SFGPL 入門

Eruhitsuji

2024-09-22

Contents

Part I.
SFGPLの概要と基礎的な文法

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．

そして，このプロジェクトでは，GitHub:https://github.com/Eruhitsuji/SFGPL において，資料やプログラムを公開している．

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/
d	/d/
s	/s/
n	/n/
l	/l/
k	/k/
g	/g/
j	/j/
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/
oa	/oa/

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, LangVar と NumberList が存在する．

BoolList，LangList，LangFunc，LangVar 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．LangObj は単語の基本構造体であり，_BaseList は List 用の基本構造が定義されている．また，DeterminerN と DeterminerV は，単語に意味を付加するためだけの単語であり，Python 上では，ただの静的な関数として表されている．

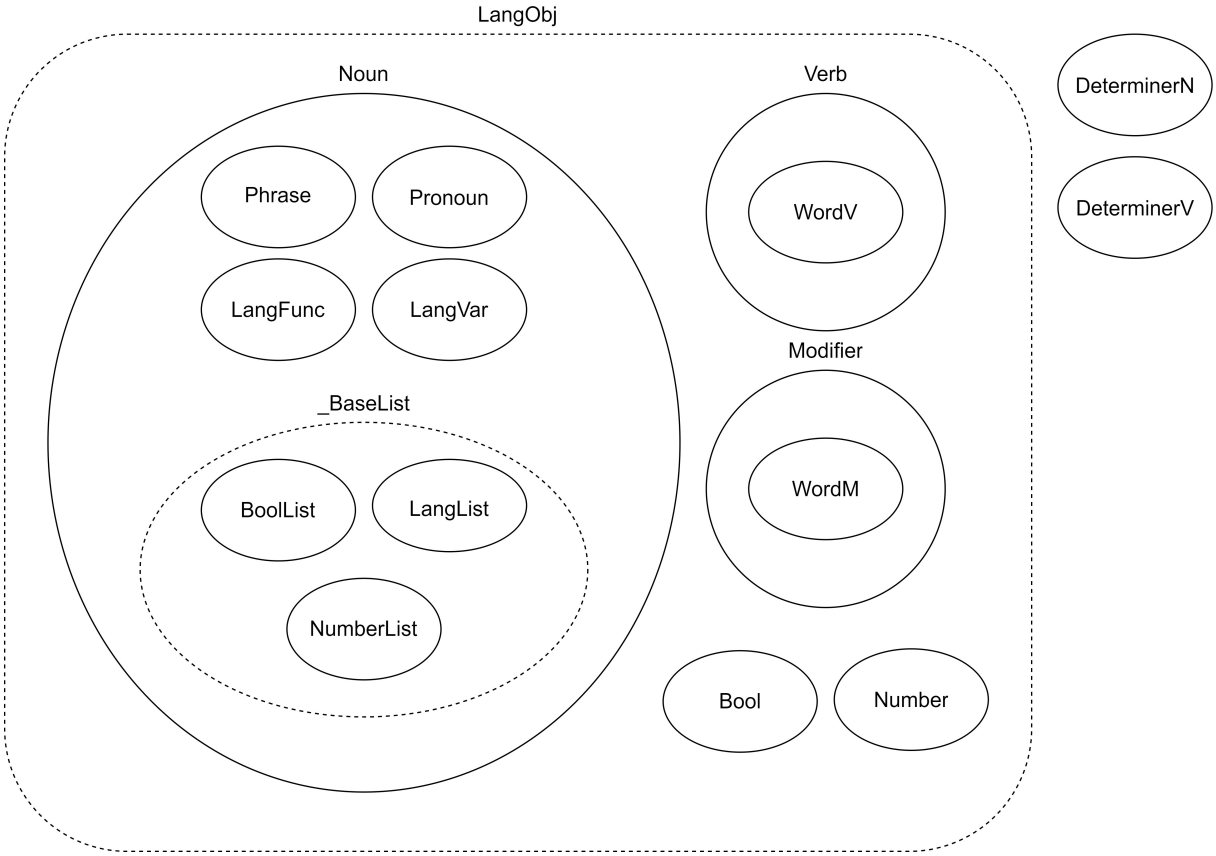

Figure 1: PartOfSpeach

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. 基本文法

この章では，SFGPL を学ぶための基本的知識や文法を説明する．特に，現在形の肯定形の基本的な文について説明する．

また前提として aboutSFGPL を読んでおくことを推奨する．さらにこの資料の全体として，例文等は英語を元に作成しているため，少し英語が分かること（中学卒業程度）が望ましい．

2.1. SFGPL 文構造の特徴について

SFGPL では英語と同様に，SOV 型で語の役割が位置で決まる言語である．また，SFGPL の最大の特徴として，文型が重視されていることが挙げられる．この文型は，それぞれどのような引数（どのような品詞の単語）を何個とるかが定義されている．そのため，文の意味が一意に決まる．また，この文型を決定する機能語が文（句）の先頭に付属する．そして，この文（句）全体では，名詞としてみなされ，入れ子的に文を作ることができる（複文）．

2.2. SFGPL の文型の具体例

2.2.1. ta による構文

まず，taを使用した例を提示する．この taでは，2 つの引数を持ち，第一引数が文の主語，第二引数が文の動詞を表す．つまり，taでは，英語の第一文型 SV と同等の文を作ることができる．

例として SFGPL で “I run.” を表すには次のようにする．

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

このとき，taは，文型が “SV” のときに付ける語である．

また，gaは，一人称代名詞 “I” を表す．

そして，sa 'run'は，動詞 “run” を表す．この sa 'run'は 2 単語から構成されている．このような借用語などでは，品詞を表す語（この場合は動詞を表す，sa）が付いている．このような品詞を表す語は，次の 3 つが存在する．

	SFGPL
名詞	fa
動詞	sa
修飾語	la

2.2.2. ma による構文

次に，maを使用した例を提示する．この maは 3 つの引数を持ち，第一引数が文の主語，第二引数が文の動詞，第三引数が主語に対する補語を表す．また，第三引数の補語は名詞でないといけない．つまり maでは，英語の第二文型 SVC と同等の文を作ることができる．

例として SFGPL で “I am a student.” を表すには次のようにする．

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

このとき，maは，文型が “SVC” のときに付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，soは，動詞が無意味であることを示す単語である．この soでは，場所によって意味が変わる．このとき例文のときは，英語の be 動詞と同等の意味となる．

そして，fa 'student'は，名詞 “student” を表す．このとき，英語などにある冠詞は SFGPL では存在しないため，何もつけなくても良い．

2.2.3. me による構文

次に，meを使用した例を提示する．この meは 3 つの引数を持ち，第一引数が文の主語，第二引数が文の動詞，第三引数が主語に対する補語を表す．また，第三引数の補語は修飾語でないといけない．つまり meでは，英語の第二文型 SVC と同等の文を作ることができる．

例として SFGPL で “I am happy.” を表すには次のようにする．

```
1 me ga so la 'happy'
```

このとき，meは，文型が “SVC” のときに付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，soは，動詞が無意味であることを示す単語である．この soでは，場所によって意味が変わる．このとき例文のときは，英語の be 動詞と同等の意味となる．

そして，la 'happy'は，修飾語 “happy” を表す．

2.2.4. te による構文

そして，teを使用した例を提示する．この teは 3 つの引数を持ち，第一引数が文の主語，第二引数が文の動詞，第三引数が目的語を表す．つまり，teでは，英語の第三文型の SVO と同等の文を作ることができる．

例として SFGPL で “I open the door.” を表すには次のようにする．

```
1 te ga sa 'open' fa 'door'
```

このとき，teは，文型が “SVO” のときに付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，sa 'open'は，動詞 “open” を表す．

そして，fa 'door'は，名詞 “door” を表す．

2.2.5. ti による構文

次に，tiを使用した例を提示する．この tiは 4 つの引数を持ち，第一引数が文の主語，第二引数が文の動詞，第三引数が間接目的語，第四引数が直接目的語を表す．つまり，tiでは，英語の第四文型の SVOO と同等の文を作ることができる．

例として SFGPL で “I give you a box.” を表すには次のようにする．

```
1 ti ga so ge fa 'box'
```

このとき，tiは，文型が “SVOO” のときに付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，soは，動詞が無意味であることを示す単語である．この soでは，場所によって意味が変わる．このとき例文のときは，英語の give と同等の意味となる．

そして，geは，二人称代名詞 “you” を表す．

さらに，fa 'box'は，名詞 “box” を表す．

2.2.6. tu による構文

次に，tuを使用した例を提示する．この tuは 4 つの引数を持ち，第一引数が文の主語，第二引数が文の動詞，第三引数が目的語，第四引数が目的語に対する補語を表す．第四引数の補語は名詞でないといけない．つまり，tuでは，英語の第四文型の SVOC と同等の文を作ることができる．

例として SFGPL で “I make you a teacher.” を表すには次のようにする．

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

このとき，tuは，文型が “SVOC” のときに付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，soは，動詞が無意味であることを示す単語である．この soでは，場所によって意味が変わる．このとき例文のときは，英語で使役動詞の make と同等の意味となる．

そして，geは，二人称代名詞 “you” を表す．

さらに，fa 'teacher'は，名詞 “teacher” を表す．

2.2.7. to による構文

次に，toを使用した例を提示する．この toは 4 つの引数を持ち，第一引数が文の主語，第二引数が文の動詞，第三引数が目的語，第四引数が目的語に対する補語を表す．第四引数の補語は修飾語でないといけない．つまり，toでは，英語の第四文型の SVOC と同等の文を作ることができる．

