SFGPL 入門

Eruhitsuji

2023-12-08

Contents

1 1. SFGPL について 6
1.1 はじめに . . . 6
1.2 SFGPL 作成の背景と目的 . . . 6
1.3 SFGPL の特徴 . . . 6
1.4 SFGPL の基本文法 . . . 6
1.4.1 SFGPL の文構造 . . . 7
1.5 SFGPL の発音 . . . 7
1.6 SFGPL の単語 . . . 8
1.6.1 SFGPL の品詞 . . . 9
1.6.2 SFGPL の機能語 . . . 10
1.6.3 SFGPL の借用語 . . . 10
1.7 SFGPL とプログラミング . . . 10

2 2. 文型 10
2.1 SFGPL の文型の一覧 . . . 10
2.2 Noun.do . . . 11
2.3 Noun.eq . . . 11
2.4 Noun.haveP . . . 11
2.5 Noun.doT . . . 12
2.6 Noun.give . . . 12
2.7 Noun.makeN と Noun.makeM . . . 12
2.8 Noun.have . . . 13
2.9 Noun.belong . . . 13
2.10 Noun.gt . . . 13
2.11 Noun.hearSay . . . 13
2.12 文構造を使用した名詞の修飾方法 . . . 13
2.12.1 強調形 . . . 14
2.13 単語集 . . . 14

3 3. 否定文 15
3.1 単語集 . . . 15

4 4. 疑問文 15
4.1 単語集 . . . 16

5 5. 命令文 16
5.1 単語集 . . . 16

6 6. 複文 16
6.1 並列節 17
6.2 従属節 17
6.2.1 一般的な従属節 17
6.2.2 副詞節 18
6.3 単語集 18

7 7. 詳細な文法 19
7.1 文章を修飾する方法 19
7.1.1 英語における前置詞的な用法 19
7.2 比較表現の文法 20
7.2.1 比較級 20
7.2.2 最上級 20
7.2.3 同級 21
7.3 単語集 21

8 8. 単語 21
8.1 借用語について 22
8.1.1 借用語と借用元の言語 22
8.2 固有単語について 23
8.2.1 固有単語のルール 23
8.3 限定詞について 23
8.3.1 名詞限定詞 24
8.3.2 動詞限定詞 24
8.4 無意味単語について 24
8.5 代名詞について 25
8.6 数値や論理的に使われる語 25

9 9. 動詞の活用 25
9.1 動詞の時制 25
9.1.1 動詞の拡張時制 26
9.2 相 27
9.2.1 一般的な進行形 28
9.3 完了形 29
9.4 受動態 29
9.5 その他の動詞の修飾 30
9.6 単語集 30

10 10. 修飾語 **30**
10.1 修飾語について 30
10.2 比較表現 30
10.3 各品詞に対する修飾語 30
10.4 修飾語の応用 31
10.5 単語集 31

11 11. 品詞変換 **31**
11.1 動詞から名詞 32
11.2 名詞から修飾語 32
11.3 動詞から修飾語 32
11.4 単語集 33

12 12. 接続詞 **33**
12.1 単語集 34

13 13. 代名詞 **35**
13.1 代名詞一覧 35
13.2 代名詞の応用 35

14 14. 名詞限定詞 **36**
14.1 単語集 37

15 15. 動詞限定詞 **37**
15.1 単語集 37

16 16. Bool 関連クラス **38**
16.1 Bool 型について 38
16.2 BoolList 型について 38
16.3 単語集 40

17 17. LangList **40**
17.1 単語集 41

18 18. LangFunc **41**

19 19. 数字の表現方法 **42**
19.1 Number クラス 42
19.2 NumberList クラス 42
19.3 単語集 44

20 20. 例文 **44**

21 21. バージョンについて **58**
21.1 バージョンの命名規則 58
21.2 バージョン更新内容について 58

1 1. SFGPL について

1.1 はじめに

SFGPL は “Simple Functional General Purpose Language” の略で，自然言語を形式化するための言語である．この言語は，文の構造や意味を容易に解釈でき，かつコミュニケーションができるようにするために考案した言語である．

また，この言語は私が趣味で作成したものであり，厳密に検証を行っていないため不備等がある可能性がある．

1.2 SFGPL 作成の背景と目的

多くの自然言語の文法では，多くの例外や存在し，学習者を悩ませることが多い．また，それを解決するために，世界共通語を目的とした人工言語が提案されたが，それらは多くの自然言語と同様に曖昧な意味や複数の解釈ができる場合が存在する．特に，接続詞や関係代名詞などを含む，長くて複雑な文章は解釈が困難であることが多い．それらを解決するために，形式的，論理的に理解できるような言語を目的として作成した人工言語が SFGPL である．

1.3 SFGPL の特徴

SFGPL では，関数型の言語で，また関数のとる引数の型が厳密に定義されている．SFGPL では，文構造それぞれに関数が割り振られているため，主語，述語，目的語，補語などの文法上の役割が分かりやすくなっている．また，文構造を組み合わせることによって複雑な文章を作成することができる．

1.4 SFGPL の基本文法

- SFGPL には機能語と少しの単語のみが存在し，厳密に定義された意味を持つ．その他の単語は他の言語から借用される．
- 機能語の後にはいくつかの引数が付き，その引数によって意味が決まる．
- 原則として，各引数は 1 つの単語または 1 つのオブジェクトに対応するが，原語が複数の単語である場合は，アンダースコアで接続することで 1 つの単語とみなすことができる．
- 借用語の前後にはシングルクォーテーションを付けることで区別する．
- 文法上の性や数などの区別をすることはなく，また，冠詞も存在しない．
- 文末にはセミコロン (;) を付ける．ただし，単文の場合は省略可能である．

1.4.1 SFGPL の文構造

SFGPL の語順は SVO であるが，文頭に文の構造を決定する機能語が付属する．また，SFGPL では固有語によって，文構造が厳密に定義されている．以下の表は，SFGPL で表現できる文構造の表である．また使用方法等の詳細は，文型に記述してある．

		単語	関数	引数	補足
1	S V	ta	Noun.do	S,V	
2	S V C	ma	Noun.eq	S,V,C	C が名詞
2	S V C	me	Noun.haveP	S,V,C	C が修飾語
3	S V O	te	Noun.doT	S,V,O	
4	S V O1 O2	ti	Noun.give	S,V,O1,O2	
5	S V O C	tu	Noun.makeN	S,V,O,C	C が名詞
5	S V O C	to	Noun.makeM	S,V,O,C	C が修飾語
-	A has B	mi	Noun.have	A,V,B	A が B を所有している
-	A belongs to B	mu	Noun.belong	A,V,B	A が B に所属している
-	A is more B than C	mo	Noun.gt	A,V,B,C	A が C より B である
-	According to C, A V B	moa	Noun.hearSay	A,V,B,C	B という内容を C という情報源から，A は F する

1.5 SFGPL の発音

SFGPL の固有単語においては，発音の例外が存在しない．また，以下の表の国際音声記号 (IPA) は発音例である．

SFGPL の子音は次の表のようなものがある．

表記	IPA
p	/p/
b	/b/
f	/f/
m	/m/
t	/t/

表記	IPA
d	/d/
s	/s/
n	/n/
l	/l/
k	/k/
g	/g/
w	/w/

一方，SFGPL の母音は次の表のようなものがある．SFGPL の固有語は，数少ない単語を除いて二重母音は存在しない．

表記	IPA
a	/a/
e	/e/
i	/i/
u	/u/
o	/o/

また，借用語は借用語固有の発音で読む．

1.6 SFGPL の単語

SFGPL の単語は主に，SFGPL の固有の単語と借用語に分かれる．

固有単語は，主に文構造に必要な機能語と，動詞と修飾語の基礎単語が存在する．またそれ以外は，借用語が使用される．

そして，SFGPL の文構造では，品詞の場所が決定されており，それに従った品詞の単語を使わなければならない．

1.6.1 SFGPL の品詞

SFGPL の品詞は名詞 (Noun)，動詞 (Verb)，修飾詞 (Modifier) の三種類がある．また，名詞のサブクラスとして句 (Phrase)，代名詞（Pronoun），Bool 配列型（BoolList），LangList，LangFunc と NumberList が存在する．

BoolList，LangList，LangFunc は一般的な文以外に論理的な文を作る際に使用される．そして，真偽を表す Bool 型が存在する．

NumberList は主に数詞として使われる．また，基数詞としての Number クラスが存在する．この Number クラスは通常単体で使われない．

さらに，名詞や動詞を修飾する特殊な語として，名詞限定語（DeterminerN）と動詞限定語（DeterminerV）が存在する．

それぞれの品詞にはそれぞれ特有の関数（機能語）が存在し，それによって品詞の変更や意味の決定などが行われる．その他に，基礎単語を実装する，単語（Word）が存在する．単語は，品詞別に動詞の単語の “WordV”，修飾語の単語の “WordM” が存在する．

名詞は，あらゆる物体，物質，人物，場所などのあらゆる概念を表す語である．動詞は，あらゆる動作，作用，状態，存在などを表す語である．修飾語は，他の語を修飾する語である．SFGPL では形容詞と副詞の区別はつけない．

