| | |
|---|---|
| Title | Modifying the Learning Algorithm on the Kirby's Model for Simultaneous Bilingual Case |
| Author(s) | Namchisawatwong, Teerat |
| Citation | |
| Issue Date | 2012-09 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/10781 |
| Rights | |
| Description | Supervisor:Satoshi Tojo，情報科学研究科，修士 |

# Modifying the Learning Algorithm on the Kirby' s Model for Simultaneous Bilingual Case

By NAMCHAISAWATWONG Teerat

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Satoshi Tojo

September, 2012

# Modifying the Learning Algorithm on the Kirby' s Model for Simultaneous Bilingual Case

By NAMCHAISAWATWONG Teerat (1010211)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Satoshi Tojo

and approved by
Professor Satoshi Tojo
Professor Hiroyuki Iida
Professor Akira Shimazu

August, 2012 (Submitted)

## Abstract

To study the language evolution, there are many frameworks for investigating the cultural evolution of linguistic structure. One of them is a famous Iterated Learning Model (ILM) that named Kirby's model.

Kirby's model can work well in various experiments. But it cannot simulate some of case studies such as case studies of Latin language evolution. I am interested in this problem and set the objective of the research that is to design and implement language separator in order to make the Kirby's model which supports bilingual environments.

Language separator is a feature helping agent in Kirby's model differentiate two languages by observing the frequency of the co-occurrence of vocabularies and sentence structure. And Language separator will be successful, when languages in former generation can be transmitted to the next generation.

For this objective, we purposed the method. First of all, we modified the Kirby's model according to bilingual education in the real world. We give more definition about language in modified Kirby's model. We change the start generation from agent with blank grammar to agent with ideal grammar. We set a new characteristic of speaker agent and learner agent. We modify the speak process and invention to make agent can speak two language without mixing them. We modify the learning process by adding scoring system to evaluate the relation score of each rule named front end process. The mechanism of front end process is observing co-occurrence of using rules. And we implemented the separator to help agent differentiate the languages by using relation score table that got from front end process. The separator will be applied when learner agent change role to be speaker agent.

After applying our proposed method, languages in former generation can be transmitted to the next generation. It shows the best performance when using the grammar that has no common area between two languages as grammar of n-th generation. And it shows the good performance when using the grammar that the number of rules in common area between two languages is grammar of n-th generation less than four.

# Acknowledgement

I would like to express my gratitude to those who gave me the opportunity to complete this research. This research could not have been finished without Professor Satoshi Tojo who not only served as my supervisor but also encouraged and challenged me throughout my academic program, Professor Hiroyuki Iida and Professor Akira Shimazu who took time from their busy schedules to serve me on my dissertation committee and gave me a lot of constructive criticisms, and Assistant Professor Katsuhiko Sano and Mr. Shingo Hagiwara who gave me a lot of useful guidances and opinions to continue my research. Moreover, I would like to express my thanks to every member of our lab for the encouragement and support on my work and learning experience. Last, I would like to thank JAIST and Japan for giving me such good experiences.

# Contents

# List of Figures

# Chapter 1

# Introduction

To study the language evolution, there are many frameworks for investigating the cultural evolution of linguistic structure. One of them is a famous Iterated Learning Model (ILM) that named `Kirby's model` [1-5]

Kirby's model can work well in various experiments. But it cannot simulate some of case studies such as case studies of Latin language evolution. I am interested in this problem and set the objective of the research that is to design and implement some conditions settings for make the Kirby's model support bilingual environments.

This thesis is organized as follows: in Chapter 2 we explain Kirby's Iterated Learning Model and it's problem. And we mention the goal of the research at the end of this chapter. In Chapter 3 we explain about our Proposed method. Chapter 4 presents the details of our experiment and gives example result of them and analyze the results. And conclusion and future work are explained in Chapter 5.

# Chapter 2

# Kirby's Model and Problem

## 2.1 Kirby's Model

### 2.1.1 Iterated Learning Model

Iterated Learning Model (ILM) is a framework for investigating the cultural evolution of linguistic structure. In the experiment by Kirby named Kirby's Model, there are two types of agents that are speaker agent and learner agent. The speaker agent represents the mother in real world. And the learner agent represents the infant in the real world. The next paragraph will show what do the model simulates.

The mother (speaker agent) tries to speak to her infant (listener agent) by utterance which is phased her intention into her grammars. And the infant tries to learn language by using some operations to manage a received pair of intention and utterance. At last, the infant is able to build own grammar that may or may not be similar to mother's grammar. From this point, it will be counted it as one generation. And then the infant becomes a new mother of next generation, and a new infant with no any rule in her grammar is added to the experiment. This process is iterated generation by generation. And finally, a certain generation would acquire a compact, limited number of grammar rules.

### 2.1.2 Grammatical representation

All agents in the model must have a grammar that is the set of rules. And the context-free grammar is used for represent the rules in grammar, as follow.

$$S/eat(cat,fish) \rightarrow \texttt{cdsai} \qquad (1)$$

Grammar1

There is one rule in grammar1 that is $S/eat(cat,fish) \rightarrow$ `cdsai`. It means that intention where is *eat(cat,fish)* or "cat eat fish" in real world can be expressed by the utterance where is `cdsai`. The 'S' symbol means the category of this rule. To generate the utterance, there are many way to generate them. For example, follow rules in grammar2 can also generate the same utterance by phasing the same intention into different grammar that contains different rules.

$$S/p(x,fish) \rightarrow A/x\texttt{sa}\ B/p \qquad (1)$$
$$A/cat \rightarrow \texttt{cd} \qquad (2)$$
$$B/eat \rightarrow \texttt{i} \qquad (3)$$

Grammar2

The variable x of rule (1) in grammar2 is substituted for a category A and the variable p is substituted for a category B. By using these previous rules in grammar2, the intention *eat(cat,fish)* can be express by the utterance `cdsai`. For easy to understand, the figure2.1(a) is shown the phrased tree when *eat(cat,fish)* was phrased by grammar1 and the figure2.1(b) is shown the phrased tree when *eat(cat,fish)* was phrased by grammar2.

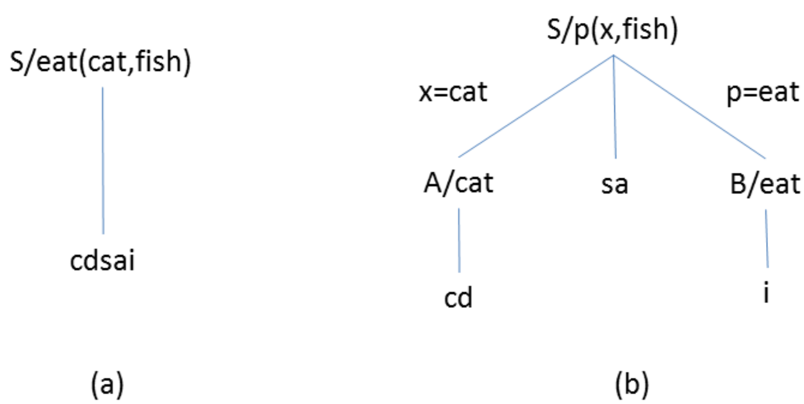


Figure 2.1: phrased trees of *eat(cat,fish)*

The rule(1) in grammar1 is the rule that have no any variables. This kind of rule is called holistic rule. The rule with some variables such as rule(1) in grammar(2) is called a compositional rule. And the rules that represent a vocabulary such as rule(2) and rule(3) in grammar2 are called lexical rule.

### 2.1.3 Learning process and Learning operation

The learning process is the core of any model of language acquisition. In Kirby' model, the learning process is used by learner agent.

In the first stage of learning process, the agent's grammar contains no any rules. The learner agent will receive only two data, heard utterance and speaker's intention, from speaker agent such as <`cdsai`; *S/eat(cat,fish)*>. The learner will not know the rules that speaker used in her speaking process. So, the learner can create a simple holistic rule to describe the received data<`cdsai`; *S/eat(cat,fish)*> like *S/eat(cat,fish)* $\rightarrow$ `cdsai`

But it wills not a good idea, if all rules in grammar are holistic rules. So, to make grammar more compositional in Kirby's model, there are some operations that change the holistic rules to compositional rules and lexical rules.

There are three operations that is chunk operation, merge operation and replace operation.

### Chunk

Chunk operation will be applied when a condition "the two rules will be the same if either one or both of them chunked a sequence of terminals" is met.

The chunk operation will create the new rules that contain substring of non-terminals on the right hand side of the old rule, and adjusting the old rule to refer to the new one. For example, grammar3 can be chunked to be grammar4.

$$\textit{S/eat(cat,fish)} \rightarrow \texttt{cdsai} \qquad (1)$$
$$\textit{S/eat(dog,fish)} \rightarrow \texttt{xopsai} \qquad (2)$$

Grammar3

$$\textit{S/eat(x,fish)} \rightarrow A/x\texttt{sai} \qquad (1)$$
$$\textit{A/cat} \rightarrow \texttt{cd} \qquad (2)$$
$$\textit{A/dog} \rightarrow \texttt{xop} \qquad (3)$$

Grammar4

In grammar3, the rule(1) and rule(2) are almost same and they meet the condition as said before. So, that chunk operation can be applied. And the chunked result are shown as grammar4.

The rule(1) in grammar4 is the rule that be adjusted from rule(1) and rule(2) in grammar3. The rule(2) and rule(3) in grammar4 is the added rules that contain substring of non-terminals on the right hand side of the rule(1) and rule(2) in grammar3.

### Merge

Merge operation will be applied when a condition "the two rules will be the same if two category symbols were merged" is met.

The merge operation will select one of the categories of those two rules and rewrite the other one to be the same as it throughout the grammar. For example, category A and C in grammar5 can be merged into be category A in grammar6.

$$
\begin{aligned}
&\textit{S/p(x,fish)} \rightarrow A/x\texttt{sa}B/p && (1)\\
&A/cat \rightarrow \texttt{cd} && (2)\\
&B/eat \rightarrow \texttt{I} && (3)\\
&\textit{S/bite(x,snake)} \rightarrow \texttt{kp}C/x\texttt{o} && (4)\\
&C/cat \rightarrow \texttt{cd} && (5)\\
&C/dog \rightarrow \texttt{xop} && (6)
\end{aligned}
$$

Grammar5

$$
\begin{aligned}
&\textit{S/p(x,fish)} \rightarrow A/x\texttt{sa}B/p && (1)\\
&A/cat \rightarrow \texttt{cd} && (2)\\
&B/eat \rightarrow \texttt{I} && (3)\\
&\textit{S/bite(x,snake)} \rightarrow \texttt{kp}A/x\texttt{o} && (4)\\
&A/dog \rightarrow \texttt{xop} && (5)
\end{aligned}
$$

Grammar6

In grammar5, both rule(2) and rule(5) are lexical rules that have the same meanings and string but different categories. So, the category A and B will be merged into one category that is shown as generation6.

The rule(2) in grammar6 is the rule that cause this time merging rule(2) and rule(5) in grammar5. The category B of rule(4) and rule(6) in generation6 was replaced by category A as rule(4) and rule(5) in generation6 in order.

**Replace**

Replace operation will be applied when a lexical rule can be embedded in another holistic rule or compositional rule.