例として SFGPL で “I make you happy.” を表すには次のようにする．

```
1 to ga so ge la 'happy'
```

このとき，toは，文型が “SVOC” のときに付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，soは，動詞が無意味であることを示す単語である．この soでは，場所によって意味が変わる．このとき例文のときは，英語で使役動詞の make と同等の意味となる．

そして，geは，二人称代名詞 “you” を表す．

さらに，la 'happy'は，修飾語 “happy” を表す．

2.2.8. mi による構文

次に，miを使用した例を提示する．この miは 3 つの引数を持ち，第一引数が文の主語（持ち主），第二引数が文の動詞，第三引数が目的語（持ち物）を表す．そのため miでは “S has O” という文を表すことができる．

例として SFGPL で “I have a box.” を表すには次のようにする．

```
mi ga so fa 'box'
```

このとき，miは，所有を表す文を作る際に付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，soは，動詞が無意味であることを示す単語である．この soでは，場所によって意味が変わる．このとき例文のときは，英語で have と同等の意味となる．

さらに，fa 'box'は，名詞 “box” を表す．

2.2.9. mu による構文

次に，muを使用した例を提示する．この muは 3 つの引数を持ち，第一引数が文の主語（所属している人や物），第二引数が文の動詞，第三引数が目的語（所属先）を表す．そのため muでは “S belongs to O” という文を表すことができる．

例として SFGPL で “I belong to a school.” を表すには次のようにする．

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

このとき，muは，所属していることを表す文を作る際に付ける語である．

また，gaは，一人称代名詞 “I” を表す．

次に，soは，動詞が無意味であることを示す単語である．この soでは，場所によって意味が変わる．このとき例文のときは，英語で “belong to” と同等の意味となる．

さらに，fa 'school'は，名詞 “school” を表す．

2.2.10. その他の構文

その他の構文は，文型で示されている．

2.3. 修飾方法

2.3.1. 名詞の修飾方法

名詞の修飾方法としては，主に，修飾語を用いる方法と名詞を用いる方法の 2 種類が存在する．

名詞に対して修飾語で修飾する方法　たとえば，ある箱に対してそれが大きいことを表すとき，英語では “The box is big.” と表す．同様に SFGPL でも SVC を表す `me`を用いて，次のように表せる．

```
me fa 'box' so wan
```

このとき，`wan`は大きいという意味である．

名詞に対して名詞で修飾する方法　この方法は，英語の前置詞 “of” や日本語の助詞 “の” を使用するような場面で使われる．しかし，SFGPL では簡潔に記すことができず，このような場合でも英語の関係代名詞のような文によって修飾する（複文）．

例えば，“My box is big.” を表すには，次のような文で表す．

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

この文では，主文の主語に `mi ga so san fa 'box'`という文が入れ子的に入っている．この `mi ga so san fa 'box'`は “I have a box.” という意味であり，それが主語であることを示している．また，特にその文の中で重要な語を `san`を使うことで強調することができる．そのため，この文では `san fa 'box'`で “box” を強調している．

総合的に，直訳すると “[I have a **box**] is big.” という意味になり，結果的に “My box is big.” と同じ意味になる．

2.3.2. 動詞の修飾方法

単純な動詞の修飾方法　動詞の修飾方法は，`na`を使用して，修飾できる．この `na`では第一引数が動詞，第二引数が修飾語となる．

例えば，“I quickly run.” を表すには，次のようにする．

```
ta ga na sa 'run' la 'quickly'
```

このとき，`la 'quickly'`は “quickly” という意味である．また，`na sa 'run'la 'quickly'`では “quickly run” という意味で，動詞の “run” を修飾語の “quickly” で修飾している．

名詞句を修飾語に変換し動詞を修飾する方法　SFGPL ではある名詞句を修飾語に変換し，それを動詞を修飾することができる．これは，英語の前置詞を用いた副詞化と同様の方法である．

まず，SFGPL には，品詞間の変換ができる語があり，この場合では，名詞から修飾語に変換する `li`が使われる．またこの用法では，`li`と並行して，名詞句の意味を限定する名詞限定詞が使われ，意味の限定が行われる．

例えば，“I go to Tokyo.” を SFGPL で表すには次のようにする．

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

まず，`sa 'go'`は英語の “go” を表していて，そしてその行き先は動詞を修飾することで表される．特に，`li pun fa 'Tokyo'`の 4 語で “to Tokyo” を表す．その中の`li`が名詞から修飾語に変換する語，`pun`は場所を表す名詞限定詞である．その 2 語と場所を表す `fa 'Tokyo'`（東京）を組み合わせることで，“to Tokyo” という意味を表している．

2.3.3. 修飾詞の修飾方法

修飾詞の修飾方法は，`ka`を使用して表現する．この `ka`では第一引数の修飾語に対して第二引数の修飾語が修飾する．

例えば，SFGPL で “Your box is a little big.” を表すには次のようにする．

```
me mi ge so san fa 'box' so ka wan la 'little'
```

このとき, `wan`(= “big”) に `la 'little'`(= “a little”) が修飾しているため, この`ka wan la 'little'`では “a little big” という意味となる．

2.4. 単語集

English	SFGPL
I	ga
run	sa ‘run’
student	fa ‘student’
happy	la ‘happy’
open	sa ‘open’
door	fa ‘door’
you	ge
box	fa ‘box’
teacher	fa ‘teacher’
school	fa ‘school’
big	wan
quickly	la ‘quickly’

English	SFGPL
go	sa ‘go’
Tokyo	fa ‘Tokyo’
a little	la ‘little’

Part II.
SFGPLの構文

3. 文型

3.1. 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
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

3.2. Noun.do (ta)

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'
```

3.3. Noun.eq (ma)

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'
```

3.4. Noun.haveP (me)

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'
```

3.5. Noun.doT (te)

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'
```

3.6. Noun.give (ti)

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'
```

3.7. Noun.makeN (tu) と Noun.makeM (to)

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'
```

3.8. Noun.have (mi)

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'
```

3.9. Noun.belong (mu)

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'
```

3.10. Noun.gt (mo)

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．

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

3.11. Noun.hearSay (moa)

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．

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

3.12. 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．

```
1 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．

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

3.12.1. 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．

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

3.13. 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’
big	wan
yours	sen ge
book	fa ‘book’
Japan	fa ‘Japan’
in East Asia	li fun pun me fa ‘Asia’ so la ‘east’

4. 否定文と否定表現

4.1. 否定文

通常の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．この文の全体を否定し，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．

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

4.2. 動詞の否定

SFGPL では，文章全体を否定する以外に動詞だけを否定することもできる．文章全体を否定する場合と動詞だけを否定する場合は，意味が異なる場合がある．特に英語などの衛星枠付け言語では，それぞれの意味の解釈が異なる場合がある．

例えば，次のように “I don’t make a table.” という場合は，文全体の否定と動詞のみの否定の意味がほとんど同義である．

文全体	pa te ga sa ‘make’ fa ‘table’
動詞のみ	te ga pa sa ‘make’ fa ‘table’

また，次のように “I didn’t run to my school.” では，文全体の否定と動詞のみの否定の意味が異なる．

文全体	di pa ta ga na sa ‘run’ li pun mu ga so san fa ‘school’
動詞のみ	di ta ga na pa sa ‘run’ li pun mu ga so san fa ‘school’

文章全体の否定の場合では，「私は学校に走って行った」以外の事象すべてを表している，つまり，「私は学校に歩いて行った」や「私は学校に行かなかった」などの意味も含意している．

しかし，動詞のみの否定の場合では，「私は学校に行く」事象の中で「走る」以外の行動をしたという意味になる．つまり，「私は学校に歩いて行った」などの他の手段である場合は含意するが，「私は学校に行かなかった」は含意しない．

一方で日本語などの動詞枠付け言語では，「走って行く」などの複合動詞によって表現するため，このような意味の差異がなくなる傾向がある．このときの「走って行く」という複合動詞では，英語と違い「走る」という動作の手法と「行く」という動作の結果が含まれている．

4.3. 修飾語の否定形

ある修飾語において，`ke`を付けることによって，その修飾語の対義語を表すことができる．

例えば，“big” という意味の `wan`に対する対義語の “small” を表す場合，`ke wan`とすることで表すことができる．

“My table is small.” を SFGPL で表すと次のようになる．

```
me mi ga so san fa 'table' so ke wan
```

4.4. 単語集

English	SFGPL
I	ga
table	fa ‘table’
make	sa ‘make’
run	sa ‘run’
my school	mu ga so san fa ‘school’
big	wan

5. 疑問文

5.1. 諾否疑問文

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．

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

このような疑問文を返答する場合は次のように，Bool.B2N`pis`を使用して命題が正か偽かを示すことで表す．

```
pis mi ga so pen gi pos
pis mi ga so pen gi pas
```

5.2. 疑問詞疑問文

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．疑問詞は疑問代名詞 `wa`と名詞限定詞を組み合わせて表す．

“Who has a table?” は次のように表す．

```
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
```

5.3. 単語集

English	SFGPL
you	ge
table	fa ‘table’
true	pos
false	pas
who	ben wa
what	pen wa

6. 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'
```

6.1. 単語集

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

7. 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．

7.1. 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
```

7.2. 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．

7.2.1. 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．

7.2.2. 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'
```