Python ライブラリ SFGPL では品詞ごとにクラスが存在する．

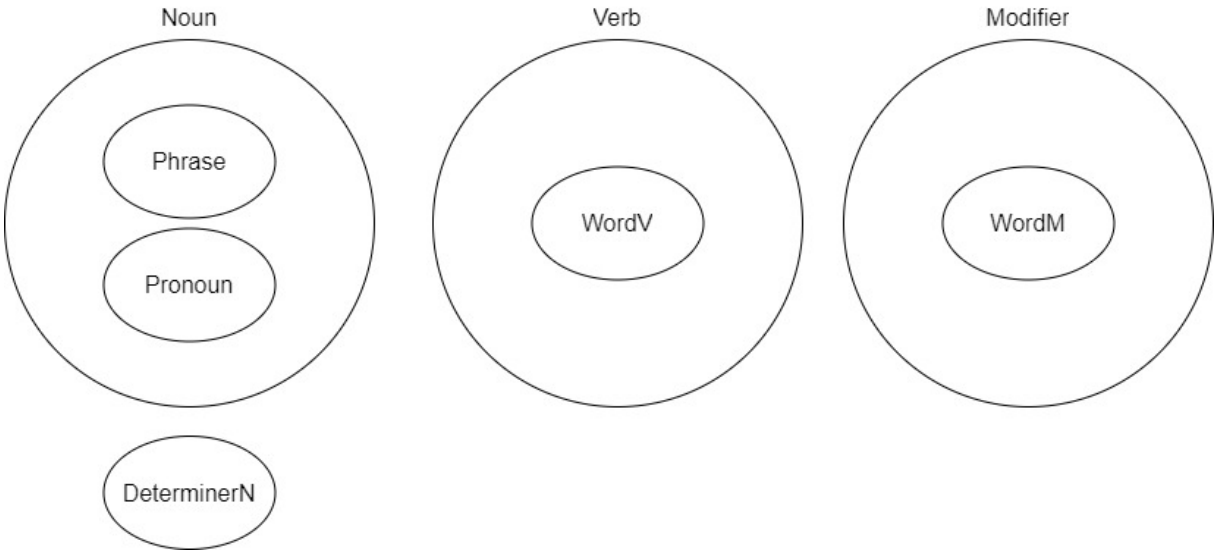

Figure 1: PartOfSpeach

1.6.2 SFGPL の機能語

機能語により，文の役割や品詞等の決定がされる．機能語の機能，役割や意味は引数内でのみしか適応されない．

この機能語らは，Python の関数と一対一となっている．また，引数の数も決まっていて，引数の場所によって，役割が決まっている．

機能語の一覧と利用方法は，dict.csv に記述されている．

1.6.3 SFGPL の借用語

SFGPL で存在しない単語は借用語を使用する．借用語は，英語などの世界でよく使われる言語から借用することが好ましいが，相手に伝わる単語であれば問題とはならないはずである．ただし，借用語は原型で使い，活用がある場合は SFGPL の機能語で行うことを推奨している．

1.7 SFGPL とプログラミング

SFGPL の文は，Python オブジェクトに書き換えることができる．このプロジェクトは，SFGPL が定義されているファイルが含まれている．Python で SFGPL を使用するためには，SFGPL.py をインポートすることで使用することができる．使用例は samples の Python ファイルに記載されている．また，Python での SFGPL ライブラリの詳細な実行方法は，How_to_Use_SFGPL_in_Python.ipynb に記載されている．

2 2. 文型

2.1 SFGPL の文型の一覧

SFGPL では，文を構成するためには，必ず文型を決定する機能語が文の先頭に付属する．SFGPL では以下の表のような文型が存在し，それらの文の組み合わせにより，文自体が構成される．また，単語の修飾なども行われる．

		単語	関数	引数	補足
1	S V	ta	Noun.do	S,V	
2	S V C	ma	Noun.eq	S,V,C	C が名詞
2	S V C	me	Noun.haveP	S,V,C	C が修飾語

		単語	関数	引数	補足
3	S V O	te	Noun.doT	S,V,O	
4	S V O1 O2	ti	Noun.give	S,V,O1,O2	
5	S V O C	tu	Noun.makeN	S,V,O,C	C が名詞
5	S V O C	to	Noun.makeM	S,V,O,C	C が修飾語
-	A has B	mi	Noun.have	A,V,B	A が B を所有している
-	A belongs to B	mu	Noun.belong	A,V,B	A が B に所属している
-	A is more B than C	mo	Noun.gt	A,V,B,C	A が C より B である
-	According to C, A V B	moa	Noun.hearSay	A,V,B,C	B という内容を C という情報源から，A は F する

2.2 Noun.do

Noun.do taでは，特に，英語の第一文型と同じ形の S が主語，V が動詞で，主語が何かの動作をするという．単純な文章を表すことができる．“I run.” を SFGPL で表すには，次のようになる．

```
ta ga sa 'run'
```

2.3 Noun.eq

Noun.eq maでは，特に，英語の第二文型である “S is C” に相当し，その中でも，補語 C が名詞であるものを表す表すことができる．また，この構文では S と C が等価であることも示している．V が英語で be 動詞に相当する場合，動詞として soを使用する．“This is a table.” を SFGPL で表すには，次のようになる．

```
ma gu so fa 'table'
```

“You become a teacher.” を SFGPL で表すには，次のようになる．

```
ma ge sa 'become' fa 'teacher'
```

2.4 Noun.haveP

Noun.haveP meでは，特に，英語の第二文型である “S is C” に相当し，その中でも，補語 C が修飾語であるものを表すことができる．また，この構文では S が C という性質や状態であるというこ

とを表す．V が英語で be 動詞に相当する場合，動詞として soを使用する．“The table is red.” を SFGPL で表すには，次のようになる．

```
me fa 'table' so la 'red'
```

“You look sad.” を SFGPL で表すには，次のようになる．

```
me ge sa 'look' la 'sad'
```

2.5 Noun.doT

Noun.doT teでは，特に，英語の第三文型に相当し，S が主語，V が動詞，O が目的語である．“I study English.” を SFGPL で表すには，次のようになる．

```
te ga sa 'study' fa 'English'
```

2.6 Noun.give

Noun.give tiでは，特に，英語の第四文型に相当し，S が主語，V が動詞，O1 と O2 が目的語である．特に，この構文では，S が O1 に O2 を V するという意味となる．V が英語で “give” に相当する場合，動詞として soを使用する．“I give you a table.” を SFGPL で表すには，次のようになる．

```
ti ga so ge fa 'table'
```

2.7 Noun.makeN と Noun.makeM

Noun.makeN tuと Noun.makeM toでは，特に，英語の第五文型に相当し，S が主語，V が動詞，O が目的語，C が補語である．Noun.makeN は C が名詞，Noun.makeM は C が修飾語のときに使用する．この構文では “S が O を C にさせる” という意味になる．V が英語で “make” に相当する場合，動詞として soを使用する．

“I make you a teacher.” を SFGPL で表すには，次のようになる．

```
tu ga so ge fa 'teacher'
```

“I make you sad.” を SFGPL で表すには，次のようになる．

```
to ga so ge la 'sad'
```

2.8 Noun.have

Noun.have `mi`は “A が B を所有している” という意味になる．V が英語で “have” に相当する場合，動詞として `so`を使用する．“I have a table.” を SFGPL で表すには，次のようになる．

```
mi ga so fa 'table'
```

2.9 Noun.belong

Noun.belong `mu`は “A が B に所属している” という意味になる．V が英語で “belong to” に相当する場合，動詞として `so`を使用する．“I belong to a school.” を SFGPL で表すには，次のようになる．

```
mu ga so fa 'school'
```

2.10 Noun.gt

Noun.gt `mo`は “A は C より B である” という意味になる．このとき A と B が比較対象の名詞，C は修飾語である．V が英語で be 動詞に相当する場合，動詞として `so`を使用する．“The bed is bigger than yours.” を SFGPL で表すには，次のようになる．

```
mo fa 'bed' so wan sen ge
```

2.11 Noun.hearSay

Noun.hearSay `moa`は “B という内容を C という情報源から，A は V する” という意味になる．このとき，A は情報を受け取った人や物，V は動詞，B は情報の内容，C は情報源の人や物である．V が英語で hear，say や see などの伝聞に関する動詞に相当する場合，動詞として `so`を使用する．“According to the book, I saw that Japan is located in East Asia.” を SFGPL で表すには，次のようになる．

```
di moa ga so ta fa 'Japan' na ne sa 'locate' li fun pun me fa 'Asia' so
    la 'east' fa 'book'
```

2.12 文構造を使用した名詞の修飾方法

SFGPL では名詞の修飾を行う際に，これらの文構造を使用する．また，文が生成されたとき，その全体は名詞となり，それを別の文に埋め込むことができる．

“Your table is red.” を SFGPL で表すには，次のようになる．

```
me mi ge so fa 'table' so la 'red'
```

このように “You have table” である `mi ge so fa 'table'`が主語となり，そのテーブルが赤い `la 'red'`ということが説明できる．また，同等の意味である，“You have red table.” は次のように表すことができる．

```
mi ge so me fa 'table' so la 'red'
```

