Discourse generation is a critical task of natural language generation. In this thesis, we introduce an approach to discourse generation for qualitative causal probabilistic domains that incorporates argumentation into the generation process. The discourse generation process uses three modules: a qualitative causal probabilistic domain model, a genre-specific discourse grammar, and a normative argument generator. The model of discourse generation has been implemented for the domain of clinical genetics. In conjunction with GenIE, a prototype intelligent system for generating the first draft of a patient letter on behalf of a genetic counselor, the discourse grammar exploits general information about clinical genetics as well as documentation about a specific patient's case provided by a genetic counselor, to create discourse plans. The argument generator generates arguments for the claims passed to it from the discourse grammar using domain-independent argument strategies. An important contribution of the thesis is a modification of the argument generator to support interactive argument exploration.
Argument Generation for a Biomedical Domain

by

Kanyamas Navoraphan

A thesis submitted to the Graduate Faculty of
North Carolina State University
In partial fulfillment of the
Requirements for the degree of
Master of Science

Computer Science

Raleigh, North Carolina

2008

APPROVED BY:

_________________________________________  _______________________________________
Dr. Robert D. Rodman                      Dr. R. Michael Young

_________________________________________  _______________________________________
Dr. Nancy L. Green                      Dr. James C. Lester
Co-Chair of Advisory Committee              Chair of Advisory Committee
DEDICATION

To life, and what remains of it

Carpe diem: Seize the day

- Horace -
BIOGRAPHY

Kanyamas Navoraphan was born in Bangkok, Thailand, in 1976. She received her Bachelor of Arts degree in English from Chulalongkorn University, Bangkok, Thailand in 1996. She then obtained a second Bachelor of Science degree in Computer Science from University of Hawaii at Manoa in Honolulu, Hawaii in 2000. After working as a Software Engineer with Synopsys Inc., Mountain View, California for five years, from July 2000 to August 2005, she joined the graduate program at the Computer Science Department in North Carolina State University in the fall of 2005. Her dream is to combine the knowledge in both the language and the computer science fields in the area of natural language processing, specifically machine translation.
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the following people without whose help and support I could not have achieved what I have.

My advisor, Dr. Nancy L. Green, for her continued guidance and support, and especially for spending endless hours perfecting my thesis.

My co-advisor, Dr. James C. Lester, for introducing me to Dr. Green, and for his help in reviewing the final draft of my thesis.

The members of my committee, Dr. Robert D. Rodman and Dr. R. Michael Young for their academic support.

My project partner, Rachael Dwight, for insightful discussions throughout the course of the project.

My friend, Valuncha Paterson, for her help in formatting my thesis, and for giving me the much-needed encouragement throughout the course of my thesis writing.

My friend, Jeremy Paterson, for his help in reviewing my thesis.

My friend, Emilie Wang, for her help in providing me valuable feedbacks during my thesis defense preparation.

My parents, for providing me the greatest gift of education, for their unconditional love and understanding, and for being there for me, always.

The rest of my family and friends, for their love and support, without whom I would not be who I am today.

This material is based upon work supported by the National Science Foundation under CAREER Award No. 0132821.
# TABLE OF CONTENTS

LIST OF TABLES ........................................................................................................ vii

LIST OF FIGURES .................................................................................................. viii

INTRODUCTION ................................................................................................. 1

1.1. Genetics Background .................................................................................. 2

1.2. Related Work ............................................................................................... 3

1.3. Thesis Organization ................................................................................... 6

DOMAIN MODEL ................................................................................................. 7

2.1. Qualitative Probabilistic Network ............................................................. 7

2.2. Implementation ......................................................................................... 8

2.3. Qualitative Relations ............................................................................... 14

OVERVIEW OF KNOWLEDGE BASES ............................................................ 18

3.1. Cystic Fibrosis ......................................................................................... 18

3.2. Achondroplasia ....................................................................................... 22

3.3. Familial Hypercholesterolemia ............................................................... 26

3.4. Phenylketonuria ..................................................................................... 28

DISCOURSE GRAMMAR ..................................................................................... 31

4.1. Overview .................................................................................................. 32

4.2. Discourse Plan ........................................................................................ 34

ARGUMENT GENERATION .............................................................................. 39

5.1. Elements of Arguments ........................................................................... 39

5.2. Argument Schemes ................................................................................. 40
LIST OF TABLES

Table 2-1: A full set of node types and their descriptions [Green, 2005]............................9

Table 2-2: Domain definition and corresponding scalar values. .................................10

Table 4-1: List of nucleus-satellite RST relations used in the Discourse Grammar. ..........33

Table 4-2: List of multinuclear RST relations used in the Discourse Grammar. ..............34

Table 5-1: The six elements of argument in Toulmin’s argument structure [Toulmin, 2003]. ........................................................................................................................................39

Table 6-1: List of possible choices and corresponding actions in Argument Generation state. ........................................................................................................................................60

Table A-1: Non generic events in the Cystic Fibrosis KB. .................................................97

Table A-2: Generic events in the Cystic Fibrosis KB. ......................................................99

Table A-3: Non-generic events in the Achondroplasia KB. ..............................................100

Table A-4: Generic events in the Achondroplasia KB. ............................................... 102

Table A-5: Non-generic events in the Familial Hypercholesterolemia KB. ..................103

Table A-6: Generic events in the Familial Hypercholesterolemia KB. .........................105

Table A-7: Non-generic events in the Phenylketonuria KB. ...........................................105

Table A-8: Generic events in the Phenylketonuria KB. ..................................................106
LIST OF FIGURES

Figure 2-1: Examples of $S^+$ and $S^-$ relations. ................................................................. 14

Figure 2-2: Example of a $Y^+$ relation. ................................................................. 15

Figure 2-3: Example of a $Y^-$ relation. ................................................................. 16

Figure 3-1: The pretest state of the Cystic Fibrosis KB. ................................................................. 19

Figure 3-2: The posttest state of the Cystic Fibrosis KB. ................................................................. 20

Figure 3-3: The future state of the Cystic Fibrosis KB. ................................................................. 21

Figure 3-4: The pretest state of the Achondroplasia KB. ................................................................. 23

Figure 3-5: The posttest state of the Achondroplasia KB. ................................................................. 24

Figure 3-6: The future state of the Achondroplasia KB. ................................................................. 25

Figure 3-7: The pretest state of the Familial Hypercholesterolemia KB. ................................................................. 26

Figure 3-8: The posttest state of the Familial Hypercholesterolemia KB. ................................................................. 27

Figure 3-9: The posttest state of the Phenylketonuria (PKU) KB. ................................................................. 28

Figure 3-10: The future state of the Phenylketonuria (PKU) KB. ................................................................. 29

Figure 4-1: The discourse generation process. ........................................................................... 31

Figure 4-2: A rough organization of the dplan. ........................................................................... 35

Figure 5-1: The RST representation of an argument [Green, 2007]. ................................................................. 40

Figure 5-2: The applicability constraint of the E2C argument scheme. ................................................................. 41

Figure 5-3: Constraints of a variant of the E2C argument scheme. ................................................................. 42

Figure 5-4: Another variant of the E2C argument scheme. ................................................................. 43

Figure 5-5: The three applicability constraints of the NE2C argument scheme. ................................................................. 44

Figure 5-6: The E2JC argument scheme. ........................................................................... 46
Figure 5-7: The E2XORC argument scheme................................................................. 47
Figure 5-8: A sample scenario illustrating a variant of the ELIM argument scheme........ 48
Figure 5-9: An example of a complex argument generation.......................................... 50
Figure 5-10: The data exchange between the discourse generator and the argument
generator.................................................................................................................... 51
Figure 5-11: An RST tree before and after pruning....................................................... 56
Figure 6-1: FSM representing the interactive argument generator’s architecture............. 58
Figure 6-2: Detailed FSM during the Argument Generation state................................... 61
Figure 6-3: Example user interaction during KB, KB state, and node selections............. 62
Figure 6-4: Argument Listing Format.............................................................................. 63
Figure 6-5: An example of user interaction during warrant selection............................ 63
Figure 6-6: Sample Test KB. ........................................................................................ 68
Figure A-1: Section 1 of the Cystic Fibrosis dplan......................................................... 87
Figure A-2: Section 2 of the Cystic Fibrosis dplan......................................................... 88
Figure A-3: Section 3 of the Cystic Fibrosis dplan......................................................... 89
Figure A-4: Section 4 of the Cystic Fibrosis dplan......................................................... 900
Figure A-5: Section 1 of the Achondroplasia dplan....................................................... 91
Figure A-6: Section 2 of the Achondroplasia dplan....................................................... 92
Figure A-7: Section 3 of the Achondroplasia dplan....................................................... 93
Figure A-8: Section 1 of the Familial Hypercholesterolemia dplan................................. 94
Figure A-9: Section 2 of the Familial Hypercholesterolemia dplan................................. 95
Figure A-10: Section 2 of the Phenylketonuria dplan.................................................... 95
Figure A-11: Section 3 of the Phenylketonuria dplan..........................96
CHAPTER 1
INTRODUCTION

The goal of this research is to develop GenIE, a prototype intelligent system for generating the first draft of a patient letter on behalf of a genetic counselor, using general information on clinical genetics as well as specific documentation about the patient’s case provided by the genetic counselor. The work done for this thesis focuses on the reimplementation of the GenIE prototype first described in [Green, 2006], using a new discourse grammar that generates discourse plans employing Rhetorical Structure Theory (RST) relations [Mann and Thompson, 1988] as specified in [Green, 2007]. The work also includes an implementation of domain models for four new genetic diseases partly based on case studies in [Korf, 2000] and [Nussbaum, McInnes, and Willard, 2001], and implementation of an interactive argumentation driver.

Like many other natural language generation (NLG) systems [Reiter and Dale, 2000], GenIE uses a nonlinguistic domain model. Generation is performed in two stages: discourse generation, and then linguistic realization. The scope of this thesis covers the domain model and discourse generation portions, leaving the linguistic realization portion to be done in separate work.
The *domain model* is a causal probabilistic knowledge base, designed specifically to store information related to clinical genetics. *Discourse generation* is divided into two separate parts, namely the *discourse grammar*, and *argument generation*. The output from discourse generation is in the format to be used as input to the linguistic realizer at a later time. Because the work on *linguistic realization* is incomplete at this point, as part of the work done on this thesis, we have also implemented an interactive driver for the argument generator that can be used to partially validate the intermediate outputs as far as argument generation is concerned. The last part of this thesis describes the interactive driver. All implementation for this thesis was done using the logic programming language Prolog.

### 1.1. Genetics Background

According to [Baker *et al.*, 2002; Harper, 1998], genetic counseling is the process by which a patient who is at risk of a genetic disorder seeks advice on how to deal with such disease from a genetic counselor. The advice ranges in nature from the general characteristics of the disease, its symptoms and consequences, the probability of developing and transmitting it, as well as the options and procedures available to prevent or treat the disease. Genetic counselors generally work as members of a healthcare team to identify families at risk and investigate problems present in the family. Then, they go on to meet with the patient and his family to review available testing options, discuss risk of complications, interpret test results, diagnose inherited conditions, and analyze inheritance patterns as well as recurrence risks.
An important task performed by a genetic counselor is to educate the patient about his disease. It is thus important that the counselor be able to communicate in terms that are comprehensible to a lay audience. Sending letters with details summarizing the services and information provided is a way to keep the patient informed about his health conditions. Adequate understanding of the health conditions in turn enables the patient to be more actively involved in his own healthcare. This type of standard document designed to record the counselor’s reasoning for medical as well as legal purposes is called a patient letter.

1.2. Related Work

Argument generation is among the topics that attract research interest in the area of NLG due to the fact that argument is, by nature, a highly structured form of text [Reed and Long, 1998]. Zukerman et al. have done work on an argument generation system called NAG (Nice Argument Generator) [Zukerman et al., 2000], which relies on domain models that are based on a Bayesian Network formalism. Unlike NAG, GenIE’s domain model uses a qualitative probabilistic network formalism as its main representation. Also, whereas NAG generates arguments using standard deductive and Bayesian argumentation strategies, GenIE’s argument strategies are based on analysis of a corpus of genetic counseling patient letters.

[Reed and Long, 1997] present a framework that supports the modeling of argument structure using a hierarchical planner. First, an abstract plan is generated through operators that encode deductive argument schemes. These operators complete the plan by fulfilling its communicative as well as topic manipulation goals. Then, a set of heuristics is applied to
maximize persuasive effects as well as coherency of the argument. In their subsequent work, [Reed and Long, 1998] describe *Rhetorica*, an abstraction-based planning system that uses deductive, refutation, and inductive generalization operators to generate complex argument structure. The planning occurs at a level that is more abstract than RST. This extra layer is introduced specifically to represent argumentative relationships that cannot be expressed with RST. Like *Rhetorica*, GenIE also uses argument schemes to support its generation of arguments at a level that is more abstract than RST.

[Branting *et al.*, 1999] describe a legal document drafting system that relies on both domain knowledge as well as discourse knowledge to generate legal arguments. Similar to GenIE, the system uses a discourse grammar to define the organization of the document, and then fills in the content with legal reasoning. However, in GenIE, we separate the task of domain reasoning from argument generation.