7.3. 名詞による名詞の修飾

ある名詞 X と Y において，Y が X を修飾するとき，日本語では “Y の X”，英語では “Y X” または “X of Y” と表されるが SFGPL では，主に 3 種類の用法を使い分けて使用する．SFGPL では，先述のように，従属節で修飾をすることが多いが，名詞を名詞で修飾する場合も例外ではない．そのため名詞の修飾方法は，`ma`，`mi`と `mu`で使い分けられる．

7.3.1. Noun.eq (ma)

まず，`ma`では，主に修飾語と被修飾語が同等のもののときに使われる．例えば “This pen is big.” を SFGPL で表すには次のようにする．

```
me ma gu so san fa 'pen' so wan
```

このとき，“this” と “pen” は同等のものを指している．そのため，`ma`が使われる．

7.3.2. Noun.have (mi)

次に，`mi`では，主に何かが何かを持っているときに使われる．SFGPL で “My pen is big.” を表すには次のようにする．

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

7.3.3. Noun.belong (mu)

また，`mu`では，主に何かがなにかに所属しているときに使われる．SFGPL で “My school is big.” を表すには次のようにする．

```
me mu ga so san fa 'school' so wan
```

7.4. 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’
this	gu
pen	fa ‘pen’
school	fa ‘school’

8. 動詞の活用方法

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．

8.1. 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．

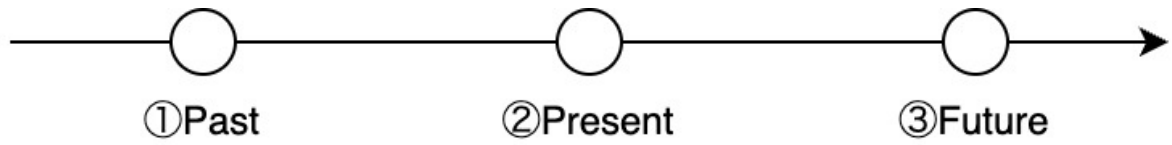

Figure 2: BasingPoint

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．

また，②の現在形は，何も付属しないことで，通常は現在のことを表す．しかし，本来は不定時制であり，特に時制を必要としない場合にも使われる．

8.1.1. 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

8.2. 動作の時間軸に関する相

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．

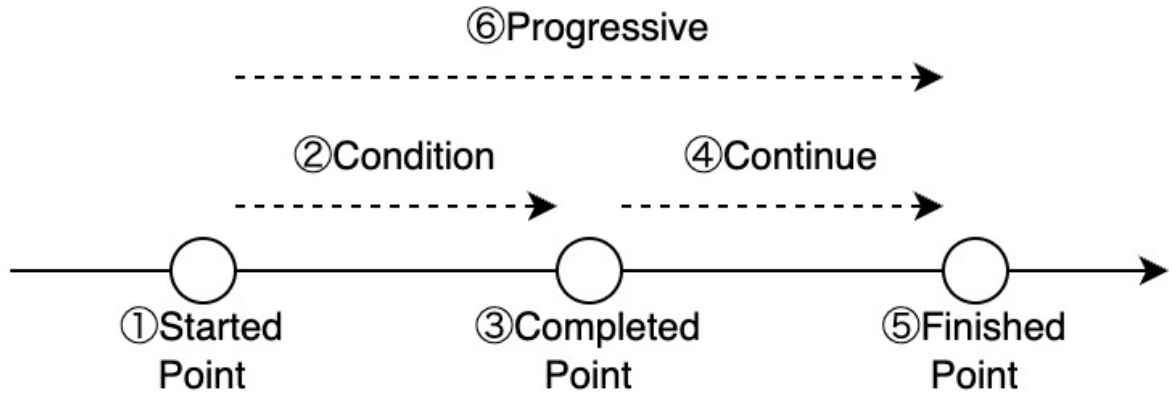

Figure 3: ProgressiveForm

“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’
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’

①起動相，③完結相，⑤終了相では，ある動作に対しての瞬間の一点だけを表す．

②経過相，④継続相，⑥進行相は，ある動作に対しての期間を表す．⑥進行相は②経過相と④継続相が含まれた，不明瞭な期間を表す．また，一部の動詞では，それぞれとの相との間が一瞬であり，ほとんど区別できない場合がある．

これらの相は，現在形以外にも，過去形，未来形にできる．“I begin wear a dress.” を過去形，未来形にすると次のようになる．

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

また，原則としてこれらの相単体では，ある時点にフォーカスをした動作を表している．特に，その時点より過去からその動作が続いている場合を強調して示すためには，それらの相に加えて完了形を使用し表現する．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．

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

8.2.1. 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'
```

8.3. 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．

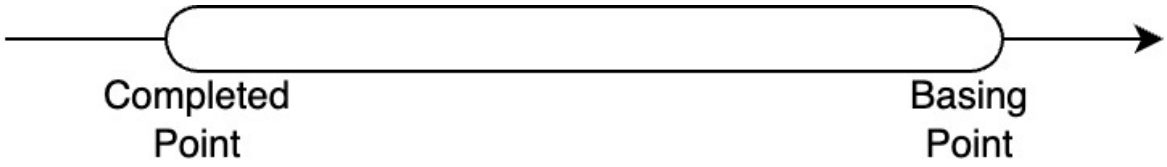

Figure 4: PerfectForm

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．

8.4. SFGPL の時間表現のまとめ

SFGPL の時間表現に関しては，次の表のようなものが存在する．

基本時制	拡張時制	完了相	完結相
-	-	-	-
di	bak	nu	tak
du	bik		tek
	bok		tik
			tuk
			tok
			ni

このように，SFGPL では 3×4×2×7=168 通りの時間表現が存在し，あらゆる場面に対して表現することが可能である．

8.5. 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．

```
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．

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

8.6. 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．

8.7. 単語集

English	SFGPL
I	ga
live	sa ‘live’

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

9. 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．

9.1. 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'
```

9.1.1. 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

English	Meaning	SFGPL
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
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

9.2. 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．

9.2.1. 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
```

9.2.2. 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 ki 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 'bag' so ka ki wan li fun mu ga so san fa 'class'
```

また，「N 番目に X な」を表現するとき，修飾語に数値を付与して `ka X li N`のように表す．序数を使用した “My bag is the second biggest in my class.” は次のように表現する．

```
me mi ga so san fa 'bag' so ka ki ka wan li mal pil li fun mu ga so san
    fa 'class'
```

9.2.3. 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 'bag' sen lan gi so ka wan wen
```

9.3. 通時的な文

習慣や周期的な事柄，不変の事実などの恒常的な事柄や事実を表すには，現在と同様に時制をつけないことで表現する．

SFGPL で “I cook every day.” を表すには次のようにする．

```
ta ga na sa 'cook' li pin me fa 'day' so la 'every'
```

また，SFGPL で “The Earth revolves around the Sun.” を表すには次のようにする．

```
ta fa 'Earth' na sa 'revolve' li tun tin fa 'Sun'
```

そして，SFGPL で “English is spoken all over the world.” を表すには次のようにする．

```
ta fa 'English' na ne sa 'speak' li fun dan fa 'world'
```

9.4. 存在を表すときの文法

ただ単に存在することだけを表す文を作成するときには，genを使用して表す．これは英語の There is/are の構文と同じ意味となる．例えば “There is a book on this table.” は次のように表せる．

```
ta fa 'book' na gen li pun ma gu so fa 'table'
```

9.5. 主題優勢言語的な文法

日本語や中国語，朝鮮語，インドネシア語などの主に東アジアの言語によく見られる，主題優勢言語のような文を作成することができる．主題優勢言語は，通常の主語の他に，文の主題を提示できるような文法が存在する言語である．これにより，主題と主語の両方を含む文を容易に表現できる．SFGPL では，東アジア諸言語のような明確な方法ではないが，簡易的に主題を含む文を作成できる．

9.5.1. 主題もしくは主語の片方を含む文

主題もしくは主語の片方を含む文は，文型と同様に文を構築する．

9.5.2. 主題と主語の両方を含む文

主題と主語の両方を含む文は，次のように表現する．このときの “T” は主題，“C” はコメント（主題を説明する文や単語等）で構成される．

```
ma T so C
```

例として，日本語の「象は鼻が長い」を SFGPL で表現するには次のようになる．

```
ma fa '象' so me fa '鼻' so la '長い'
ma fa 'elephant' so me fa 'nose' so la 'long'
```

9.6. 単語集

English	SFGPL
I	ga
go	sa ‘go’
to Tokyo	li pun fa ‘Tokyo’

English	SFGPL
bag	fa ‘bag’
big	wan
yours(possessive)	sen ge
my class	mu ga so san fa ‘class’
his(possessive)	sen lan gi
cook	sa ‘cook’
every day	me fa ‘day’ so la ‘every’
the Earth	fa ‘Earth’
revolve	sa ‘revolve’
the Sun	fa ‘Sun’
English	fa ‘English’
speak	sa ‘speak’
all over the world	li fun dan fa ‘world’
book	fa ‘book’
on this table	li pun ma gu so fa ‘table’
象	fa ‘象’
鼻	fa ‘鼻’
長い	fa ‘長い’
elephant	fa ‘elephant’
nose	fa ‘nose’
long	la ‘long’

Part III.
SFGPLの単語

10. 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．

10.1. 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．ただし，半角ダブルクオーテーション (") と半角スペース ( ) は使用できない．また，半角シングルクォーテーション (') は，借用語であることを示す記号であるため，最初と最後に付けたい場合は注意が必要である．