2.12.1 強調形

特に文中で主語以外の単語を強調したい場合には，強調形 `san`を使用する事もできる．“Your table is red.” の table を強調形にするためには次のようにする．

```
me mi ge so san fa 'table' so la 'red'
```

2.13 単語集

English	SFGPL
I	ga
run	sa ‘run’
this	gu
table	fa ‘table’
red	la ‘red’
you	ge
become	sa ‘become’
teacher	fa ‘teacher’
look	sa ‘look’
sad	la ‘sad’
study	sa ‘study’
English	fa ‘English’
school	fa ‘school’
bed	fa ‘bed’

English	SFGPL
big	wan
yours	sen ge
book	fa ‘book’
Japan	fa ‘Japan’
in East Asia	li fun pun me fa ‘Asia’ so la ‘east’

3 3. 否定文

否定文を作成するためには `pa`を使用する．この語は，文章に付属することで否定文を作る．“I have a table.” は SFGPL では `mi ga so fa 'table'`である．それを否定文の “I don’t have a table.” という意味にする場合，SFGPL では次のように表現できる．

```
1 pa mi ga so fa 'table'
```

3.1 単語集

English	SFGPL
I	ga
table	fa ‘table’

4 4. 疑問文

疑問文を作成するためには `da`を使用する．この単語を文につけると疑問文になる．“You have a table.” は SFGPL では `mi ge so fa 'table'`である．それを疑問文の “Do you have a table?” という意味にする場合，SFGPL では次のように表現できる．

```
1 da mi ge so fa 'table'
```

また，疑問詞を含む疑問文の場合，不定のところを疑問詞に置き換えることで表す．

“Who has a table?” は次のように表す．

```
1 da mi ben wa so fa 'table'
```

“What do you have?” は次のように表す.

```
da mi ge so pen wa
```

4.1 単語集

English	SFGPL
you	ge
table	fa ‘table’
who	ben wa
what	pen wa

5 5. 命令文

命令文を作成するためには `de`を使用する．この単語を文につけると命令文になる．“You buy a table.” は SFGPL では `te ge sa 'buy'fa 'table'`である．それを命令文の “Buy a table, you!” という意味にする場合，SFGPL では次のように表現できる.

```
de te ge sa 'buy' fa 'table'
```

5.1 単語集

English	SFGPL
you	ge
buy	sa ‘buy’
table	fa ‘table’

6 6. 複文

SFGPL では 1 つの文の中に，複数のを組み合わせて表す文を作成することができる.

6.1 並列節

2 つ以上の文を並列に接続するためには，接続詞が使用される．

SFGPL で，“I went to Tokyo and I was shopping there.” を表すには次のようにする．

```
ba di ta ga na sa 'go' li pun fa 'Tokyo' di ta ga na ni sa 'shop' li pun gu
```

また，英語のような時制の一致をするにはこのように節ごとに時制を活用させるが，SFGPL では文全体に基本時制を活用させることができる．

```
di ba ta ga na sa 'go' li pun fa 'Tokyo' ta ga na ni sa 'shop' li pun gu
```

6.2 従属節

主節内の名詞に対して従属的に修飾するためには，その名詞の代わりにその名詞を説明する文を入れることで実現できる．また，SFGPL では一般的に，名詞を修飾する場合には従属節を使用することが多い．

6.2.1 一般的な従属節

SFGPL で，“My bag is big.” を表すには次のようにする．またこのときの “My bag” は，“I have a bag” であるというように表現する．そしてこのとき，“bag” が修飾されている名詞であるため，その名詞には san を付ける．

```
me mi ga so san fa 'bag' so wan
```

また，意味がほぼ同じである，“I have a bag is big.” を表すには次のようにする．またこのときは，“a bag is big” の “bag” は従属節の主語となっているため，san を付けなくても良い．

```
mi ga so me fa 'bag' so wan
```

そして，“I give you the desk I built.” を表すには次のようにする．

```
ti ga so ge di te ga sa 'build' san fa 'desk'
```

このように従属節だけの時制を変えることもできる．

6.2.2 副詞節

副詞節で述語や文全体に対して修飾することができる．SFGPL で “I ate sushi, when I went to Tokyo.” を表すには次のようにする．

```
di te ga na sa 'eat' li ta ga na sa 'go' li pun fa 'Tokyo' fa 'sushi'
```

また，SFGPL で “I went grocery shopping while my kids were sleeping.” を表すためには次のようにする．

```
di ta ga na sa 'go' ba li ma fi ni sa 'shop' so fa 'grocery' li ta mi
    ga so san don fa 'kid' ni sa 'sleep'
```

6.3 単語集

English	SFGPL
I	ga
go	sa ‘go’
to Tokyo	li pun fa ‘Tokyo’
shop (Verb)	sa ‘shop’
there	pun gu
bag	fa ‘bag’
big	wan
you	ge
build	sa ‘build’
desk	fa ‘desk’
eat	sa ‘eat’
sushi	fa ‘sushi’
grocery	fa ‘grocery’
kid	fa ‘kid’
sleep	sa ‘sleep’

7 7. 詳細な文法

SFGPL は基本的に，文型に記されているような文法は厳守する必要があるが，その他はユーザ側である程度決めてよい．しかし，模範的な文法を本章で記述しておく．

7.1 文章を修飾する方法

文章全体に対して修飾するためには，基本的にその文内の動詞を naを使用することで修飾する．例えば，“I go to Tokyo.” という例文では，“to Tokyo” の部分が修飾語となる．その際 SFGPL では次のように表現する．

```
1 ta ga na sa 'go' li pun fa 'Tokyo'
```

また，別の方法として，meを使う方法もある．

```
1 me ta ga sa 'go' so li pun fa 'Tokyo'
```

7.1.1 英語における前置詞的な用法

特に，英語における前置詞のように動詞を修飾する場合，liと DeterminerN を使用して表現する．英語の前置詞と SFGPL の一例を次の表に示す．

English	Meaning	SFGPL
at/in/on/to/from	Time	li pin
at/in/on/to/from	Place	li pun
for	Reason	li pon
for	Way/Means	li ban
from	Start	li fan
to	End	li fen
between/among	Section	li fin
in	In	li fun
into	Into	li tun fun
out	Out	li fon
up/over	Move&Above	li tun man
above	Above	li man

English	Meaning	SFGPL
down	Move&Below	li tun men
under	On&Below	li min men
below	Below	li men
on	On	li min
right	Right	li mun
left	Left	li mon
near	Near	li tin
by/about	By/About	li tan tin
with	With	li ten tin

7.2 比較表現の文法

SFGPL では，英語における比較級を使った比較表現は，moによって定義されているが，最上級や同級による比較は定義されていない．このような文は次のように表すことを推奨する．

7.2.1 比較級

“A is B(-er) than C” のような比較表現は，moによって表現する．“My bag is bigger than yours.” は，次のように表現する．

```
mo mi ga so san fa 'big' so wan sen ge
```

7.2.2 最上級

“A is the B(-est) in/of C” のような比較表現は，次のような構文で表現する．

```
me A V ka B li fun C
```

“My bag is the biggest in my class.” は，次のように表現する．

```
me mi ga so san fa 'big' so ka wan li fun mu ga so san fa 'class'
```

7.2.3 同級

“A is as B as C” のような比較表現は，次のような構文で表現する．このとき，“似ている” という意味の wenを使って表現する．

```
me ba A C V ka B wen
```

“My bag is as big as his.” は，次のように表現する．

```
me ba mi ga so san fa 'big' sen lan gi so ka wan wen
```

7.3 単語集

English	SFGPL
I	ga
go	sa ‘go’
to Tokyo	li pun fa ‘Tokyo’
bag	fa ‘bag’
big	wan
yours(possessive)	sen ge
my class	mu ga so san fa ‘class’
his(possessive)	sen lan gi

8 8. 単語

SFGPL の単語は，基本的に使い方が定まっている．例えば，借用語を使用する方法などが決まっている．本章ではこれら単語の種類と使用方法について記載する．また，単語の詳細は dict.csv に記述されている．

また，SFGPL の単語では，基本的に，冠詞，数，性，格などによる変形は行われない．数や性を示したいときには，名詞限定詞を使用する．

8.1 借用語について

SFGPL は基礎単語以外は借用語にて代用する．名詞，動詞，修飾語にて，借用語を使用するためには，次の表のように表す．

元の語	品詞	SFGPL
apple	名詞	fa ‘apple’
open	動詞	sa ‘open’
tall	修飾語	la ‘tall’

これらの単語を使った例を次に表す．

English	SFGPL
I have an apple.	mi ga so fa ‘apple’
I open a door.	te ga sa ‘open’ fa ‘door’
I am tall.	me ga so la ‘tall’

8.1.1 借用語と借用元の言語

借用語はあらゆる言語より借用することが可能である．ただし，話者間双方で理解できる単語を選ぶことが望ましい．

例えば，あらゆる言語の「言語」という単語を SFGPL に借用するには次の表のようにする．

Language	Raw Word	SFGPL
English	language	fa ‘language’
Japanese	言語	fa ‘言語’
Spanish	idioma	fa ‘idioma’
French	langue	fa ‘langue’
Russian	я з ы к	fa ‘я з ы к’
Portuguese	linguagem	fa ‘linguagem’
Esperanto	lingvo	fa ‘lingvo’

