

# Chapter 7

## A Schema Theory for TAG3P

Since John Holland proposed his schema theorem in the mid-1970s [Hol1975, Gol1989], schemata are often used to explain why GAs work. Schemata are seen as similarity templates representing entire groups of chromosomes in the population [Lan2002]. Holland's schema theorem describes how the schemata propagate from generation to generation under the influence of selection, crossover and mutation. Although there have been a number of criticisms among GA researchers over the schema theorem [Alt1995, Kar1995, Wol1996, Vos1999], it is still a simple, easy-to-understand, and concise description of the way GAs conduct their search. The problem, as stated in [Rad1997], is not the schema theorem but rather its over-interpretation. In any case, a schema theorem is usually the first stepping stone in understanding the behavior of GAs. A brief introduction to the concepts and terminology commonly used when describing the schema theorem in GAs is given in Appendix B.

In this chapter, we first briefly survey the schema theorems in genetic programming and grammar guided genetic programming. Then, we define the concept of a schema on TAG-based representation, and show that it unifies the three important aspects of a schema on a syntactically constraint domain. Finally, we present a simple schema theorem, estimating the expected lower bound for the propagation of a schema in a TAG3P system using fitness-proportionate selection, subtree crossover, and subtree mutation.

## 7.1 Schema Theory in GP

In GP, the structure of program chromosomes is usually more complicated than the linear (and binary) representation in GAs. Consequently, as noted in [ORe1995], the extension to GP of some concepts in GA schemata such as "order" and "defining length" is not straightforward. O'Reilly and colleagues suggested that different representations will inevitably lead to different concepts of schemata [ORe1995]. Consequently, there have been a number of different attempts to define a schema and subsequently a schema theorem in GP.

In the early days of GP, Koza ([Koz92]), Altenberg ([Alt1994]), and O'Reilly and Oppacher ([ORe1995]) were the first researchers to try to define a schema for GP systems using expression tree representation. The latter two papers also gave schema theorems based on their concepts of GP schemata. A detailed survey of these schema concepts was given in [Lan2002]. However, as pointed out in [Lan2002], their concepts of schemata as components of programs (usually as subtrees or subtree fragments) although reflecting the component aspect of schemata in GAs, do not resemble the subspace aspect of GA schemata. In other words, GA schemata can potentially match components anywhere in the program tree. This problem complicates the computation of schema propagation in the corresponding schema theorems.

For an effective definition, Langdon and Poli ([Lan2002]) argued that a GP schema should be a component of program trees and should also represent a subspace of program trees. In GAs, this is not an issue because of their fixed and linear chromosome structure, so that the Holland definition of a schema satisfies both requirements at the same time. It is not obvious how to achieve this for GP with tree-based structure.

In [Ros1997], Rosca was the first person to introduce the concept of positional schemata in GP, by defining them as rooted schemata. Based on Rosca's idea, Poli and Langdon ([Pol1997, Pol1998]) developed a new theory of fixed size and shape schemata in GP, in which they resurrected the concepts of "order" and

of “defining length” of a schema. A schema theorem was subsequently proven for this type of GP schema using some rather specialised operators [Pol1997, Pol1998, Lan2002]. Their theory for fixed shape and size schemata is given in Appendix B of this thesis. They subsequently went further to derive a series of exact schema theorems [Lan2002].

## 7.2 Schema Theory in GGGP

In grammar guided genetic programming, the program trees are generated and delimited by a grammar. Therefore, the concept of schema should relate the schemata to the formalism. Moreover, we argue that, apart from the above two requirements for a schema, it is important that the schema also represent a sub-language of the formalism. If it is a sub-language, all the properties of the formalism language transfer to the schemata, making their syntactical properties uniform with the language of the whole search space (i.e. the tree set of the formalism). If we can define schemata to be at once components of programs, sub-search-spaces, and sub-languages, then we can interpret the propagation of schemata as propagating good templates, sampling good parts of the search space, and/or focusing on relevant sub-languages.

To the best of our knowledge, there has been only one attempt at a schema theory for standard grammar guided genetic programming, namely that proposed in Whigham’s ([Whi1995c]) schema theorem. Some recent attempts have been made in GE but no schema theorem for GE has been presented [Rya2004].