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’

10.1.1. 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．また，この資料では基本的に借用語は英語から借用している．

10.1.2. 借用語の明示方法

借用語がどこの言語から借用されたかを明示するために次のような単語が存在する．

Type	Word
Noun	foa
Verb	soa
Modifier	loa

名詞　英語の “language” という名詞の単語を借用するには次のようにする．

```
1 foa 'language' 'English'
```

動詞　英語の “go” という動詞の単語を借用するには次のようにする．

```
soa 'go' 'English'
```

修飾語　英語の “big” という修飾語の単語を借用するには次のようにする．

```
loa 'big' 'English'
```

10.2. 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

10.2.1. 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．

10.3. 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．

10.3.1. 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

10.3.2. 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．

10.4. 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’

10.5. 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

10.6. 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, 変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．

11. 修飾語

11.1. 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．

11.2. 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．

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

11.3. 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

11.4. 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．

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

また，punは，場所を表す限定詞である．

11.5. 単語集

English	SFGPL
I	ga

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

12. 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
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．

12.1. 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．

12.2. 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．

```
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．

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

12.3. 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．

```
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'
```

12.4. 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

English	SFGPL
sleep	sa ‘sleep’
boy	fa ‘boy’
that	go
destroy	sa ‘destroy’

13. 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．

13.1. 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’

14. 代名詞

14.1. 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

14.2. 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．

14.2.1. 疑問詞

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

14.2.2. 代名詞の複数形

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．

人称代名詞の除括性　特に一人称代名詞の複数形では，話し手や聞き手を含むか否かで区別する場合がある．SFGPL では直接これらを区別をすることはできないが，次のようにすることで区別することができる．

	聞き手を含む	聞き手を含まない
話し手を含む	`don ba ge ga`	`don ba ga gi`
話してを含まない	`don ge`	`don gi`

14.2.3. 三人称代名詞の活用例

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

	English	SFGPL
事物	it	pen gi

14.2.4. 所有代名詞と再帰代名詞

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

15. 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．

15.1. 単語集

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

16. 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．

16.1. 単語集

English	SFGPL
I	ga
see	sa ‘see’
sea	fa ‘sea’

English	SFGPL
swim	sa ‘swim’

17. 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．

17.1. 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．このような文では，全体が True として継承される．

```
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．

たとえば，`True OR False`を表すには次のようになる．

```
be pos pas
```

LangObj には通常の IFELSEbiの他に，logicIFELSEjaが存在する．この単語により，条件を満たすかどうかで内部的に実行する文章（単語）を変えることができる．“If true, I am a student.” を表すには次のようにする．

```
ja pos ma ga so fa 'student' pa ma ga so fa 'student'
```

17.2. 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
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
foas A	BoolList(A) の長さを取得する
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．

```
1  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
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 pas pos pos	3.1415927410125732

浮動小数点は，IEEE 754 の半精度，単精度，倍精度，四倍精度に対応している．そのため，それぞれ 16bit，32bit，64bit，128bit で表す必要がある．

それぞれの精度で 1/3 を表すには次のようになる．まず，16 進数で表すと次のようになる．

Type	HEX
Half	`3555`
Single	`3eaa aaab`
Double	`3FD5 5555 5555 5555`
Quadruple	`3ffd 5555 5555 5555 5555 5555 5555 5555`

これを SFGPL に変換すると次のようになる．

Type	SFGPL
Half	`tis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fas pas pas pos pos pas pos pas pos pas pos pas pos pas pos pas pos`
Single	`tis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fas pas pas pos pos pos pos pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pos`
Double	`tis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fis fas pas pas pos pos pos pos pos pos pos pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos`

Type	SFGPL
Quadruple	tis fas pas pas pos pos pos pos pos pos pos pos pos pos pos pos pas pos pos pas pos pas pos pas pos pas pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos pas pos

17.3. BoolList の日時表現

BoolList を利用して，Unix 時間に基づく日時表現をすることができる．日時表現はその精度によって以下の 3 種類存在する．

SFGPL	Type	Unit
das	yyyy-mm-dd	Day
des	yyyy-mm-dd HH:MM:SS	Second
dis	yyyy-mm-dd HH:MM:SS.nnnnnnnnn	Nano Second

これらの表現は `1970-01-01 00:00:00.000000000`が基準で，それぞれ日単位，秒単位，ナノ秒単位での差分によって日時を表す．また，これらは UTC 時間が基準となっている．

例えば，`2024-09-19 09:27:27`を `des`で表すには次のようにする．

まず，この時間の Unix 時間は `1726738047`である．これを 2 進数に変換すると，`0110 0110 1110 1011 1110 1110 0111 1111`となる．そのため BoolList に変換すると次のようになる．

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

さらに，`des`を使用して日時に変換すると次のようになる．

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

これによって，`2024-09-19 09:27:27`を SFGPL で表すことができる．

17.4. 単語集

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

18. 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
foat A	LangList(A) の長さを取得する
tat A B C	LangList を使用した繰り返し用の関数
tet A B	LangList の全要素に対して一定の処理を行う関数

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 append を使用して作成する一例である．

```
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
```

18.1. LangList での繰り返し処理

LangList を繰り返し処理するための `tat`を使用することで LangList を使用した繰り返し処理を行うことができる．以下に表す *x* はすべて同じ LangList で，While 関数内の処理で使用される変数となっている．

第一引数 A は，ループの初期値を設定する．この A の値が，最初に *x* に代入される．

第二引数 B は，定義済み LangFunc の名前を設定する．この B の名前の関数は，ループ上での条件文となり，この関数の引数には *x* が代入される．また，この関数の戻り値の LangList の 0 番目の要素は，条件を満たすかを表す Bool 型とし，この値が True の場合はループを継続する．

第三引数 C は，定義済み LangFunc の名前を設定する．この C の名前の関数は，繰り返し実行される処理内容となり，この関数の引数には *x* が代入される．そして，この関数では更新された *x* を戻り値に設定する．

ループ終了時は最終的な *x* の結果が出力される．

次は，tatを使用した例である．

```
1 pat fa 'condition_func' fit fat sal tel mal pol fet pit tol mal pal
2 pat fa 'process_func' fit fit fit fat tal fet pit tol mal pal mal pel
     tal fet pit tol mal pel mel pel pal til fet pit tol mal pil mal pil
3 tat fit fit fit fat mal pal mal pal mal pel fa 'condition_func' fa '
     process_func'
```

まず 1 行目では，条件文の関数の設定を行っている．この条件文の関数は “condition_func” とし，4-x[0]>=0が True の間は，ループを実行するように定義している．

2 行目では，処理文の関数の設定を行っている．この処理文の関数は “process_func” とし，それぞれの要素の更新を行う．更新する内容は，[x[0]+1,x[1]+10,x[2]*2]としている．

3 行目で，実際に繰り返しを実行している．このとき初期値として，[0,0,1]を与えている．

18.2. LangList の map 関数

LangList のすべての要素に対して，一定の処理を行う関数 tetが存在する．このとき，第一引数に適応する LangList A，第二引数に一定の処理を行うための関数名 B を指定する．

このとき，B の関数には，LangList 型で [それぞれの要素のデータ, その要素のindex(NumberList), LangList A]が引数として渡される．また，B の関数を実行した結果の LangList[0] の値が，新たな要素の値として使われる．

次に，tetを使用して，全要素に 1 を足すためには次のようにする．

```
1 pat fa 'map_func' fit fat tal fet pit tol mal pal mal pel
2 tet fit fit fit fat mel pel pal mel pel pel mel pel pil fa 'map_func'
```

1 行目で，処理用の関数を設定している．この処理内容は，それぞれの要素のデータに対して，1 を足す処理を行っている．

2 行目で，実際に [10,11,12]を代入し，すべての要素に対して 1 を足す処理を実行している．このとき，[10,11,12]を表すためには，fit fit fit fat mel pel pal mel pel pel mel pel pilとすることで表現できる．

18.3. 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’

19. 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．

```
1 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．

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

20. LangVar

SFGPL では LangList を格納する変数を作成できる．

単語	説明
bat A B	A という名前の変数に LangList B を代入する
bot A	A という名前の変数を取得する

var1 という名前の変数に LangList["apple","banana"]を格納するには次のようにする．

```
bat fa 'var1' fit fit fat fa 'apple' fa 'banana'
```

また，var1 を取得するには次のようにする．

```
bot fa 'var1'
```

21. 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．

21.1. 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

21.2. 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
foal A	NumberList(A) の長さを取得する