The replace operation will replace a variable to the substring in right hand side holistic rule or compositional rule that similar to the right hand side of the lexical rule. For example, in the grammar7, rule(2) can be embedded in rule(1). So, the result after replaced is shown as grammar8

$$
\begin{aligned}
&\textit{S/bite(dog,snake)} \rightarrow \texttt{kpxopo} && (1)\\
&A/dog \rightarrow \texttt{xop} && (2)
\end{aligned}
$$

Grammar7

$$
\begin{aligned}
&\textit{S/bite(x,snake)} \rightarrow \texttt{kp}A/x\texttt{o} && (1)\\
&A/dog \rightarrow \texttt{xop} && (2)
\end{aligned}
$$

Grammar8

### 2.1.4 Speaking process and Invention

Speaking process and invention are used by speaker agent. The speaker agent tries to express some intention by using speaking process. An output of this process is an utterance. But in some case that speaker agent cannot express some intentions. Because her grammar has not enough rules to generate utterance, the invention will help her to create new rule in her grammar and the speaking process will be used again.

#### Speaking process

Speaking process is the process that is used for phase an intention to an utterance. In Kirby's model, the beginning of this process is to select an intention at random from the meaning space.

The meaning space is a set of all possible meaning in the experiment. For example, in degree0 experiment with 5 verbs and 5 nouns, there are 100 meanings (5 possible verbs X 5 possible nouns as subject X 4 possible nouns as object) in meaning space. Because the reflexive meanings, subject and object are the same noun, like *eat(cat,cat)* are prohibited in this experiment. And in degree0 experiment, only one verb is allowed in each meaning. So the meanings that have two or more verbs like *dog(see,eat(cat,fish))* are prohibited.

After selecting an intention at random from the meaning space, the selected intention will be phased by using speaker agent's grammar. If an utterance can be generated by this way, then speak process is finished. But sometime the utterance cannot be generated. It means that her grammar has not enough rules to generate utterance. In this case, the invention process will start working.

#### Invention

The invention process is an important process for introducing random new words for chunks of meaning in later. The invention process will be used when speaker agent's grammar has not enough rules to generate utterance.

The beginning of this process is trying to find the closest intention that the speaker has a way of producing. And then try phrasing new intention by using the phrase tree of found closest intention. The parse tree will show the parts of the string that correspond to the wrong parts of the meaning. These parts of the string are excised, and replaced with a random sequence of symbols.

$$S/eat(x,fish) \rightarrow A/x\texttt{sai} \quad (1)$$
$$S/bite(bird,x) \rightarrow \texttt{k}A/x\texttt{nxz} \quad (2)$$
$$A/cat \rightarrow \texttt{cd} \quad (3)$$
$$A/dog \rightarrow \texttt{xop} \quad (4)$$

Grammar9

First example, speaker agent has the grammar as grammar9. And speaker is asked to produce a string for the intention *eat(bird,fish)*. The nearest intention to this that the speaker can produce strings for are *eat(cat,fish)* or *eat(dog,fish)*. So the rule(1) will be picked first, then phrased tree will be created like figure2.2(a).

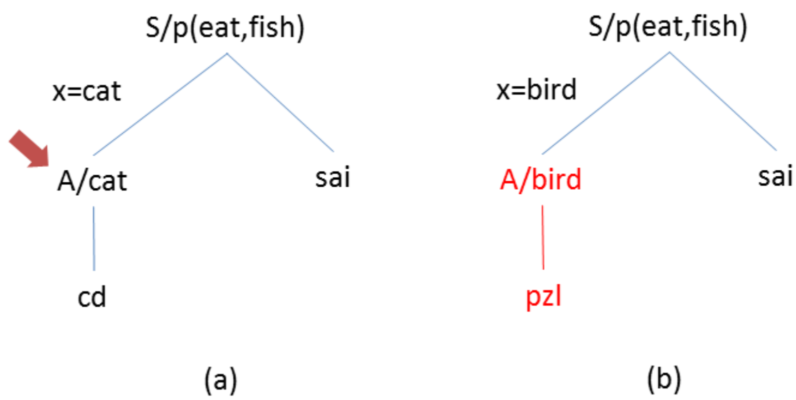


Figure 2.2: phrased trees when invent *eat(bird,fish)*

The wrong part of this tree is the part that introduces the meaning cat. So this part will be deleted and replaced this with a random sequence of characters that have length 1-3 characters. So, the an invented rule for the intention *eat(bird,fish)* might be $A/bird \rightarrow$ `pzl` as seen in rule(5) of grammar10. And the intention *eat(bird,fish)* might be expressed by utterance `pzlsai` by using phrase tree like figure2.2(b)

$$S/eat(x,fish) \rightarrow A/x\texttt{sai} \quad (1)$$
$$S/bite(bird,x) \rightarrow \texttt{k}A/x\texttt{nxz} \quad (2)$$
$$A/cat \rightarrow \texttt{cd} \quad (3)$$
$$A/dog \rightarrow \texttt{xop} \quad (4)$$
$$A/bird \rightarrow \texttt{pzl} \quad (5)$$

Grammar10

Second example, speaker agent has the grammar as grammar9. And speaker is asked to produce a string for the intention *bite(snake,fish)*. The nearest intention to this that the speaker can produce strings for are *eat(cat,fish)* or *eat(dog,fish)*. So the rule(1) will be picked first, then phrased tree will be created like figure2.3(a).

The wrong part of this tree is the root of the tree. So this tree deleted and replaced this with a random sequence of characters that have length 1-3 characters. So, the an invented rule for the intention *bite(snake,fish)* might be $bite(snake,fish) \rightarrow$ `yv` as seen in rule(5) of grammar11. And the intention *bite(snake,fish)*might be expressed by utterance `yv` by using phrase tree like figure2.3(b)

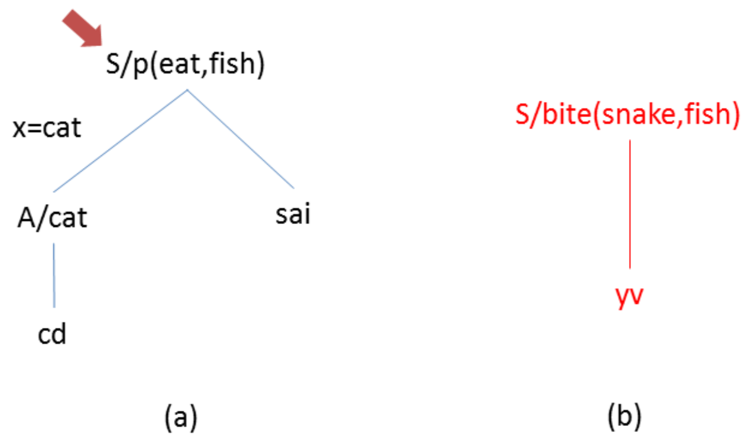


Figure 2.3: phrased trees when invent *bite(snake,fish)*

$$
\begin{aligned}
&S/eat(x,fish) \rightarrow A/x\texttt{sai} && (1)\\
&S/bite(bird,x) \rightarrow \texttt{k}A/x\texttt{nxz} && (2)\\
&A/cat \rightarrow \texttt{cd} && (3)\\
&A/dog \rightarrow \texttt{xop} && (4)\\
&S/bite(snake,fish) \rightarrow \texttt{yv} && (5)
\end{aligned}
$$

Grammar11

### 2.1.5 Simulation cycle

Each generation in simulation goes through the following steps:

1. The speaker agent tries to speak for 50 times.

- The speaker agent selects an intention at random from meaning space.

- If the speaker agent can generate a string for that meaning using own grammar, she does so, otherwise she invents a string. If the speaker has invented a string, the speaker uses that utterance as input to learner agent's learning

- The learner agent learns by receiving utterance and speaker agent's intention.

2. The speaker agent's grammar is logged and then it is deleted from the simulation.

3. The learner agent becomes the new speaker agent, and a new learner agent with a blank grammar is added to the simulation.


Figure 2.4: Simulation cycle of Kirby's model

```
generation ≔ 0;
speaker ≔ new agent
learner ≔ new agent
repeat{
        for 1 to 50 do{
                intension ≔ random from meaning space
                if(speaker.tryspeak(intension)){
                        utterance ≔ speaker.speak(intension)
                }else{
                        speaker.invent(intension)
                        utterance ≔ speaker.speak(intension)
                }
                learner.learn(utterance, intension)
        }
        speaker.saveLog()
        generation++
        speaker ≔ learner
        learner ≔ new agent
}until 200th generation
```

Algorithm1: Pseudo-code of Kirby's model

## 2.2 Problem

Kirby's model was designed for learning language evolution. It can work well in various experiments. And many researchers developed this model in many ways such as adding some bias for agent's learning, modifying the model to becoming more resembled to human learning in the real world, developing the performance of the model etc.

But Kirby's model cannot simulate the evolution of language in bilingual environments. It cannot describe the language separation in evolution of language history such as the case studies of Latin language evolution that was separated to French, Italian, Spanish, etc. after Roman Empire collapsed.

Kirby's model was designed for the learning language evolution of only one language. So it cannot work in bilingual environment. Because in Kirby's model, one intention can be expressed by one utterance. But in bilingual environment, one intention can be expressed by many utterances.

And I have tried to input the grammar that contains rules from two languages. The result is it merges both languages in to one language. Some of utterances were neglected by speaker agent and they have not ever been learned by learner agent.

From this problem, I have set the objective of the research that is to design and implement language separator for make the Kirby's model support bilingual environments.

Language separator is a feature helping agent in Kirby's model differentiate two languages by observing the frequency of the co-occurrence of vocabularies and sentence structure. Language separator will be successful, when languages in former generation can be transmitted to the next generation

# Chapter 3

# Proposed method

To reach the objective of the research, that is to design and implement some conditions settings for make the Kirby's model support bilingual environment, the cause of this problem was thought first. Because Kirby's model was not designed for bilingual environment, so the model will be modified first.

After finishing modifying the model, the model still did not support bilingual because the agent could not differentiate the difference of two languages. In this step, the idea to add some feature such as Front end process and separator were thought.

## 3.1 Modified Kirby's model

To modify the Kirby's model for make it support bilingual environment, the knowledge about bilingual education[6-10] will be focused. Because Kirby's model simulate human's language learning in environment that contain only one language, so we have to know the human's language learning in bilingual environment for modifying the model.

The bilingual is a person who can speak at least two languages fluently. To educate the bilingual infant, there are some keys of Bilingual Education, for example the parents must be fluent speaker of that language. The parents must not mix two languages in each sentence. It means that using code-switching is prohibits. After learning, infant may confuse about two languages, but when time goes by, they can differentiate these two languages automatically. Etc.

### 3.1.1 Definition of L1, L2, L0 and UN in modified Kirby's model

In Modified Kirby's model, there are two languages that are language1 and language2. Each rule in grammars have to be defined the language number that which language is it. But some rules cannot be defined that they are rules in language1 or language2. And some rules can be defined that they are both rules in language1 and language2.

For easy to understand, these four symbols will be used to indicate the language of rules that are L1, L2, L0 and UN. And Venn diagram will be used to explain the relation of these four symbols as figure3.1(a). The universe is a set that has all rules in grammar as elements.