Whigham’s concept of schemata for CFG-GP (context-free grammar (guided) genetic programming) was based on the notion of partial derivation trees in CFGs. Each schema, in his definition, is a partial derivation tree stemming from some non-terminal symbols of the formalism. However, as pointed out in [Lan2002], Whigham’s Schemata are program components but not program sub-search-spaces. This is because his schemata are non-positioned. Consequently, they can occur several times in the one program tree. Figure 7.1 illustrates this

situation. Moreover, since a Whigham schema is only a partial derivation tree, the string set (and tree set) of the set of programs containing a given schema does not define a sub-language of the formalism. Nevertheless, this definition of schemata for CFG-GP led to a simple schema theorem [Whi1995c, Whi1996]. Appendix B contains a brief review of the CFG-GP schema theory.

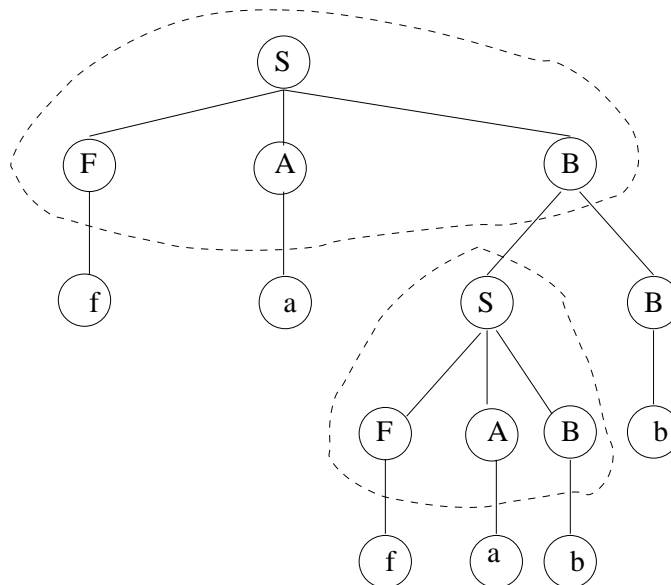


Figure 7.1: An example for Whigham's schema, where a schema can match more than once in an individual

### 7.3 Schema Definition in TAG3P

The concept of schemata in TAG-based representation is inspired by the rooted schemata of [Ros1997, Pol1997, Pol1998]. Each schema is a template for TAG derivation trees, in which some adjoining addresses are closed or opened for adjunctions. The definition of a schema in TAG-based representation is given as follows:

**Definition 1** *A schema in TAG-based representation is a TAG derivation tree. However, all leaf nodes of a schema are labeled with #, where a leaf # means it can be replaced with either a NULL node or a TAG sub-derivation tree, whose*

root is a  $\beta$ -tree that can be adjoined at the address represented by the link between the leaf and its parent node

Figure 7.2 depicts an example of a schema on TAG-based representation.

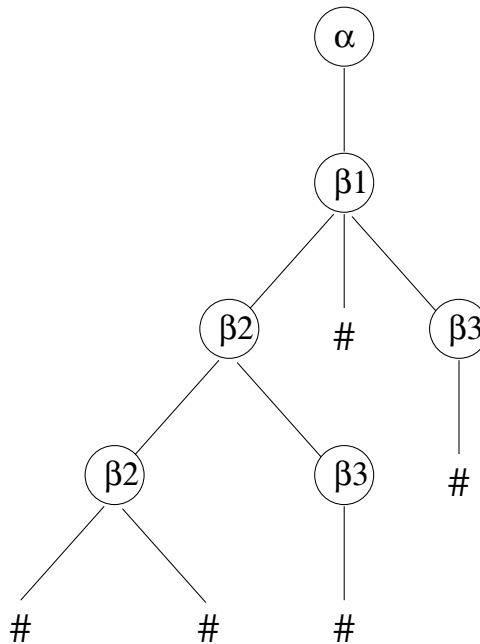


Figure 7.2: An example for a schema on TAG-based representation

Since a schema in TAG-based representation is a rooted, labeled tree, it can occur at most once in each program. Therefore, it satisfies the two requirements for a schema definition proposed by Poli and Langdon. It represents a component of TAG-based representation programs, and it represents a search subspace of TAG derivation trees. It remains to be shown that the set, consisting of all derivation trees in which a given schema occurs, is a sub-language of the TAG used to generate the programs.

**Lemma 1.** *Let  $\gamma$  be a completed tree derived from some initial  $\alpha$  S-trees of a TAG  $G_{lex} = \{\Sigma, N, I, A, S\}$ . Then, the tree set consisting of all trees which can be derived from  $\gamma$  by repeated adjunctions and/or substitutions is a sub-language of  $G_{lex}$ .*

**Proof.** It is obvious that any tree  $\gamma_1$  derived from  $\gamma$  is also derived from some

initial S-trees of  $G_{lex}$  because  $\gamma$  itself is derived from them. Therefore, the tree set derived from  $\gamma$  using adjunctions and/or substitution must be a subset of the  $G_{lex}$  tree language. To see that it is also a TAL, we note that  $\gamma$  is also a completed S-tree, since it is derived from some initial S-trees. Therefore the set can be generated using TAG  $G'_{lex} = \{\Sigma, N, I', A, S\}$ , where  $\Sigma, N, A, S$  are as in  $G_{lex}$ , and  $I' = \gamma$ . That completes the proof.