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
```

21.2.1. 数値計算

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

	Python	SFGPL
Addition	`A+B`	tal A B
Subtraction	`A-B`	tel A B
Multiplication	`A*B`	til A B
Division	`A/B`	tul A B
Power	`A**B`	dal A B
Int Division	`A//B`	del A B
Remainder	`A%B`	dil A B
Minus	`-A`	sel A
Absolute value	`abs(A)`	sil A

21.2.2. BoolList と NumberList の相互変換

次の表のように BoolList と NumberList を相互に変換する関数が存在する．

Type	SFGPL	from	to
Int	tol	NumberList	BoolList
Int	tos	BoolList	NumberList
Float	dol	NumberList	BoolList
Float	dos	BoolList	NumberList

整数型における相互変換　整数として相互変換する関数 `tol`と `tos`が存在する．これらの変換で扱われる数値は，BoolList を整数型 (`tes`) として見做される．つまり，このときの BoolList の値は，2 進数の 2 の補数表現方法と同等である．これらの値は，NumberList によって，四則演算等の数

値計算が行われた場合も適応できる．ただし，NumberList が除算結果などにより実数となっている場合は，変換ができずエラーとなる．

浮動小数点型（実数）における相互変換　浮動小数点（実数）として相互変換する関数 `dol`と `dos` が存在する．これらの変換で扱われる数値は，BoolList を浮動小数点型 (`tis`) として見做される．つまり，このときの BoolList の値は，IEEE754 における半精度，単精度，倍精度，四倍精度の浮動小数点表現方法が用いられる．また，NumberList から BoolList に変換する際には，64bit の倍精度浮動小数点数として変換され，BoolList に格納される．

21.2.3. 実数の扱い方

実数を扱うためには，NumberList の除算 (`tul`) を使用する方法と，BoolList で浮動小数点数を表しそれを NumberList に変換する方法がある．

例えば 3.14 を除算によって表すには次のようにする．

```
1 tul mil pul pel pol mil pel pal pal
```

同様に，3.14 を倍精度浮動小数点で表す場合は次のようにする．

```
1 dos fos fos fos mos pas pos pas pas pas pas pas pas mos pas pas pas pas
      pos pas pas pos fos mos pas pas pas pos pos pos pos pas mos pos pas
      pos pos pos pas pas pas fos fos mos pas pos pas pos pas pas pas pos
      mos pos pos pos pas pos pas pos pos fos mos pos pas pas pas pas pos
      pas pos mos pas pas pas pos pos pos pos pos
```

21.2.4. 正の数の判定

NumberList で正の数かを判定するには，`sal`を使用する．これによって，SFGPL の Bool 型が出力され，0 以上の数の場合が `pos`と同値になる．例えば，4 と-4 の 2 つの場合において正の整数か判定するには次のようにする．

```
1 sal mal pol
2 sal sel mal pol
```

21.3. 単語集

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

Part IV.
付録

22. 英語由来以外の借用語を使う例

SFGPL では英語由来以外の借用語を使うこともできる．その場合，基本的には英語の場合と同様の使い方でよい．しかし，語順は変えられないため，元の言語の語順と異なる可能性がある．

22.1. 日本語由来の借用語

例えば，「私はりんごを持っている。」という文を作るには次のようにする．

```
mi ga so fa 'りんご'
```

また，複文が含んでいる文「私の鞄は赤い。」という文は次のようにする．

```
me mi ga so san fa '鞄' so la '赤い'
```

22.2. エスペラント由来の借用語

エスペラントの語を借用語として使用する場合は，原則として品詞語尾を除いた形を使用することを推奨する．

例えば，“Mi havas pomon.” という文を作るには次のようにする．