このように，様々な言語から借用することができる．

8.2 固有単語について

SFGPL では，動詞と修飾語については，いくつかの固有単語が用意されている．WordV，WordM クラスでは，SFGPL に固有に存在する単語群である．

これらの単語群は，品詞が既に決定していおり，また広い意味を持っているため汎用性は高いが，意味の詳細の特定はしにくいものである．

次の表は，固有単語の例である．

English	SFGPL
create	kan
big	wan

これらの単語を使った例を次に表す．

English	SFGPL
I create a door.	te ga kan fa ‘door’
The apple is big.	me fa ‘apple’ so wan

8.2.1 固有単語のルール

SFGPL の固有単語に関しては一意性があり，異なる意味の単語は異なる発音となる．また音節構造は，一単語一音節（CV または CVC）である．

8.3 限定詞について

文法上の機能として，単語を単純に修飾する語である限定詞が存在する．限定詞には，名詞を限定する名詞限定詞と，動詞を限定する動詞限定詞が存在する．

8.3.1 名詞限定詞

SFGPL には名詞限定詞が存在する．これは，元々名詞を修飾する特別な語である．しかし，限定詞自体の意味をそのまま名詞にすることもできる．そのためには，foを使用する．使用例を次の表に表す．

English	SFGPL
human	ben fo

これらの単語を使った例を次に表す．

English	SFGPL
I am human.	ma ga so ben fo

8.3.2 動詞限定詞

SFGPL には動詞限定詞が存在する．これは，動詞を修飾する特別な語である．そしてこれらは，動詞の時制や相として使われる語や，助動詞的に動詞の意味を付加するものが存在している．

8.4 無意味単語について

SFGPL には，意味を付加しない単語が存在する．これらの単語は，品詞ごとに存在し，文法上必要なときに使われる．

	SFGPL
Noun	fo
Verb	so
Modifier	lo

はじめに，無意味名詞の foでは，名詞限定詞をそのままの意味で表すときによく使われる．また，無意味動詞の soは，特に文型で，動詞が必要ない場合など使われる．一方，無意味修飾詞 loは，あまり使われない．これらの例を次に表す．

English	SFGPL
I am human.	ma ga so ben fo
I have an apple.	mi ga so fa ‘apple’

8.5 代名詞について

SFGPL では代名詞が存在する．代名詞は次の表のようなものがある．

	English	SFGPL
一人称代名詞	I	ga
二人称代名詞	you	ge
三人称代名詞	he/she/it	gi
近称代名詞	this	gu
遠称代名詞	that	go
疑問代名詞	what	wa
不定代名詞	something	we

8.6 数値や論理的に使われる語

SFGPL には，数値的な単語や真偽値に関する単語，リストに関する単語，関数に関する単語が存在している．これらの単語は，一般的な文ではあまり使われないが，論理的なことを示す際に使われる．

9 9. 動詞の活用

SFGPL では，時制や相，助動詞などの動詞を修飾する語が備わっている．これらの語は，主に，動詞に直接付属し修飾するものと，文全体に修飾するものが存在する．

9.1 動詞の時制

SFGPL では以下の図のような動詞の時制が存在する．

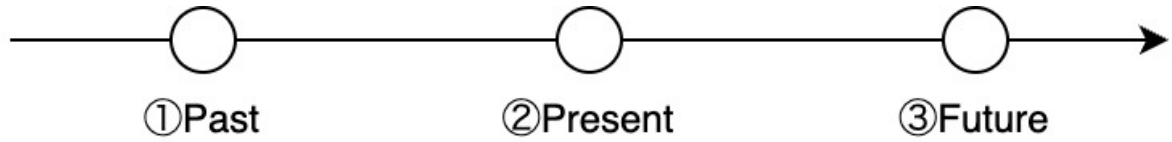

Figure 2: BasingPoint

このように，SFGPL では①過去形，②現在形，③未来形の 3 つの時制が存在する．これら時制は動詞の活用として基礎的なもので，文の時間の基準点となる．時制を使用した例文は次の表のようになる．

時制	English	SFGPL
①過去形	I lived in Tokyo.	di ta ga na sa ‘live’ li pun fa ‘Tokyo’
②現在形	I live in Tokyo.	ta ga na sa ‘live’ li pun fa ‘Tokyo’
③未来形	I will live in Tokyo.	du ta ga na sa ‘live’ li pun fa ‘Tokyo’

特に `di`と `du`では，文章自体に付属し形容する．

9.1.1 動詞の拡張時制

前項で説明した動詞は，一番基本的な動詞の時制の表し方である．しかし，SFGPL では DeterminV クラスににによって，主に時制を組み合わせるための単語が存在する．また，この DeterminerV クラスによる拡張時制は，Phrase クラスによる基礎時制より優先度が低く，基本的には基礎時制で文全体の時制を表す．以下の表は拡張時制を表す単語である．

時制	単語
①過去形	bak
②現在形	bik
③未来形	bok

これらの時制を組み合わせることで，未来過去形や過去未来形などの複合時制を作ることができる．次の例は，未来の時点で過去を表す未来過去形の例である．

```
1 du ta ga na bak sa 'live' li pun fa 'Tokyo'
```

まとめると SFGPL における時制は以下の表のようなものが存在する．以下の表の列名は Phrase による基礎時制の種類，行名は DeterminerV による拡張時制の種類を表している．また，`A/B`で A は基礎時制，B 拡張時制を表す，

	Past Tense	-	Future Tense
-	di/-	-/-	du/-
Past Tense	di/bak	-/bak	du/bak
Present Tense	di/bik	-/bik	du/bik
Future Tense	di/bok	-/bok	du/bok

9.2 相

SFGPL では下の図のように，①起動相，②経過相，③完結相，④継続相，⑤終了相，⑥進行相の 6 つが存在する．

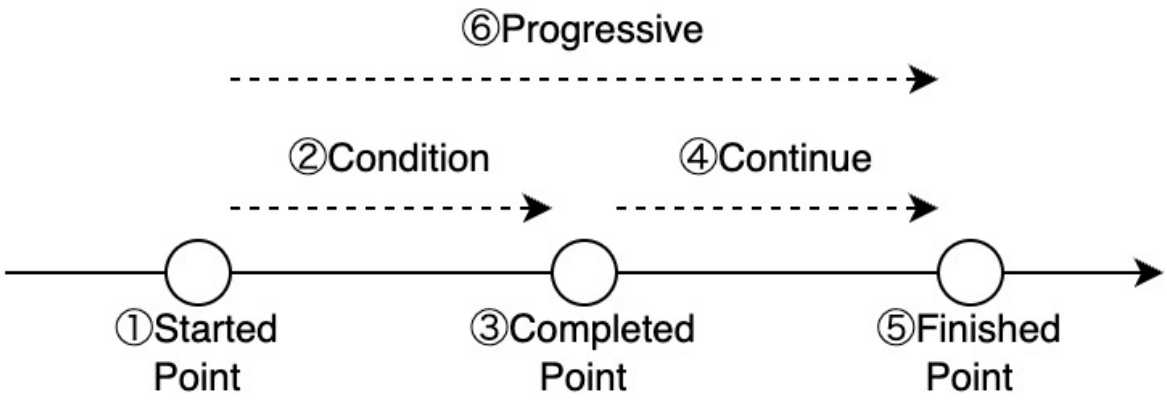

Figure 3: ProgressiveForm

“I wear dress” という意味の `te ga sa 'wear'fa 'dress'`について，それぞれの相での例文を次の表に示す．

相	単語	English	SFGPL
①起動相	tak	I begin wear a dress.	te ga tak sa ‘wear’ fa ‘dress’
②経過相	tek	I am (in the process of) wearing a dress.	te ga tek sa ‘wear’ fa ‘dress’

相	単語	English	SFGPL
③完結相	tik	I wear a dress. (I just finished wearing it.)	te ga tik sa ‘wear’ fa ‘dress’
④継続相	tuk	I am wearing a dress. (The state in which it is worn.)	te ga tuk sa ‘wear’ fa ‘dress’
⑤終了相	tok	I finish wear a dress. (I stopped wearing it.)	te ga tok sa ‘wear’ fa ‘dress’
⑥進行相	ni	I am wearing a dress.	te ga ni sa ‘wear’ fa ‘dress’

これらの相は，現在形以外にも，過去形，未来形にできる．⑥進行相は②経過相と④継続相が含まれている．また③完結相と⑤終了相が同じである場合もある．“I begin wear a dress.” を過去形，未来形にすると次のようになる．

```
di te ga tak sa 'wear' fa 'dress'
du te ga tak sa 'wear' fa 'dress'
```

また，原則として相単体では，時間の幅はなく，その瞬間だけを表す．時間の幅を表す場合は，完了形を付け加える．進行形に完了形を加えた “I have been wearing a dress.” を表すには，次のようになる．

```
te ga nu ni sa 'wear' fa 'dress'
```

9.2.1 一般的な進行形

SFGPL では前節の①〜⑤のような相を考えずに，⑥のように単純な進行形にすることができる．SFGPL は次のように，“I am wearing the dress.” という意味の進行形を表すことができる．

```
te ga ni sa 'wear' fa 'dress'
```