[Grasso *et al.*, 2000] present Daphne, a nutrition counseling system that uses an argumentation theory framework to provide healthy nutrition advice. Like GenIE, though the system has been implemented to work in a healthcare-related domain, the domain knowledge was kept as a well-defined component separate from the argumentative tactics that are expressed in general terms and thus make Daphne extendable to other domains. Unlike GenIE, Daphne is intended to generate persuasive arguments in interactive dialogue with users based on informal argumentation theories [Perelman and Olbrechts-Tyteca, 1969] rather than normative arguments in text.
The goal of the GenIE project [Green, 2005; Green, 2006; Green, 2007; Green, submitted] is to investigate generation of scientific arguments for lay audiences. The testbed for the research is the development of a natural language generation system for drafting genetic counseling patient letters. After collecting a corpus of patient letters from genetic counselors, Green developed and formally validated the inter-coder reliability of a coding scheme for annotating genetic counseling patient letters [Green, 2005]. The coding scheme provides a conceptual model of single-factor genetic disease that is used in expert-lay communication in this domain. This conceptual model of genetic disease was used to guide design of the domain model in the GenIE architecture [Green, 2006]. The domain model is represented as a qualitative probabilistic network. This architecture also makes use of a genre-dependent discourse grammar and non-genre-specific argument schemes that map formal (domain-independent) properties of a qualitative probabilistic network to functional elements of Toulmin’s [Toulmin, 2003] model of argument (e.g., data, claim, and warrant). In a study of pragmatic features of argumentation in the corpus, [Green, 2007] presents extensions to Rhetorical Structure Theory [Mann and Thompson, 1988] incorporating Toulmin’s components of arguments. Finally, [Green, submitted] presents a dialogue game for challenging and asking follow-up questions to arguments generated in the GenIE architecture.
1.3. Thesis Organization

This thesis is organized as follows: Chapter 2 discusses the domain model. Chapter 3 describes in detail the four knowledge bases used to test the system. Discourse generation is presented in two chapters: Chapter 4 presents the first part of generation using the discourse grammar, and Chapter 5 presents the second part, argument generation. Chapter 6 describes the interactive driver, a tool used to test the functionality of the system as far as argument generation is concerned. Chapter 7 concludes the discussion and notes directions for future work.
CHAPTER 2

DOMAIN MODEL

The domain model is a knowledge base (KB) containing information about a patient’s case and general information about genetic disease. This chapter describes the implementation of the KB as a qualitative probabilistic network (QPN). The next chapter explains the four KBs implemented for this thesis.

2.1. Qualitative Probabilistic Network

A qualitative probabilistic network is a directed acyclic graph whose nodes represent random variables and whose edges represent the dependencies between the random variables. The idea of a QPN was introduced as an abstraction of a Bayesian Belief Network, where numerical probability relations are replaced with qualitative influences and synergies [Wellman, 1990, Druzdzel and Henrion, 1993]. QPNs are thus ideal for situations where numerical probabilities are either unavailable or unnecessary to complete the tasks at hand.

For GenIE, we need a model of the patient’s case that can be used to generate arguments in the patient letters for the diagnosis and other conclusions of medical experts. For example, given the diagnosis reported by the genetic counselor, GenIE will search its KB for evidence to explain that diagnosis in the letter. Thus, instead of requiring numerical probabilities for
domain reasoning, it is sufficient for the domain model in GenIE to use qualitative constraints as described in [Green, 2006]. The constraints are described in section 2.3.

2.2. Implementation

A KB contains one or more QPNs. Each network is a directed acyclic graph (DAG). Each graph describes the disease at a specific period of time. Each node of the DAG represents either a random or a decision variable, depending on its type. In GenIE, a random variable node is simply called a node whereas a decision variable node is called a dnode. Each node consists of five different attributes, namely

- node identifier (ID)
- type
- domain
- role
- node description

Node IDs are unique within each KB. For consistency purposes, we use the same naming convention across all KBs. The first part of a node ID represents its type, whereas the second part denotes the genealogical role of the person the node is describing, followed by an optional third part which is a numerical index to distinguish a group of nodes that refer to the same abstract variable type for the same family member. For example, we can expect a node that is labeled gp to describe the proband’s (patient’s) genotype (p for proband and g for
Similarly, the nodes \( sp1 \) and \( sp2 \) will most likely describe two different symptoms \( (s \) for symptom\) the proband is experiencing.

### Table 2-1: A full set of node types and their descriptions [Green, 2005].

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>node</th>
<th>dnode</th>
</tr>
</thead>
<tbody>
<tr>
<td>history</td>
<td>Demographic or predispositional factor (e.g., gender, ethnicity, age, family history of a specific mutation, environmental risk factor)</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>genotype</td>
<td>A pair of alleles(^1) of a gene inherited from parents</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>genotype_germline</td>
<td>A pair of alleles of a gene resulting from mutation rather than inheritance</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>event</td>
<td>Event that results in a change in an individual’s genotype at different points in his life cycle and/or in different cells</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>biochemistry</td>
<td>Manifestation of an individual’s genotype at the biochemical level, e.g., a shortened protein</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>physiology</td>
<td>Manifestation of an individual’s genotype at the physiological level, e.g. in metabolism</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>symptom</td>
<td>Manifestation of an individual’s genotype that is observable without performing a test, e.g., some type of birth defects</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>test</td>
<td>Event of performing a test</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>test result</td>
<td>Result of a test</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>complication</td>
<td>Undesirable side effect of a test, e.g., fetal damage</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

\(^1\) An allele is one of the alternate forms of a specific gene. Each allele is an individual member of a gene pair and is inherited from one parent.
Table 2-1 (continued).

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>node</th>
<th>dnode</th>
</tr>
</thead>
<tbody>
<tr>
<td>behavior</td>
<td>An activity performed by an individual which can be observed</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>treatment</td>
<td>A specific procedure used for the cure of a disease or a condition</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

A small set of node types was identified in [Green, 2005] to cover the extent of KB contents used in this research. Table 2-1 lists the full set of all node types including their brief descriptions. The set reflects a simplified model used by genetic counselors to communicate with their clients.

The domain attribute of a node indicates the range of values the node is allowed to take.

Table 2-2 below shows the domain definition with corresponding scalar values.

Table 2-2: Domain definition and corresponding scalar values.

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Scalar Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>recessive_mut</td>
<td>[0, 1, 2]</td>
</tr>
<tr>
<td>dominant_mut</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>boolean</td>
<td>[true, false]</td>
</tr>
<tr>
<td>normality</td>
<td>[normal, abnormal]</td>
</tr>
<tr>
<td>toxicity</td>
<td>[normal, toxic]</td>
</tr>
<tr>
<td>decision</td>
<td>[recommended, done]</td>
</tr>
</tbody>
</table>
Table 2-2 (continued).

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Scalar Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>nninteger</td>
<td>any non-negative integer</td>
</tr>
<tr>
<td>real_0_1</td>
<td>any real number between 0 and 1</td>
</tr>
</tbody>
</table>

The *role* and *node description* attributes of a node indicate the role in the family tree and the specific concept represented by the node, respectively. Every KB consists of one or more node definitions for the role *proband*. Following standard convention in genetics, other roles such as *father*, *mother*, *future_sibling*, and *offspring* are specified relative to the *proband*. Like a node’s ID, its role and description together can be used to uniquely distinguish one node from the others.

A graph is defined by the KB state at a particular time (pretest, posttest, future, etc.), and a list of arcs it contains. Each directed arc that connects one node to another has a unique identifier with respect to the KB. The same arc may occur in more than one graph, and does not necessarily occur in every graph. Each graph has one root node called *superroot*, of which other nodes are descendants. This special dummy node was introduced in the effort to keep each graph a connected network\(^2\). Without *superroot*, we would have ended up with dangling decision nodes, specifically *dnodes* of type *test*, which typically do not connect to any other nodes in the *pretest* graphs. This will be illustrated in the next chapter.

\(^2\) Note that none of the graphs shown in the next chapter includes this dummy *superroot* node since it does not serve any descriptive purpose.
A node may have different states, representing its possible states at a specific time. Each node state is explicitly defined with three components, including a node ID, a KB state, and an assignment list. In the current implementation, the assignment list contains at most one element. Each element of an assignment list is a tuple, consisting of an assignment and its corresponding probability. An assignment is in the form of \([Op, Value]\) for numeric values, where \(Op\) is one of \(\{=, \geq, =<, >, <\}\), or \(Value\) alone if nonnumeric. A probability is of the form \([Ptype, Pvalue]\) where \(Ptype\) is the name of a defined probability value format, and \(Pvalue\) is the probability value. The probability value represents the degree of belief in different states of a node. Probability type can be either qualitative or quantitative. In the current implementation, \(Ptype\) is qualitative, with an allowable scale of \(\{\text{very_low}, \text{low}, \text{high}, \text{very_high}\}\). For decision nodes where no probability is required, an empty list is put in the place of its probability. Examples of \(state\) definitions include the following:

- \(state(tstp, \text{pretest}, [[[\text{recommended}, []]])\).
  
  Paraphrase: In the pretest graph, the proband’s test is recommended.

- \(state(sm, \text{posttest}, [[[\text{false}, [\text{qual, high}]])\).
  
  Paraphrase: In the posttest graph, the belief that the mother does not have symptoms is high.

- \(state(gp, \text{future}, [[[=, 2], [\text{qual, very_high}])])\).
  
  Paraphrase: In the future graph, the belief that the proband’s genotype has 2 mutations is very high.
In the current implementation, no more than one state predicate is defined per variable per KB state. However, it is possible for a variable’s state to change from one KB state to the next. For example, in the pretest KB state, the state of a test node is generally defined to be recommended, whereas in the posttest state, the node’s state usually changes to done.

Apart from concepts represented by nodes and the relationships among them represented by arcs, the domain model also stores epidemiological statistics related to the concepts used in the network. For example, in the Achondroplasia KB (described in the next chapter), we have:

- \text{statistic}\{\text{cprob}([[G, [=, 1]]], [\text{range}, [0.03, 0.07]], [[C, true]]), \text{research}\) :-
  
  \text{node}(G, \text{genotype}, \text{dominant\_mut, Who, ‘FGFR3’}),
  \text{node}(C, \text{complication}, \text{Boolean, Who, ‘sudden death in first year’}).

  \[ P(\text{sudden death in first year} = \text{true} \mid \text{FGFR3 genotype} = 1) = 3.7\% \]

  \text{Paraphrase: The rate of sudden death during the first year for Achondroplasia patient having 1 mutation of the FGFR3 genotype is 3.7\%.}

The information, though it plays no role in domain reasoning, will be used as backing for an argument. This process will be described in detail in the next chapter.

For information that is neither a fact nor derivable by argument, the domain model also contains a status predicate to indicate that it is assumed. For example, in the Cystic Fibrosis KB (described in the next chapter), we have:

- \text{status}(\text{ep1, pretest, assumed}).
2.3. Qualitative Relations

To support reasoning, the domain model uses qualitative constraints based upon formal relations of qualitative influence, product synergy, and additive synergy [Druzdzel and Henrion, 1993]. In this section, we describe the constraints used in the four KBs covered in this thesis.

![Diagram of qualitative relations](image)

*Figure 2-1: Examples of $S^+$ and $S^-$ relations.*

Nodes are connected to one another by directed arcs that represent a qualitative influence, either positive or negative. A qualitative influence from node $A$ to node $B$ expresses how the value of $A$ influences the probability of a value of $B$. Specifically, node $A$ having a *positive qualitative influence* on node $B$, denoted $S^+ ([A, V_A], [B, V_B])$, means that the state of $A$ reaching its threshold value $V_A$ makes it more likely for $B$ to reach its threshold value $V_B$. For example, a parent having one or more mutated alleles of the CFTR gene increases the likelihood that the offspring will inherit one mutated allele from that parent. On the other
hand, node $C$ having a *negative qualitative influence* on node $D$, denoted $S^-$ ([C, $V_C$], [D, $V_D$]), means that the state of $C$ reaching its threshold value $V_C$ makes it less likely for $D$ to reach its threshold value $V_D$. For example, a lung transplantation done on a patient makes it less likely he will experience a respiratory failure.

![Diagram of Y+ relation](image)

*Figure 2-2: Example of a $Y^+$ relation.*

Product and additive synergies model interaction between various influences, describing the relation between a set of variables and their direct descendant in a graph. An additive synergy expresses how the value of nodes $A$ and $B$ jointly influences the probabilities of node $C$ [Wellman, 1990]. A *positive additive synergy* of nodes $A$ and $B$ on their common child node $C$, denoted $Y^+$ ([A, $V_A$], [B, $V_B$], [C, $V_C$]), expresses that the state of $A$ reaching its threshold value $V_A$ makes it more likely for the state of $B$ reaching its threshold value $V_B$ to influence $C$ to reach its threshold value $V_C$. For example, a CF patient having bacteria in his lung secretion increases the chance of his viscous lung secretion resulting in a respiratory infection. Similarly, a *negative additive synergy* of nodes $A$ and $B$ on their common child
node $C$, denoted $Y^-$ ([[A, $V_A$], [B, $V_B$]], [C, $V_C$]), expresses that the state of $A$ reaching its threshold value $V_A$ makes it less likely for the state of $B$ reaching its threshold value $V_B$ to influence $C$ to reach its threshold value $V_C$. For example, a CF patient taking caloric supplements decreases the chance of his malabsorption resulting in a growth failure.