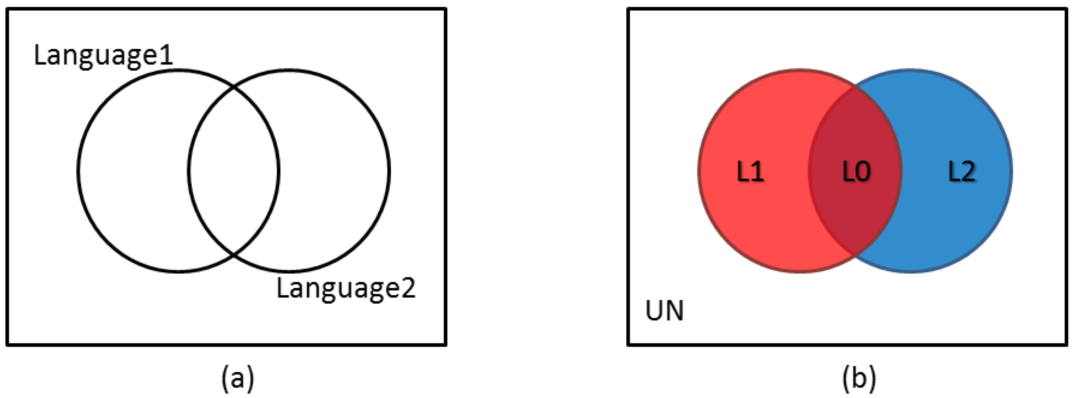


Figure 3.1: Venn diagram of definition of L1, L2, L0 and UN

L1 set means set of rules that can be indicated in only language1. It can be express by Venn diagram as red color area in figure3.1(b). L2 set means set of rules that can be indicated in only language2. It can be express by Venn diagram as blue color area in figure3.1(b). L0 set means set of rules that can be both language1 and language2. It can be express by Venn diagram as violet color area in figure3.1(b). And UN set means set of rules that have be not decided the language yet and the rule that cannot be both language1 and language2. It can be express by Venn diagram as white color area in figure3.1(b).

When implement modified Kirby's model, we have to add a variable to each rules in grammar. This variable will store a value to indicate the language of rules represent to the four symbols (L1, L2, L0 and UN).

### 3.1.2 Start generation in modified Kirby's model

In original Kirby's model, the start generation is zero generation. The grammar of speaker agent in zero generation is an empty set. It means that there is no any rule in the grammar of speaker agent in zero generation. So, when experiment is started, the speaker agent with empty grammar and learner agent with empty grammar will be added to simulation. Then the speaker agent will try to speak with her empty grammar. Of course, speaker agent cannot generate an utterance and the new holistic rule that contain a sequence of random character will be invented into her grammar.

But in modified Kirby's model, the start generation should not be zero generation. And grammar of speaker agent in start generation should not be an empty set. Because modified Kirby's model use for describe some phenomena in the evolution of language in bilingual environments such as language separation. So the grammar of start generation should be the grammar of well-developed language.

So I decided to define the start generation in modified Kirby's model as n-th generation. The definition of n-th generation is the generation that speaker agent has an ideal grammar of well-developed language.

The ideal grammar is the grammar that can express all intentions in meaning space with least possible number of rules.

$S/p(x,y) \rightarrow A/x\texttt{d}B/pA/y$
$A/noun0 \rightarrow$ `uj`
$A/noun1 \rightarrow$ `mp`
$A/noun2 \rightarrow$ `nod`
$A/noun3 \rightarrow$ `q`
$A/noun4 \rightarrow$ `uuyg`
$B/verb5 \rightarrow$ `rfs`
$B/verb6 \rightarrow$ `cg`
$B/verb7 \rightarrow$ `cnf`
$B/verb8 \rightarrow$ `z`
$B/verb9 \rightarrow$ `s`

Grammar12

For example, the ideal grammar of degree0 experiment of the original Kirby's model can be written like grammar12. It contains eleven rules that are one of compositional rule and five lexical rules that are represented as noun vocabulary and other five lexical rules that are represented as verb vocabulary. These eleven rules can express all intensions in meaning space.

Next examples are ideal grammar of degree0 experiment of the modified Kirby's model. The ideal grammar in modified Kirby's model has rules in L1 and L2. And also L0 rules may be existed up to experiment. And there are no any UN rules in ideal grammar.

Grammar13 show rules in the ideal grammar of degree0 experiment of modified Kirby's model that have no any L0 rules. Grammar13 contains twenty two rules that are eleven rules in L1 and eleven rules in L2. These eleven L1 rule can be divided to one of L1 compositional rule of and five L1 lexical rules that are represented as noun vocabulary and other five L1 lexical rules that are represented as verb vocabulary. And L2 rules can be also divided in the same way. And these twenty two rules can express all intentions in meaning space in by utterances in language1 and utterances in language2.

| | |
|---|---|
| $S/p(x,y) \rightarrow A/x\texttt{d}B/pA/y$ | [L1] |
| $A/noun0 \rightarrow \texttt{uj}$ | [L1] |
| $A/noun1 \rightarrow \texttt{mp}$ | [L1] |
| $A/noun2 \rightarrow \texttt{nod}$ | [L1] |
| $A/noun3 \rightarrow \texttt{q}$ | [L1] |
| $A/noun4 \rightarrow \texttt{uuyg}$ | [L1] |
| $B/verb5 \rightarrow \texttt{rfs}$ | [L1] |
| $B/verb6 \rightarrow \texttt{cg}$ | [L1] |
| $B/verb7 \rightarrow \texttt{cnf}$ | [L1] |
| $B/verb8 \rightarrow \texttt{z}$ | [L1] |
| $B/verb9 \rightarrow \texttt{s}$ | [L1] |
| $S/p(x,y) \rightarrow A/yB/p\texttt{rg}A/x\texttt{o}$ | [L2] |
| $A/noun0 \rightarrow \texttt{d}$ | [L2] |
| $A/noun1 \rightarrow \texttt{mx}$ | [L2] |
| $A/noun2 \rightarrow \texttt{k}$ | [L2] |
| $A/noun3 \rightarrow \texttt{cs}$ | [L2] |
| $A/noun4 \rightarrow \texttt{t}$ | [L2] |
| $B/verb5 \rightarrow \texttt{xg}$ | [L2] |
| $B/verb6 \rightarrow \texttt{sfp}$ | [L2] |
| $B/verb7 \rightarrow \texttt{o}$ | [L2] |
| $B/verb8 \rightarrow \texttt{xvi}$ | [L2] |
| $B/verb9 \rightarrow \texttt{zhi}$ | [L2] |

Grammar13

Grammar14 show rules in the ideal grammar of degree0 experiment of modified Kirby's model that has two L0 rules. Grammar14 contains twenty rules that are nine rules in L1, nine rules in L2 and two rules in L0. These nine L1 rule can be divided to one of L1 compositional rule of and seven L1 lexical rules that are represented as noun or verb vocabulary. And L2 rules can be also divided in the same way. And these twenty rules can express all intentions in meaning space in by utterances in language1 and utterances in language2.

| | |
|---|---|
| $S/p(x,y) \rightarrow A/x\texttt{d}B/pA/y$ | [L1] |
| $A/noun0 \rightarrow \texttt{uj}$ | [L1] |
| $A/noun1 \rightarrow \texttt{mp}$ | [L1] |
| $A/noun2 \rightarrow \texttt{nod}$ | [L1] |
| $A/noun4 \rightarrow \texttt{uuyg}$ | [L1] |
| $B/verb5 \rightarrow \texttt{rfs}$ | [L1] |
| $B/verb6 \rightarrow \texttt{cg}$ | [L1] |
| $B/verb7 \rightarrow \texttt{cnf}$ | [L1] |
| $B/verb9 \rightarrow \texttt{s}$ | [L1] |
| $S/p(x,y) \rightarrow A/yB/p\texttt{rg}A/xo$ | [L2] |
| $A/noun0 \rightarrow \texttt{d}$ | [L2] |
| $A/noun1 \rightarrow \texttt{mx}$ | [L2] |

| | |
|---|---|
| $A/noun2 \rightarrow$ `k` | [L2] |
| $A/noun4 \rightarrow$ `t` | [L2] |
| $B/verb5 \rightarrow$ `xg` | [L2] |
| $B/verb6 \rightarrow$ `sfp` | [L2] |
| $B/verb7 \rightarrow$ `o` | [L2] |
| $B/verb9 \rightarrow$ `zhi` | [L2] |
| $A/noun3 \rightarrow$ `q` | [L0] |
| $B/verb8 \rightarrow$ `xvi` | [L0] |

Grammar14

### 3.1.3 Speaker agent in modified Kirby's model

From bilingual education in the real world, we found that mothers know that they can speak two languages. And they can indicate the vocabularies and sentence structures that from which languages. So, in Modified Kirby's model, All of rules in speaker agent's grammar must be defined the language by using L1, L2, L0 and UN.

Expressivity is the ratio of the utterable meanings to the whole meaning space. When we want to calculate expressivity of speaker agent in modified Kirby's model, we have to mention that which expressivity of which language that we want to know.

In case that calculating the expressivity of language1, first we have to find the number of utterable meanings by trying phrasing all meanings in the whole meaning space to language1 rules in agent's grammar. It means only L1 and L0 will be used. And then expressivity of language1 is ratio of the number of utterable meanings in language1 to the whole meaning space.

And also in case that calculating the expressivity of language2, we do the same process, but we try phrasing all meanings in the whole meaning space to language2 rules in agent's grammar. So, in this case, only L2 and L0 will be used.

### 3.1.4 Learner agent in modified Kirby's model

From bilingual education in the real world, we found that infants do not know that they have learnt two languages. They thought that they know only one language. They cannot indicate the vocabularies and sentence structures that from which languages. We can prove this hypothesis by observing the infant speaking. They often mix two languages in to one language. So, in Modified Kirby's model, All of rules in learner agent's grammar will be defined the language as UN.

To calculate expressivity of learner agent in modified Kirby's model, first we have to find the number of utterable meanings by trying phrasing all meanings in the whole meaning space to undefined rules in agent's grammar. It means only UN will be used. And then expressivity is ratio of the number of utterable meanings to the whole meaning space.

### 3.1.5 Speaking process and invention in modified Kirby's model

Speaking process and invention are used by speaker agent. Compare to original Kirby's model, there are some modified point in modified Kirby's model.

#### Speaking process

Speaking process is the process that is used for phase an intention to an utterance. The beginning of this process in modified Kirby's model is same as the original Kirby's model that is selecting an intention at random from the meaning space. After selecting an intention at random from the meaning space, next step is to randomly choose the language that agent is going to use. Then then selected intention will be phased by using speaker agent's grammar in the chosen language. Because in the real world, mothers must not use code-switching to talk with their infant, it means mothers must not mix two languages in each sentence. So in each uttered utterance will be generated from the rules in the same language.

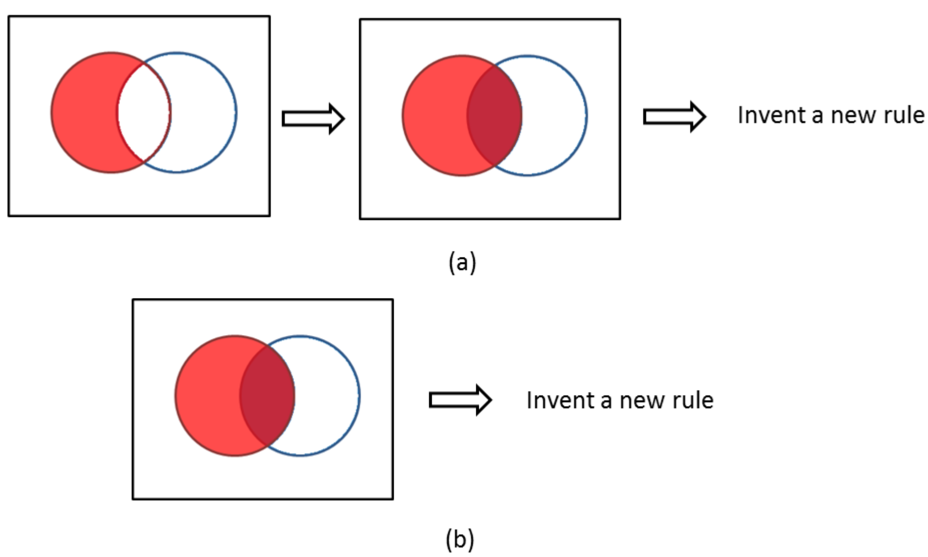


Figure 3.2: Procedure of speaking in Language1

If chosen language is language1, only rules in L1 and L0 will be used in this phasing. But there is a priority of using rule. In case of lexical rules, first we will use the rules in L1 only. If an utterance cannot be generated then allow agent use rules in both L1 and L0. If agent still cannot generate an utterance. It means that her grammar has not enough rules to generate utterance. So the invention process will start working. The flow of speaking process of lexical rules is shown as figure3.2(a). In case of compositional rules, first we will use the rules in both L1 and L0. If agent still cannot generate an utterance, the invention process will start working. The flow of speaking process of compositional rules is shown as figure3.2(b).