The sub-language aspect of a schema in TAG-based representation is stated in the following theorem:

**Theorem 3** *The set of derived trees of the programs initiated by a TAG-based representation schema is a sub-language of the formalism ( $G_{lex}$ ) used to generate the schema.*

**Proof.** For any schema  $H$ , if all leaves are replaced with NULL, it becomes a derivation tree of  $G_{lex}$ . Because of the non-fixed arity property, the derived tree of this derivation tree is a completed tree  $\gamma$  of S-type. Thus, for each  $G_{lex}$  derivation tree matching schema  $H$ , its derived trees are completed trees derived from  $\gamma$  using adjunction and/or substitutions. By Lemma 1, this tree set is a subset of the  $G_{lex}$  tree set, and it is also a TAL. Therefore the set consisting of all derived trees of programs ( $G_{lex}$  derivation trees) matching a given schema  $H$  constitutes a sub-language of  $G_{lex}$ . That completes the proof.

With theorem 3, we can conclude that a schema in TAG-based representation, as defined in Definition 1, unifies all three aspects of a schema on syntactically constrained domains.

## 7.4 A Schema Theorem for TAG3P

In this section, a simple schema theorem for TAG-based representation schema is derived. The operators used are fitness-proportionate selection, subtree crossover, and subtree mutation. We note that a schema  $H$  can only propagate to the next

generation if the individuals matching it in the population are selected and transferred to the next generation without being disrupted by genetic operators (of course, some new instances of the schema may be generated through crossover or mutation).

### 7.4.1 Expected Number of Individuals Matching a Schema after Selection

The aim of a schema theorem is to estimate the rate at which individuals matching a schema are propagated to the next generation. The first stage of this process, selection, is largely independent of representation. To be precise, so long as each individual can match with a schema at most once, and using fitness proportionate selection, the expected number of individuals matching schema  $H$  at generation  $t$  which are selected for generation  $t + 1$  is independent of the type of representation. The proof of this proposition can be found in [Hol1975, Gol1989, Lan2002].

**Proposition 1.** The number ( $N_s$ ) of individuals matching a schema  $H$  being selected with fitness-proportionate selection at generation  $t$  is:

$$E[m(H, t + 1)] = m(H, t) \times \frac{f(H, t)}{\bar{f}(t)} \quad (7.1)$$

Where  $m(H, t)$  is the number of individuals matching schema  $H$  at generation  $t$ ;  $f(H, t)$  is the average fitness of individuals matching  $H$  at generation  $t$ ; and  $\bar{f}(t)$  is the average fitness of all individuals in the population at generation  $t$ .

### 7.4.2 Schema Disruption due to Genetic Operators

After selection, individuals are transformed using genetic operators. The possibility arises that the children of an individual  $h$  matching a schema  $H$  may no longer match  $H$ . This can happen if and only if the genetic code in  $h$  that matches  $H$  is destroyed by the genetic operators, usually by choosing operator point(s) in that part, and the new genetic code generated by the operators does

not compensate this loss (i.e does not make  $h$  match  $H$  again). For each  $h \in H$ , we can formulate an upper bound for the probability that its children do not match schema  $H$  as follows:

**Proposition 2.** The upper bound for the disruptive probability of schema  $H$  on an individual  $h$  after the application of a genetic operator on  $h$  is  $\frac{o(H)}{n(h)}$ . Where  $o(H)$  and  $n(H)$  are the numbers of non-# nodes in  $H$  and  $h$  respectively.

**Proof.** Since the number of non-#nodes in an individual  $h$  matching  $H$  is  $n(h)$ , there are  $n(h)$  points for genetic operators to act on. The disruption of  $H$  on the children of  $h$  (after genetic operations) happens if the chosen points are in the matching region of  $h$  and  $H$ , which has  $o(H)$  nodes. Therefore the probability for a point in the matching region to be chosen is  $\frac{o(H)}{n(h)}$ . This is an upper bound because there is a possibility that even when the points for genetic operations are chosen in the match region, the children of  $h$  might still match  $H$ .