A negative product synergy of nodes $A$ and $B$ on their common child node $C$, denoted $X^-$ ([[A, $V_A$], [B, $V_B$]], [C, $V_C$]), expresses that either the state of $A$ reaching its threshold value $V_A$ or the state of $B$ reaching its threshold value $V_B$ makes it more likely for $C$ to reach its threshold value $V_C$. Negative product synergy can be used to represent mutually exclusive alternative diagnoses that could account for the same symptom, or to represent autosomal dominant inheritance, an inheritance pattern where inheriting one mutated allele of a genotype (from either parent) may lead to health problems. Examples of an $X^-$ relationship can be seen in either the Achondroplasia KB or the Familial Hypercholesterolemia KB.

*Figure 2-3: Example of a $Y$ relation.*
In contrast, autosomal recessive inheritance, an inheritance pattern in which two mutated copies of a gene, one inherited from each parent, can be represented by zero product synergy. A zero product synergy of nodes $A$ and $B$ on their common child node $C$, denoted $X^0([A, V_A], [B, V_B], [C, V_C])$, expresses that together the state of $A$ reaching its threshold value $V_A$ and the state of $B$ reaching its threshold value $V_B$ make it more likely for $C$ to reach its threshold value $V_C$. Examples of an $X^0$ relationship can be seen in either the Cystic Fibrosis KB or the Phenylketonuria KB, both of which are described in the next chapter.
CHAPTER 3
OVERVIEW OF KNOWLEDGE BASES

We use a total of four sample knowledge bases (KB) to test the functionality of our system. All KBs were partly based on case studies found in clinical genetics textbooks. Each describes a specific patient case of a particular genetically transmitted disease which is either autosomal recessive or autosomal dominant. In an autosomal recessive disorder, the patient inherits the recessive trait from both parents who are carriers, each contributing one of the similar alleles. On the other hand, in autosomal dominant inheritance, the patient needs only one copy of the abnormal gene from either parent in order for the trait to become apparent. In this chapter, we describe each KB in detail, and provide graphs of each KB that show the relationships among concepts. In each graph, the blue arrows represent $S^+$ relations, the red arrows $S^-$, while the X and Y relations are explicitly tagged.

3.1. Cystic Fibrosis

The Cystic Fibrosis (CF) KB [Nussbaum, McInnes, and Willard, 2001] describes a proband who, prior to testing, was suspected of having cystic fibrosis, an autosomal recessive disorder caused by inheriting two mutated alleles of the CFTR (cystic fibrosis transmembrane conductance regulator) gene, one from each parent. CF is most commonly found in Caucasian populations. The disease frequency is 1 per 2500, whereas a carrier frequency is approximately 1 in 25 [Nussbaum, McInnes, and Willard, 2001]. For a couple with one
affected child, the risk of cystic fibrosis recurrence in future offspring is 25% [Korf, 2000]. Studies show that the false-negative rate of carrier screening is about 1-2% [Nussbaum, McInnes, and Willard, 2001].

Figure 3-1: The pretest state of the Cystic Fibrosis KB.

Both parents were of northern European ancestry, but neither showed any CF symptoms. The preliminary diagnosis was based on the fact that two different symptoms of CF were
observed: respiratory infections and growth failure. Bacteria in the proband’s lung secretion is assumed to have enabled a viscous lung secretion to cause the respiratory infections. The growth failure, on the other hand, was believed to have been caused by malabsorption resulting from an abnormal level of pancreas enzyme due to the CFTR mutation. The clinic recommended that a sweat test be done on the proband. Figure 3-1 illustrates the relationships among all concepts described at this state.

Figure 3-2: The posttest state of the Cystic Fibrosis KB.
After testing, the initial diagnosis was further confirmed by the abnormal Sodium Chloride (NaCl) level revealed by the sweat test. Two separate treatments were recommended, including an antibiotic to help inhibit the growth of the bacteria in lung secretion, as well as a clearance of the secretions. Figure 3-2 shows the KB at this particular state.

Figure 3-3: The future state of the Cystic Fibrosis KB.
The future KB state is depicted in Figure 3-3. It represents the information that the proband may need a lung transplantation by the age of 30, the recommendation of the use of caloric supplements to inhibit growth failure and a pancreas enzyme replacement to remedy the abnormal enzyme level, and the prediction that it is highly likely that a future sibling of the proband will inherit the disease as well.

3.2. Achondroplasia

In the Achondroplasia KB [Nussbaum, McInnes, and Willard, 2001], the proband was referred to the clinic since he had typical physical features of achondroplasia3 at birth. Neither the parents nor the three paternal half-siblings of the proband showed any signs of achondroplasia. There was also no family history of this autosomal dominant disease on either the mother’s or father’s side. The clinic suspected that the father may have developed a newly mutated gene which in turn was transmitted to the child. This was based on the fact that the father was 45 years old. Studies show that the fathers of children affected with achondroplasia tend to be older than the population mean for fathers at the time of conception of the child [Korf, 2000]. A high rate of mutation is found in the male population aged 35 and above, whereas no such effect is seen on the female counterpart. A DNA test as well as a radiograph and a fetal ultrasound were recommended. Figure 3-4 shows the KB at this state.

---

3 Achondroplasia is characterized by abnormal bone growth that results in short stature with disproportionately short arms and legs, as well as a large head [Nussbaum, McInnes, and Willard, 2001].
The DNA test revealed both the G1138A and the G1138C mutations in the proband’s FGFR3 (Fibroblast Growth Factor Receptor 3) gene. A positive result of the G1138A mutation typically yields a 98% accuracy rate in the diagnosis of achondroplasia, whereas approximately 1-2% of the patients testing positive for the G1138C mutation end up having the disease [Nussbaum, McInnes, and Willard, 2001]. The radiograph test further confirmed the radiographic features of achondroplasia. Furthermore, the fetal ultrasound indicated both
fetal rhizomelia (a disproportion in the length of the upper arms and thighs) and fetal macrocephaly (an abnormal largeness of the head). Figure 3-5 illustrates the KB at this state.

![Figure 3-5: The posttest state of the Achondroplasia KB.](image)

The clinic further predicted that it is possible the proband will encounter complications in his symptoms which may lead to sudden death during the first year of age. Studies show that the rate of sudden death during the first year for achondroplasia patients is approximately 3-7%
[Nussbaum, McInnes, and Willard, 2001]. The counselor also recommended specific treatments for complications during infancy and early childhood, as well as for complications during later childhood and early adulthood. Figure 3-6 shows this future KB state.

Figure 3-6: The future state of the Achondroplasia KB.
3.3. Familial Hypercholesterolemia

The pretest state of the Familial Hypercholesterolemia KB.

In the Familial Hypercholesterolemia KB [Nussbaum, McInnes, and Willard, 2001], the proband was referred to the clinic after he experienced a symptom of myocardial infarction (commonly known as a heart attack). The counselor initially diagnosed him with familial hypercholesterolemia, a genetic disorder characterized by very high LDL (low-density lipoprotein) cholesterol and early cardiovascular disease. Studies show that this autosomal dominant disease is more prevalent in the French Canadian population. Even though the proband has neither French-Canadian ancestry nor any family history of the disease, several risk factors were evident, including obesity, smoking, and physical inactivity. The counselor
also suspected that all three offspring of the proband may have also inherited the disease. Thus both a gene test and an LDL test were recommended for the proband himself and an LDL test was recommended for all of his offspring. Figure 3-7 shows the relationships between the concepts described at this state.

*Figure 3-8: The posttest state of the Familial Hypercholesterolemia KB.*
The proband’s gene test confirmed a mutation of the LDLR gene in one allele. The LDL test also revealed an elevated level of LDL cholesterol in the proband, as well as two out of three of his offspring. Approximately 5% of the patients who have high LDL end up actually having the disease [Nussbaum, McInnes, and Willard, 2001]. The counselor recommended a drug treatment along with a low-cholesterol diet for the proband. At the same time, he was encouraged to stop smoking, undergo a weight loss, and start exercise regularly. Figure 3-8 shows the KB at this state.

### 3.4. Phenylketonuria

![Figure 3-9: The posttest state of the Phenylketonuria (PKU) KB.](image)

In the Phenylketonuria (PKU) KB [Korf, 2000], the proband was initially referred to the clinic for a phenylalanine concentration test following an abnormal result of a standard newborn screening. The test result revealed an abnormally high phenylalanine level, confirming the original suspicion of PKU. Since PKU is an autosomal recessive disorder, the
counselor believed both parents were carriers, each contributing one mutated gene to their offspring, even though neither of them displayed any PKU symptoms. The counselor recommended that a low phenylalanine diet be started for the proband. This is very important, especially for the first few years of life during which brain development is taking place. The KB at this state is depicted in Figure 3-9.

![Figure 3-10: The future state of the Phenylketonuria (PKU) KB.](image-url)
In the future, the counselor believed the proband had a chance to experience damage to her central nervous system, which in turns would lead to a delayed cognitive development. Furthermore, the counselor also predicted that the proband was likely to give birth to an offspring who would have been exposed to an abnormally high phenylalanine level in the fetal environment, resulting in the damage to development of its major organ systems, including the brain. It is common for such infant to have low birth weight, congenital heart disease, and inadequate brain development at the time of birth. A low phenylalanine diet was recommended prior to pregnancy. Figure 3-10 illustrates the KB at this future state.
The discourse generation process used in GenIE involves three separate modules: a qualitative causal probabilistic domain model, a genre-specific discourse grammar, and a normative argument generator [Green, 2006]. The first component was discussed in Chapter 2. This chapter presents the second component – the discourse grammar. The final component – the argument generator, will be discussed in detail in the next chapter.

*Figure 4-1: The discourse generation process.*
As illustrated in Figure 4-1 above, the discourse grammar extracts information from the knowledge base to construct a claim. The claim is then passed on to the argument generator which in turn returns a list of arguments that support the claim. An output of the DG is a discourse plan that is later converted into an English text by the linguistic realizer. Accompanying the discourse plan is an event model containing the event propositions referred to by the discourse plan. The final English text output is the first draft of a patient letter generated by GenIE.

4.1. Overview

The discourse grammar (DG) determines the organization of the patient letter to be generated, initially as a generic outline and later instantiated with information specific to a patient’s case. The DG rules are based upon an analysis of the corpus done in a previous study, [Green, 2006], as well as a description of standard practice in genetic counseling [Baker et al., 2002]. The format of the DG output has been designed so that it can be easily turned into natural language sentences.

The DG generates a data structure called the discourse plan or dplan, a list of trees whose internal nodes specify discourse relations, and whose leaves specify propositions in the form of events [Green, 2007]. Discourse relations are based upon a modified version of Mann and Thompson’s Rhetorical Structure Theory (RST) [Mann and Taboada, 2007], a theory of text organization that proves to be useful for applications in several areas of discourse analysis.
and text generation. Table 4-1 shows the list of nucleus-satellite RST relations used in the DG, whereas Table 4-2 shows the list of multinuclear RST relations used.

### Table 4-1: List of nucleus-satellite RST relations used in the Discourse Grammar.

<table>
<thead>
<tr>
<th>Relation Name</th>
<th>Nucleus</th>
<th>Satellite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribution</td>
<td>A claim</td>
<td>The person to whom a claim is attributed</td>
</tr>
<tr>
<td>Background</td>
<td>Text whose understanding is being facilitated</td>
<td>Text for facilitating understanding</td>
</tr>
<tr>
<td>Condition</td>
<td>Action or situation whose occurrence results from the occurrence of the conditioning situation</td>
<td>Conditioning situation</td>
</tr>
<tr>
<td>Evaluation</td>
<td>A situation</td>
<td>An evaluative comment about the situation</td>
</tr>
<tr>
<td>Evidence</td>
<td>A claim</td>
<td>Information intended to increase the reader’s belief in the claim</td>
</tr>
<tr>
<td>Purpose</td>
<td>An intended situation</td>
<td>The intent behind the situation</td>
</tr>
</tbody>
</table>

---

4 All but the first relation were extracted from a complete list in [Mann and Taboada, 2007].
Table 4-2: List of multinuclear RST relations used in the Discourse Grammar.\(^5\)

<table>
<thead>
<tr>
<th>Relation Name</th>
<th>Each Nucleus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conjunction</td>
<td>The items are conjoined to form a unit in which each item plays a comparable role</td>
</tr>
<tr>
<td>Disjunction</td>
<td>An item presents a (not necessarily exclusive) alternative for the other(s)</td>
</tr>
<tr>
<td>List</td>
<td>An item comparable to others linked to it by the List relation</td>
</tr>
<tr>
<td>Narration(^6)</td>
<td>Each argument is in temporal order</td>
</tr>
</tbody>
</table>

Event propositions are represented in the form of the four-place predicate:

\[
event(ID, Modality, Semantics, Graph)
\]

where \(ID\) uniquely identifies the event, \(Modality\) consists of a list of features that describe the event such as time, duration, polarity, probability, etc., \(Semantics\) describes the event in terms of its type of action and the semantic roles of the individuals involved in the action, and \(Graph\) tells which KB state the event is based upon.