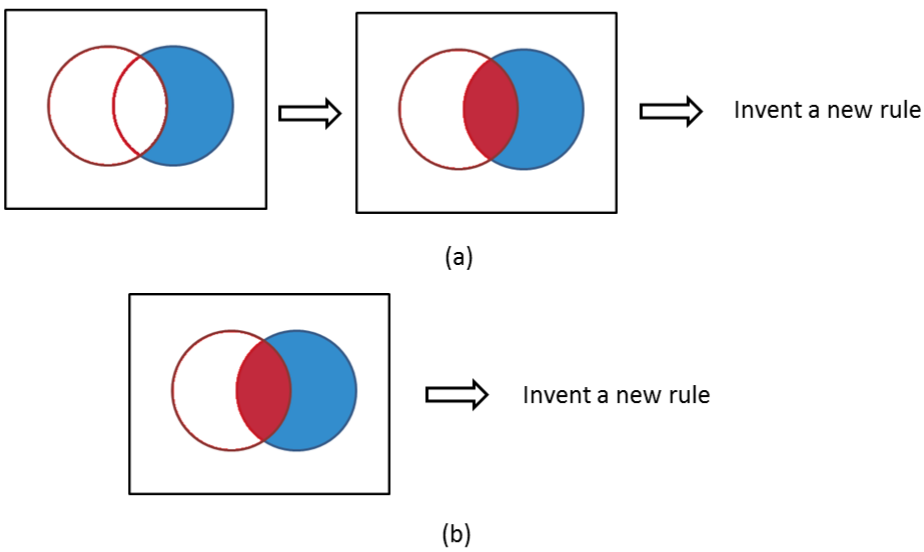


Figure 3.3: Procedure of speaking in Language2

If chosen language is language2, only rules in L2 and L0 will be used in this phasing. But there is a priority of using rule. In case of lexical rules, first we will use the rules in L2 only. If an utterance cannot be generated then allow agent use rules in both L2 and L0. If agent still cannot generate an utterance. It means that her grammar has not enough rules to generate utterance. So the invention process will start working. The flow of speaking process of lexical rules is shown as figure3.3(a). In case of compositional rules, first we will use the rules in both L2 and L0. If agent still cannot generate an utterance, the invention process will start working. The flow of speaking process of compositional rules is shown as figure3.3(b).

#### Invention

In modified Kirby's model, the invention process will be used by speaker agent who can differentiate two languages when speaker agent's grammar has not enough rules to generate utterance. The the invention process of modified Kirby's model is quite same as the invention process of original Kirby's model.

But in modified Kirby's model, speaker agent can differentiate two languages. So the new invented rules must be defined the language. To define the language for the new invented rules, it is up to the language that related in this time inventing. If invention worked when speaking process in language1 is processing, the new invented rule will be defined as L1. And if invention worked when speaking process in language2 is processing, the new invented rule will be defined as L2.

### 3.1.6 Learning process in modified Kirby's model

From the bilingual education in the real world, infants may confuse about two languages after learning. They can know the relation between each rule that may come from the same language by observing the frequency of the co-occurrence of vocabularies and sentence structure. And when time goes by, they can differentiate these two languages automatically.

The learning process is used by learner agent who cannot indicate the language of each rule in own grammar. In modified Kirby's model, learning process is same as original Kirby's model. But some feature was added that are front end process and separator.

The front end process is a process that simulates the observing the frequency of the co-occurrence of vocabularies and sentence structure in the real world. In modified Kirby's model, the front end process will evaluate the relation score of each pair of rules. Front end process will be applied every time after learning process finish. More detail about front end process was written in chapter 3.2 Front end process.

And separator is a process that simulates the growth of human that they can differentiate these two languages automatically after growth. In modified Kirby's model, separator will help learner agent differentiate the languages by using relation score that got from front end process. Finally separator will define the language of each rule in grammar as L1, L2, L0 and UN. Separator will be applied at once when learner agent change role into speaker agent. More detail about separator was written in chapter 3.3 Separator.

### 3.1.7 Simulation cycle in modified Kirby's model

Each generation in simulation goes through the following steps:

1. The speaker agent tries to speak for 100 times.
   - The speaker agent selects an intention at random from meaning space.
   - The speaker agent randomly choose the language that agent is going to speak.
   - If the speaker agent can generate a string for that meaning using own grammar, she does so, otherwise she invents a string. If the speaker has invented a string, the speaker uses that utterance as input to learner agent's learning.

- The learner agent learns by receiving utterance and speaker agent's intention.
- The learner agent reviews her learning. (Front end process works in this step)

2. The speaker agent's grammar is logged and then it is deleted from the simulation.
3. The learner agent apply separator.
4. The learner agent becomes the new speaker agent , and a new learner agent with a blank grammar is added to the simulation.

```
generation ≔ n;
speaker ≔ agent with grammar from n-th generation
learner ≔ new agent
repeat{
        for 1 to 100 do{
                intension ≔ random from meaning space
                language ≔ random between language1 and language2
                if(speaker.tryspeak(intension, language)){
                        utterance ≔ speaker.speak(intension, language)
                }else{
                        speaker.invent(intension, language)
                        utterance ≔ speaker.speak(intension, language)
                }
                learner.learn(utterance, intension)
                learner.review(utterance, intension)
        }
        speaker.saveLog()
        learner.applySeparator()
        generation++
        speaker ≔ learner
        learner ≔ new agent
}until n+200th generation
```

Algorithm2 : Pseudo-code of modified Kirby's model

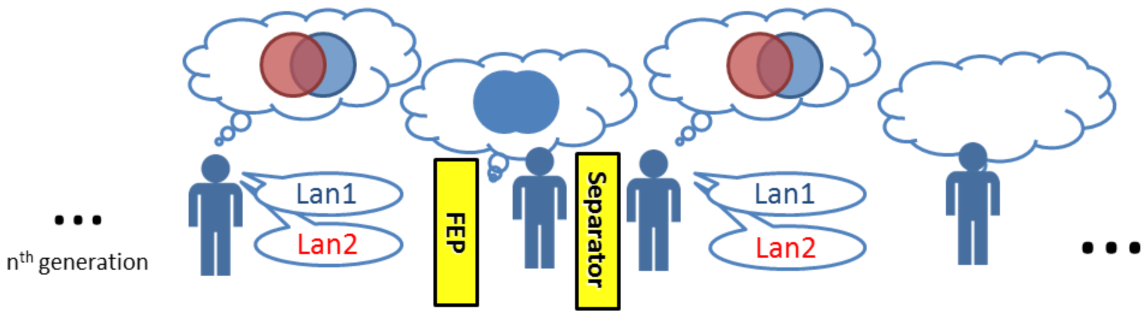


Figure 3.4: Simulation cycle of modified Kirby's model

## 3.2 Front end process

Front end process simulates the infant's thinking that observes the frequency of the co-occurrence of vocabularies and sentence structure to help differentiating the language in the future. So, in modified Kirby's model, front end process will be applied every time after learning process finish. The function of front end process is to evaluate the relation score of each pair of rules.

The meaning of the relation score is the possibility that both rules are same language. The higher score means there is higher possibility that both rules are same language. The score zero means the possibility that both rules are same language are low. For example, the relation score between rule(3) and rule(4) is five. And relation score between rule(3) and rule(5) is zero. This may imply that rule(3) and rule(4) tend to the same language but rule(5) is not. The relation score will be stored in relation score table as figure3.5.

| | rule1 | rule2 | rule3 | rule4 | rule5 | ... | rule n |
|---|---|---|---|---|---|---|---|
| rule1 | | 1 | 0 | 2 | 3 | | |
| rule2 | | | 0 | 0 | 0 | | |
| rule3 | | | | 5 | 0 | | |
| rule4 | | | | | 0 | | |
| rule5 | | | | | | | |
| ... | | | | | | | |
| rule n | | | | | | | |

Figure 3.5: Example of relation score table

The process to evaluate the relation score is call "review". After finish learning, learner agent has to review by try to express speaker's intention by using own grammar. If the utterance is same as speaker's utterance, learner agent will add relation score to each pairs of related rules in utterance generating process.

$$S/eat(cat,fish) \rightarrow \texttt{cdsai} \qquad (1)$$

Grammar15

Example1, imagine that there is a learner agent that have a grammar as grammar15. And she has to learn some knowledge from speaker agent that is `xopsai` utterance express *S/eat(cat,fish)* → `cdsai` intention. While learning, textitS/eat(dog,fish) → `xopsai` was added to her grammar and it did chunk operation with rule(1) in grammar15. And after finish learning process, learner agent has grammar like grammar16.

$$
\begin{array}{ll}
S/eat(x,fish) \rightarrow A/x\texttt{sai} & (1) \\
A/cat \rightarrow \texttt{cd} & (2) \\
A/dog \rightarrow \texttt{xop} & (3)
\end{array}
$$

Grammar16

Then review process start working. Learner agent try to express *eat(dog,fish)* intention by using own grammar. And she found that she used rule(1) and rule(3) to generate `xopsai` utterance. So the relation score between rule(1) and rule(3) will be added. And the relation score table can be drawn as figure3.6

| | rule1 | rule2 | rule3 |
|---|---|---|---|
| rule1 | | 0 | 1 |
| rule2 | | | 0 |
| rule3 | | | |

Figure 3.6: Relation score table of grammar16

Example2, (continue from example1) imagine that there is a learner agent that have a grammar as grammar16. And she has to learn some knowledge from speaker agent that is `kpmwo` utterance express *bite(snake,rat)* intention. While learning, *S/bite(snake,rat)* → `kpmwo` was added to her grammar. And after finish learning process, learner agent has grammar like grammar17.

$S/eat(x,fish) \rightarrow A/x\texttt{sai}$ (1)
$A/cat \rightarrow \texttt{cd}$ (2)
$A/dog \rightarrow \texttt{xop}$ (3)
$S/bite(snake,rat) \rightarrow \texttt{kpmwo}$ (4)

Grammar17

Then review process start working. Learner agent try to express *bite(snake,rat)* intention by using own grammar. And she found that she used only rule(4) to generate `kpmwo` utterance. So the relation score will not be added. And the relation score table can be updated as figure3.7

| | rule1 | rule2 | rule3 | rule4 |
|---|---|---|---|---|
| rule1 | | 0 | 1 | 0 |
| rule2 | | | 0 | 0 |
| rule3 | | | | 0 |
| rule4 | | | | |

Figure 3.7: Relation score table of grammar17

Example3, (continue from example2) imagine that there is a learner agent that have a grammar as grammar17. And she has to learn some knowledge from speaker agent that is `kpcdo` utterance express *bite(snake,cat)* intention. While learning, $S/bite(snake,cat) \rightarrow$ `kpcdo` was added to her grammar and it did chunk operation with rule(4) in grammar17. Then a product of this chunk that is $B/cat \rightarrow \texttt{cd}$ did merge operation with rule(2) in grammar17. And after finish learning process, learner agent has grammar like grammar18.