```
mi ga so fa 'pom'
```

また，“Mia sako estas ruĝa.” という文は次のようにする．

```
me mi ga so san fa 'sak' so la 'ruĝ'
```

23. 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.
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.

SFGPL	Python	Translation
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 office workers.
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 office workers.
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 office workers.
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.

SFGPL	Python	Translation
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.
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.

SFGPL	Python	Translation
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.
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.

SFGPL	Python	Translation
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.
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.

SFGPL	Python	Translation
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?
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?

SFGPL	Python	Translation
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.
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.

SFGPL	Python	Translation
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.
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.

SFGPL	Python	Translation
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.
di pa te ga sa ‘know’ di te ben we ni sa ‘call’ ga	`Phrase.past(LangObj.NOT(Noun.doT(Pronoun.I(),Verb("'know'"),Phrase.past(Noun.doT(DeterminerN.human(Pronoun.indefinite()),Verb.progressive(Verb("'call'")),Pronoun.I())))))`	I didn’t know that someone was calling me.
pe di pa ta ga na nak sa ‘go’ li pun ma go so san fa ‘event’ pa mi ga so fa ‘car’	`LangObj.Because(Phrase.past(LangObj.NOT(Noun.do(Pronoun.I(),Verb.add(DeterminerV.Possible(Verb("'go'")),Modifier.N2M(DeterminerN.place(Noun.eq(Pronoun.distal(),Verb.none(),DeterminerN.stressed(Noun("'event'")))))))),LangObj.NOT(Noun.have(Pronoun.I(),Verb.none(),Noun("'car'"))))`	I could not go to that event because I do not have a car.

SFGPL	Python	Translation
di to ge na so li pon di te ge soa ‘eat’ ‘English’ te ga soa ‘make’ ‘English’ san foa ‘cake’ ‘English’ fa ‘Mary’ loa ‘sad’ ‘English’	`Phrase.past(Noun.makeM (Pronoun.you(),Verb. add(Verb.none(), Modifier.N2M(DeterminerN.reason(Phrase.past(Noun.doT(Pronoun.you(),Verb. borrowed("'eat'","' English'"),Noun.doT(Pronoun.I(),Verb. borrowed("'make'","' English'"),DeterminerN .stressed(Noun. borrowed("'cake'","' English'")))))))),Noun ("'Mary'"),Modifier. borrowed("'sad'","' English'")))`	You made Mary sad, for you ate the cake I made.
di te ga na soa ‘meet’ ‘English’ li pin di te ga soa ‘go’ ‘English’ fi soa ‘shop’ ‘English’ fa ‘Mary’	`Phrase.past(Noun.doT( Pronoun.I(),Verb.add( Verb.borrowed("'meet'" ,"'English'"),Modifier .N2M(DeterminerN.time( Phrase.past(Noun.doT( Pronoun.I(),Verb. borrowed("'go'","' English'"),Noun.V2N( Verb.borrowed("'shop'" ,"'English'")))))), Noun("'Mary'")))`	I met Mary when I went shopping.

SFGPL	Python	Translation
di te ga na soa ‘meet’ ‘English’ li pin di te ga soa ‘go’ ‘English’ fi soa ‘shop’ ‘English’ fa ‘Mary’	`Phrase.past(Noun.doT(Pronoun.I(),Verb.add(Verb.borrowed("'meet'","'English'"),Modifier.N2M(DeterminerN.time(Phrase.past(Noun.doT(Pronoun.I(),Verb.borrowed("'go'","'English'"),Noun.V2N(Verb.borrowed("'shop'","'English'")))))),Noun("'Mary'")))`	I met Mary when I went shopping.
ta fa ‘apple’ na gen li pun ma gu so fa ‘table’	`Noun.do(Noun("'apple'"),Verb.add(WordV.exist(),Modifier.N2M(DeterminerN.place(Noun.eq(Pronoun.proximal(),Verb.none(),Noun("'table'"))))))`	There is an apple on this table.
ta don fa ‘apple’ na gen li pun ma gu so fa ‘table’	`Noun.do(DeterminerN.plural(Noun("'apple'")),Verb.add(WordV.exist(),Modifier.N2M(DeterminerN.place(Noun.eq(Pronoun.proximal(),Verb.none(),Noun("'table'"))))))`	There are apples on this table.
di ti ga so mi ga so don fa ‘student’ don fa ‘apple’	`Phrase.past(Noun.give(Pronoun.I(),Verb.none(),Noun.have(Pronoun.I(),Verb.none(),DeterminerN.plural(Noun("'student'"))),DeterminerN.plural(Noun("'apple'"))))`	I gave my students apples.

SFGPL	Python	Translation
di te ben we sa ‘eat’ fa ‘apple’	`Phrase.past(Noun.doT(DeterminerN.human(Pronoun.indefinite()),Verb("'eat'"),Noun("'apple'")))`	Someone ate an apple.
ta len gi na sa ‘sing’ la ‘beautifully’	`Noun.do(DeterminerN.female(Pronoun.he()),Verb.add(Verb("'sing'"),Modifier("'beautifully'")))`	She sings beautifully.
di ta don gi na sa ‘arrive’ la ‘late’	`Phrase.past(Noun.do(DeterminerN.plural(Pronoun.he()),Verb.add(Verb("'arrive'"),Modifier("'late'"))))`	They arrived late.
ta lan gi na sa ‘work’ la ‘hard’	`Noun.do(DeterminerN.male(Pronoun.he()),Verb.add(Verb("'work'"),Modifier("'hard'")))`	He works hard.
ta ma bin gi so san fa ‘cat’ ni sa ‘sleep’	`Noun.do(Noun.eq(DeterminerN.animal(Pronoun.he()),Verb.none(),DeterminerN.stressed(Noun("'cat'"))),Verb.progressive(Verb("'sleep'")))`	The cat is sleeping.
te ga sa ‘like’ fa ‘coffee’	`Noun.doT(Pronoun.I(),Verb("'like'"),Noun("'coffee'"))`	I like coffee.
ta len gi na sa ‘run’ la ‘fast’	`Noun.do(DeterminerN.female(Pronoun.he()),Verb.add(Verb("'run'"),Modifier("'fast'")))`	She runs fast.

SFGPL	Python	Translation
mi don ga so fa ‘plan’	`Noun.have(DeterminerN.plural(Pronoun.I()),Verb.none(),Noun("'plan'"))`	We have a plan.
te lan gi sa ‘play’ fa ‘guitar’	`Noun.doT(DeterminerN.male(Pronoun.he()),Verb("'play'"),Noun("'guitar'"))`	He plays the guitar.
te len gi sa ‘play’ fa ‘guitar’	`Noun.doT(DeterminerN.female(Pronoun.he()),Verb("'play'"),Noun("'guitar'"))`	She plays the guitar.
ta ma gi so san fa ‘phone’ ni sa ‘ring’	`Noun.do(Noun.eq(Pronoun.he(),Verb.none(),DeterminerN.stressed(Noun("'phone'"))),Verb.progressive(Verb("'ring'")))`	The phone is ringing.
ta mi ga so san fa ‘phone’ ni sa ‘ring’	`Noun.do(Noun.have(Pronoun.I(),Verb.none(),DeterminerN.stressed(Noun("'phone'"))),Verb.progressive(Verb("'ring'")))`	My phone is ringing.
te don gi ni sa ‘watch’ fa ‘movie’	`Noun.doT(DeterminerN.plural(Pronoun.he()),Verb.progressive(Verb("'watch'")),Noun("'movie'"))`	They are watching a movie.

24. 辞書

詳細は dict.csv を参照.

ID	word	func	How to use	Japanese	English
0	pa	`LangObj.NOT`	pa A	A でない	not A
1	pe	`LangObj.Because`	pe A B	A なぜならば B	A because B
2	pi	`LangObj.IF`	pi A B	もし A ならば B である	if A, B
3	pu	`LangObj.So`	pu A B	A だから B	A so B
4	po	`LangObj.But`	po A B	A しかし B	A but B
5	ba	`LangObj.AND`	ba A B	A かつ B	A and B
6	be	`LangObj.OR`	be A B	A または B	A or B
7	bi	`LangObj.IFELSE`	bi A B C	もし A ならば B である，そうでなければ C である	If A, B, otherwise C
8	bu	`LangObj.NAND`	bu A B	A かつ B でない	A nand B
9	bo	`LangObj.NOR`	bo A B	A または B でない	A nor B
10	ja	`LangObj.logicIFELSE`	ja A B C	もし A ならば B を出力，そうでなければ C を出力する	If A, output B, otherwise output C
11	fa	`Noun`	fa A	A は存在する	There be A / A exist
12	foa	`Noun.borrowed`	foa A B	A という名詞の単語を B という言語から借用する	Borrowing noun words called A from the language B

ID	word	func	How to use	Japanese	English
13	fi	Noun.V2N	fi A	動詞から名詞に変換する	Converting verbs to nouns.
14	fu	Noun.M2N	fu A	修飾語から名詞に変換する	Converting modifiers to nouns.
15	fo	Noun.none	fo	品詞が名詞の無意味の語を作る	The part of speech makes the noun nonsensical
16	ma	Noun.eq	ma A F B	A は B で (F) ある	A F(Verbs such that A means equal to B) B
17	me	Noun.haveP	me A F B	A が B という性質が (F) ある	A F(Verbs such that A means equal to B) B
18	mi	Noun.have	mi A F B	A は B を所有している/A は B を含んでいる	A have B/ A include B
19	mu	Noun.belong	mu A F B	A は B に所属している/A は B に含まれている	A belongs to B/ A is included in B
20	mo	Noun.gt	mo A F B C	A は C より（B という比較基準で）大きい	A is more B than C

ID	word	func	How to use	Japanese	English
21	moa	Noun.hearSay	moa A F B C	B という内容を C という情報源から，A は F する	A(Subject) F(Verb) that B(Content) according to C(Source)
22	ta	Noun.**do**	ta A F	A は F（〜する）	A F(do)
23	te	Noun.doT	te A F B	A は B を F（〜する）	A F(do) B
24	ti	Noun.give	ti A F B C	A は B に C を F（〜与える）	A F(give) B C
25	tu	Noun.makeN	tu A F B C	A は B を C の状態に F（〜する）	A F(make) B C[Noun]
26	to	Noun.makeM	to A F B C	A は B を C の状態に F（〜する）	A F(make) B C[Modifier]
27	da	Phrase.interrogative	da A	A（〜ですか）[疑問]	A(interrogative form)
28	de	Phrase.imperative	de A	A（〜しろ）[命令]	A(imperative form)
29	di	Phrase.past	di A	A（〜した）[過去]	A(did)
30	du	Phrase.future	du A	A（〜するだろう/する予定である）[未来]	A(will do / be going to do)
31	sa	Verb	sa A	A（〜する）行為が存在する	There is an act of A

ID	word	func	How to use	Japanese	English
32	soa	`Verb.borrowed`	soa A B	A という動詞の単語を B という言語から借用する	Borrowing verb words called A from the language B
33	si	`Verb.M2V`	si A	修飾語から動詞に変換する	Converting from modifiers to verbs.
34	su	`Verb.N2V`	su A	名詞から動詞に変換する	Converting from nouns to verbs.