4.2. Discourse Plan

The starting rule of the DG generates four main sections of a patient letter in the form of a discourse plan or \(dplan\). Figure 4-2 below shows a rough organization of the \(dplan\). The four main sections of the \(dplan\) are

\(^5\) The first three relations were extracted from a complete list in [Mann and Taboada, 2007].
\(^6\) Narration is a discourse relation similar to RST sequence.
- Preliminary diagnosis: `doc_pretest`
- Final diagnosis: `doc_diagnosis`
- Origin of genetic condition: `doc_source`
- Future risk: `doc_sibling_risk`

![Diagram]

**Figure 4-2: A rough organization of the dplan.**

The four sections together form an initial outline of the information to be presented in the letter, including various claims that require an argument to support their validity. Each of the claims is then passed to the argument generator described in the next chapter. The information returned by the argument generator is added to the initial outline, thus completing the structure that will later be transformed by the linguistic realizer into English text.
The first section, *doc_pretest*, narrates the events involving the referral event that brought the proband to the clinic leading to a preliminary diagnosis of what genotype(s) caused the symptoms. The DG consults the KB to get specific information on the list of symptoms the proband was experiencing as well as the list of genotypes the proband was suspected of having, then constructs the claim that the said genotype was responsible for the said symptom(s). Also provided is the list of tests that had been done on the proband, together with corresponding test results. The final optional subsection handles the case when the preliminary diagnosis has been disconfirmed by the test results.

The second section of the letter, *doc_diagnosis*, presents the final diagnosis following the testing done on the proband. This posttest diagnosis is later backed up by the results of specific tests done on the proband when the DG invokes the argument generator to get arguments to support the claim.

The third section, *doc_source*, describes the source of the proband’s genotype. The structure varies depending whether the disease of interest is an autosomal recessive or autosomal dominant disorder. For an autosomal recessive case, information on both parents is included, whereas for an autosomal dominant case, only the information relevant to the carrier is included.
The fourth section, *doc_sibling_risk*, provides information on potential risk of disease inheritance for the proband’s future siblings. The risk is presented in terms of the probability that a future offspring of the same pair of parents may inherit the same disease.

In the case where no particular information is available, the entire section will be replaced with a null clause ([] – an empty list in Prolog), indicating that such detail is to be ignored during English text generation. With it being the most important piece of information, the second section, *doc_diagnosis*, is the only section that is required in the *dplan*. Without a proper diagnosis, there would be no reason for a patient letter to be generated in the first place. Detailed examples of *dplans* for the four KBs covered in this thesis are shown in the appendix section.

Each of the DG rules used to construct the *dplan* consists of one or more nonterminals and/or predicates provided in the KB’s Application Program Interface (API). The predicates are used to access specific details in the KB that are to be included in the content of the patient letter. For example, the predicate *get_symptoms_set(Graph, Who, SymptomList)* returns a list of symptoms the person *Who* was experiencing at the time of *Graph* (pretest, posttest, or future). Similarly, *get_genotype_set*, *get_test_set*, and *get_test_result_set*, each return a list of mutated genes, tests, and test results for the requested person and time, respectively.

Once information is obtained from the KB through the APIs, corresponding event propositions are then created. Each event is passed on to the argument generator, which in
turns supplies arguments that support the claim. The next chapter describes the process of argument generation in more detail.
CHAPTER 5
ARGUMENT GENERATION

The argument generator (AG) generates arguments for the claims passed to it from the DG, using non-domain-specific argument strategies [Green, 2006]. The arguments justify the conclusions of the genetic counselor about the patient’s case. The following sections explain first the layout of arguments according to Toulmin’s model and then the argument schemes used in GenIE. The last section presents some implementation details of the AG.

5.1. Elements of Arguments

According to Toulmin’s model of argument structure [Toulmin, 2003], every acceptable argument shares the same layout of basic elements, consisting of six interrelated components, as listed in Table 5-1 below.

<table>
<thead>
<tr>
<th>Element of Argument</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim</td>
<td>The assertion that is to be established</td>
</tr>
<tr>
<td>Data</td>
<td>The evidence supporting the claim</td>
</tr>
<tr>
<td>Warrant</td>
<td>The reasoning that justifies the claim</td>
</tr>
<tr>
<td>Backing</td>
<td>The credentials that further support the warrant</td>
</tr>
<tr>
<td>Element of Argument</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Rebuttal</td>
<td>The condition that undermines the warrant or the backing</td>
</tr>
<tr>
<td>Qualifier</td>
<td>The degree of certainty expressed for the claim</td>
</tr>
</tbody>
</table>

In GenIE, argument strategies map formal properties of the domain model to the data and warrant supporting a claim and to the backing of a warrant [Green, 2006]. The next section describes the different argument schemes. Each argument is translated into an RST tree format after it is generated before adding it to the dplan. Figure 5-1 illustrates the RST representation of an argument [Green, 2007].

Figure 5-1: The RST representation of an argument [Green, 2007].

5.2. Argument Schemes

An analysis of the corpus done in a previous study in [Green, 2006] resulted in seven different argument strategies to cover the argument generation needed for our domain. All of the arguments generated for each sample KB were constructed using one or more of these
seven argument schemes. A number of arguments were formed as sequences of arguments. An example of such complex arguments is elaborated in Section 5.3. Let us first look at each of the seven argument schemes in detail.

**Argument Scheme: E2C (Effect to Cause)**

**Claim:** \( A \geq V_A \)

**Data:** \( B \geq V_B \)

**Warrant:** \( S^+ ([A, V_A], [B, V_B]) \)

**Applicability Constraint:** \( \neg (\exists C \ X^- ([C, V_C], [A, V_A], [B, V_B]), C \geq V_C) \)

![Figure 5-2: The applicability constraint of the E2C argument scheme.](image-url)

The Effect-to-Cause argument scheme justifies a belief in a claim based on a possible causal effect. In this strategy, the claim that the value of \( A \) has reached its threshold value \( V_A \) is supported by the data that the value of \( B \) has reached its threshold value \( V_B \). The warrant is \( S^+ ([A, V_A], [B, V_B]) \), i.e., that the state \([A, V_A]\) could be responsible for the state \([B, V_B]\). An
applicability constraint is that the following condition does not hold: there exists another variable $C$ such that the relation $X^-$ ($[[C, V_C], [A, V_A]], [B, V_B]$) holds and the value of $C$ has reached its threshold value $V_C$. This constraint deals with the possibility that the state $[C, V_C]$ is the actual cause for the state $[B, V_B]$.

Figure 5-3: Constraints of a variant of the E2C argument scheme.

A variant of this particular argument scheme covers the case where there exists a chain of $S^+$ relations from $A$ to $B$. A constraint similar to the one described above holds at each and every juncture of the $S^+$ relations along the path from $A$ to $B$, i.e. any such relation as $X^-$ ($[[X_{i+1}, V_{X_{i+1}}], [A, V_A]], [X_i, V_{X_i}]$) or $X^-$ ($[[X_{j+1}, V_{X_{j+1}}], [X_j, V_{X_j}]], [B, V_B]$) is not allowed. Figure 5-3 illustrates this E2C subcase.
Another variant of the E2C argument scheme represents the case where there exists a chain of one or more \( S^+ \) relations from \( A \) to \( B \) and the last section of the \( S^+ \) chain forms a \( Y^+ \) relation. The support in this case is the data that:

1) the value of \( B \) has reached its threshold \( V_B \)

2) the value of a necessary enabler \( N \) has reached its threshold \( V_N \)

The warrant in the case of a chain of length 1 is \( Y^+ ([N, V_N], [A, V_A], [B, V_B]) \), i.e., that \( N \) can enable \( A \) to influence \( B \). In the case of a chain of length \( > 1 \), the warrant is \( S^+ ([A, V_A], [X_i, V_{X_i}]) \) and \( Y^+ ([N, V_N], [X_n, V_{X_n}], [B, V_B]) \), i.e., that \( A \) has a positive influence on \( X_i \) where \( X_i \) is along the path from \( A \) to \( X_n \) and \( N \) can enable \( X_n \) to influence \( B \). The same constraint holds along the path of the \( S^+ \) chain. Figure 5-4 illustrates both variants.

**Argument Scheme: NE2C (No Effect to Cause)**

Claim: \( A < V_A \)

Data: \( B < V_B \)

Warrant: \( S^+ ([A, V_A], [B, V_B]) \)
Applicability Constraints:

1) \( \neg (\exists C Y^+ ([C, V_C], [A, V_A], [B, V_B]), C < V_C) \)

2) \( \neg (\exists C Y^- ([C, V_C], [A, V_A], [B, V_B]), C \geq V_C) \)

3) \( \neg (\exists C X^0 ([C, V_C], [A, V_A], [B, V_B]), C < V_C) \)

Figure 5-5: The three applicability constraints of the NE2C argument scheme.

The No-Effect-to-Cause argument scheme justifies a belief that the state of the potential causal variable \( A \) is below its threshold value \( V_A \), based on the absence of any effect on \( B \), i.e., \( B \)'s value is lower than its threshold value \( V_B \). The warrant for this particular strategy is \( S^+ ([A, V_A], [B, V_B]) \), i.e., that the state \([A, V_A]\) could result in the state \([B, V_B]\). There are three independent applicability constraints. First, if there exists another variable \( C \) such that the relation \( Y^+ ([C, V_C], [A, V_A], [B, V_B]) \) holds but the value of \( C \) has not reached its threshold value \( V_C \), then \( A \) has not been enabled to cause \( B \). Thus, it is not valid to make any claim regarding the state of \( A \). Second, if there exists another variable \( C \) such that the relation \( Y^- ([C, V_C], [A, V_A], [B, V_B]) \) holds and the value of \( C \) has reached its threshold value \( V_C \), then it is possible that \( C \) has prevented \( B \)'s value from reaching its threshold. Third, if there exists another variable \( C \) such that the relation \( X^0 ([C, V_C], [A, V_A], [B, V_B]) \) holds, i.e., \( A \) and \( C \)
are jointly required for $B$ to reach its threshold, but the value of $C$ has not reached its threshold value $V_C$, then $C$’s value being under its threshold value $V_C$ can be responsible for the value of $B$ not reaching its threshold.

**Argument Scheme: INC–RISK (Increased Risk)**

Claim: $B \geq V_B$

Data: $A \geq V_A$

Warrant: $S^+ ([A, V_A], [B, V_B])$

Applicability Constraints:

1) $\neg (\exists C Y^+ ([C, V_C], [A, V_A], [B, V_B]), C < V_C)$

2) $\neg (\exists C Y^- ([C, V_C], [A, V_A], [B, V_B]), C \geq V_C)$

The Increased-Risk argument scheme reasons forward to establish a belief. The claim that the value of $B$ has reached its threshold value $V_B$ is supported by the data that the value of $A$ has reached its threshold value $V_A$. The warrant is $S^+ ([A, V_A], [B, V_B])$, i.e., that the state $[A, V_A]$ increases the likelihood of the state $[B, V_B]$. There are two independent applicability constraints, similar to the first two constraints of the NE2C argument scheme described above.

**Argument Scheme: E2JC (Effect to Joint Cause)**

Claim: $(A \geq V_A) \land (B \geq V_B)$

Data: $C \geq V_C$

Warrant: $X^0 ([A, V_A], [B, V_B], [C, V_C])$
The Effect-to-Joint-Cause argument scheme justifies a belief in a claim of joint responsibility, based on a possible effect. The claim, that the state of $A$ has reached its threshold value $V_A$ and that the state of $B$ has also reached its threshold value $V_B$, is supported by the data that the value of $C$ has reached its threshold value $V_C$. The warrant is $X^0 (\left[ A, V_A \right], \left[ B, V_B \right], \left[ C, V_C \right])$, i.e., that together the states $[A, V_A]$ and $[B, V_B]$ could be jointly responsible for the state $[C, V_C]$.

**Argument Scheme: E2XORC (Effect to Exclusive–Or Cause)**

Claim: \[ A \geq V_A \text{ xor } B \geq V_B \]

Data: \[ C \geq V_C \]

Warrant: \[ X^- (\left[ A, V_A \right], \left[ B, V_B \right], \left[ C, V_C \right]) \]
The Effect-to-Exclusive-Or-Cause argument scheme justifies a claim that either the state of $A$ has reached its threshold value $V_A$ or that the state of $B$ has reached its threshold value $V_B$. The warrant is $X^\rightarrow ([A, V_A], [B, V_B], [C, V_C])$, i.e., that the states $[A, V_A]$ and $[B, V_B]$ are mutually exclusive possible causes of the state $[C, V_C]$. The data is that the value of $C$ has reached its threshold value $V_C$. The strategy can be extended to cover the case when there are more than two alternative causes as well. Currently in our system, this argument scheme works with either two or three alternative causes.

**Argument Scheme: ELIM (Elimination)**

Claim: $A = V_{A1}$

Data: $(A = V_{A1}) \lor (A = V_{A2}),$  
$\neg (A = V_{A2})$

The Elimination argument scheme is based on the *unit resolution* inference rule [Russell and Norvig, 2003]. If the variable $A$ has as its allowable range of value $[V_{A1}, V_{A2}, \ldots, V_{Ax}, V_{Ay}]$, 

\[ 47 \]
and we know the value of $A$ is not equal to any of the $x$ out of the total $y$ values (with $x = y - 1$), i.e. $[V_{A2}, \ldots, V_{Ax}, V_{Ay}]$, then we can infer by logical inference that it must be equal to the only remaining value, $V_{AI}$.

Figure 5-8: A sample scenario illustrating a variant of the ELIM argument scheme.

A variant of the ELIM argument scheme is illustrated in Figure 5-8. Instead of eliminating one possible value for a single variable, two mutually exclusive sets of variable assignment are being considered. Formally, the relationship can be stated as follows:

Claim: \hspace{1cm} X = V_X

Data: \hspace{1cm} (X = V_X) \text{ xor } (Y = V_Y), \hspace{1cm} \neg (Y = V_Y)

As shown in Figure 5-8, a mutated gene on either the mother’s or the father’s side is mutually exclusively responsible for the mutated gene causing an autosomal dominant disease found in the proband. Knowing that the father does not have the mutated gene, we can eliminate the
fact that the proband inherited the disease from his father and instead conclude that he inherited the disease from his mother.