$S/eat(x,fish) \rightarrow A/x\texttt{sai}$ (1)
$A/cat \rightarrow \texttt{cd}$ (2)
$A/dog \rightarrow \texttt{xop}$ (3)
$S/bite(snake,x) \rightarrow \texttt{kp}A/x\texttt{o}$ (4)
$A/rat \rightarrow \texttt{mw}$ (5)

Grammar18

Then review process start working. Learner agent try to express *bite(snake,cat)* intention by using own grammar. And she found that she used used rule(2) and rule(4) to generate `kpcdo` utterance. So the relation score between rule(2) and rule(4) will be added. And the relation score table can be drawn as figure3.8

| | rule1 | rule2 | rule3 | rule4 | rule5 |
|---|---|---|---|---|---|
| rule1 | | 0 | 1 | 0 | 0 |
| rule2 | | | 0 | 0 | 1 |
| rule3 | | | | 0 | 0 |
| rule4 | | | | | 0 |
| rule5 | | | | | |

Figure 3.8: Relation score table of grammar18

## 3.3 Separator

From bilingual education, after learning, infant may confuse about two languages, but when they growth, they can differentiate these two languages automatically. In modified Kirby's model, separator is a clustering method[11-12] that simulates the growth of human. Separator will help learner agent differentiate the languages by using relation score that got from front end process. Separator will be applied at once when learner agent change role into speaker agent. The function of separator is to define the language of each rule in grammar as L1, L2, L0 or UN.

To define the language of each rule in grammar, the separator can be implemented follow these step.

1. Find the core of L1 and L2

- Core of L1 is a composition rule that has highest sum of score, and core of L2 is a composition rule that has second highest sum of score.

- Sum of score of each rule = sum of relation score between itself and other valid rules.

- Define language of rule that is core of L1 as L1 and define language of rule that is core of L2 as L2

2. Create a list of pairs of rule

- Sort by relation score

3. Repeat these step until the list is empty and all pains in list can be decided the language.

- consider the language of pairs of rule in the list

- If it has pattern like (UN,L1) (L1,UN) then define the undefined rule as L1 and remove this pair from the list. Because it means the undefined rule tend to be L1.

- If it has pattern like (UN,L2) (L2,UN) then define the undefined rule as L2 and remove this pair from the list Because it means the undefined rule tend to be L2.

- If it has pattern like (L1,L1) (L1,L0) (L0,L1) (L2,L2) (L2,L0) (L0,L2) (L0,L0) then do nothing and remove this pair from the list. Because it means there is no conflict in this pair of relation.

- If it has pattern like (L1,L2) (L2,L1), then calculate sum of score of both rule and change the language of the rule that has less sum of score to L0. Then remove this pair of relation. Because it means there is a conflict with these two rules. To avoid the conflict, one of them should change the language to L0.

- If it has pattern like (UN,UN) (L0,UN) (UN,L0), it means this pair cannot be decided the language now, skip this pair and consider this pair again when it can be decide the language.

```
define(highestOfSum(tableOfRelationScore),L1)
define(secondHighestOfSum(tableOfRelationScore),L2)
list ≔ queryList(tableOfRelationScore)
pos ≔ 0 //top of list
repeat{
        pair ≔ list.get ();
        switch(language of rule in pair){
                // define undefined rule
                case (L1,UN):   define(pair.y, L1)
                                pos ≔ 0      list.remove(pair)      break();
                case (UN,L1):   define(pair.x, L1)
                                pos ≔ 0      list.remove(pair)      break();
                case (L2,UN):   define(pair.y, L2)
                                pos ≔ 0      list.remove(pair)      break();
                case (UN,L2):   define(pair.x, L2)
                                pos ≔ 0      list.remove(pair)      break();

                // no conflict
                case (L1,L1):
                case (L1,L0):
                case (L0,L1):
                case (L2,L2):
                case (L2,L0):
                case (L0,L2):
                case (L0,L0):   pos ≔ 0      list.remove(pair)      break();

                // conflict
                case (L1,L2):
                case (L2,L1):   define(lowerSum(pair,L0)
                                pos ≔ 0      list.remove(pair)      break();

                // not yet decided (temporary skip this pair)
                case (UN,UN):
                case (L0,UN):
                case (UN,L0):   pos++        break();
        }
}while list is not empty and all pains in list can be decided the language.
```

Algorithm3 : Pseudo-code of separator

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | UN | L1 | UN | UN | L2 | UN | UN | UN | UN |

(a)

List : (1,2) (5,6) (2,7) (2,4) (1,7) (4,5) (3,9) (3,4) (3,5) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | UN | L2 | UN | UN | UN | UN |

(b)

List : (5,6) (2,7) (2,4) (1,7) (4,5) (3,9) (3,4) (3,5) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | UN | L2 | L2 | UN | UN | UN |

(c)

List : (2,7) (2,4) (1,7) (4,5) (3,9) (3,4) (3,5) (5,9)

Figure 3.9: Language table and rule number list (first loop to third loop)

Example of applying separator, assuming agent has nine rules in her grammar. The core of L1 is rule(2) and core of L2 is rule(5). And agent has a list of pairs of rule as figure3.9(a). Color of rule number represent the language of that rule. (red = L1, blue=L2, violet=L0 and black=UN)

First, the pair that has highest relative score that is (1,2) will be considered. The language of rule(1) is UN and the language of rule(2) is L1. It means rule(1) tend to be L1 same as rule(2). So rule(1) is defined as L1. And pair (1,2) is removed from list as shown in figure3.9(b).

Next, the pair (5,6) will be considered. The language of rule(5) is L2 and the language of rule(6) is UN. It means rule(6) tend to be L3 same as rule(5). So rule(6) is defined as L2. And pair (5,6) is removed from list as shown in figure3.9(c)

Then pair(2,7) and pair(2,4) are considered. And rule(7) and rule(4) are defined as L1. And then these two pair are removed from the list as shown in figure3.10(a) and figure3.10(b) in order.

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | UN | L2 | L2 | L1 | UN | UN |

(a)

List : (2,4) (1,7) (4,5) (3,9) (3,4) (3,5) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | L1 | L2 | L2 | L1 | UN | UN |

(b)

List : (1,7) (4,5) (3,9) (3,4) (3,5) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | L1 | L2 | L2 | L1 | UN | UN |

(c)

List : (4,5) (3,9) (3,4) (3,5) (5,9)

Figure 3.10: Language table and rule number list (fourth loop to sixth loop)

Next, in figure3.10(b), the pair (1,7) will be considered. Both language of rule(1) and language of rule(7) are L1. It means there is no conflict in this case. So we do nothing , just removed (1,7) from list and go to next step.

Then, the pair (4,5) in figure3.10(c) will be considered. The language of rule(4) is L1 but the language of rule(5) is L2. The conflict occurred. The solution to solve this conflict is to change language of one of them to be L0, because L0 can co-occur with both L1 and L2. The changed rule is the one that weaker. It means the one that has lower sum of relation score will be change language. In this case, the sum of relation score of rule(4) is lower than rule(5). So the language of rule(4) will change to be L0 as shown in figure3.11(a).

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | L0 | L2 | L2 | L1 | UN | UN |

(a)

List : (3,9) (3,4) (3,5) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | L0 | L2 | L2 | L1 | UN | UN |

(b)

List : (3,9) (3,4) (3,5) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | UN | L0 | L2 | L2 | L1 | UN | UN |

(c)

List : (3,9) (3,4) (3,5) (5,9)

Figure 3.11: Language table and rule number list (seventh loop to ninth loop)

Next, in figure3.11(a), the pair (3,9) will be considered. Both language of rule(3) and language of rule(9) are UN. So, in this case we still cannot decide the language of rule now. And the pair (3,9) and consider pair (3,4).

In figure3.11(b), the pair (3,4) is considered. The language of rule(3) is UN and the language of rule(4) is L0. So, this case has also not yet decided. We will skip this pair and go to the next pair that is (3,5) in figure3.11(c)

The pair (3,5) is considered. The undefined rule(3) are tend to be L2 same as rule(5). So rule(3) are defined as L2. Then pair (3,5) are removed from the list. And next is to go back to consider pair (3,9) again as shown in figure3.12(a)

Let's consider the pair (3,9) again. This time, the language of rule(3) is L2. So we can decide the language of rule(9). It is defined as L2 same as rule(3). Then remove pair(3,9) and consider(3,4) again as shown in figure3.12(c). There is no conflict in this time considering. So remove (3,4) and consider (5,9) next. It also has on conflict. So do nothing just remove pair(5,9) from the list. Then the list is empty as shown in figure3.12(d) Then separator process is finished.

From figure3.12(d), separator can differentiate the rule by using the score from front end process. The result is that there are tree rules in L1 (1,2,7), four rules in L2 (3,5,6,9), one rule in L0 (4) and one undefined rule (8).

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | L2 | L0 | L2 | L2 | L1 | UN | UN |

(a)

List : (3,9) (3,4) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | L2 | L0 | L2 | L2 | L1 | UN | L2 |

(b)

List : (3,4) (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | L2 | L0 | L2 | L2 | L1 | UN | L2 |

(c)

List : (5,9)

| Rule No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Language | L1 | L1 | L2 | L0 | L2 | L2 | L1 | UN | L2 |

(d)

List :

Figure 3.12: Language table and rule number list (tenth loop to thirteenth loop)

# Chapter 4

# Experiments, Result and Analysis

To show the performance of Proposed method, the result of experiments will be compared. There are 3 experiments in this research.

1. Kirby's model in bilingual environment
2. Modified Kirby's model (n-th generation grammar has no rules that can be both languages)
3. Modified Kirby's model (n-th generation grammar has rules that can be both languages)

We compare experiment1 with experiment2 to observe the effectiveness of proposed method. And compare experiment2 with experiment3 to observe the performance.

## 4.1 Kirby's model in bilingual environment

In this experiment, we use the Kirby's model in bilingual environment. The n-th generation contains two languages that has no rules that can be both languages (no L0 rules)as grammar19. By the way, both speaker agent and learner agent can't differentiate that there are two languages in this experiment. The front end process and separator are not applied in this experiment.

| | | | | | |
|---|---|---|---|---|---|
| $S/p(x,y) \rightarrow A/xgA/yB/p$ | | [L1] | $S/p(x,y) \rightarrow A/xB/pA/y$ | | [L2] |
| $A/noun0 \rightarrow$ | `sy` | [L1] | $A/noun0 \rightarrow$ | `oac` | [L2] |
| $A/noun4 \rightarrow$ | `n` | [L1] | $A/noun1 \rightarrow$ | `v` | [L2] |
| $A/noun2 \rightarrow$ | `iw` | [L1] | $A/noun2 \rightarrow$ | `bz` | [L2] |
| $A/noun1 \rightarrow$ | `fsr` | [L1] | $A/noun3 \rightarrow$ | `aw` | [L2] |
| $A/noun3 \rightarrow$ | `u` | [L1] | $A/noun4 \rightarrow$ | `r` | [L2] |
| $B/verb5 \rightarrow$ | `d` | [L1] | $B/verb5 \rightarrow$ | `xk` | [L2] |
| $B/verb6 \rightarrow$ | `bd` | [L1] | $B/verb6 \rightarrow$ | `lq` | [L2] |
| $B/verb8 \rightarrow$ | `oft` | [L1] | $B/verb7 \rightarrow$ | `t` | [L2] |
| $B/verb7 \rightarrow$ | `k` | [L1] | $B/verb8 \rightarrow$ | `bqp` | [L2] |
| $B/verb9 \rightarrow$ | `ke` | [L1] | $B/verb9 \rightarrow$ | `ybr` | [L2] |

Grammar19

The result got worse since generation n+1. The number of rule increased from 28 to 108 and expressivity decrease from 100% to 76%. The grammar in n-th generation and grammar in n+1 th generation are almost completely different because agent cannot make a complex compositional rule and agent often do operation with the wrong place. When compare grammar in n+1 th generation to the grammar in n-th generation by phasing same intensions, some generated utterances was different. Some generated utterances did not change. The grammar of experiment1 in generation n+1 in detail was written in Appendix A.1

In the generation n+9, grammar has 38 rules with expressivity 100%. But when compare to the grammar in n-th generation by phasing same intensions, almost of generated utterances was different. The grammar of experiment1 in generation n+9 in detail was written in Appendix A.2

From this experiment, It may conclude that agent combined both language into one language and invent it into own new language.

