2\texttt{:}\\ \quad s_2\\ \texttt{else:}\\ \quad s_3\end{array}\right]\!\right]^{\texttt{Python}} = \begin{array}{l}\texttt{if } \mathcal{T}[\![c_1]\!]^{\texttt{Python}} \texttt{ then}\\ \quad \mathcal{T}[\![s_1]\!]^{\texttt{Python}}\\ \texttt{elsif } \mathcal{T}[\![c_2]\!]^{\texttt{Python}} \texttt{ then}\\ \quad \mathcal{T}[\![s_2]\!]^{\texttt{Python}}\\ \texttt{else}\\ \quad \mathcal{T}[\![s_3]\!]^{\texttt{Python}}\\ \texttt{end}\end{array}$$

再帰降下法による翻訳規則の制定(簡易版) 演算

$$\mathcal{T}[\![(c)]\!]^{\texttt{Python}} = (\mathcal{T}[\![c]\!]^{\texttt{Python}})$$
$$\mathcal{T}[\![(e)]\!]^{\texttt{Python}} = (\mathcal{T}[\![e]\!]^{\texttt{Python}})$$

$$\mathcal{T}[\![\ c_1\ \odot\ c_2\ ]\!]^{\texttt{Python}} = \ (\mathcal{T}[\![c_1]\!]^{\texttt{Python}}\ \mathcal{T}_{\texttt{C\_OP}}[\![\odot]\!]^{\texttt{Python}}\ \mathcal{T}[\![c_2]\!]^{\texttt{Python}})$$

$$\mathcal{T}[\![\ e_1\ \odot\ e_2\ ]\!]^{\texttt{Python}} = \ (\mathcal{T}[\![e_1]\!]^{\texttt{Python}}\ \mathcal{T}_{\texttt{OP}}[\![\odot]\!]^{\texttt{Python}}\ \mathcal{T}[\![e_2]\!]^{\texttt{Python}})$$
$$\mathcal{T}[\![\ e_1\ /\ e_2\ ]\!]^{\texttt{Python}} = \ (\mathcal{T}[\![e_1]\!]^{\texttt{Python}}\ \mathcal{T}_{\texttt{OP}}[\![/]\!]^{\texttt{Python}}\ \mathcal{T}[\![e_2]\!]^{\texttt{Python}}\texttt{.to\_f})$$
$$\mathcal{T}[\![\ e_1\ //\ e_2\ ]\!]^{\texttt{Python}} = \ (\mathcal{T}[\![e_1]\!]^{\texttt{Python}}\ \mathcal{T}_{\texttt{OP}}[\![//]\!]^{\texttt{Python}}\ \mathcal{T}[\![e_2]\!]^{\texttt{Python}})\texttt{.floor}$$

再帰降下法による翻訳規則の制定(簡易版) 演算子

$$\mathcal{T}_{\texttt{OP}}[\![\ +\ ]\!]^{\texttt{Python}} = +$$
$$\mathcal{T}_{\texttt{OP}}[\![\ -\ ]\!]^{\texttt{Python}} = -$$
$$\mathcal{T}_{\texttt{OP}}[\![\ *\ ]\!]^{\texttt{Python}} = *$$
$$\mathcal{T}_{\texttt{OP}}[\![\ /\ ]\!]^{\texttt{Python}} = /$$
$$\mathcal{T}_{\texttt{OP}}[\![\ //\ ]\!]^{\texttt{Python}} = /$$
$$\mathcal{T}_{\texttt{OP}}[\![\ **\ ]\!]^{\texttt{Python}} = **$$
$$\mathcal{T}_{\texttt{OP}}[\![\ \%\ ]\!]^{\texttt{Python}} = \%$$

$$\mathcal{T}_{\texttt{C\_OP}}[\![<]\!]^{\texttt{Python}} = <$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![>]\!]^{\texttt{Python}} = >$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![==]\!]^{\texttt{Python}} = ==$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![<=]\!]^{\texttt{Python}} = <=$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![>=]\!]^{\texttt{Python}} = >=$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![<>]\!]^{\texttt{Python}} = !=$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![!=]\!]^{\texttt{Python}} = !=$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![\texttt{or}]\!]^{\texttt{Python}} = \texttt{or}$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![\texttt{and}]\!]^{\texttt{Python}} = \texttt{and}$$
$$\mathcal{T}_{\texttt{C\_OP}}[\![\texttt{not}]\!]^{\texttt{Python}} = \texttt{not}$$

まとめ

- コスト削減の為のソフトウェアの提案
- 製作するソフトウェアの理論の作成

  抽象構文木を翻訳するソフトウェア

- 翻訳規則の実装

  四則演算とif文

今後の課題

繰り返し構文や配列等の抽象構文木の翻訳を

実装することである.

参考文献

[1]"The 2017 Top Programming Languages".IEEE SPEC-TRUM,<https://spectrum.ieee.org/computing/soft-ware/the-2017-top-programming-languages>(accessed 2017-10-4)

[2]清水崇之,小川翔二郎,松原俊一,Duerst, M.: 情報学広場:情報処理学会電子図書館,情報処理学会(オンライン)<https://ipsj.ixsq.nii.ac.jp/ej/index.php?active_action=repository_view_main_item_detail&page_id=13&block_id=8&item_id=107791&item_no=1>(2011)

[3]中田育男:コンパイラ,pp.11-139,産業図書(1981)

参考文献

[4]mtSystems (Migration TechnologySystems)GmbH,mtSystems, <https://www.mtsystems.com>(accessed 2017-09-27)

[5]Python Software Foundation:Python 3.6.1 ドキュメント,Python, <https://docs.python.jp/3/index.html>(参照 2017-09-27)

[6]まつもとゆきひろ:Ruby2.4.0 リファレンスマニュアル,オブジェクト指向スクリプト言語 Ruby リファレンスマニュアル,<https://docs.rubylang.org/ja/latest/doc/index.html>(参照 2017-09-27)