### 7.4.3 A Schema Theorem for TAG3P

Given the previous two propositions, we can state a simple schema theorem for TAG3P with fitness-proportionate selection, subtree crossover and subtree mutation as genetic operators.

**Theorem 4** *The propagation of each schema  $H$  in the population satisfies the following bound:*

$$E[m(H, t + 1)] \geq m(H, t) \times \frac{f(H, t)}{\bar{f}(t)} \times \left[ 1 - (p_c + p_m) \sum_{h \in H} \frac{o(H)}{n(h)} \frac{f(h)}{\sum_{h \in H} f(h)} \right] \quad (7.2)$$

where

$m(H, t)$  is the number of individuals in the population which match schema  $H$  at generation  $t$ .

$f(H, t)$  is the mean fitness of all individuals matching schema  $H$ .

$\bar{f}(t)$  is the mean fitness of all individuals in the population.

$p_m$  is the subtree mutation probability.

$p_c$  is the subtree crossover probability.

$o(H), n(h)$  are the number of non-# nodes respectively in schema  $H$ , and in individual  $h \in H$ .

$E[m(H, t + 1)]$  is the expected number of individuals matching the schema  $H$  at generation  $t + 1$ .

$\Sigma$  The sum is taken over the multiset of all individuals in the population that match schema  $H$  (a multiset is a generalisation of a set in which each element may occur multiple times. It is necessary to use multisets here because each individual might appear several times in the population).

**Proof of Theorem 4.** The lower bound for the number of individuals representing schema  $H$  in the next generation ( $m(H, t + 1)$ ) is determined by two factors. The first is the expected number of individuals  $h \in H$  chosen by fitness-proportionate selection – according to proposition 1, this is  $m(H, t) \times \frac{f(H, t)}{(\bar{f})(t)}$ . The second factor is the number of selected individuals where the schema may be disrupted by genetic operators. From proposition 2, an upper bound for the probability of disruption of a schema in an individual  $h$  is  $\frac{o(H)}{n(H)}$ , given that  $h$  is selected. The probability that  $h$  is selected from the individuals in  $m(H, t)$  is  $\frac{f(h)}{\sum_{h \in H} f(h)}$ . Therefore, an upper bound for the probability that each individual  $h \in H$  is disrupted by the genetic operators is  $(p_m + p_c) \frac{o(H)}{n(h)} \frac{f(h)}{\sum_{h \in H} f(h)}$ . Summing over  $h \in H$ , we get an overall upper bound for the probability that at least one  $h \in H$  is disrupted by the genetic operators. Subtracting from 1 gives us a lower bound on the probability that all  $h \in H$  get through to the next generation still matching  $H$ . Combining the two factors, the inequality of the theorem is derived. That completes the proof.

From the theorem, it can be seen that if  $f(H, t)$  is high compared to  $\bar{f}(t)$ , and  $o(H)$  is small (compared to  $n(h)$ ), the lower bound (the left hand side of the

inequality) increases. We note that the smaller the starting  $\gamma$  tree in lemma 1, the larger the sub-language of  $G_{lex}$  it produces. That is, when  $o(H)$  is small, the sub-language defined by the derived tree sets from the individuals matching  $H$  is large. Thus the schema theorem may be restated as: “genetic search on TAG-based representation, using fitness proportionate selection, subtree crossover, and subtree mutation, has a bias toward individuals belonging to schemata that are short in length, high in fitness, and large in terms of sub-language”.

## 7.5 Conclusion

In this chapter, we presented a concept of schemata for TAG-based representation. We showed that these schemata embody all three aspects of schemata on syntactically constrained domains, namely search subspaces, program sub-components, and formalism sub-languages. A simple theorem was given for estimating a lower bound for the expected number of individuals instantiating a schema in the next generation, based on the number in the current generation (using fitness-proportionate selection, subtree crossover and subtree mutation). The theorem gives a rough estimate of how each schema propagates during the evolutionary process. It helps to explain the behavior of TAG3P in terms of schema sampling, in which the evolutionary process of TAG3P favours schemata that are high in fitness and of low order. From the perspective of sub-language sampling, it favours larger sub-languages.

The schemata defined in this chapter are variable in shape and size. Consequently, the process of calculating the lower bound must take into account the size of every individual in the population. This makes the calculation more complicated and results in a relatively underestimated lower bound. We believe that it should be possible to define schemata with fixed shape and size, similar to the approach of [Lan2002]. By so doing, it may be possible to derive a schema theorem which is more descriptive, simpler to calculate, and subject to a better lower bound. We leave this for future work.