35	so	`Verb.none`	so	品詞が動詞の無意味の語を作る	The part of speech makes the verb nonsensical
36	na	`Verb.add`	na A B	A（〜する）に B（〜く/〜に）[副詞]という意味を付加する	Adding the meaning of B to A [verb]
37	ne	`Verb.passive`	ne A	A（〜される）[受動]	A(be done)
38	ni	`Verb.progressive`	ni A	A（〜している）[継続]	A(be doing)
39	nu	`Verb.perfective`	nu A	A（〜したことのある）[経験/完了/結果/継続]	A(have done)

ID	word	func	How to use	Japanese	English
40	la	`Modifier`	la A	A（〜な/〜の/〜に/〜く）[形容詞/副詞]という修飾語が存在する	There is a modifier [adjective/adverb] called A
41	loa	`Modifier.borrowed`	loa A B	A という修飾詞の単語を B という言語から借用する	Borrowing modifier words called A from the language B
42	li	`Modifier.N2M`	li A	A（〜の/〜に/〜で）	(of/ in/ at/ on/ by/ with etc.) A
43	lu	`Modifier.V2M`	lu A	動詞から修飾語に変換する	Converting verbs to modifiers.
44	lo	`Modifier.none`	lo	品詞が修飾語の無意味の語を作る	The part of speech makes the modifier nonsensical
45	ka	`Modifier.add`	ka A B	A に B[副詞]という意味を付加する	Adding the meaning of B to A [modifier]
46	ke	`Modifier.Neg`	ke A	A でなく	not A
47	ki	`Modifier.Very`	ki A	とても A である	very A
48	pan	`DeterminerN.biology`	pan A	名詞を「A が何らかの人や生物」と限定する	Limit nouns to ‘A is some kind of person or creature’

ID	word	func	How to use	Japanese	English
49	pen	`DeterminerN.thing`	pen A	名詞を「A が何らかの物や概念」と限定する	Limit nouns to 'A is some object or concept'
50	pin	`DeterminerN.time`	pin A	名詞を「A が何らかの時間」と限定する	Limit a noun to 'A is some time'
51	pun	`DeterminerN.place`	pun A	名詞を「A が何らかの場所」と限定する	Limit a noun to 'A is some place'
52	pon	`DeterminerN.reason`	pon A	名詞を「A が何らかの理由」と限定する	Limit a noun to 'A is some reason'
53	ban	`DeterminerN.method`	ban A	名詞を「A が何らかの方法や道具や手段」と限定する	Limit nouns to 'A is some way or tool or means'
54	ben	`DeterminerN.human`	ben A	名詞を「A が何らかの人間」と限定する	Limit nouns to 'A is some kind of human being'
55	bin	`DeterminerN.animal`	bin A	名詞を「A が何らかの動物」と限定する	Limit nouns to 'A is some kind of animal'
56	bun	`DeterminerN.plant`	bun A	名詞を「A が何らかの植物」と限定する	Limit the noun to 'A is some kind of plant'

ID	word	func	How to use	Japanese	English
57	bon	DeterminerN.material	bon A	名詞を「A が何らかの材料」と限定する	Limit the noun to 'A is some kind of material'
58	fan	DeterminerN.start	fan A	名詞を「A が何らかの始点」と限定する	Limit a noun to 'A is some starting point'
59	fen	DeterminerN.end	fen A	名詞を「A が何らかの終点」と限定する	Limit a noun to 'A is some end point'
60	fin	DeterminerN.section	fin A	名詞を「A が何らかの区間」と限定する	Limit a noun to 'A is some interval'
61	fun	DeterminerN.In	fun A	名詞を「A が何らかの中」と限定する	Limit nouns to 'A is in some'
62	fon	DeterminerN.Out	fon A	名詞を「A が何らかの外」と限定する	Limit nouns to 'A is out some'
63	man	DeterminerN.above	man A	名詞を「A が何らかの上」と限定する	Limit nouns to 'A above some'
64	men	DeterminerN.below	men A	名詞を「A が何らかの下」と限定する	Limit nouns to 'A is below some'
65	min	DeterminerN.on	min A	名詞を「A が何らかに接地している」と限定する	Limit nouns to 'A is grounded to something'

ID	word	func	How to use	Japanese	English
66	mun	DeterminerN.right	mun A	名詞を「A が何らかの右」と限定する	Limit nouns to 'A is some right'
67	mon	DeterminerN.left	mon A	名詞を「A が何らかの左」と限定する	Limit nouns to 'A is some left'
68	tan	DeterminerN.affect	tan A	名詞を「A が何らかの影響を与えるものや関連していること」と限定する	Limit the noun to 'something that A affects or is related to in some way'
69	ten	DeterminerN.affected	ten A	名詞を「A が何らかの影響が与えられるもの」と限定する	Limit noun to 'something that A is affected by in some way'
70	tin	DeterminerN.near	tin A	名詞を「A が近くにあるものや関連しているもの」と限定する	Limit the noun to 'something that A is near or related to'.
71	tun	DeterminerN.move	tun A	名詞を「A が横切る」や「A が通る」「A が向かう」と動きのあるものに限定する	Limit nouns to those with motion, such as 'A crosses', 'A passes' or 'A heads'.
72	ton	DeterminerN.stop	ton A	名詞を静止のあるものに限定する	Limit nouns to those with static
73	dan	DeterminerN.all	dan A	名詞を「すべての A」と限定する	Limit the noun to 'all A'

ID	word	func	How to use	Japanese	English
74	den	DeterminerN.many	den A	名詞を「多くの A」と限定する	Limit the noun to ‘many A’, ‘much A’ or ‘a lot of A’
75	din	DeterminerN.some	din A	名詞を「いくつかの A」と限定する	Limit the noun to ‘some A’, ‘a few A’ or ‘several A’
76	dun	DeterminerN.one	dun A	名詞を「ある（一つの）A」と限定する	Limit the noun to ‘a certain A’, ‘one certain A’.
77	don	DeterminerN.plural	don A	名詞を複数形にする	Making nouns plural
78	san	DeterminerN.stressed	san A	名詞を強調形する	Stressed form of nouns.
79	sen	DeterminerN.possessive	sen A	所有代名詞を作成する	Creating possessive pronouns
80	sin	DeterminerN.reflexive	sin A	再帰代名詞を作成する	Creating recursive pronouns
81	sun	DeterminerN.etc	sun A	名詞を「A など」と限定する	Limit nouns to ‘A etc.’
82	son	DeterminerN.**abstract**	son A	～的/～のようなもの/大体の～/おおよそ～ぐらい/名詞を抽象化する	something like A/ Abstracting nouns

ID	word	func	How to use	Japanese	English
83	nan	DeterminerN.front	nan A	名詞を「A が何らかの空間的に前」と限定する	Limit nouns to ‘A is in front of some space’
84	nen	DeterminerN.behind	nen A	名詞を「A が何らかの空間的に後ろ」と限定する	Limit nouns to ‘A is behind in some space’
85	nun	DeterminerN.future	nun A	名詞を「A が何らかの時間的に未来」と限定する	Limit nouns to ‘A is some time in the future’
86	non	DeterminerN.past	non A	名詞を「A が何らかの時間的に過去」と限定する	Limit nouns to ‘A is some time in the past’
87	lan	DeterminerN.male	lan A	名詞を「A が何らかの男性や雄」と限定する	Limit noun to ‘A is some male’
88	len	DeterminerN.female	len A	名詞を「A が何らかの女性や雌」と限定する	Limit the noun to ‘A is some female’
89	lin	DeterminerN.every	lin A	名詞を「A があらゆる何らかのもの」と限定する	Limit the noun to ‘A is every something’
90	lun	DeterminerN.each	lun A	名詞を「A がそれぞれの何らかのもの」と限定する	Limit the noun to ‘A is each something’

ID	word	func	How to use	Japanese	English
91	lon	DeterminerN.other	lon A	名詞を「A が他の何らかのもの」と限定する	Limit the noun to 'A is other something'
92	pak	DeterminerV.Estimation10	pak A	100%の確率で A する	100% probability A
93	pek	DeterminerV.Estimation75	pek A	75%の確率で A する	75% probability A
94	pik	DeterminerV.Estimation50	pik A	50%の確率で A する	50% probability A
95	puk	DeterminerV.Estimation25	puk A	25%の確率で A する	25% probability A
96	pok	DeterminerV.Estimation0	pok A	0%の確率で A する	0% probability A
97	fak	DeterminerV.Frequency100	fak A	100%ぐらいの頻度で A する	100% frequently A
98	fek	DeterminerV.Frequency75	fek A	75%ぐらいの頻度で A する	75% frequently A

ID	word	func	How to use	Japanese	English
99	fik	DeterminerV.Frequency50	fik A	50%ぐらいの頻度で A する	50% frequently A
100	fuk	DeterminerV.Frequency25	fuk A	25%ぐらいの頻度で A する	25% frequently A
101	fok	DeterminerV.Frequency0	fok A	0%ぐらいの頻度で A する	0% frequently A
102	tak	DeterminerV.Start	tak A	A し始める	Someone starts doing something
103	tek	DeterminerV.Condition	tek A	A している途中である	Someone is in the middle of doing something
104	tik	DeterminerV.Complete	tik A	A している途中だったが完了した	Someone was in the middle of doing something but has completed
105	tuk	DeterminerV.Continue	tuk A	A している状態が続いている	Someone is still doing something
106	tok	DeterminerV.End	tok A	A し終える	Someone finishes doing something
107	bak	DeterminerV.past	bak A	過去には A であった	In the past it was A
108	bik	DeterminerV.present	bik A	現在 A である	In the present it is A

ID	word	func	How to use	Japanese	English
109	bok	DeterminerV.future	bok A	未来には A だろう	In the future it will be A
110	nak	DeterminerV.Possible	nak A	A できる/A することが可能である	can
111	nek	DeterminerV.Ability	nek A	A する能力がある	can
112	nik	DeterminerV.Will	nik A	A しよう	will/ shall
113	nuk	DeterminerV.Obligation	nuk A	A すべきだ	should/ ought to
114	nok	DeterminerV.Necessary	nok A	A する必要がある	need to
115	lak	DeterminerV.Duty	lak A	A しなければならない	must/ have to
116	lek	DeterminerV.forced	lek A	外部からの強い力で強制的に A させられる	be forced to A by a strong external force
117	lik	DeterminerV.want	lik A	A したい/A することを願望する	want to A
118	luk	DeterminerV.dare	luk A	あえて A する/思い切って A する/A する勇気がある	dare A
119	lok	DeterminerV.allow	lok A	A することを許す	allow to A
120	kak	DeterminerV.easy	kak A	A しやすい	be easy to A
121	kek	DeterminerV.hard	kek A	A しにくい	be hard to A

ID	word	func	How to use	Japanese	English
122	kik	DeterminerV.habit	kik A	習慣的に A する	Habitually A
123	kuk	DeterminerV.Polite	kuk A	A します（丁寧表現）	Make the verb a polite expression
124	kok	DeterminerV.Respect	kok A	A される（尊敬表現）	Make the verb a respectful expression
125	gak	DeterminerV.volitional	gak A	意識的に A する	Consciously A
126	gek	DeterminerV.nonVolitiona	gek A	無意識的に A する	Unconsciously A
127	gik	DeterminerV.Requests	gik A	A してください	will/ would/ can/ could
128	guk	DeterminerV.Permission	guk A	A してもよいですか	can/ may
129	gok	DeterminerV.Suggestion	gok A	A しましょうか	shall
130	ga	Pronoun.I	ga	私	I
131	ge	Pronoun.you	ge	あなた	you
132	gi	Pronoun.he	gi	彼/彼女/それ	he/ she/ it
133	gu	Pronoun.proximal	gu	これ	this
134	go	Pronoun.distal	go	それ/あれ	that

ID	word	func	How to use	Japanese	English
135	wa	Pronoun.interrogative	wa	どれ	what
136	we	Pronoun.indefinite	we	どれか	something
137	kan	WordV.create	kan	生み出す/作る/産む	create/ make/ bear
138	ken	WordV.destroy	ken	破壊する/壊す/死ぬ	destroy/ break/ die
139	kin	WordV.act	kin	行動する/動く/実行する/歩く/働く	act/ move/ do/ walk/ work
140	kun	WordV.turn	kun	回る/回転する/急ぐ/走る	turn/ rotate/ hurry/ run