進行形を表す `ni`は動詞に付属する．これらは，現在形以外にも，過去形，未来形にできる．“I am wearing the dress.” を過去形，未来形にすると次のようになる．

```
di te ga ni sa 'wear' fa 'dress'
du te ga ni sa 'wear' fa 'dress'
```

9.3 完了形

SFGPL では，以下の図のような，英語と同等な完了形が存在する．

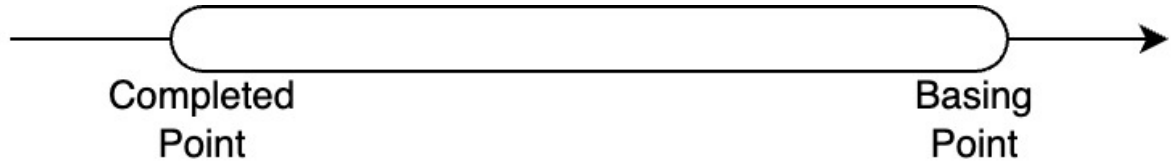

Figure 4: PerfectForm

この完了形では過去に起こったことが続いていることを表す際に使用する．3 つの時制に対する完了形の例は次のようになる．

時制	English	SFGPL
①過去完了形	I had lived in Tokyo.	di ta ga nu na sa ‘live’ li pun fa ‘Tokyo’
②現在完了形	I have lived in Tokyo.	ta ga nu na sa ‘live’ li pun fa ‘Tokyo’
③未来完了形	I will have lived in Tokyo.	du ta ga nu na sa ‘live’ li pun fa ‘Tokyo’

完了形を表す nuでは，動詞自体に付属し，修飾する．

9.4 受動態

SFGPL は次のように，“The dress is worn.” という意味の受動態を表すことができる．

```
1 ta fa 'dress' ne sa 'wear'
```

受動態を表す neは動詞に付属する．これらは，現在形以外にも，過去形，未来形にできる．“The dress is worn.” を過去形，未来形にすると次のようになる．

```
1 di ta fa 'dress' ne sa 'wear'
2 du ta fa 'dress' ne sa 'wear'
```

9.5 その他の動詞の修飾

DeterminerV クラス内の関数では，その他の動詞の修飾をすることができる．また，それらは，英語の助動詞と似ている．

9.6 単語集

English	SFGPL
I	ga
live	sa ‘live’
in Tokyo	li pun fa ‘Tokyo’
wear	sa ‘wear’
dress	fa ‘dress’

10 10. 修飾語

10.1 修飾語について

SFGPL には形容詞と副詞の違いがなく，修飾する語はすべて修飾語となる．

修飾語は，修飾の反対の意を表すための単語が用意されている．それによって，英語の “big” に対応する `wan`を `ke wan`とすることで，“small” という意味にすることができる．

10.2 比較表現

SFGPL には 2 項の名詞に対しての比較を行う文の `mo`が存在する．`mo A F B C`で，“A は C より B である．” という意味となる．

“My table is bigger than yours.” のような比較表現は次のようにして表す．

```
1 mo mi ga so san fa 'table' so wan sen ge
```

10.3 各品詞に対する修飾語

各品詞を単純に修飾語で修飾するためには，次の表になる．

	SFGPL
Noun	me Noun Modifier
Verb	na Verb Modifier
Modifier	ka Modifier Modifier

10.4 修飾語の応用

修飾語では，英語の前置詞と名詞含む句を修飾語として代用する事ができる．このとき，名詞を修飾語に変換する `li`と，名詞限定詞がよく組み合わされて表現される．例えば，“I live in Tokyo.”と表す場合は，次のように表せる．

```
1 ta ga na sa 'live' li pun fa 'Tokyo'
```

また，`pun`は，場所を表す限定詞である．

10.5 単語集

English	SFGPL
I	ga
table	fa ‘table’
yours	sen ge
live	sa ‘live’
in Tokyo	li pun fa ‘Tokyo’

11 11. 品詞変換

SFGPL では，名詞，動詞，修飾語の相互の品詞を変換することができる．以下の表は SFGPL で品詞変換する語の一覧である．

	変換前の品詞	変換後の品詞	単語
V2N	動詞	名詞	fi
M2N	修飾語	名詞	fu

	変換前の品詞	変換後の品詞	単語
M2V	修飾語	動詞	si
N2V	名詞	動詞	su
N2M	名詞	修飾語	li
V2M	動詞	修飾語	lu

特に，動詞から名詞，名詞から修飾語はよく使用される．

11.1 動詞から名詞

動詞から名詞は “This is building.” のように使用される．

```
1 ma gu so fi sa 'build'
```

また元の単語の動詞は動詞の活用に従って事前に活用させることも可能である．

11.2 名詞から修飾語

名詞から修飾語は，英語の前置詞と名詞が組み合わされた句と同等の意味を作成するときに使われる．またそのときは，liと限定詞 (DeterminerN) が組み合わされて使用する．“I live in Tokyo.” を SFGPL にすると次のようになる．このとき，punは場所を表す限定詞である．

```
1 ta ga na sa 'live' li pun fa 'Tokyo'
```

また，名詞を抽象化する単語の sonと組み合わせることで，“〜的な” という意味にすることができる．“My daughter has a cat-like stuffed toy.” を SFGPL で表すには次のようになる．

```
1 mi mi ga so san fa 'daughter' so me me fa 'toy' so lu ne sa 'stuff' so
    li son fa 'cat'
```

11.3 動詞から修飾語

動詞から修飾語に変換すると，印欧語族に多く見られる分詞に相当する用法を使用できる．また元の単語の動詞は動詞の活用に従って事前に活用させることも可能である．

“There is a sleeping boy.” を SFGPL で表すには次のようにする．

```
1 ma pun go so me fa 'boy' so lu ni sa 'sleep'
```

また，“I lived in that destroyed building.” を表すには次のように表現する．

```
di ta ga na sa 'live' li pun ma go so san me fi sa 'build' so lu ne sa 'destroy'
```

11.4 単語集

English	SFGPL
this	gu
build	sa ‘build’
I	ga
live	sa ‘live’
in Tokyo	li pun fa ‘Tokyo’
daughter	fa ‘daughter’
cat	fa ‘cat’
stuffed	lu ne sa ‘stuff’
toy	fa ‘toy’
there	pun go
sleep	sa ‘sleep’
boy	fa ‘boy’
that	go
destroy	sa ‘destroy’

12 12. 接続詞

SFGPL は，文と文や単語と単語を繋ぐものとして接続詞が存在する．SFGPL の主な接続詞として次のようなものがある．

Word	English Word	English	SFGPL
pe	because	I go to a store, because I want it.	pe ta ga na sa ‘go’ li pun fa ‘store’ te ga sa ‘want’ pen gi
pu	so	I want it, so I go to a store.	pu te ga sa ‘want’ pen gi ta ga na sa ‘go’ li pun fa ‘store’
pi	if	I go to a store, if I want it.	pi ta ga na sa ‘go’ li pun fa ‘store’ te ga sa ‘want’ pen gi
po	but	I want it, but I don’t go to a store.	po te ga sa ‘want’ pen gi pa ta ga na sa ‘go’ li pun fa ‘store’
ba	and	I go to a store, and I go to a library.	ba ta ga na sa ‘go’ li pun fa ‘store’ ta ga na sa ‘go’ li pun fa ‘library’
be	or	I go to a store, or I go to a library.	I go to a store, or I go to a library.

また，`ba fa 'store'fa 'library'`や`be fa 'store'fa 'library'`のように，単語同士でも接続することができる．

12.1 単語集

English	SFGPL
I go to a store	ta ga na sa ‘go’ li pun fa ‘store’
I don’t go to a store	pa ta ga na sa ‘go’ li pun fa ‘store’
I want it	te ga sa ‘want’ pen gi
I go to a library	ta ga na sa ‘go’ li pun fa ‘library’
store	fa ‘store’
library	fa ‘library’

13 13. 代名詞

13.1 代名詞一覧

代名詞は次の表のようなものがある．

	English	SFGPL
一人称代名詞	I	ga
二人称代名詞	you	ge
三人称代名詞	he/she/it	gi
近称代名詞	this	gu
遠称代名詞	that	go
疑問代名詞	what	wa
不定代名詞	something	we

13.2 代名詞の応用

SFGPL の代名詞は原則として人，生物，物，概念，場所，時間，理由，方法等の区別はされない．そして，性別や，数による区別も存在しない．これらの区別をする場合は，名詞限定詞を使用することで限定できる．

疑問詞に対する名詞限定詞の使用方法は次の表となる．

English	SFGPL
what	pen wa
who	ben wa
when	pin wa
where	pun wa
why	pon wa
how	ban wa