**Argument Scheme: CR (Conjunction Reduction)**

Claim: \((A = V_{A1}) \lor (A = V_{A2})\)

Data: \((A = V_{A1} \lor A = V_{A2}) \land (B = V_{B1} \lor B = V_{B2})\)

The Conjunction-Reduction argument scheme is based on the *and-elimination* inference rule [Russell and Norvig, 2003]. If we know that the value of \(A\) is one of \([V_{A1}, V_{A2}]\) and the value of \(B\) is one of \([V_{B1}, V_{B2}]\), then we can infer that the value of variable \(A\) is equal to either \(V_{A1}\) or \(V_{A2}\). Based on the principle of logical conjunction, having the conjunctive clause evaluated to *true* implies that each and every one of the conjuncts must be *true*. The next section will explain in more detail how this particular argument scheme was implemented in GenIE. Note that in our system, the logical connectives *and*, *exclusive-or*, and *not*, used in this argument scheme, as well as others, are represented as *log_and*, *log_or*, and *log_not* respectively.

**5.3. Complex Argument Generation**

As mentioned earlier, it is possible for an argument to be formed as a sequence of successive arguments. The data of an argument may itself serve as the claim in a subsequent recursive search for an argument. In this section, we provide an example of a series of arguments formed as a result of the search for an argument in the CF KB.
Figure 5-9 shows an argument for the claims that the mother has exactly one mutated CFTR allele \((gm = 1)\) and the father has exactly one mutated CFTR allele \((gf = 1)\). The argument for the claim about the mother is similar to the argument regarding the father so we will only describe the former. According to the ELIM argument scheme, an argument for the claim \(gm = 1\) can be made if there exists data that \(\neg(gm = 2)\) and that \((gm = 1 \lor gm = 2)\). An NE2C argument for the claim that \(\neg(gm = 2)\), i.e. \((gm < 2)\), can be made from the data that the mother has no symptoms of CF. By the CR argument scheme, an argument for the claim that
An E2JC argument for the claim \(((gm = 1 \lor gm = 2) \land (gf = 1 \lor gf = 2))\) can be made from the data that \((gp = 2)\).

For a dplan containing this argument, see Figure A-3 in the appendix section.

5.4. Implementation

![Diagram](image)

*Figure 5-10: The data exchange between the discourse generator and the argument generator.*

When the DG passes claims to the AG, the claims are first translated from event propositions (see Chapter 4) into a format that is compatible with the KB, using the API predicate `translate_from_event`. Each argument is later translated from KB format back into RST tree format using the API predicate `translate_to_event`. It is possible for a claim to have more than one argument. If such is the case, the AG returns a complete list of all arguments supporting the same claim. On the other hand, if no argument can be constructed for the
specified claim, the empty list is returned. Figure 5-10 above illustrates the data exchange between the DG and the AG.

In order to prevent the system from repeating the attempt to prove the same claim, internally, a list of proven claims is maintained throughout the length of program execution. When the AG receives a request to prove a certain claim, the claim is first checked against the proven claims list to see whether it has previously been proven. The AG proceeds to prove the claim only when the claim is not already in the list. The process helps to avoid redundancy in the resulting dplan.

With the argument generation process being recursive, we need a proper terminating case to avoid the possibility of execution going into an infinite loop. One way is to check whether the data is an observable fact. Otherwise, we try to recursively prove the validity of the data, forcing the invocation of the predicate get_argument to end once the same claim is encountered twice. To achieve this, during recursion, a list of previously encountered subgoals, with the current claim as the most current subgoal added to it, is passed with the recursive call. This list of previously encountered subgoals is different from the proven claims list in the sense that a subgoal represents a partially proven claim whereas a proven claim implies that the claim has once been completely proven. Both conditions serve as terminating cases to the recursion. The first ends it with a failure, i.e., execution fails when the same subgoal is encountered twice, signaling an endless loop. The latter, on the other
hand, ends it with a success, i.e. the claim can be considered valid since it has been completely proven before.

To prove the Conjunction-Reduction argument scheme as described in the previous section, instead of a brute-force method, we use a heuristic search in the attempt to match the argument pattern with information presented in the KB. With the claim being that a variable node $A$ has a value of either $V_{A1}$ or $V_{A2}$, we first try to match the node as part of an $X^0 ([A, V_A], [B, V_B], [C, V_C])$ relation. Once matched, the expected value ($V_A$) is confirmed with those in the original claim (either $V_{A1}$ or $V_{A2}$) before the resulting conjunction-reduction argument is formed.

In the current implementation of the letter version of the AG, all applicability constraints are internally checked before each argument is formed. Before an $S^+$ (positive influence) relation for the E2C, NE2C, and INC-RISK argument schemes is matched, the nodes forming such relation are pre-screened to make sure they are not part of any other qualitative relations including $Y^+, Y^-, X^-$, or $X^0$, with the only exception being the subcase of the E2C argument scheme which allows a $Y^+$ relation to be present in the last section of the $S^+$ chain. The mapping for all relations also checks to confirm that each expected component’s value matches that of the corresponding node’s value at the time in question.

The Increased-Risk argument scheme forms what is considered a relatively weak argument, when compared to the other strategies. As its name implies, this type of argument
characterizes just an increase in risk. To suppress the weaker arguments generated by this particular argument scheme, we included a parameter in the `get_argument` function to represent the resulting argument’s strength. The DG controls the degree of argument strength required by each of its sections. In the current implementation, we only allow weak arguments in the third section of the `dplan`, the `doc_source`.

As mentioned previously, after it is generated, each argument is translated from KB format back into RST format using the API predicate `translate_to_event`. In order to avoid duplicate events, the translation process first checks whether an event already exists by searching for matching events in the event model (see Chapter 4). If a match is found, the event ID is reused. Otherwise, a new event is created to represent the event.

To represent a newly created event, we need a way to dynamically generate unique event identifiers. To achieve this, we maintain a list of numerical indices for the generated event IDs. The list is first initialized as an empty list, and then it gets incrementally populated as new events get created. To generate a new event ID, we take the length of the current list, increment it by one, then concatenate it with the prefix `ge`. The list is then updated by appending the newly generated index to the old list. This way, the very first event generated will be `ge1`, since the length of the list is initially zero. The prefix `ge` is used with the intention to distinguish generated events from default events, whose prefix is `e`. The default events are those events generated by the DG, most of which exist for all generated patient letters.
Event propositions representing the data or claim of an argument describe specific individuals and situations. Consider the example below. Both event propositions are similar. The first one is a non-generic event about a specific individual (the proband), and the second is a generic event about non-specific individuals.

- event(ge1, [time(pretest), duration(continuing)], [action(experience), experiencer(proband), attribute([gp, 'CFTR', =, 2], disjunctive)], pretest).
  
  Paraphrase: At pretest, the proband has two mutated CFTR alleles.

- event(ge2, [time(generic), duration(continuing)], [action(experience), experiencer(generic), generic_attribute([genotype, 'CFTR', =, 2]), generic).
  
  Paraphrase: Someone has two mutated CFTR alleles.

For events representing the joint responsibility (X⁰) and exclusive-or responsibility relations, found only in description of inherited genotypes, the experiencer component is explicitly labeled as either generic_mother, or generic_father, as opposed to the usual generic, so that later it is possible to distinguish which parent is under discussion.

As can be seen in the previous section, some argument schemes from logic do not have warrants. Incorporating arguments generated from these schemes into the final RST tree results in null leaves that represent the empty warrants. To get rid of these empty leaves, we prune the original RST tree once, getting rid of an empty branch and promote its sibling node in place of the former parent. Figure 5-11 illustrates the tree transformation that takes place during the pruning process.
Figure 5-11: An RST tree before and after pruning.
One contribution of this thesis involves the modification of the argument generator described in the previous chapter for use with an interactive driver that allows a user to interact with the argument generator in real time. The interactive version is designed to be used as a temporary means to test the functionality of the system until the natural language realizer is finished.

The argument generator used in the interactive version is slightly different from that used in the letter version. Only the first three argument schemes described in section 5.2, including their subcases, are supported in the interactive version, namely E2C, NE2C, and INC-RISK. To accommodate stepping through the argumentation process, the check done internally for each applicability constraint was removed. Instead, all constraints are displayed to the user as “critical questions” [Green, submitted]. The user can explore the argument’s critical question if desired. In contrast to the letter version, the arguments generated by the interactive version do not get translated back into RST format. They are presented to the user in the KB format.

The interactive driver is modeled as a finite state machine, through which the user has total control over the argument generation process. The user has an option to query an argument for a particular claim by probing further into its data, warrant, or critical questions. On the
other hand, the user can also choose to challenge the current claim by exploring its counterclaims to see if any arguments are available to invalidate the claim.

6.1. Finite State Machine

The interactive argument generator is modeled as a finite state machine (FSM), as shown in Figure 6-1 below. Each user input action leads to the next state in the FSM.

* d/w/q/c selections are not among valid actions in certain scenarios

Figure 6-1: FSM representing the interactive argument generator’s architecture.
In the Knowledge Base Selection state, the user is given a list of five knowledge bases to choose from. These include the same four knowledge bases we use to test the letter version of GenIE, namely Cystic Fibrosis, Achondroplasia, Familial Hypercholesterolemia, and Phenylketonuria (PKU), as well as an extra Sample Test knowledge base which was added initially for unit testing. Upon valid KB selection, the control is transitioned to the next state. If, however, an invalid selection is encountered, the control loops back to the current state and the user is asked to perform her selection again.

Next, in the KB State Selection state, the user is presented with a list of corresponding states for the selected KB. All KBs include pretest, the state of time before any types of testing has been done to the proband. A selected number of KBs include a second state – posttest, which represents the state of time after testing has been done to the proband, as well as a third state – future, which represents the state of time in the future after the diagnosis has been made. The control transitions to the next state once a valid KB state selection has been made by the user.

In the Node Selection state, a complete list of all nodes present in the selected KB for the selected KB state is presented to the user. Once a valid node is selected, the user finds herself in the next state – the Claim Selection state. The choice of claims for a particular node depends on the node’s designated domain and the domain’s range of potential values. For example, if a node belongs to the boolean domain, then there will be two different possible
claims, one with the node’s value being *true*, and another with the node’s value being *false*. After the user selects a claim, the FSM enters the next state – the Argument Generation state.

<table>
<thead>
<tr>
<th>Choice</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>To explore the data part of the argument</td>
</tr>
<tr>
<td>w</td>
<td>To explore the warrant part of the argument</td>
</tr>
<tr>
<td>q</td>
<td>To explore the critical question part of the argument</td>
</tr>
<tr>
<td>c</td>
<td>To see the counterclaims for the current claim</td>
</tr>
<tr>
<td>n</td>
<td>To select a new node in the same KB</td>
</tr>
<tr>
<td>s</td>
<td>To select a different state for the same KB</td>
</tr>
<tr>
<td>k</td>
<td>To start over with another KB</td>
</tr>
<tr>
<td>x</td>
<td>To end this query by exiting the program</td>
</tr>
</tbody>
</table>

In the Argument Generation state, there are two possible scenarios. If one or more arguments are generated for the claim, the user has a choice of eight possible actions to lead her to the next state. A complete list of choices and the action to which each of them corresponds can be found below in Table 6-1. If no arguments are generated for the claim or if the claim is an observable fact, the user can choose only among the last five options listed above. A claim is considered observable if it is about an *observable* node type, which includes Behavior, Complication, History, Symptom, and Test Result.
In the case where one or more arguments is generated for the current claim, the user may select a new claim from the components of one of the arguments: \( d \) for data, \( w \) for warrant, \( q \) for critical question, or \( c \) for counterclaim. The control loops back to the same state – the argument generation state, with the selected component now being the current claim. Figure 6-2 below shows the detailed FSM architecture around the argument generation state for this particular scenario.

![Figure 6-2: Detailed FSM during the Argument Generation state.](image)

The only accepting state of this particular FSM is the Exit state, which can be reached by the user selecting action \( x \) to exit the query, after one or more claim has been selected.
6.2. User Interface

The user interface for the interactive argument generator has been kept simple in an easy-to-read textual format. After loading the program, the user starts a query by typing in the goal `start_query` to the Prolog command prompt. Figure 6-3 below shows a typical set of steps that guide the user through her selection of KB, KB state, and node respectively.

![Image of Prolog interface]

Figure 6-3: Example user interaction during KB, KB state, and node selections.

Once arguments for a claim have been generated, they are listed in an easy-to-read format as demonstrated in Figure 6-4 below. As described in the previous chapter, an argument consists of a claim, a list of data, a list of warrants, and a list of critical questions.
Numerical indices are assigned to each list item for the purpose of user selection. When no arguments can be generated for the given claim or when the claim is considered to be an observable fact, such information is explicitly relayed to the user.

Figure 6-4: Argument Listing Format.

Figure 6-5: An example of user interaction during warrant selection.
Another interesting user interface feature can be seen during data/warrant/critical question selection. Since there can be multiple arguments, as well as multiple data/warrants/critical questions, we need to break the user input collection process into two parts, one to identify the argument selected and the other to identify the data/warrant/critical question selected. An example of user interaction during this stage is demonstrated in Figure 6-5.