141	kon	WordV.receive	kon	感じ取る/受信する/受け取る/入れる/摂取する/取得する/得る/習う/聞く/見る/食べる/飲む	receive/ accept/ acquire/ get/ learn/ hear/ see/ listen/ look at/ watch/ eat/ drink
142	gan	WordV.stimulate	gan	発する/発信する/発射する/出す/送信する/送る/教える/刺激する/言う/話す/攻撃する	emit/ transmit/ put out/ send/ give/ teach/ stimulate/ say/ speak/ attack

ID	word	func	How to use	Japanese	English
143	gen	`WordV.exist`	gen	ある/いる/存在する/生きている/住んでいる/留まる/止まっている/休む	be/ exist/ live/ stay/ be stopping/ get rest
144	gin	`WordV.use`	gin	使う/使用する	use
145	gun	`WordV.change`	gun	変わる/なる/成長する/移行する/移動する	change/ become/ grow/ transfer
146	wan	`WordM.big`	wan	大きい/長い/広い/高い/多い/重い	big/ long/ wide/ tall/ many/ heavy/ large
147	wen	`WordM.near`	wen	近い/親しい/似ている/好きである	near/ familiar/ close to/ similar/ like
148	win	`WordM.good`	win	良い/新しい/若い/美しい	good/ new/ young/ beautiful
149	won	`WordM.bright`	won	明るい/白い/色鮮やかな	bright/ white/ colourful
150	pas	`Bool.false`	pas	偽	False (Boolean)
151	pos	`Bool.true`	pos	真	True (Boolean)
152	pis	`Bool.B2N`	pis A B	A は B である（B は真偽）	A is B (B is true or false)
153	fas	`BoolList`	fas	真偽のリスト（BoolList）を作成する	Create a list of true/ false (BoolList)

ID	word	func	How to use	Japanese	English
154	fes	BoolList.get	fes A B	BoolList(A) の B 番目の値を取得する	Gets the B-th value of BoolList(A)
155	fis	BoolList.append	fis A B	BoolList に 1 つの Bool を末尾に加える	Add one Bool to the end of the BoolList
156	fus	BoolList.slice	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	Get the B-th through C-th lists for a BoolList (A).
157	fos	BoolList.add	fos A B	2 つの BoolList を結合する	Combine two BoolLists
158	foas	BoolList.len	foas A	BoolList の長さを取得する	Get the length of the BoolList
158	mas	BoolList.twoBit	mas A B	2 つ Bool の値からなる BoolList を作成する	Create a BoolList consisting of 2 Bool values
159	mis	BoolList.fourBit	mis A B C D	4 つ Bool の値からなる BoolList を作成する	Create a BoolList consisting of 4 Bool values
160	mos	BoolList.**byte**	mos X1 X2 X3 X4 X5 X6 X7 X8	8 つ Bool の値からなる BoolList を作成する	Create a BoolList consisting of 8 Bool values

ID	word	func	How to use	Japanese	English
161	tas	`BoolList.NaturalNum`	tas A	BoolList を 2 進数の自然数とみなす	BoolList is considered a binary natural number
162	tes	`BoolList.Int`	tes A	BoolList を 2 進数の整数とみなす	BoolList is considered a binary integer
163	tis	`BoolList.Float`	tis A	BoolList を 2 進数の浮動小数とみなす	BoolList is considered a binary floating number
164	tus	`BoolList.ASCII`	tus A	BoolList を ASCII 文字とみなす	BoolList is considered an ASCII character
165	tos	`BoolList.IntBL2NL`	tos A	整数の BoolList を NumberList に変換する	Convert an integer BoolList to a NumberList
166	das	`BoolList.UnixTimeD`	das A	BoolList を日単位の UnixTime とする	BoolList as UnixTime in days
167	des	`BoolList.UnixTimeDT`	des A	BoolList を秒単位の UnixTime とする	BoolList as UnixTime in seconds
168	dis	`BoolList.UnixTimeDTN`	dis A	BoolList をナノ秒単位の UnixTime とする	BoolList as UnixTime in nanoseconds

ID	word	func	How to use	Japanese	English
169	dos	`BoolList.FloatBL2NL`	dos A	浮動小数点の実数の BoolList を NumberList に変換する	Convert an Floating-point real numbers BoolList to a NumberList
169	pat	`LangFunc.setFunc`	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	Set up a function that returns B named A with a certain LangList as an argument.
170	pit	`LangFunc.arg`	pit	`LangFunc.setFunc()` の引数用に使用する	Used for `LangFunc.setFunc()` arguments
171	pot	`LangFunc.runFunc`	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	Execute the configured LangFunc named A with argument B
172	bat	`LangVar.set`	bat A B	グローバル変数として A という名前の変数を定義し，LangList B を代入する	Define a variable named A as a global variable and assign LangList B to it.
173	bot	`LangVar.get`	bot A	定義された A という名前のグローバル変数を取得する	Obtain the defined global variable named A

ID	word	func	How to use	Japanese	English
174	fat	`LangList`	fat	LangObj のリスト LangList を作成する	Create a list of LangObj (LangList)
175	fet	`LangList.get`	fet A B	`LangList(A)` の B 番目の値を取得する	Gets the B-th value of `LangList(A)`
176	fit	`LangList.append`	fit A B	LangList に 1 つの LangObj を末尾に加える	Add one LangObj to the end of the LangList
177	fut	`LangList.slice`	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	Get the B-th through C-th lists for a LangList (A).
178	fot	`LangList.add`	fot A B	2 つの LangList を結合する	Combine two LangLists
179	foat	`LangList.len`	foat A	LangList の長さを取得する	Get the length of the LangList
180	tat	`LangList.While`	tat A B C	繰り返し処理を行う	Repeat processing
181	tet	`LangList.map`	tet A B	LangList A のすべての要素に対して，B という名前の LangFunc を適用する	Apply a LangFunc named B to all elements of LangList A
182	pal	`Number.zero`	pal	0	0
183	pel	`Number.one`	pel	1	1

ID	word	func	How to use	Japanese	English
184	pil	Number.two	pil	2	2
185	pul	Number.three	pul	3	3
186	pol	Number.four	pol	4	4
187	bal	Number.five	bal	5	5
188	bel	Number.six	bel	6	6
189	bil	Number.seven	bil	7	7
190	bul	Number.eight	bul	8	8
191	bol	Number.nine	bol	9	9
192	fal	NumberList	fal	Number のリスト NumberList を作成する	Create a list of Number (NumberList)
193	fel	NumberList.get	fel A B	NumberList(A) の B 番目の値を取得する	Gets the B-th value of NumberList(A)
194	fil	NumberList.append	fil A B	NumberList に 1 つの Number を末尾に加える	Add one Number to the end of the NumberList
195	ful	NumberList.slice	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	Get the B-th through C-th lists for a NumberList (A).

ID	word	func	How to use	Japanese	English
196	fol	NumberList.add	fol A B	2 つの NumberList を結合する	Combine two NumberLists
197	foal	NumberList.len	foal A	NumberList の長さを取得する	Get the length of the NumberList
198	mal	NumberList.digit1	mal A	10 進数 1 桁からなる NumberList を作成する	Create a NumberList consisting of one decimal digit
199	mel	NumberList.digit2	mel A B	10 進数 2 桁からなる NumberList を作成する	Create a NumberList consisting of two decimal digit
200	mil	NumberList.digit3	mil A B C	10 進数 3 桁からなる NumberList を作成する	Create a NumberList consisting of three decimal digit
201	mul	NumberList.digit4	mul A B C D	10 進数 4 桁からなる NumberList を作成する	Create a NumberList consisting of four decimal digit
202	mol	NumberList.digit5	mol A B C D E	10 進数 5 桁からなる NumberList を作成する	Create a NumberList consisting of five decimal digit

ID	word	func	How to use	Japanese	English
203	tal	NumberList.calcAdd	tal A B	2 つの NumberList に対して加算をする	Perform addition on two NumberLists
204	tel	NumberList.calcSub	tel A B	2 つの NumberList に対して減算をする	Perform subtraction on two NumberLists
205	til	NumberList.calcMul	til A B	2 つの NumberList に対して乗算をする	Perform multiplication on two NumberLists
206	tul	NumberList.calcDiv	tul A B	2 つの NumberList に対して除算をする	Perform division on two NumberLists
207	tol	NumberList.IntNL2BL	tol A	整数の NumberList を BoolList に変換する	Convert an integer NumberList to a BoolList
208	dal	NumberList.calcPow	dal A B	2 つの NumberList に対して累乗をする	Performs a power over two NumberLists
209	del	NumberList.calcIntDiv	del A B	2 つの NumberList に対して整数除算をする	Perform integer division on two NumberLists
210	dil	NumberList.calcMod	dil A B	2 つの NumberList に対して剰余をする	Performs remainder with respect to two NumberLists

ID	word	func	How to use	Japanese	English
211	dol	NumberList.FloatNL2BL	dol A	浮動小数点の実数の NumberList を BoolList に変換する	Convert a Floating-point real numbers NumberList to a BoolList
212	sal	NumberList.isPN	sal A	正の数かを判定する	Determine if it is a positive number
213	sel	NumberList.minus	sel A	符号を反転させる	Reversing the sign
214	sil	NumberList.abs	sil A	整数の絶対値を取得する	Obtaining the absolute value of an integer

25. 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．

25.1. 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

Version	Update	Contents
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

25.2. バージョン更新内容について

Version	Update contents
1.0.0	正式版公開
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

Version	Update contents
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
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
5.0.10	ドキュメントの追加・修正
5.0.11	ドキュメントの追加・修正
5.0.12	ドキュメントの追加・修正
5.0.13	ドキュメントの追加・修正
5.0.14	ドキュメントの追加・修正

Version	Update contents
5.0.15	ドキュメントの追加・修正
5.0.16	ドキュメントの追加・修正
5.0.17	ドキュメントの追加・修正
5.0.18	ドキュメントの追加・修正
5.1.0	LangObj.logicIFELSE() と NumberList.isPN() を追加
5.1.1	ドキュメントの追加・修正
5.1.2	ドキュメントの追加・修正
5.1.3	ドキュメントの追加・修正
5.1.4	ドキュメントの追加・修正
5.1.5	ドキュメントの追加・修正
5.1.6	ドキュメントの追加・修正
5.1.7	ドキュメントの追加・修正
5.2.0	ドキュメントに辞書を追加
5.2.1	ドキュメントの追加・修正
5.3.0	SFGPLLib.checkType() の追加
5.3.1	ドキュメントの追加・修正
6.0.0	LangVar の追加
6.1.0	SFGPL の構造化に関連する関数の追加
6.1.1	ドキュメントと SFGPL.py の追加・修正
7.0.0	単語と関連ライブラリの追加
7.1.0	リスト関連クラスにおけるリスト長さの関数を追加
7.2.0	`LangList.map`追加
7.2.1	ドキュメントの追加・修正
7.2.2	ドキュメントの追加・修正
7.2.3	ドキュメントの追加・修正
7.3.0	BoolList における Unix 時間と様々な浮動小数点数の対応
7.3.1	ドキュメントの追加・修正

Version	Update contents
7.3.2	ドキュメントの追加・修正
7.4.0	BoolList と NumberList 間の浮動小数点の変換の追加，NumberList の演算種類の追加
7.4.1	ドキュメントの追加・修正