また，複数形を明示するためには donを使用する．例えば，“We” を表すには don ga とする．

SFGPL では性の区別が存在しない．また，人と物の区別も存在しない．例えば，三人称代名詞の男性，女性，事物を明示するためには，次のようにする．

	English	SFGPL
男性	he	lan gi
女性	she	len gi
事物	it	pen gi

さらに，所有代名詞，再帰代名詞を作成するには `sen`や `sin`を使用する．次の表は，一人称代名詞の所有代名詞，再帰代名詞である．

	English	SFGPL
所有代名詞	mine	sen ga
再帰代名詞	myself	sin ga

14 14. 名詞限定詞

名詞限定詞は，名詞を修飾するための最も単純なものである．また，代名詞と使われたり，名詞から修飾語に変換するときに使用する `li`と一緒に使われることが多い．

次の表は名詞限定詞の例である．

Word	Base Meaning	English	SFGPL
lan	male	He is student.	ma lan gi so fa ‘student’
len	female	She is student.	ma len gi so fa ‘student’
don	plural	They are student.	ma don gi so fa ‘student’
pun	place	I go to Tokyo.	ta ga na sa ‘go’ li pun fa ‘Tokyo’
pin	time	I go today.	ta ga na sa ‘go’ li pin fa ‘today’

また，名詞限定詞は複数個付加することができる．

一般的に，名詞限定詞 A,B と名詞 N の場合で，`A B N`という句は，「A の (B の N)」という意味になる．

14.1 単語集

English	SFGPL
he/she/they	gi
student	fa ‘student’
I	ga
go	sa ‘go’
Tokyo	fa ‘Tokyo’
today	fa ‘today’

15 15. 動詞限定詞

動詞限定詞は，動詞を修飾するための最も単純なものである．これらは，英語の助動詞に相当する．次の表は動詞限定詞の例である．

Word	Base Meaning	English	SFGPL
nak	possible	I can see a sea.	te ga nak sa ‘see’ fa ‘sea’
nek	ability	I can swim.	ta ga nek sa ‘swim’
nuk	obligation	I should swim.	ta ga nuk sa ‘swim’
nok	necessary	I need to swim.	ta ga nok sa ‘swim’
lak	duty	I must swim.	ta ga lak sa ‘swim’
lik	want to	I want to swim.	ta ga lik sa ‘swim’

また，相などの，動詞の活用をすることもできる．

15.1 単語集

English	SFGPL
I	ga
see	sa ‘see’

English	SFGPL
sea	fa ‘sea’
swim	sa ‘swim’

16 16. Bool 関連クラス

SFGPL には Bool に関連したクラス，Bool 型と，BoolList 型が存在する．これらのクラスは，真偽値や，数値などを表すために使われる．

16.1 Bool 型について

Bool 型は，真偽を表すためのクラスである．Bool 型の False と True は次のように表される．

	word
False	pas
True	pos

また，pisを使用して，Bool 型と名詞を次のように接続することで，ある名詞に対する真偽を表すことができる．次の文は “It is true that I am a student.” という例を表す．

```
1 pis ma ga so fa 'student' pos
```

そして，Bool 型では，LangObj に備わっている，NOT pa，OR be，AND ba，NOR boと NAND buを使用することもできる．そして，それら関数は論理演算をすることができる．

16.2 BoolList 型について

BoolList では，真偽値の配列を作成することができる．BoolList には以下のような関数が存在している．

単語	説明
fas	真偽のリスト（BoolList）を作成する
fes A B	BoolList(A) の B 番目の値を取得する

単語	説明
fis A B	BoolList(A) に 1 つの Bool(B) を末尾に加える
fus A B C	A という BoolList に対して，B 番目から C 番目までのリストを取得する
fos A B	2 つの BoolList(A,B) を結合する
mas A B	2 つ Bool の値 (A,B) からなる BoolList を作成する
mis X1~X4	4 つ Bool の値 (x1~x4) からなる BoolList を作成する
mos X1~X8	8 つ Bool の値 (x1~x8) からなる BoolList を作成する
tas A	BoolList(A) を 2 進数の自然数とみなす
tes A	BoolList(A) を 2 進数の整数とみなす
tis A	BoolList(A) を 2 進数の浮動小数とみなす
tus A	BoolList(A) を ASCII 文字とみなす

次のようにすることによって，4byte のデータを使用することができる．

```
fos fos mos pas pos pas pas pas pas pas pas mos pas pos pas pas pos pas
    pas pos fos mos pas pas pas pas pos pos pos pos mos pos pos pas pos
    pos pas pos pos
```

これは，2 進数で `0100 0000 0100 1001 0000 1111 1101 1011`を表している．また，次のようにすることで，数値として使うことができる．

Type	SFGPL	Value
自然数	tas fos fos mos pas pos pas pas pas pas pas pas mos pas pos pas pas pos pas pas pos fos mos pas pas pas pas pos pos pos pos mos pos pos pas pos pos pas pos pos	1078530011
整数	tes fos fos mos pas pos pas pas pas pas pas pas mos pas pos pas pas pos pas pas pos fos mos pas pas pas pas pos pos pos pos mos pos pos pas pos pos pas pos pos	1078530011

Type	SFGPL	Value
浮動小数点	tis fos fos mos pas pos pas pas pas pas pas pas mos pas pos pas pas pos pas pas pos fos mos pas pas pas pas pos pos pos pos mos pos pos pas pos pos pas pos pos	3.1415927410125732

16.3 単語集

English	SFGPL
I am a student	ma ga so fa ‘student’

17 17. LangList

SFGPL では基本的なデータ構造型として，LangList 型が存在する．LangList には，以下の関数が存在している．

単語	説明
fat	LangObj のリスト LangList を作成する
fet A B	LangList(A) の B 番目の値を取得する
fit A B	LangList(A) に 1 つの LangObj(B) を末尾に加える
fut A B C	A という LangList に対して，B 番目から C 番目までのリストを取得する
fot A B	2 つの LangList を結合する

LangList は，LangObj を継承しているすべてのクラスを格納することができる．次は LangList を作成する一例である．

```
1 fit fit fit fit fit fat ga fa 'pen' sa 'go' la 'happy' ma ga so fa '
    student'
```

また, この LangList から最初の値を取得するには次のようにする. このとき `fis fas pas`は BoolList における 0 を表している．

```
fet fit fit fit fit fit fat ga fa 'pen' sa 'go' la 'happy' ma ga so fa
    'student' fis fas pas
```

17.1 単語集

English	SFGPL
I	ga
pen	fa ‘pen’
go	sa ‘go’
happy	la ‘happy’
I am a student	ma ga so fa ‘student’

18 18. LangFunc

SFGPL では基本的な関数型として，LangFunc 型が存在する．LangFunc には，以下の関数が存在している．

単語	説明
pat A B	ある LangList を引数とする A という名前の B を返す関数を設定する
pit	pat の引数用に使用する
pot A B	設定した A という名前の LangFunc を引数 B として実行する

LangFunc は，patによって関数を設定する．また，pitを patの第二引数内の文内に含ませることができる．それによって，関数実行時に実際の値が代入されて処理される．また，patの第一引数は関数名を表す．そして，関数名は重複して付けることはできない．以下は，関数設定の例を示す．

```
pat fa 'xor' fit fat bu bu fet pit mas pas pas bu fet pit mas pas pas
    fet pit mas pas pos bu bu fet pit mas pas pas fet pit mas pas pos
    fet pit mas pas pos
```

この関数は, ある LangList に対して, 0 番目と 1 番目の XOR を取る関数である. この関数に (false,false) を与えるときは，次のようにする．

```
pot fa 'xor' fit fit fat pas pas
```

19 19. 数字の表現方法

SFGPL では 10 進数の数値を表すために，Number と NumberList クラスが存在する．

19.1 Number クラス

Number クラスは，基数詞用のクラスであり，単体では使用されない．このクラスでは以下の表のように，0~9 までの値が定義されている．

Meaning	SFGPL
0	pal
1	pel
2	pil
3	pul
4	pol
5	bal
6	bel
7	bil
8	bul
9	bol

19.2 NumberList クラス

通常の数詞として使う場合には NumberList クラスを使用する．このクラスは基数詞のデータをリストに格納することができる．数値の表現方法は，大きい桁から順に 0 番目から格納し，10 進数の数値を表す．

NumberList クラスにはリスト型の関数として次の表のようなものが存在する．ただしこれらの関数は，下記に記述する数値計算した後の NumberList には適用することができない．

単語	説明
fal	Number のリスト NumberList を作成する
fel A B	NumberList(A) の B 番目の値を取得する
fil A B	NumberList に 1 つの Number を末尾に加える
ful A B C	A という NumberList に対して，B 番目から C 番目までのリストを取得する
fol A B	2 つの NumberList を結合する

また，1~5 桁の整数を作るためには，以下の表のような専用の関数が用意されている．

単語	説明
mal	10 進数 1 桁からなる NumberList を作成する
mel	10 進数 2 桁からなる NumberList を作成する
mil	10 進数 3 桁からなる NumberList を作成する
mul	10 進数 4 桁からなる NumberList を作成する
mol	10 進数 5 桁からなる NumberList を作成する

SFGPL で，“I have five apples.” を表すには次のようにする．

```
mi ga so ma fa 'apple' so mal bal
```

また，“I have fifteen apples.” を表すには次のようにする．

```
mi ga so ma fa 'apple' so mel pel bal
```

さらに，10 進数で 5 桁より大きな数値を表すためには，次のように `fol`を使い，2 つの NumberList を結合することで実現できる．次の文は SFGPL で “Japan has 125416877 people.” を表している．

```
mi fa 'Japan' so ma fa 'people' so fol mul pel pil bal pol mol pel bel
    bul bil bil
```

そして，次の表のように NumberList では四則演算を行う関数が存在する．

	SFGPL
Addition	tal
Subtraction	tel

	SFGPL
Multiplication	til
Division	tul

加えて，次の表のように整数の BoolList と NumberList を相互に変換する関数が存在する．

SFGPL	from	to
tol	NumberList	BoolList
tos	BoolList	NumberList

19.3 単語集

English	SFGPL
I	ga
apple	fa ‘apple’
Japan	fa ‘Japan’
people	fa ‘people’

20 20. 例文

以下の表は，SFGPL の例文を示す．

SFGPL	Python	Translation
ma ga so me fa ‘worker’ so li pun fa ‘office’	Noun.eq(Pronoun.I( ) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”)))))	I am an office worker.