To avoid invalid selection of KBs, KB states, nodes, claims, as well as counterclaims, data, warrant, and critical questions, every user input is verified to ensure it is legal. In the case when a mistake is made, the user will first be informed that an invalid selection has been encountered, then will be presented with the same choices and asked to choose again.

At this point, the current user interface design does not let the user step back to a previous stage. For example, let say three different arguments have been generated for claim A. The user decides to further explore the first data of the first argument as the next claim – B. After making her selection, she is not satisfied with the new arguments generated for claim B and wants to go back and inspect the original arguments for claim A in a different way. The only way to do that is to select ‘n’ as the next move, reselect the same node – A, then the same claim, and have the arguments regenerated. This may seem like a rather laborious way to achieve such goal, however, the design greatly simplifies the FSM and helps minimizes the amount of data we need to maintain in memory during runtime.
6.3. Implementation

As depicted in the FSM model in the previous section, the user is initially asked to choose a KB from the list presented. With a valid KB selection, the prolog file corresponding to the KB gets loaded internally. For node selection, only the nodes present in the selected KB at the selected time are listed for the user to choose from.

A choice of possible claims is determined by the node being selected. To generate the list of claims, we first check the domain of the selected node then generate one claim for each of the possible values. For example, if the node belongs to domain boolean, two claims will be generated, one for the node being true, and another for the node being false. On the other hand, if the node is of domain recessive_mut, three claims will be generated, one for the node being 0, one for the node being 1, and yet another for the node being 2. We need to keep track of the list of claims generated, so that later we can come back and do a lookup to match the right claim with the numerical index entered by the user.

Once a proof of a claim has been requested by the user, it is first checked to see whether it qualifies as an observable fact. If so, there is no need to prove the validity of the claim. The user is informed of the observation and asked to select a next move. On the other hand, if the claim is not observable, an attempt will be made to find argument(s) that supports the claim.

In preparation for argument generation, an inspection is done to see whether it is possible to represent this particular claim in a different format. Generally, this is true when the claim is
in the format of \([Node, Operator, Value_X]\) and \(Node\) belongs to a binary domain with two possible values: \(Value_X\) and \(Value_Y\). To say that the node’s value is equal to \(Value_X\) is generally the same as saying that it is \(not\) equal to \(Value_Y\). In this case, arguments will be generated for both the original claim \([Node, Operator, Value_X]\) as well as the alternate claim \(log_not[Node, Operator, Value_Y]\).\(^7\) The reason behind such transformation is because the NE2C argument scheme in particular expects an argument in the format of \(log_not[Node, Operator, Value]\). Rearranging the original claim into this particular format covers the chance that an NE2C argument will be generated.

In a non-binary domain, however, similar assumption cannot be made. If an allowable range of values in the domain includes \([Value_X, Value_Y, Value_Z]\), then saying that a node’s value is equal to \(Value_X\) is neither the same as saying that it is \(not\) equal to \(Value_Y\) nor the same as saying that it is \(not\) equal to \(Value_Z\). By saying that a node’s value is \(not\) equal to \(Value_Y\), it is equally likely that it may be equal to either \(Value_X\) or \(Value_Z\).

In the case that one or more arguments can be generated for the claim selected by the user, she will be given a complete list of the next possible moves, listed in Table 5-1. However, if the claim is considered an observable fact, or if no arguments can be generated for the claim, the list of possible next moves is reduced to those regarded as \(general moves\), which is essentially everything in Table 5-1 excluding the first three choices.

\(^7\) \(log_not\) is the internal representation we use for the logical operator \(not\), indicating the negation of the statement that follows it.
A counterclaim is by definition a claim that is incompatible to the original claim. When the user chooses to explore a counterclaim of the current claim, she will be presented with a list of all possible counterclaims. To generate a list of counterclaims, we inspect the node’s domain to obtain the list all possible values for that particular domain. Each counterclaim is then constructed using a value from the list that does not match the value stated in the current claim. For example, if the domain permits values in the range of [0, 1, 2] and the current claim states that the node’s value is equal to 2, then the counterclaim list will consist of two different claims, one with the node’s value equal to 0, and the other with the node’s value equal to 1. On the other hand, if the current claim states that the node’s value is not equal to a certain value $X$, then the only possible counterclaim for that will be a claim that the node’s value is equal to that same value $X$.

If the user decides to choose the option $k$ for her next move, indicating that she wants to start over with a new knowledge base, a cleanup operation is done to get the argument generator ready to accept queries on the new KB. Prolog’s built-in *abolish* utility serves this purpose nicely by removing all traces of the predicates defined in the previous KB. This operation is necessary since all KBs share similar predicates as described in Chapter 3. Moreover, since the same convention is used to determine node IDs in all KBs, we end up with the same node ID representing totally different concepts from one KB to another.
6.4. Example Dialogues

In this section, we provide two example dialogues to illustrate how the interactive driver works. The first uses the Achondroplasia KB as a reference, showing arguments for claims related to the genotype of the proband’s parents. The second uses the Sample Test KB to show other capabilities of the interactive driver not covered in the first example. Figure 6-6 shows the Sample Test KB as a reference.

![Diagram of Sample Test KB]

Figure 6-6: Sample Test KB.

Example Dialogue #1: Achondroplasia

Welcome to SWI-Prolog (Multi-threaded, Version 5.6.25)
Copyright (c) 1990-2006 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under certain conditions. Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- start_query.
% API.pl compiled 0.03 sec, 66,132 bytes
% ArgumentGenerator.pl compiled 0.00 sec, 3,760 bytes

Please select a knowledge base:
1 for Cystic Fibrosis
2 for Achondroplasia
3 for Familial Hypercholesterolemia
4 for Phenylketonuria (PKU)
5 for Sample Test

68
Please pick the state of the KB you selected:
  b for before the test
  a for after the test
  f for future
|
|: a.

Please pick a node for the KB state you selected:
  bp for proband’s biochemistry: FGFR3 protein
  ef for father’s event: mutation
  gf1 for father’s genotype: FGFR3
  gf2 for father’s genotype_germline: FGFR3 germline
  gm for mother’s genotype: FGFR3
  gp for proband’s genotype: FGFR3
  gs1 for paternal_half_sib1’s genotype: FGFR3
  gs2 for paternal_half_sib2’s genotype: FGFR3
  gs3 for paternal_half_sib3’s genotype: FGFR3
  hf1 for father’s history: family history of skeletal dysplasia, birth defects, genetic disorders
  hf2 for father’s history: age
  hm for mother’s history: family history of skeletal dysplasia, birth defects, genetic disorders
  pp for proband’s physiology: bone development
  sf for father’s symptom: Achondroplasia symptoms
  sm for mother’s symptom: Achondroplasia symptoms
  ss1 for paternal_half_sib1’s symptom: Achondroplasia symptoms
  ss2 for paternal_half_sib2’s symptom: Achondroplasia symptoms
  ss3 for paternal_half_sib3’s symptom: Achondroplasia symptoms
  tresp1 for proband’s test_result: fetal rhizomelia
  tresp2 for proband’s test_result: fetal macrocephaly
  tresp3 for proband’s test_result: G1138A mutation in FGFR3 gene
  tresp4 for proband’s test_result: G1138C mutation in FGFR3 gene
  tresp5 for proband’s test_result: radiographic features of Achondroplasia
  tsp1 for proband’s test: DNA test
  tsp2 for proband’s test: radiograph at birth
  tsp3 for proband’s test: fetal ultrasound
|
|: gm.

Please pick a claim for the node you selected:
  1 for [gm, =, 0]
  2 for [gm, =, 1]
|
|: 1.

Argument #1
CLAIM:
  log_not([gm, =, 1])
DATA:
  1) log_not([sm, [], true})
WARRANT:
  1) has_pos_infll([gm, =, 1], [sm, [], true})
CRITICAL QUESTION:
  1)[]

Please pick your next move:
Enter d to select a data
Enter w to select a warrant
Enter q to select a critical question
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator

Please pick a counterclaim:
1 for \([gm, =, 1]\)
2 for \(\log\_\text{not}(gm, =, 0)\)

|: 1.

Argument #1
CLAIM:
\([gm, =, 1]\)
DATA:
1) \([sp1, [], true]\)
WARRANT:
1) has_pos_infl\(\{gm, =, 1\}, \{gp, =, 1\}\)
2) has_pos_infl\(\{gp, =, 1\}, \{bp, [], abnormal\}\)
3) has_pos_infl\(\{bp, [], abnormal\}, \{pp, [], abnormal\}\)
4) has_pos_infl\(\{pp, [], abnormal\}, \{sp1, [], true\}\)
CRITICAL QUESTION:
1) \([xor\_\text{resp}(\{gm, =, 1\}, \{gf2, =, 1\}, \{gf1, =, 1\}\), \{gp, =, 1\}], \{gf2, =, 1\}\]
2) \([xor\_\text{resp}(\{gm, =, 1\}, \{gf2, =, 1\}, \{gf1, =, 1\}\), \{gp, =, 1\}], \{gf1, =, 1\}\]

Argument #2
CLAIM:
\([gm, =, 1]\)
DATA:
1) \([tresp1, [], true]\)
2) \([tstp3, [], done]\)
WARRANT:
1) has_pos_infl\(\{gm, =, 1\}, \{gp, =, 1\}\)
2) has_pos_infl\(\{gp, =, 1\}, \{bp, [], abnormal\}\)
3) has_pos_infl\(\{bp, [], abnormal\}, \{pp, [], abnormal\}\)
4) enables\(\{\{tstp3, [], done\}, \{pp, [], abnormal\}\}, \{tresp1, [], true\}\)
CRITICAL QUESTION:
1) \([xor\_\text{resp}(\{gm, =, 1\}, \{gf2, =, 1\}, \{gf1, =, 1\}\), \{gp, =, 1\}], \{gf2, =, 1\}\]
2) \([xor\_\text{resp}(\{gm, =, 1\}, \{gf2, =, 1\}, \{gf1, =, 1\}\), \{gp, =, 1\}], \{gf1, =, 1\}\]

Argument #3
CLAIM:
\([gm, =, 1]\)
DATA:
1) \([tresp2, [], true]\)
2) \([tstp3, [], done]\)
WARRANT:
1) has_pos_infl\(\{gm, =, 1\}, \{gp, =, 1\}\)
2) has_pos_infl\(\{gp, =, 1\}, \{bp, [], abnormal\}\)
3) has_pos_infl\(\{bp, [], abnormal\}, \{pp, [], abnormal\}\)
4) enables\(\{\{tstp3, [], done\}, \{pp, [], abnormal\}\}, \{tresp2, [], true\}\)
CRITICAL QUESTION:
1) \([xor\_\text{resp}(\{gm, =, 1\}, \{gf2, =, 1\}, \{gf1, =, 1\}\), \{gp, =, 1\}], \{gf2, =, 1\}\]
2) \([xor\_\text{resp}(\{gm, =, 1\}, \{gf2, =, 1\}, \{gf1, =, 1\}\), \{gp, =, 1\}], \{gf1, =, 1\}\]

Argument #4
CLAIM:
\([gm, =, 1]\)
DATA:
1) [tresp3, [], true]
2) [tstp1, [], done]

WARRANT:
1) has_pos_infl([gm, =, 1], [gp, =, 1])
2) enables([tstp1, [], done], [gp, =, 1], [tresp3, [], true])

CRITICAL QUESTION:
1) [xor_resp([gm, =, 1], [gf2, =, 1], [gf1, =, 1], [gp, =, 1], [gf2, =, 1])]
2) [xor_resp([gm, =, 1], [gf2, =, 1], [gf1, =, 1], [gp, =, 1], [gf1, =, 1])]

Argument #5
CLAIM:
[gm, =, 1]

DATA:
1) [tresp5, [], true]
2) [tstp2, [], done]

WARRANT:
1) has_pos_infl([gm, =, 1], [gp, =, 1])
2) has_pos_infl([gp, =, 1], [bp, [ ], abnormal])
3) has_pos_infl([bp, [ ], abnormal], [pp, [ ], abnormal])
4) enables([tstp2, [], done], [pp, [ ], abnormal], [tresp5, [], true])

CRITICAL QUESTION:
1) [xor_resp([gm, =, 1], [gf2, =, 1], [gf1, =, 1], [gp, =, 1], [gf2, =, 1])]
2) [xor_resp([gm, =, 1], [gf2, =, 1], [gf1, =, 1], [gp, =, 1], [gf1, =, 1])]

Please pick your next move:
Enter d to select a data
Enter w to select a warrant
Enter q to select a critical question
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: q.

Please enter N for the N-th argument: 5.

Please enter M for the M-th critical question: 1.