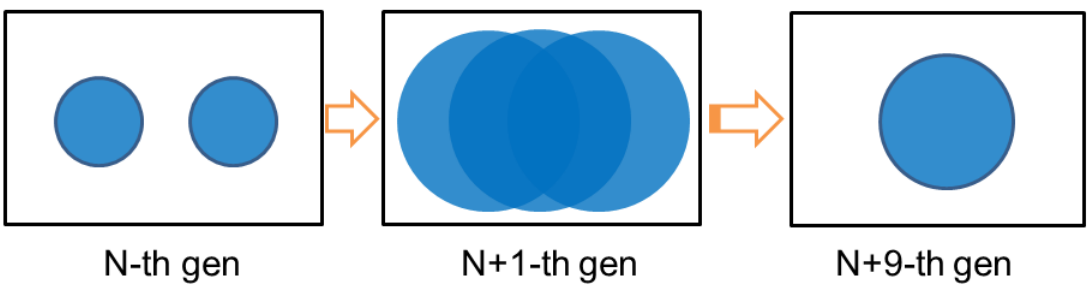


Figure 4.1: Result of experiment1

## 4.2 Modified Kirby's model (n-th generation grammar has no rules that can be both languages)

In this experiment, we use the Modified Kirby's model in bilingual environment. The n-th generation contains two languages that has no rules that can be both languages (no L0 rules) as grammar18. Although the experiment2 use the same input as experiment1, but in this experiment will use Proposed modified Kirby's model instead of original Kirby's model. So speaker agents can differentiate that there are two languages in their grammar. And the front end process and separator are also applied in this experiment.

The result in generation n+1, there are 23 rules in grammar that are 11rules in L1 and 11 rules in L2 and 1 rule in UN. And the agent can maintain the expressivity in both two languages at 100%. The grammar of experiment2 in generation n+1 in detail was written in Appendix A.3

Not only the result in generation n+1, but also the result in generation n+20, there are 24 rules in grammar that are 11 rules in L1 and 11 rules in L2 and 2 rules in UN. And the agent can maintain the expressivity in both two languages at 100%. The grammar of experiment2 in generation n+20 in detail was written in Appendix A.4

From this experiment, It can be assumed that the front end process and separator works and produces results as expected with input that n-th generation grammar has no rules that can be both languages. And it can prove that the front end process and separator can work when compare the result from this experiment to the result of experiment1.

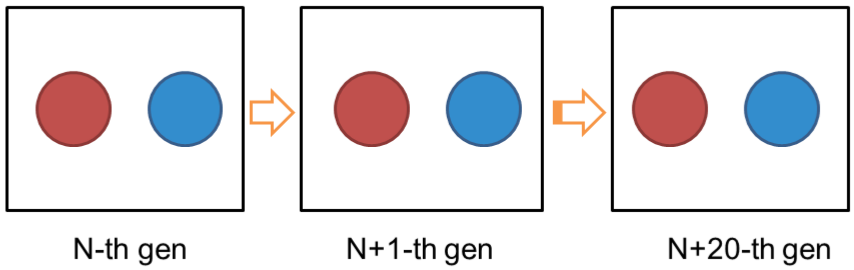


Figure 4.2: Result of experiment2

## 4.3 Modified Kirby's model (n-th generation grammar has rules that can be both languages)

In this experiment, we use the Modified Kirby's model in bilingual environment. The n-th generation contains two languages that has some rules that can be both languages. It means that L0 rule will appear in grammar of n-th generation. This experiment try using grammars of n-th generation that have various number of L0 rules from 1 and increase it to 2, 3 and more.

First, we decide to use the grammar that contain 10 rules in L1, 10 rule in L2 and 1 rule in L0 as n-th generation like grammar20

| | | | |
|---|---|---|---|
| $S/p(x,y) \rightarrow A/xgA/yB/p$ | [L1] | $A/noun1 \rightarrow$ `v` | [L2] |
| $A/noun4 \rightarrow$ `n` | [L1] | $A/noun2 \rightarrow$ `bz` | [L2] |
| $A/noun2 \rightarrow$ `iw` | [L1] | $A/noun3 \rightarrow$ `aw` | [L2] |
| $A/noun1 \rightarrow$ `fsr` | [L1] | $A/noun4 \rightarrow$ `r` | [L2] |
| $A/noun3 \rightarrow$ `u` | [L1] | $B/verb5 \rightarrow$ `xk` | [L2] |
| $B/verb5 \rightarrow$ `d` | [L1] | $B/verb6 \rightarrow$ `lq` | [L2] |
| $B/verb6 \rightarrow$ `bd` | [L1] | $B/verb7 \rightarrow$ `t` | [L2] |
| $B/verb8 \rightarrow$ `oft` | [L1] | $B/verb8 \rightarrow$ `bqp` | [L2] |
| $B/verb7 \rightarrow$ `k` | [L1] | $B/verb9 \rightarrow$ `ybr` | [L2] |
| $B/verb9 \rightarrow$ `ke` | [L1] | | |
| $S/p(x,y) \rightarrow A/xB/pA/y$ | [L2] | $A/noun0 \rightarrow$ `sy` | [L0] |

Grammar20

The result in generation n+1, there are 25 rules in grammar that are 10 rules in L1 and 10 rules in L2, 1 rule in L0 and 4 rules in UN. And the agent can maintain the expressivity in both two languages at 100 %. The more detail was written in Appendix A.5

Next, we change the grammar of n-th generation by increase the number of L0 rule from 1 to 2, 3, 4, 5 and 6 and decrease the number L1 rule and L2 rule from 10 to 9, 8, 7, 6 and 5 in order.

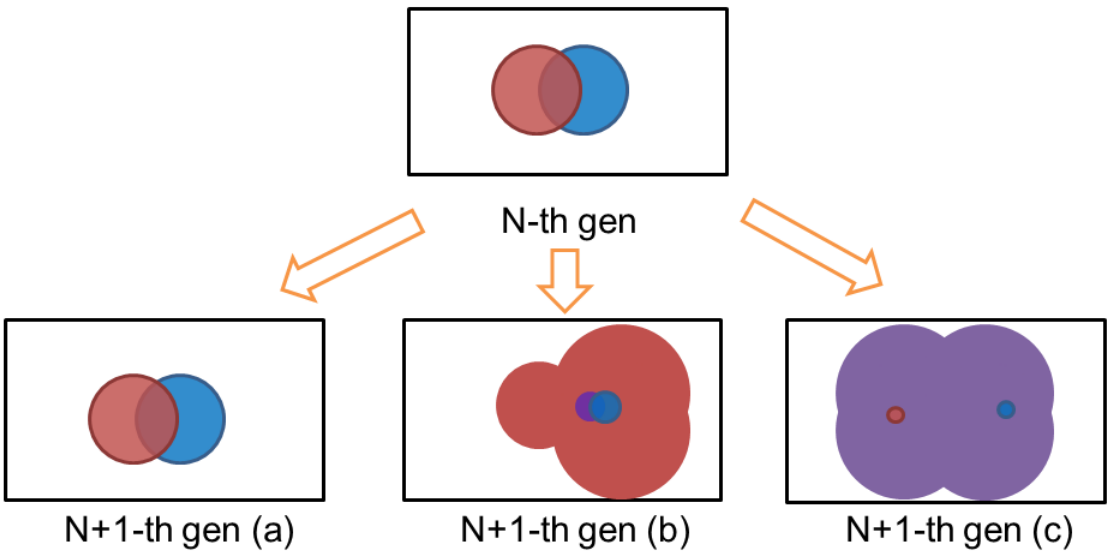


Figure 4.3: Result of experiment3

We found that front end process and separator can work as expect in the case of the number of L0 rule in grammars of n-th generation is 1, 2, 3. But when the number of L0 rule in grammars of n-th generation is 4, front end process and separator can show their good performance, but sometime they cannot work well. And when the number of L0 rule in grammars of n-th generation is increased more than4, the chance of failure is also increased.

The reasons that result was not as expected is the common use area of two languages (L0) is too wide. It causes learner agent use chunk operation with two rules from different language and the composition rules that should be L0 will added to grammar. This composition rules may have high opportunity to co-occurrence with other rules and sometime they have high some of relation score. When separator applied, separator will pick up the core of L1 and L2. If the composition rules that should be L0 are picked up to be core of L1 or L2, then the results will be not as expected.

# Chapter 5

# Conclusion

This research, we set the objective is to design and implement some conditions settings for make the Kirby's model support bilingual environments. For this objective, we Proposed the method. First of all, we modified the Kirby's model according to bilingual education in the real world. Then front end process was design as scoring system to evaluate the relation score of each rule by observing co-occurrence of using rules. And we implemented the separator to help agent differentiate the languages by using relation score table that got from front end process.

After applying our proposed method, languages in former generation can be transmitted to the next generation. It shows the best performance when using the grammar that has no common area between two languages as grammar of n-th generation. The key of our proposed method is to differentiate two languages by observing the frequency of the co-occurrence of vocabularies and sentence structure.

Our future work, we plan to develop the performance of our proposed method by adding some bias to prevent learner agent use operations with rules from different language

# Bibliography


[1] Kirby, S. : Learning, bottlenecks and the evolution of recursive syntax. Linguistic Evolution through Language Acquisition. Cambridge University Press, Cambridge (2002)

[2] Ryuichi Matoba, Makoto Nakamura, Satoshi Tojo : Efficiency of the Symmetry Bias in Grammar Acquisition. Information and Computation (2010)

[3] Kenny Smith and James R. Hurford : Language Evolution in Populations: extending the Iterated Learning Model, ECAL03 (Vol.2801,p.507-516) (2003)

[4] M. Nakamura et al., Multilayered Formalisms for Kanguage Cintract, In: EvolangIX Workshop: Constructive Approaches to Language Evolution, Five Approaches to Language Evolution: Proceedings of the Workshop of the 9th International Conference on the Ecolution of Language, pp.145-147, Kyoto,2012

[5] Shingo Hagiwara : LEILA system

[6] Grace M. Libardo Alvarez, M.S., CCC-SLP : A Child's Journey to Bilingualism Simultaneous Dual LanguageDevelopment. Multilingual Living Magazine (2006)

[7] Yunisrina Qismullah Yusuf : A Case Study on Infant Bilingual Acquisition, MJAL 1 (2009)

[8] Kirby, S. (2007). The evolution of language. In Dunbar, R. and Barrett, L., editors, Oxford Handbook of Evolutionary Psychology, pages 669?681.

[9] Hagen, L. Kirk."The Bilingual Brain: Human Evolution and Second Language Acquisition" Evolutionary Psychology. 2008.