SFGPL	Python	Translation
ma ge so me fa ‘worker’ so li pun fa ‘office’	Noun.eq(Pronoun.you( ) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”)))))	You are an office worker.
ma lan gi so me fa ‘worker’ so li pun fa ‘office’	Noun.eq(DeterminerN.male(Pronoun.he( )) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”)))))	He is an office worker.
ma len gi so me fa ‘worker’ so li pun fa ‘office’	Noun.eq(DeterminerN.female(Pronoun.he( )) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”)))))	She is an office worker.
ma don ga so me fa ‘worker’ so li pun fa ‘office’	Noun.eq(DeterminerN.plural(Pronoun.I( )) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”)))))	We are an office worker.
ma don ge so me fa ‘worker’ so li pun fa ‘office’	Noun.eq(DeterminerN.plural(Pronoun.you( )) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”)))))	You are an office worker.

SFGPL	Python	Translation
ma don gi so me fa ‘worker’ so li pun fa ‘office’	Noun.eq(DeterminerN.plural(Pronoun.he( )) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”))))))	They are an office worker.
di ma ga so me fa ‘worker’ so li pun fa ‘office’	Phrase.past(Noun.eq(Pronoun.I( ) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”))))))	I was an office worker.
du ma ga so me fa ‘worker’ so li pun fa ‘office’	Phrase.future(Noun.eq(Pronoun.I( ) , Verb.none( ) , Noun.haveP(Noun(“ ‘worker’ ”) , Verb.none( ) , Modifier.N2M(DeterminerN.place(Noun(“ ‘office’ ”))))))	I will be an office worker.
ta ga na sa ‘go’ li pun mu ga so san fa ‘school’	Noun.do(Pronoun.I( ) , Verb.add(Verb(“ ‘go’ ”) , Modifier.N2M(DeterminerN.place(Noun.belong(Pronoun.I( ) , Verb.none( ) , DeterminerN.stressed(Noun(“ ‘school’ ”)))))))	I go to my school.

SFGPL	Python	Translation
di ta ga na sa ‘go’ li pun mu ga so san fa ‘school’	Phrase.past(Noun.do(Pronoun.I( ) , Verb.add(Verb(“ ‘go’ ”) , Modifier.N2M(DeterminerN.place(Noun.belong(Pronoun.I( ) , Verb.none( ) , DeterminerN.stressed(Noun(“ ‘school’ ”))))))))	I went to my school.
du ta ga na sa ‘go’ li pun mu ga so san fa ‘school’	Phrase.future(Noun.do(Pronoun.I( ) , Verb.add(Verb(“ ‘go’ ”) , Modifier.N2M(DeterminerN.place(Noun.belong(Pronoun.I( ) , Verb.none( ) , DeterminerN.stressed(Noun(“ ‘school’ ”))))))))	I will go to my school.
te ga sa ‘read’ fa ‘book’	Noun.doT(Pronoun.I( ) , Verb(“ ‘read’ ”) , Noun(“ ‘book’ ”))	I read a book.
di ti ga na sa ‘send’ li pin fa ‘yesterday’ lan gi fa ‘letter’	Phrase.past(Noun.give(Pronoun.I( ) , Verb.add(Verb(“ ‘send’ ”) , Modifier.N2M(DeterminerN.time(Noun(“ ‘yesterday’ ”)))) , DeterminerN.male(Pronoun.he( )) , Noun(“ ‘letter’ ”)))	I sent him a letter yesterday.
di tu ga so lan gi fa ‘teacher’	Phrase.past(Noun.makeN(Pronoun.I( ) , Verb.none( ) , DeterminerN.male(Pronoun.he( )) , Noun(“ ‘teacher’ ”)))	I made him a teacher.

SFGPL	Python	Translation
di to ga so lan gi la ‘happy’	Phrase.past(Noun.makeM(Pronoun.I( ) , Verb.none( ) , DeterminerN.male(Pronoun.he( )) , Modifier(“ ‘happy’ ”)))	I made her happy.
mo lan gi so la ‘tall’ ga	Noun.gt(DeterminerN.male(Pronoun.he( )) , Verb.none( ) , Modifier(“ ‘tall’ ”) , Pronoun.I( ))	He is taller than me.
di te ga na sa ‘put’ li pun min fa ‘table’ ba fa ‘apple’ fa ‘peach’	Phrase.past(Noun.doT(Pronoun.I( ) , Verb.add(Verb(“ ‘put’ ”) , Modifier.N2M(DeterminerN.place(DeterminerN.on(Noun(“ ‘table’ ”))))) , LangObj.AND(Noun(“ ‘apple’ ”) , Noun(“ ‘peach’ ”))))	I put an apple and a peach on the table.
ta ga na sa ‘go’ li pun fa ‘Osaka’	Noun.do(Pronoun.I( ) , Verb.add(Verb(“ ‘go’ ”) , Modifier.N2M(DeterminerN.place(Noun(“ ‘Osaka’ ”)))))	I go to Osaka.
di ta ga na sa ‘go’ li pun fa ‘Osaka’	Phrase.past(Noun.do(Pronoun.I( ) , Verb.add(Verb(“ ‘go’ ”) , Modifier.N2M(DeterminerN.place(Noun(“ ‘Osaka’ ”))))))	I went to Osaka.
du ta ga na sa ‘go’ li pun fa ‘Osaka’	Phrase.future(Noun.do(Pronoun.I( ) , Verb.add(Verb(“ ‘go’ ”) , Modifier.N2M(DeterminerN.place(Noun(“ ‘Osaka’ ”))))))	I will go to Osaka.

SFGPL	Python	Translation
te ga sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”))	I create a table.
te ga sa ‘create’ ma gu so san fa ‘table’	Noun.doT(Pronoun.I( ) , Verb(“ ‘create’ ”) , Noun.eq(Pronoun.proximal( ) , Verb.none( ) , DeterminerN.stressed(Noun(“ ‘table’ ”))))	I create this table.
pa te ga sa ‘create’ fa ‘table’	LangObj.NOT(Noun.doT(Pronoun.I( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)))	I don’t create a table.
te ge sa ‘create’ fa ‘table’	Noun.doT(Pronoun.you( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”))	You create a table.
da te ge sa ‘create’ fa ‘table’	Phrase.interrogative(Noun.doT(Pronoun.you( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)))	Do you create a table?
da di te ge sa ‘create’ fa ‘table’	Phrase.interrogative(Phrase.past(Noun.doT(Pronoun.you( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”))))	Did you create a table?
da te ben wa sa ‘create’ fa ‘table’	Phrase.interrogative(Noun.doT(DeterminerN.human(Pronoun.interrogative( )) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)))	Who create the table?

SFGPL	Python	Translation
da te ge sa ‘create’ pen wa	Phrase.interrogative(Noun.doT(Pronoun.you( ) , Verb(“ ‘create’ ”) , DeterminerN.thing(Pronoun.interrogative( ))))	What do you create?
da te ge na sa ‘create’ li pin wa fa ‘table’	Phrase.interrogative(Noun.doT(Pronoun.you( ) , Verb.add(Verb(“ ‘create’ ”) , Modifier.N2M(DeterminerN.time(Pronoun.interrogative( )))) , Noun(“ ‘table’ ”)))	When do you create the table?
da te ge na sa ‘create’ li pon wa fa ‘table’	Phrase.interrogative(Noun.doT(Pronoun.you( ) , Verb.add(Verb(“ ‘create’ ”) , Modifier.N2M(DeterminerN.reason(Pronoun.interrogative( )))) , Noun(“ ‘table’ ”)))	Why do you create the table?
de te we sa ‘create’ fa ‘table’	Phrase.imperative(Noun.doT(Pronoun.indefinite( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)))	Create a table!
di te ga sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)))	I created a table.
du te ga sa ‘create’ fa ‘table’	Phrase.future(Noun.doT(Pronoun.I( ) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)))	I will create a table.