Argument #1
CLAIM:
[gf2, =, 1]

DATA:
1) [ef, [ ], true]

WARRANT:
1) has_pos_infl([ef, [ ], true], [gf2, =, 1])

CRITICAL QUESTION:
1) []

Argument #2
CLAIM:
[gf2, =, 1]

DATA:
1) [sp1, [ ], true]

WARRANT:
1) has_pos_infl([gf2, =, 1], [gp, =, 1])
2) has_pos_infl([gp, =, 1], [bp, [ ], abnormal])
3) has_pos_infl([bp, [ ], abnormal], [pp, [ ], abnormal])
4) has_pos_infl([pp, [ ], abnormal], [sp1, [ ], true])
ARGUMENT #3
CLAIM: 
\[ gf2, =, 1 \]
DATA: 
1) \[ tresp1, [], true \]
2) \[ tstp3, [], done \]
WARRANT: 
1) \( \text{has\_pos\_infl}(gf2, =, 1), (gp, =, 1) \)
2) \( \text{has\_pos\_infl}(gp, =, 1), (bp, [], abnormal) \)
3) \( \text{has\_pos\_infl}(bp, [], abnormal), (pp, [], abnormal) \)
4) \( \text{enables}(tstp3, [], done), (pp, [], abnormal), (tresp1, [], true) \)

ARGUMENT #4
CLAIM: 
\[ gf2, =, 1 \]
DATA: 
1) \[ tresp2, [], true \]
2) \[ tstp3, [], done \]
WARRANT: 
1) \( \text{has\_pos\_infl}(gf2, =, 1), (gp, =, 1) \)
2) \( \text{has\_pos\_infl}(gp, =, 1), (bp, [], abnormal) \)
3) \( \text{has\_pos\_infl}(bp, [], abnormal), (pp, [], abnormal) \)
4) \( \text{enables}(tstp3, [], done), (pp, [], abnormal), (tresp2, [], true) \)

ARGUMENT #5
CLAIM: 
\[ gf2, =, 1 \]
DATA: 
1) \[ tresp3, [], true \]
2) \[ tstp1, [], done \]
WARRANT: 
1) \( \text{has\_pos\_infl}(gf2, =, 1), (gp, =, 1) \)
2) \( \text{enables}(tstp1, [], done), (gp, =, 1), (tresp3, [], true) \)

ARGUMENT #6
CLAIM: 
\[ gf2, =, 1 \]
DATA: 
1) \[ tresp5, [], true \]
2) \[ tstp2, [], done \]
WARRANT: 
1) \( \text{has\_pos\_infl}(gf2, =, 1), (gp, =, 1) \)
2) \( \text{has\_pos\_infl}(gp, =, 1), (bp, [], abnormal) \)
3) \( \text{has\_pos\_infl}(bp, [], abnormal), (pp, [], abnormal) \)
4) \( \text{enables}(tstp2, [], done), (pp, [], abnormal), (tresp5, [], true) \)
1) \( \text{xor}_\text{resp}([\text{gf}_2, =, 1], [\text{gm}, =, 1], [\text{gf}_1, =, 1]), [\text{gm}, =, 1] \)

2) \( \text{xor}_\text{resp}([\text{gf}_2, =, 1], [\text{gm}, =, 1], [\text{gf}_1, =, 1]), [\text{gp}, =, 1] \)

Please pick your next move:
Enter d to select a data
Enter w to select a warrant
Enter q to select a critical question
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator

|: d.

Please enter N for the N-th argument: 1.

Please enter M for the M-th data: 1.

Argument #1
CLAIM:
[ef, [], true]
DATA:
1) \([\text{hf}_2, =, 45]\)
WARRANT:
1) \(\text{has}_\text{pos}_\text{infl}(\text{hf}_2, >=, 35), [\text{ef}, [], true]\)
CRITICAL QUESTION:
1) {}

Argument #2
CLAIM:
[ef, [], true]
DATA:
1) \([\text{sp}_1, [], true]\)
WARRANT:
1) \(\text{has}_\text{pos}_\text{infl}([\text{ef}, [], true], [\text{gf}_2, =, 1])\)
2) \(\text{has}_\text{pos}_\text{infl}([\text{gf}_2, =, 1], [\text{gp}, =, 1])\)
3) \(\text{has}_\text{pos}_\text{infl}([\text{gp}, =, 1], [\text{bp}, [], \text{abnormal}])\)
4) \(\text{has}_\text{pos}_\text{infl}([\text{bp}, [], \text{abnormal}], [\text{pp}, [], \text{abnormal}])\)
5) \(\text{has}_\text{pos}_\text{infl}([\text{pp}, [], \text{abnormal}], [\text{sp}_1, [], true])\)
CRITICAL QUESTION:
1) \(\text{xor}_\text{resp}([\text{gf}_2, =, 1], [\text{gm}, =, 1], [\text{gf}_1, =, 1]), [\text{gm}, =, 1] \)
2) \(\text{xor}_\text{resp}([\text{gf}_2, =, 1], [\text{gm}, =, 1], [\text{gf}_1, =, 1]), [\text{gp}, =, 1] \)

Argument #3
CLAIM:
[ef, [], true]
DATA:
1) \([\text{tr}_\text{esp}_1, [], true]\)
2) \([\text{tstp}_3, [], \text{done}]\)
WARRANT:
1) \(\text{has}_\text{pos}_\text{infl}([\text{ef}, [], true], [\text{gf}_2, =, 1])\)
2) \(\text{has}_\text{pos}_\text{infl}([\text{gf}_2, =, 1], [\text{gp}, =, 1])\)
3) \(\text{has}_\text{pos}_\text{infl}([\text{gp}, =, 1], [\text{bp}, [], \text{abnormal}])\)
4) \(\text{has}_\text{pos}_\text{infl}([\text{bp}, [], \text{abnormal}], [\text{pp}, [], \text{abnormal}])\)
5) \(\text{enables}([\text{tstp}_3, [], \text{done}], [\text{pp}, [], \text{abnormal}], [\text{tr}_\text{esp}_1, [], true])\)
CRITICAL QUESTION:
1) \(\text{xor}_\text{resp}([\text{gf}_2, =, 1], [\text{gm}, =, 1], [\text{gf}_1, =, 1]), [\text{gm}, =, 1] \)
2) \(\text{xor}_\text{resp}([\text{gf}_2, =, 1], [\text{gm}, =, 1], [\text{gf}_1, =, 1]), [\text{gp}, =, 1] \)
Argument #4
CLAIM:
[ef, [], true]
DATA:
1) [tresp2, [], true]
2) [tstp3, [], done]
WARRANT:
1) has_pos_infl([ef, [], true], [gf2, =, 1])
2) has_pos_infl([gf2, =, 1], [gp, =, 1])
3) has_pos_infl([gp, =, 1], [bp, [], abnormal])
4) has_pos_infl([bp, [], abnormal], [pp, [], abnormal])
5) enables([[tstp3, [], done], [pp, [], abnormal]], [tresp2, [], true])
CRITICAL QUESTION:
1) [xor_resps([gf2, =, 1], [gm, =, 1], [gf1, =, 1]), [gp, =, 1], [gm, =, 1]]
2) [xor_resps([gf2, =, 1], [gm, =, 1], [gf1, =, 1]), [gp, =, 1], [gf1, =, 1]]

Argument #5
CLAIM:
[ef, [], true]
DATA:
1) [tresp3, [], true]
2) [tstp1, [], done]
WARRANT:
1) has_pos_infl([ef, [], true], [gf2, =, 1])
2) has_pos_infl([gf2, =, 1], [gp, =, 1])
3) enables([[tstp1, [], done], [gp, =, 1]], [tresp3, [], true])
CRITICAL QUESTION:
1) [xor_resps([gf2, =, 1], [gm, =, 1], [gf1, =, 1]), [gp, =, 1], [gm, =, 1]]
2) [xor_resps([gf2, =, 1], [gm, =, 1], [gf1, =, 1]), [gp, =, 1], [gf1, =, 1]]

Argument #6
CLAIM:
[ef, [], true]
DATA:
1) [tresp5, [], true]
2) [tstp2, [], done]
WARRANT:
1) has_pos_infl([ef, [], true], [gf2, =, 1])
2) has_pos_infl([gf2, =, 1], [gp, =, 1])
3) has_pos_infl([gp, =, 1], [bp, [], abnormal])
4) has_pos_infl([bp, [], abnormal], [pp, [], abnormal])
5) enables([[tstp2, [], done], [pp, [], abnormal]], [tresp5, [], true])
CRITICAL QUESTION:
1) [xor_resps([gf2, =, 1], [gm, =, 1], [gf1, =, 1]), [gp, =, 1], [gm, =, 1]]
2) [xor_resps([gf2, =, 1], [gm, =, 1], [gf1, =, 1]), [gp, =, 1], [gf1, =, 1]]

Please pick your next move:
Enter d to select a data
Enter w to select a warrant
Enter q to select a critical question
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: x.

Thank you for using Interactive Argument Generator!
Example Dialogue #2: Sample Test

Welcome to SWI-Prolog (Multi-threaded, Version 5.6.25)
Copyright (c) 1990-2006 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic), or ?- apropos(Word).

1 ?- start_query.
% API.pl compiled 0.03 sec, 66,068 bytes
% ArgumentGenerator.pl compiled 0.00 sec, 3,760 bytes

Please select a knowledge base:
  1 for Cystic Fibrosis
  2 for Achondroplasia
  3 for Familial Hypercholesterolemia
  4 for Phenylketonuria (PKU)
  5 for Sample Test
|: 5.
% Test_KB.pl compiled 0.00 sec, 8,020 bytes

Please pick the state of the KB you selected:
  b for before the test
|: b.

Please pick a node for the KB state you selected:
  a for proband's history: A
  b for proband's history: B
  c for proband's complication: C
  d for proband's genotype: D
  e for proband's genotype: E
  f for proband's symptom: F
  g for proband's symptom: G
  h for proband's symptom: H
  i for proband's symptom: I
|: d.

Please pick a claim for the node you selected:
  1 for [d, [], true]
  2 for [d, [], false]
|: 2.

Argument #1
CLAIM:
  log_not([d, [], true])
DATA:
  1) log_not([f, [], true])
WARRANT:
  1) has_pos_infl([d, [], true], [f, [], true])
CRITICAL QUESTION:
  1) [enables([c, [], true], [d, [], true]), [f, [], true]), log_not([c, [], true])]

Please pick your next move:
Enter d to select a data
Enter w to select a warrant
Enter q to select a critical question
Enter c to select a counterclaim
Enter n  to select a new node within this KB
Enter s  to select a new state for this KB
Enter k  to select a new KB
Enter x  to exit Interactive Argument Generator
|: d.

Please enter N for the N-th argument: 1.

Please enter M for the M-th data: 1.

Claim log_not([f, [], true]) is Observed.

Please pick your next move:
Enter c  to select a counterclaim
Enter n  to select a new node within this KB
Enter s  to select a new state for this KB
Enter k  to select a new KB
Enter x  to exit Interactive Argument Generator
|: n.

Please pick a node for the KB state you selected:
  a  for proband's history: A
  b  for proband's history: B
  c  for proband's complication: C
  d  for proband's genotype: D
  e  for proband's genotype: E
  f  for proband's symptom: F
  g  for proband's symptom: G
  h  for proband's symptom: H
  i  for proband's symptom: I
|: c.

Please pick a claim for the node you selected:
  1 for [c, [], true]
  2 for [c, [], false]
|: 2.

Claim [c, [], false] is Observed.

Please pick your next move:
Enter c  to select a counterclaim
Enter n  to select a new node within this KB
Enter s  to select a new state for this KB
Enter k  to select a new KB
Enter x  to exit Interactive Argument Generator
|: n.

Please pick a node for the KB state you selected:
  a  for proband's history: A
  b  for proband's history: B
  c  for proband's complication: C
  d  for proband's genotype: D
  e  for proband's genotype: E
  f  for proband's symptom: F
  g  for proband's symptom: G
  h  for proband's symptom: H
  i  for proband's symptom: I
|: d.
Please pick a claim for the node you selected:
1 for [d, [], true]
2 for [d, [], false]
|: 1.

Argument #1
CLAIM:
[d, [], true]
DATA:
1) [g, [], true]
WARRANT:
1) has_pos_infl([d, [], true], [g, [], true])
CRITICAL QUESTION:
1) []

Argument #2
CLAIM:
[d, [], true]
DATA:
1) [h, [], true]
WARRANT:
1) has_pos_infl([d, [], true], [h, [], true])
CRITICAL QUESTION:
1) [xor_resp([e, [true], [d, [true], [h, [true]]], [e, [true]], [e, [true]]])

Please pick your next move:
Enter d to select a data
Enter w to select a warrant
Enter q to select a critical question
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: d.

Please enter N for the N-th argument: 2.

Please enter M for the M-th data: 1.

Claim [h, [true]] is Observed.

Please pick your next move:
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: n.

Please pick a node for the KB state you selected:
 a for proband's history: A
 b for proband's history: B
 c for proband's complication: C
 d for proband's genotype: D
 e for proband's genotype: E
 f for proband's symptom: F
 g for proband's symptom: G
 h for proband's symptom: H
i  for proband’s symptom: I
|: e.

Please pick a claim for the node you selected:
1 for [e, [], true]
2 for [e, [], false]
|: 1.

Argument #1
CLAIM:
[e, [], true]
DATA:
1) [a, [], true]
WARRANT:
1) has_pos_infl([a, [], true], [e, [], true])
CRITICAL QUESTION:
1) [inhibits([b, [], true], [a, [], true], [e, [], true]), [b, [], true]]

Argument #2
CLAIM:
[e, [], true]
DATA:
1) [h, [], true]
WARRANT:
1) has_pos_infl([e, [], true], [h, [], true])
CRITICAL QUESTION:
1) [xor_resp([d, [], true], [e, [], true], [h, [], true]), [d, [], true]]

Argument #3
CLAIM:
[e, [], true]
DATA:
1) [i, [], true]
WARRANT:
1) has_pos_infl([e, [], true], [i, [], true])
CRITICAL QUESTION:
1) []

Please pick your next move:
Enter d to select a data
Enter w to select a warrant
Enter q to select a critical question
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: q.