[10] Janet Goldenstein Ahler. "THE EVOLUTION OF BILINGUAL EDUCATION IN AN AMERICAN INDIAN COMMUNITY: A DECADE OF EVALUATION AS APPLIED ANTHROPOLOGY" , Great Plains Research 4 (August 1994) p.293-303

[11] Qunhui Li.,"Clustering by Intersection-Mergin", UW Statistics Technical Report 451, with Qunhua Li

[12] Hua Yu, Automatically Determining Number of Clusters. Information Retrieval (CMU CS11-741) Final Report, Apr.1998

# Appendix A

## A.1 The grammar of experiment1 in generation n+1

Generation = n+1
Number of rules = 108
Expressivity = 76%
***********************
141 S/p(x,noun1) → B/xR/p
45 S/p(noun0,noun3) → oC/p
193 S/p(noun0,noun3) → syX/p
116 S/p(noun1,y) → O/pB/y
119 S/p(noun2,y) → A/ywgvyO/pr
53 S/p(noun2,noun1) → uE/p
197 S/p(noun2,noun4) → iwY/p
146 S/p(noun4,noun3) → rgawyO/pr
147 S/verb5(x,noun4) → B/xdn
92 S/p(noun4,noun0) → nL/p
56 S/verb5(noun0,y) → F/yxk
151 S/verb5(noun0,noun1) → jrs
33 S/verb5(noun0,noun2) → f
169 S/verb5(noun1,noun3) → fsrdaw
7 S/verb5(noun1,noun4) → pyg
190 S/verb5(noun2,y) → W/yv
149 S/verb5(noun2,noun4) → ow
173 S/verb5(noun3,y) → ugU/y
200 S/verb5(noun3,y) → awZ/y
185 S/verb5(noun3,noun1) → s
77 S/verb5(noun3,noun0) → la
153 S/verb5(noun4,noun2) → ngbzd
6 S/verb5(noun1,noun2) → hs
139 S/verb6(x,y) → K/xbdB/y
64 S/verb6(noun0,noun4) → hf
150 S/verb6(noun1,noun2) → fsrgbzlq

40 S/verb6(noun2,noun0) → khi
12 S/verb6(noun1,noun2) → nt
18 S/verb6(noun3,noun1) → gw
74 S/verb6(noun3,noun2) → seg
168 S/verb6(noun4,noun3) → v
84 S/verb7(x,noun1) → yH/xv
85 S/verb7(x,noun2) → J/xz
127 S/verb7(x,noun3) → B/xku
95 S/verb7(noun1,y) → fsrgK/yt
13 S/verb7(noun1,noun2) → x
131 S/verb7(noun2,y) → bzgB/yk
10 S/verb7(noun2,noun4) → v
177 S/verb7(noun3,y) → awV/y
165 S/verb7(noun4,noun0) → yr
26 S/verb7(noun4,noun3) → u
138 S/verb8(x,y) → K/xgB/yoft
29 S/verb8(x,noun0) → nA/x
184 S/verb8(noun1,noun0) → vgsybqp
171 S/verb8(noun1,noun4) → dxd
81 S/verb8(noun2,y) → I/ybqp
103 S/verb8(noun2,noun0) → z
38 S/verb8(noun3,noun1) → b
3 S/verb7(noun3,noun2) → xb
196 S/verb9(x,noun3) → gW/x
23 S/verb9(noun0,noun4) → i
161 S/verb9(noun1,y) → vH/y
100 S/verb9(noun1,noun2) → rc
167 S/verb9(noun2,noun3) → iwguybr
152 S/verb9(noun2,noun3) → bzkeu
65 S/verb9(noun4,noun0) → z
154 S/verb9(noun4,noun1) → rybrfsr
28 S/verb9(noun4,noun2) → yx
2 S/verb8(noun4,noun1) → h
1 S/verb5(noun4,noun3) → zgq
31 A/noun1 → i
30 A/noun3 → s
115 B/noun0 → sy
129 B/noun0 → oac
128 B/noun2 → iw
43 B/noun2 → bz
44 B/noun4 → r
46 C/verb5 → qi
47 C/verb7 → actaw

55 E/verb8 → qz
54 E/verb9 → a
57 F/noun2 → sygiw
58 F/noun3 → oacgaw
59 F/noun4 → sygr
79 H/noun0 → t
80 H/noun2 → giwybr
164 H/noun3 → guybr
83 I/noun3 → bzgu
82 I/noun4 → iwgn
86 J/noun0 → gy
87 J/noun1 → vkb
90 K/noun3 → aw
89 K/noun4 → n
93 L/verb6 → x
94 L/verb7 → gsyk
118 O/verb5 → fsrd
117 O/verb9 → b
142 R/verb6 → gfsrlq
145 R/verb6 → bdv
144 R/verb7 → gfsrk
143 R/verb9 → ybrv
176 U/noun2 → iwd
175 U/noun3 → awxk
174 U/noun4 → nxk
182 V/noun0 → toac
180 V/noun0 → tsy
181 V/noun2 → tiw
178 V/noun2 → tbz
183 V/noun4 → gnk
179 V/noun4 → tr
191 W/noun0 → nf
192 W/noun1 → iwd
194 X/verb6 → gulq
195 X/verb9 → keaw
199 Y/verb6 → lqr
198 Y/verb8 → oftn
201 Z/noun0 → dsy
202 Z/noun4 → xkr
************************

## A.2 The grammar of experiment1 in generation n+9

Generation = n+9

Number of rules = 38

Expressivity = 100%

************************

330 S/p(x,y) → bdA/xgA/yF/p

331 S/p(x,y) → A/xgA/yF/p

200 S/p(noun0,y) → V/yF/p

54 S/p(noun3,y) → E/yF/p

333 S/verb6(x,y) → C/xA/y

312 S/verb8(x,y) → zgwbC/xgA/yofguybr

332 S/verb8(x,noun3) → A/xlqgwtoft

316 A/noun0 → sy

319 A/noun0 → oac

318 A/noun1 → j

320 A/noun3 → bdwt

317 A/noun2 → bz

315 A/noun2 → iw

314 A/noun3 → wt

313 A/noun4 → r

306 C/noun1 → bk

304 C/noun2 → iwlq

310 C/noun3 → oqi

305 C/noun4 → n

309 C/noun4 → jxi

36 E/noun0 → wtgsy

37 E/noun2 → bdwtgbz

40 F/verb5 → xk

174 F/verb6 → rd

175 F/verb7 → k

39 F/verb8 → e

153 F/verb9 → ngv

207 V/noun0 → bdsygwwtgbz

210 V/noun0 → bdoacgwwtgbz

209 V/noun1 → bdjgwwtgbz

201 V/noun1 → sygbdj

202 V/noun2 → bzgwwtgbz

206 V/noun2 → bdiwgwwtgbz

208 V/noun2 → bdbzgwwtgbz

203 V/noun3 → wtgwwtgbz

205 V/noun3 → bdwtgwwtgbz

211 V/noun3 → bdbdwtgwwtgbz

204 V/noun4 → bdrgwwtgbz

***********************

## A.3 The grammar of experiment2 in generation n+1

Generation = n+1

Number of rules = 23
Number of L1 rules = 11
Number of L2 rules = 11
Number of L0 rules = 0
Number of UN rules = 1
Expressivity in L1 = 100%
Expressivity in L2 = 100%
***********************

245 S/p(x,y) → A/xgA/yZ/p
11 A/noun0 → sy
40 A/noun2 → iw
49 A/noun3 → u
12 A/noun1 → fsr
19 A/noun4 → n
235 Z/verb5 → d
233 Z/verb7 → k
231 Z/verb6 → bd
234 Z/verb8 → oft
232 Z/verb9 → ke
language1 = 11
***********************

214 S/p(x,y) → H/xH/x/pH/y
70 H/noun0 → oac
123 H/noun1 → v
71 H/noun2 → bz
139 H/noun3 → aw
119 H/noun4 → r
220 X/verb5 → xk
215 X/verb6 → lq
216 X/verb7 → t
219 X/verb8 → bqp
217 X/verb9 → ybr
language2 = 11
***********************

language0(both 1 and 2) = 0
***********************

236 S/p(noun0,noun3) → syguZ/p
undefined = 1

************************

relationScore ruleNo ruleNo
4 11 12
6 11 19
5 11 40
10 11 49
9 11 231
3 11 232
6 11 233
4 11 234
3 11 235
50 11 245
7 12 19
7 12 40
8 12 49
5 12 231
8 12 232
5 12 233
5 12 234
3 12 235
52 12 245
4 19 40
5 19 49
5 19 231
8 19 232
3 19 233
5 19 234
1 19 235
44 19 245
6 40 49
8 40 231
3 40 232
7 40 233
4 40 234
44 40 245
11 49 231
6 49 232
3 49 233
6 49 234
3 49 235
58 49 245
3 70 71
6 70 119
4 70 123
1 70 139
28 70 214
2 70 215
4 70 216
3 70 217
3 70 219
2 70 220
3 71 119
4 71 123
3 71 139
26 71 214
2 71 215
2 71 216
5 71 217
2 71 219
2 71 220
4 119 123
4 119 139
34 119 214
2 119 215
3 119 216
4 119 217
7 119 219
1 119 220
6 123 139
36 123 214
2 123 215
4 123 216
8 123 217
1 123 219
3 123 220
28 139 214
2 139 215
3 139 216
6 139 217
3 139 219
10 214 215
16 214 216
26 214 217
16 214 219
8 214 220

38 231 245
28 232 245
24 233 245
24 234 245
10 235 245

## A.4 The grammar of experiment2 in generation n+20

Generation = n+20
Number of rules = 24
Number of L1 rules = 11
Number of L2 rules = 11
Number of L0 rules = 0
Number of UN rules = 2
Expressivity in L1 = 100%
Expressivity in L2 = 100%
***********************
233 S/p(x,y) → D/xZ/pD/y
45 D/noun0 → oac
80 D/noun1 → v
52 D/noun3 → aw
37 D/noun2 → bz
36 D/noun4 → r
237 Z/verb5 → xk
238 Z/verb7 → t
236 Z/verb9 → ybr
234 Z/verb6 → lq
235 Z/verb8 → bqp
language1 = 11
***********************
201 S/p(x,y) → A/xgA/yM/p
19 A/noun0 → sy
12 A/noun1 → fsr
126 A/noun2 → iw
133 A/noun3 → u
11 A/noun4 → n
188 M/verb5 → d
189 M/verb6 → bd
117 M/verb7 → k
187 M/verb8 → oft
118 M/verb9 → ke
language2 = 11
***********************
language0(both 1 and 2) = 0
***********************

193 S/p(noun1,y) → fsD/ygnM/p
128 S/p(noun2,noun4) → iwgnM/p
undefined = 2
************************

relationScore ruleNo ruleNo
7 11 12
7 11 19
8 11 117
7 11 118
5 11 126
8 11 133
1 11 187
6 11 188
5 11 189
54 11 201
3 12 19
2 12 117
2 12 118
2 12 126
2 12 133
3 12 187
2 12 188
5 12 189
28 12 201
4 19 117
3 19 118
9 19 126
1 19 133
4 19 187
7 19 188
2 19 189
40 19 201
5 36 37
9 36 45
6 36 52
2 36 80
44 36 233
6 36 234
6 36 235
2 36 236
3 36 237
5 36 238
5 37 45
9 37 52
4 37 80
46 37 233
3 37 234
4 37 235
3 37 236
6 37 237
7 37 238
7 45 52
3 45 80
48 45 233
4 45 234
5 45 235
5 45 236
7 45 237
3 45 238
4 52 80
52 52 233
6 52 234
3 52 235
4 52 236
8 52 237
5 52 238
26 80 233
3 80 234
2 80 235
2 80 236
4 80 237
2 80 238
6 117 126
2 117 133
22 117 201
4 118 126
4 118 133
20 118 201
2 126 133
2 126 187
2 126 188
4 126 189