SFGPL	Python	Translation
ta fa ‘table’ na ne sa ‘create’ li tan tin ga	Noun.do(Noun(“ ‘table’ ”) , Verb.add(Verb.passive(Verb(“ ‘create’ ”)) , Modifier.N2M(DeterminerN.affect(DeterminerN.near(Pronoun.I( ))))))	The table is created by me.
te ga ni sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , Verb.progressive(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I am creating a table.
te ga nu sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , Verb.perfective(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I have created a table.
du te ga pak sa ‘create’ fa ‘table’	Phrase.future(Noun.doT(Pronoun.I( ) , DeterminerV.Estimation100(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I 100% probability will create a table.
du te ga pek sa ‘create’ fa ‘table’	Phrase.future(Noun.doT(Pronoun.I( ) , DeterminerV.Estimation75(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I 75% probability will create a table.
du te ga pik sa ‘create’ fa ‘table’	Phrase.future(Noun.doT(Pronoun.I( ) , DeterminerV.Estimation50(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I 50% probability will create a table.
du te ga puk sa ‘create’ fa ‘table’	Phrase.future(Noun.doT(Pronoun.I( ) , DeterminerV.Estimation25(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I 25% probability will create a table.

SFGPL	Python	Translation
du te ga pok sa ‘create’ fa ‘table’	Phrase.future(Noun.doT(Pronoun.I( ) , DeterminerV.Estimation0(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I 0% probability will create a table.
te ga fak sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Frequency100(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I 100% frequently create a table.
te ga fek sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Frequency75(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I 75% frequently create a table.
te ga fik sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Frequency50(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I 50% frequently create a table.
te ga fuk sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Frequency25(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I 25% frequently create a table.
te ga fok sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Frequency0(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I 0% frequently create a table.
te ga bik sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.present(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I create a table.
te ga bak sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.past(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I created a table.

SFGPL	Python	Translation
te ga bok sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.future(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I will create a table.
di te ga bak sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , DeterminerV.past(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I created a table.(Past in the past at a point in time)
di te ga bik sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , DeterminerV.present(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I created a table.(Present in the past at a point in time)
di te ga bok sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , DeterminerV.future(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I would create a table.(Future in the past at a point in time)
di te ga bak sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , DeterminerV.past(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I will have created a table.(Past in the future at a point in time)
di te ga bik sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , DeterminerV.present(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I will create a table.(Present in the future at a point in time)
di te ga bok sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , DeterminerV.future(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	I will create a table.(Future in the future at a point in time)

SFGPL	Python	Translation
te ga nak sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Possible(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I can create a table.
te ga nek sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Ability(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I can create a table.
te ga nik sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Will(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I will create a table.
te ga nuk sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Obligation(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I should create a table.
te ga nok sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Necessary(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I need to create a table.
te ga lak sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Duty(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I must create a table.
te ga lek sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.forced(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I am forced to create a table.
te ga lik sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.want(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I want to create a table.

SFGPL	Python	Translation
te ga luk sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.dare(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I dare create a table.
te ga lok sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.allow(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I allow to create a table.
te ga kak sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.easy(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I am easy to create a table.
te ga kek sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.hard(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I am hard to create a table.
te ga kik sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.habit(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I habitually create a table.
te ga kuk sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.Polite(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I create a table.(polite expression)
te lan gi kok sa ‘create’ fa ‘table’	Noun.doT(DeterminerN.male(Pronoun.he( )) , DeterminerV.Respect(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	He creates a table.(respectful expression)
te ga gak sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.volitional(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I consciously create a table.

SFGPL	Python	Translation
te ga gek sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , DeterminerV.nonVolitional(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))	I unconsciously create a table.
da te ge gik sa ‘create’ fa ‘table’	Phrase.interrogative(Noun.doT(Pronoun.you( ) , DeterminerV.Requests(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	Can you create a table?
da te ga guk sa ‘create’ fa ‘table’	Phrase.interrogative(Noun.doT(Pronoun.I( ) , DeterminerV.Permission(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	May I create a table?
da te ga gok sa ‘create’ fa ‘table’	Phrase.interrogative(Noun.doT(Pronoun.I( ) , DeterminerV.Suggestion(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))	Shall I create a table?
te ga sa ‘get’ ma fa ‘information’ so te lan gi nu sa ‘create’ fa ‘table’	Noun.doT(Pronoun.I( ) , Verb(“ ‘get’ ”) , Noun.eq(Noun(“ ‘information’ ”) , Verb.none( ) , Noun.doT(DeterminerN.male(Pronoun.he( )) , Verb.perfective(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”))))	I get the information that he has create a table.

SFGPL	Python	Translation
di te ga sa ‘get’ ma fa ‘information’ so te lan gi nu sa ‘create’ fa ‘table’	Phrase.past(Noun.doT(Pronoun.I( ) , Verb(“ ‘get’ ”) , Noun.eq(Noun(“ ‘information’ ”) , Verb.none( ) , Noun.doT(DeterminerN.male(Pronoun.he( )) , Verb.perfective(Verb(“ ‘create’ ”)) , Noun(“ ‘table’ ”)))))	I got the information that he has create a table.
di te ga sa ‘get’ ma fa ‘information’ so te lan gi nu sa ‘create’ ma don fa ‘table’ so mal pul	Phrase.past(Noun.doT(Pronoun.I( ) , Verb(“ ‘get’ ”) , Noun.eq(Noun(“ ‘information’ ”) , Verb.none( ) , Noun.doT(DeterminerN.male(Pronoun.he( )) , Verb.perfective(Verb(“ ‘create’ ”)) , Noun.eq(DeterminerN.plural(Noun(“ ‘table’ ”)) , Verb.none( ) , NumberList.digit1(Number.three( )))))))	I got the information that he has create three tables.
di moa ga so te lan gi sa ‘create’ fa ‘table’ fa ‘John’	Phrase.past(Noun.hearSay(Pronoun.I( ) , Verb.none( ) , Noun.doT(DeterminerN.male(Pronoun.he( )) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)) , Noun(“ ‘John’ ”)))	According to John, I heard that he create a table.

SFGPL	Python	Translation
di moa ge so te lan gi sa ‘create’ fa ‘table’ fa ‘John’	Phrase.past(Noun.hearSay(Pronoun.you( ) , Verb.none( ) , Noun.doT(DeterminerN.male(Pronoun.he( )) , Verb(“ ‘create’ ”) , Noun(“ ‘table’ ”)) , Noun(“ ‘John’ ”)))	According to John, you heard that he create a table.

21 21. バージョンについて

このプロジェクトのバージョンは__version__.py に記載されている．特に，Python で実行する場合は，以下のコードを実行することで確認できる．

```
SFGPL.__version__.__version__
```

また，Python コードの SFGPL.SFGPLCorpus.saveJsonで出力されるコーパスの JSON ファイル内には，実行されたときのバージョンが記載される．

21.1 バージョンの命名規則

SFGPL では，A.B.Cのようなバージョンを使用し，管理している．バージョン名の変更による変更によるアップデート内容は，次のような表をもとにしている．

Version	Update	Contents
A	メインアップデート	単語やプログラム等の大きな変更がある場合
B	マイナーアップデート	単語やプログラム等の少量の変更がある場合
C	パッチアップデート	プログラムのバグ修正等による少量の変更やドキュメントの変更がある場合

21.2 バージョン更新内容について

Version	Update contents
1.0.0	正式版公開

Version	Update contents
1.0.1	例文の追加・修正
1.0.2	例文の追加・修正
1.0.3	バージョンごとの更新内容詳細の追加
1.1.0	Python での SFGPL の使い方詳細を追加
1.1.1	How_to_Use_SFGPL_in_Python.ipynb の修正
2.0.0	論理値に関するクラスを追加
2.0.1	Python プログラムの追加・修正
2.0.2	ドキュメントの追加・修正
2.1.0	BoolList.get() と BoolList.slice() を追加
3.0.0	LangList と LangFunc クラスの追加
3.0.1	How_to_Use_SFGPL_in_Python.ipynb の修正
3.1.0	LangFunc.runFunc() の修正
3.1.1	ドキュメントの追加・修正
3.1.2	ドキュメントの追加・修正
3.1.3	ドキュメントの追加・修正
4.0.0	DeterminerV クラスの追加
4.0.1	辞書の修正
4.0.2	ドキュメントの追加・修正
4.0.3	ドキュメントの追加・修正
4.0.4	ドキュメントの追加・修正
4.0.5	ドキュメントの追加・修正
4.0.6	ドキュメントの追加・修正
4.0.7	ドキュメントの追加・修正
4.0.8	ドキュメントの追加・修正
4.0.9	ドキュメントの追加・修正
4.0.10	ドキュメントの追加・修正
4.0.11	ドキュメントの追加・修正

Version	Update contents
4.0.12	ドキュメントの追加・修正
4.0.13	ドキュメントの追加・修正
4.1.0	Noun.hearSay() を追加
4.1.1	辞書の修正
4.1.2	ドキュメントの追加・修正
4.1.3	ドキュメントの追加・修正
5.0.0	Number と NumberList クラスの追加
5.0.1	ドキュメントの追加・修正
5.0.2	ドキュメントの追加・修正
5.0.3	ドキュメントの追加・修正
5.0.4	ドキュメントの追加・修正
5.0.5	ドキュメントの追加・修正
5.0.6	ドキュメントの追加・修正
5.0.7	ドキュメントの追加・修正
5.0.8	ドキュメントの追加・修正
5.0.9	ドキュメントの追加・修正