36 126 201
2 133 187
3 133 188
2 133 189
26 133 201
12 187 201
20 188 201
18 189 201
22 233 234
20 233 235
16 233 236
28 233 237
22 233 238

## A.5 The grammar of experiment3 in generation n+1 (number of L0=1)

Generation = n+1
Number of rules = 25
Number of L1 rules = 10
Number of L2 rules = 10
Number of L0 rules = 1
Number of UN rules = 4
Expressivity in L1 = 100%
Expressivity in L2 = 100%
************************
256 S/p(x,y) → A/xF/pA/y
138 A/noun1 → v
89 A/noun2 → bz
221 A/noun4 → r
74 A/noun3 → aw
225 F/verb5 → xk
193 F/verb6 → lq
234 F/verb7 → t
56 F/verb8 → bqp
57 F/verb9 → ybr
language1 = 10
************************
239 S/p(x,y) → A/xgA/yX/p
116 A/noun2 → iw
81 A/noun1 → fsr
36 A/noun3 → u
19 A/noun4 → n
240 X/verb5 → d
242 X/verb6 → bd
241 X/verb7 → k
244 X/verb8 → oft
243 X/verb9 → ke

language2 = 10
***********************

20 A/noun0 $\rightarrow$ sy
language0(both 1 and 2) = 1
***********************

151 S/p(x,noun2) $\rightarrow$ A/xG/p
62 G/verb6 $\rightarrow$ lqbz
60 G/verb7 $\rightarrow$ tbz
61 G/verb9 $\rightarrow$ giwke
undefined = 4
***********************

relationScore ruleNo ruleNo
4 19 20
4 19 36
4 19 81
6 19 116
36 19 239
4 19 240
2 19 241
5 19 242
3 19 243
4 19 244
5 20 36
4 20 56
3 20 57
3 20 74
6 20 81
7 20 89
5 20 116
5 20 138
6 20 193
6 20 221
2 20 225
6 20 234
40 20 239
3 20 240
6 20 241
1 20 242
5 20 243
5 20 244
42 20 256
5 36 81
6 36 116
40 36 239
1 36 240
7 36 241
5 36 242
2 36 243
5 36 244
6 56 74
5 56 89
4 56 138
5 56 221
24 56 256
2 57 74
4 57 89
3 57 138
8 57 221
20 57 256
7 74 89
5 74 138
5 74 193
3 74 221
2 74 225
3 74 234
36 74 256
3 81 116
36 81 239
3 81 240
4 81 241
3 81 242
3 81 243
5 81 244
4 89 138
3 89 193

6 89 221
6 89 225
6 89 234
48 89 256
40 116 239
5 116 240
7 116 241
2 116 242
3 116 243
3 116 244
6 138 193
6 138 221
6 138 225
1 138 234
40 138 256
2 193 221
22 193 256
2 221 225
4 221 234
42 221 256
18 225 256
20 234 256
16 239 240
26 239 241
16 239 242
16 239 243
22 239 244

## A.6 The grammar of experiment3 in generation n+1 (number of L0=2)

Generation = n+1
Number of rules = 26
Number of L1 rules = 9
Number of L2 rules = 9
Number of L0 rules = 2
Number of UN rules = 6
Expressivity in L1 = 100%
Expressivity in L2 = 100%
************************
391 S/p(x,y) → A/xC/pA/y
340 A/noun1 → v
338 A/noun4 → r
337 A/noun2 → bz
332 A/noun3 → aw
314 C/verb6 → lq
313 C/verb9 → ybr
311 C/verb7 → t
312 C/verb8 → bqp
language1 = 9
************************
370 S/p(x,y) → A/xgA/yC/p
335 A/noun1 → fsr
336 A/noun3 → u
334 A/noun4 → n

333 A/noun2 → iw
318 C/verb6 → bd
319 C/verb7 → k
316 C/verb8 → oft
315 C/verb9 → ke
language2 = 9
************************
339 A/noun0 → sy
317 C/verb5 → xk
language0(both 1 and 2) = 2
************************
353 S/verb5(x,y) → A/xP/y
120 P/noun1 → gfsrxk
122 P/noun1 → xkv
121 P/noun3 → guxk
119 P/noun3 → xkaw
123 P/noun4 → xkr
undefined = 6
****************

relationScore ruleNo ruleNo
5 311 332
3 311 337
3 311 338
2 311 339
5 311 340
18 311 391
2 312 332
6 312 337
4 312 338
4 312 339
6 312 340
22 312 391
5 313 332
3 313 337
4 313 338
1 313 339
5 313 340
18 313 391
5 314 332
4 314 337
6 314 338
4 314 339
9 314 340
28 314 391
3 315 333
5 315 334
5 315 335
6 315 336
3 315 339
22 315 370
3 316 334
2 316 335
5 316 336
4 316 339
14 316 370
4 317 332
3 317 333
3 317 334
5 317 335
4 317 336
2 317 337
5 317 338
13 317 339
3 317 340
22 317 370
20 317 391
2 318 333

4 318 334
3 318 335
2 318 336
3 318 339
14 318 370
4 319 333
2 319 334
4 319 335
6 319 336
6 319 339
22 319 370
5 332 337
6 332 338
3 332 339
7 332 340
42 332 391
1 333 334
6 333 335
4 333 336
1 333 339
24 333 370
6 334 335
5 334 336
5 334 339
34 334 370
2 335 336
5 335 339
38 335 370
12 336 339
46 336 370
4 337 338
4 337 339
5 337 340
36 337 391
3 338 339
9 338 340
44 338 391
7 339 340
46 339 370
34 339 391
56 340 391

## A.7 The grammar of experiment3 in generation n+1 (number of L0=3)

Generation = n+1
Number of rules = 35
Number of L1 rules = 7
Number of L2 rules = 10
Number of L0 rules = 4
Number of UN rules = 14
Expressivity in L1 = 100%
Expressivity in L2 = 90%
************************
218 S/p(x,y) → A/xgA/yE/p
95 A/noun1 → fsr
130 A/noun2 → iw
94 A/noun3 → u
93 A/noun4 → n
224 E/verb6 → bd
47 E/verb9 → ke
language1 = 7

************************

184 S/p(x,y) → A/xE/pA/y
244 S/verb9(x,y) → A/xybV/y
259 S/verb9(noun0,y) → syybrA/y
11 A/noun1 → v
161 A/noun2 → bz
162 A/noun4 → r
12 A/noun3 → aw
168 E/verb6 → lq
245 V/noun3 → raw
254 V/noun4 → rr
language2 = 10
************************

92 A/noun0 → sy
48 E/verb5 → xk
73 E/verb7 → t
106 E/verb8 → of
language0(both 1 and 2) = 4
************************

109 S/p(x,noun2) → A/xJ/p
112 J/verb5 → giwxk
110 J/verb6 → lqbz
113 J/verb7 → giwt
114 J/verb8 → giwof
111 J/verb9 → giwke
248 V/noun0 → syr
251 V/noun1 → fsrr
246 V/noun1 → vr
253 V/noun2 → bzr
252 V/noun2 → iwr
250 V/noun3 → ur
247 V/noun3 → awr
249 V/noun4 → nr
undefined = 14
************************

relationScore ruleNo ruleNo
2 11 12
3 11 48
1 11 73
2 11 92
2 11 106
4 11 161
3 11 162
5 11 168
22 11 184
4 11 244
1 11 245
1 11 254
2 11 259
3 12 48
5 12 73

4 12 92
4 12 106
7 12 161
6 12 162
7 12 168
38 12 184
2 47 92
3 47 94
1 47 95
4 47 130
10 47 218
10 48 92
5 48 93
8 48 94
7 48 95
5 48 130
7 48 161
2 48 162
20 48 184
30 48 218
7 73 92
7 73 93
4 73 94
6 73 95
5 73 130
4 73 161
1 73 162
14 73 184
26 73 218
6 92 93
9 92 94
4 92 95
10 92 106
6 92 130
5 92 161
2 92 162
4 92 168
26 92 184
50 92 218
5 92 224
5 93 94
5 93 95
7 93 106
5 93 130
42 93 218
2 93 224
5 94 95
5 94 106
6 94 130
50 94 218
5 94 224
3 95 106
5 95 130
38 95 218
2 95 224
4 106 130
2 106 161
5 106 162
14 106 184
28 106 218
44 130 218
4 130 224
2 161 162
5 161 168
36 161 184
2 161 244
1 161 254
2 161 259
5 162 168
26 162 184
2 162 244
1 162 245
2 162 259
26 168 184
18 218 224
4 244 245
4 244 254

## A.8 The grammar of experiment3 in generation n+1 (number of L0=4)

Generation = n+1
Number of rules = 35
Number of L1 rules = 8
Number of L2 rules = 12
Number of L0 rules = 5
Number of UN rules = 8
Expressivity in L1 = 100%
Expressivity in L2 = 78%
***********************
191 S/p(x,y) → C/xgC/yJ/p
157 S/p(noun3,noun0) → awJ/psy
67 C/noun2 → iw
106 C/noun3 → u
18 C/noun4 → n
192 J/verb6 → bd
73 J/verb8 → oft
155 J/verb9 → ke
language1 = 8
***********************
182 S/p(x,y) → C/xF/pC/y
104 S/p(x,noun2) → C/xA/p
96 S/p(x,noun3) → C/xN/p
209 S/verb7(noun3,y) → awtC/y
10 A/verb5 → xkbz
78 C/noun2 → bz
120 C/noun3 → aw
190 C/noun4 → r
53 F/verb6 → lq
52 F/verb8 → bqp
51 F/verb9 → ybr
98 N/verb7 → taw
language2 = 12
***********************
201 S/p(noun0,y) → syJ/pC/y
17 C/noun0 → sy
90 C/noun1 → v
72 J/verb5 → xk
193 J/verb7 → t
language0(both 1 and 2) = 5
***********************

118 S/p(x,noun0) → C/xH/p
212 S/verb7(noun2,noun0) → bztsy
25 A/verb5 → giwxk
11 A/verb7 → giwt
60 H/verb6 → gsybd
61 H/verb8 → bqpsy
99 N/verb5 → guxk
97 N/verb9 → guke
undefined = 8
************************

relationScore ruleNo ruleNo
2 10 90
6 10 104
1 10 120
5 17 18
5 17 51
4 17 52
2 17 53
4 17 67
7 17 72
2 17 73
4 17 78
12 17 90
7 17 106
1 17 120
3 17 155
22 17 182
2 17 190
48 17 191
7 17 192
5 17 193
5 18 67
4 18 72
3 18 73
5 18 90
7 18 106
8 18 155
44 18 191
4 18 192
3 18 193
8 51 78
5 51 90
4 51 120
26 51 182
4 51 190
5 52 78
7 52 90
2 52 120
20 52 182
2 52 190
4 53 78
5 53 90
8 53 120
20 53 182
1 53 190
2 67 72
2 67 73
3 67 90
7 67 106
6 67 155
38 67 191
2 67 192
7 67 193
1 72 78
3 72 90
4 72 106
1 72 120
2 72 157
20 72 191
4 72 201
1 73 90
4 73 106
12 73 191
6 78 90
4 78 96
2 78 98

7 78 120
34 78 182
2 78 201
2 78 209
4 90 104
1 90 106
4 90 120
2 90 155
34 90 182
3 90 190
34 90 191
5 90 192
7 90 193
2 90 201
8 96 98
4 96 190
2 98 190
2 104 120
5 106 155
44 106 191
4 106 192
5 106 193
28 120 182
2 120 190
1 120 193
4 120 201
24 155 191
14 182 190
2 190 209
22 191 192
26 191 193
4 193 201