Please enter N for the N-th argument: 1.

Please enter M for the M-th critical question: 1.

No arguments found for claim: [b, [], true]

Please pick your next move:
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: c.

Please pick a counterclaim:
1 for [b, [], false]
2 for log_not([b, [], true])
|: 1.

Claim [b, [], false] is Observed.

Please pick your next move:
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: n.

Please pick a node for the KB state you selected:
 a for proband's history: A
 b for proband's history: B
 c for proband's complication: C
 d for proband's genotype: D
 e for proband's genotype: E
 f for proband's symptom: F
 g for proband's symptom: G
 h for proband's symptom: H
 i for proband's symptom: I
|: g.

Please pick a claim for the node you selected:
1 for [g, [], true]
2 for [g, [], false]
|: 1.

Claim [g, [], true] is Observed.

Please pick your next move:
Enter c to select a counterclaim
Enter n to select a new node within this KB
Enter s to select a new state for this KB
Enter k to select a new KB
Enter x to exit Interactive Argument Generator
|: x.

Thank you for using Interactive Argument Generator!
CHAPTER 7

CONCLUSION

7.1. Summary

This thesis presents the discourse generation component of GenIE, a prototype intelligent system for drafting genetic counseling patient letters that contain arguments. The discourse generation process used in GenIE involves three separate modules: the domain model, the discourse grammar, and the argument generator. The output from discourse generation is in the format of a dplan, which will later be used as input to a linguistic realizer.

One of the main contributions of this thesis is the reimplemention of the Discourse Grammar, Argument Generator, and Knowledge Base API described in [Green, 2006]. This was necessary to make the system more robust and more efficient, and to accommodate the new representation of arguments in RST format described in [Green, 2007]. The second contribution is the implementation of four new KBs to enable us to more fully exercise GenIE’s argument generation. Lastly, an important contribution is the implementation of an interactive driver and modifications to the Argument Generator to enable arguments to be explored interactively.
7.2. Future Work

At the moment, KBs are created manually by system developers directly in Prolog. We would like to make it possible for genetic counselors to create a new KB by themselves, through a user-friendly interface that guides them step-by-step through the process of KB creation. The interface should allow its user to specify node information including type, domain, family role, and description. Once node creation is complete, the user should be allowed to assign the relationship between the nodes as well as provide node values. The system should also let the user input certain statistics related to the disease the KB represents.

At the time this thesis is being written, the work on the linguistic realizer is scheduled to be completed in Spring 2008. Once it is completed, outputs from the argument generator will serve as inputs to the linguistic realizer and the patient letters will be generated. A formal evaluation of GenIE is planned as the next step after that.
REFERENCES


APPENDIX A

Discourse Generation Results

This appendix presents the outputs from discourse generation for each of the four KBs. Note that the fourth section of the *dplan, doc_sibling_risk* – the one describing recurrence risk on future siblings, is available only in the Cystic Fibrosis KB, since no future risk information is available in any other KB. The first section, *doc_pretest* for preliminary diagnosis is missing in the Phenylketonuria KB as there is no information available prior to testing. Similarly, the third section, *doc_source* for origin of genetic condition, is missing in the Familial Cholesterolemia KB since no parent information is available for the case.

In order to make the output easier to understand, the *dplan* for the Cystic Fibrosis KB is presented in a way that its event descriptions are incorporated into the structure. For the remaining KBs, the *dplans* are presented in the original way as they are generated. First, the *dplans* are shown in graphical format. Then, the event listings corresponding to each of the *dplans* are presented.
Figure A-1: Section 1 of the Cystic Fibrosis dplan.
Proband having 2 mutated CF genes is responsible for symptoms.

Figure A-2: Section 2 of the Cystic Fibrosis dplan.
Figure A-3: Section 3 of the Cystic Fibrosis dplan.
Figure A-4: Section 4 of the Cystic Fibrosis dplan.
Figure A-5: Section 1 of the Achondroplasia dplan.
Figure A-6: Section 2 of the Achondroplasia dplan.
Figure A-7: Section 3 of the Achondroplasia dplan.
Figure A-8: Section 1 of the Familial Hypercholesterolemia dplan.
Figure A-9: Section 2 of the Familial Hypercholesterolemia dplan.

Figure A-10: Section 2 of the Phenylketonuria dplan.
Figure A-11: Section 3 of the Phenylketonuria dplan.
<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Time</th>
<th>Duration</th>
<th>Action</th>
<th>Destination</th>
<th>Beneficiary</th>
<th>Agent</th>
<th>Object</th>
<th>Effect</th>
<th>Experiencer</th>
<th>Instrument</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>pretest</td>
<td>pretest</td>
<td>referral</td>
<td>clinic</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e2</td>
<td>pretest</td>
<td>pretest</td>
<td>diagnosis</td>
<td>proband</td>
<td>clinic</td>
<td>e2_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>proband</td>
<td>![sp1, 'respiratory infections', <a href="false"></a>, true], ![sp2, 'growth failure', false)]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e3_1</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>proband</td>
<td>![gp, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e3_2</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>responsible</td>
<td>e3_1</td>
<td>e2_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e4</td>
<td>pretest</td>
<td>pretest</td>
<td>presume</td>
<td>clinic</td>
<td>e3_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e5_1</td>
<td>posttest</td>
<td>posttest</td>
<td>test</td>
<td>proband</td>
<td>clinic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>![sweat]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e5_2</td>
<td>posttest</td>
<td>infinitive</td>
<td>knowif</td>
<td>clinic</td>
<td>e3_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e7</td>
<td>posttest</td>
<td>posttest</td>
<td>produce</td>
<td>e5_1</td>
<td>![tresp, 'NaCl level', false, abnormal]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e10</td>
<td>posttest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>proband</td>
<td>![gp, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e11</td>
<td>posttest</td>
<td>pretest</td>
<td>continuing</td>
<td>responsible</td>
<td>e10</td>
<td>e2_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e12_1</td>
<td>posttest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>mother</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>![gm, 'CFTR', =, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e12_2</td>
<td>posttest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>Father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>![gf, 'CFTR', =, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event ID</td>
<td>Graph</td>
<td>Modality</td>
<td>Time</td>
<td>Duration</td>
<td>Polarity</td>
<td>Status</td>
<td>Action</td>
<td>Experimenter</td>
<td>Result</td>
<td>Attribute</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td>--------------</td>
<td>--------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e13</td>
<td>future</td>
<td>posttest</td>
<td>future</td>
<td>continuing</td>
<td>risk_assessment</td>
<td>experience</td>
<td>future_sib</td>
<td></td>
<td>0.25</td>
<td>[gs, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e14</td>
<td>future</td>
<td>future</td>
<td>future</td>
<td>continuing</td>
<td>experience</td>
<td>future_sib</td>
<td></td>
<td></td>
<td></td>
<td>[ss, 'CF symptoms', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e15</td>
<td>future</td>
<td>future</td>
<td>future</td>
<td>experience</td>
<td>future_sib</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[ss, 'CF symptoms', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e16</td>
<td>future</td>
<td>posttest</td>
<td>FALSE</td>
<td>experience</td>
<td>future_sib</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gs, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e17</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[sp2, 'growth failure', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e18</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[sp1, 'respiratory infections', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge1</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[ep1, 'bacteria in lung secretion', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge11</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[tresp, 'NaCl level', [], abnormal]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge12</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>assumed</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td>[tresp, sweat, [], done]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge18</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[sp2, 'growth failure', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge19</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[sp1, 'respiratory infections', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge20</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[sm, 'CF symptoms', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge21</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>assumed</td>
<td>experience</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td>[ep1, 'bacteria in lung secretion', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge22</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>FALSE</td>
<td>experience</td>
<td>mother</td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge23</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>FALSE</td>
<td>experience</td>
<td>mother</td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', &gt;=, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge24</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>assumed</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td>[gf, 'CFTR', &gt;=, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge25</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge26</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', &gt;=, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge27</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge28</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', &gt;=, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge29</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge30</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>mother</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge31</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge32</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge33</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge34</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge35</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge36</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge37</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge38</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge39</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge40</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge41</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge42</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge43</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[gm, 'CFTR', =, 2]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

98
### Table A-2: Generic events in the Cystic Fibrosis KB.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge2</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge3</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge4</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge5</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge6</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge7</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge8</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge9</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge10</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge13</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge14</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge15</td>
<td>generic</td>
<td>continuing</td>
<td>assumed</td>
</tr>
<tr>
<td>ge16</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge17</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
</tr>
<tr>
<td>ge21</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge22</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge23</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
</tr>
<tr>
<td>ge28</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge29</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge32</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge33</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
</tbody>
</table>

**Generic Attributes:***
- [genotype, 'CFTR', =, 2]
- [biochemistry, 'CFTR protein', [0], abnormal]
- [physiology, 'pancreas enzyme level', [0], abnormal]
- [physiology, malabsorption, [0], true]
- [symptom, 'growth failure', [0], true]
- [physiology, 'viscous lung secretion', [0], true]
- [event, 'bacteria in lung secretion', [0], true]
- [symptom, 'respiratory infections', [0], true]
- [test, sweat, [0], done]
- [test_result, 'NaCl level', [0], abnormal]
- [symptom, 'CF symptoms', [0], true]
- [generic_mother, genotype, 'CFTR', >=, 1]
- [generic_father, genotype, 'CFTR', >=, 1]
Table A-2 (continued).

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Time</th>
<th>Duration</th>
<th>Status</th>
<th>Probability</th>
<th>Action</th>
<th>Agent</th>
<th>Enabler</th>
<th>Object</th>
<th>Experiencer</th>
<th>Generic Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge34</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>joint_responsible</td>
<td>ge32, ge33</td>
<td>ge2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge37</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td></td>
<td>generic</td>
<td>[history, 'N European ancestry', [], true]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge38</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td></td>
<td>generic</td>
<td>[genotype, 'CFTR', &gt;=, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge39</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>low</td>
<td>responsible</td>
<td>ge37</td>
<td>ge38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A-3: Non-generic events in the Achondroplasia KB.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Time</th>
<th>Duration</th>
<th>Action</th>
<th>Destination</th>
<th>Beneficiary</th>
<th>Agent</th>
<th>Object</th>
<th>Effect</th>
<th>Experiencer</th>
<th>instrument</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>pretest</td>
<td>pretest</td>
<td>referral</td>
<td>clinic</td>
<td>proband</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e2</td>
<td>pretest</td>
<td>pretest</td>
<td>diagnosis</td>
<td>proband</td>
<td>clinic</td>
<td>e2_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e2_1</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e3_1</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>experience</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e3_2</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
<td>responsible</td>
<td>e3_1</td>
<td>e2_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e4</td>
<td>pretest</td>
<td>pretest</td>
<td>presume</td>
<td>clinic</td>
<td>e3_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e5_1</td>
<td>posttest</td>
<td>posttest</td>
<td>test</td>
<td>proband</td>
<td>clinic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e5_2</td>
<td>posttest</td>
<td>infinitive</td>
<td>knowif</td>
<td>clinic</td>
<td>e3_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event ID</td>
<td>Graph</td>
<td>Modality</td>
<td>Time</td>
<td>Duration</td>
<td>Polarity</td>
<td>Action</td>
<td>Agent</td>
<td>Object</td>
<td>Semantics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>----------</td>
<td>--------</td>
<td>-------</td>
<td>--------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e7</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td></td>
<td>produce</td>
<td>e5_1</td>
<td></td>
<td>![tresp1, 'fetal rhizomelia', [], true], ![tresp2, 'fetal macrocephaly', [], true], ![tresp3, 'G1138A mutation in FGFR3 gene', [], true], ![tresp4, 'G1138C mutation in FGFR3 gene', [], false], ![tresp5, 'radiographic features of Achondroplasia', [], true]]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e10</td>
<td>posttest</td>
<td>pretest</td>
<td>pretest</td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>e10</td>
<td></td>
<td>proband</td>
<td>![gp, 'FGFR3', =, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e11</td>
<td>posttest</td>
<td>pretest</td>
<td>pretest</td>
<td></td>
<td>continuing</td>
<td>responsible</td>
<td>e10</td>
<td></td>
<td>proband</td>
<td>![gp2, 'FGFR3 germline', =, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e12_1</td>
<td>posttest</td>
<td>pretest</td>
<td>pretest</td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>e10</td>
<td></td>
<td>father</td>
<td>![gp, 'FGFR3', =, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge8</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge8</td>
<td></td>
<td>proband</td>
<td>![sp1, 'physical features of Achondroplasia at birth', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge9</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge9</td>
<td></td>
<td>proband</td>
<td>![tresp3, 'G1138A mutation in FGFR3 gene', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge10</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge10</td>
<td></td>
<td>proband</td>
<td>![tstp1, 'DNA test', [], done]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge14</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge14</td>
<td></td>
<td>proband</td>
<td>![tresp5, 'radiographic features of Achondroplasia', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge15</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge15</td>
<td></td>
<td>proband</td>
<td>![tstp2, 'radiograph at birth', [], done]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge19</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge19</td>
<td></td>
<td>proband</td>
<td>![tresp1, 'fetal rhizomelia', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge20</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge20</td>
<td></td>
<td>proband</td>
<td>![tstp3, 'fetal ultrasound', [], done]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge24</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge24</td>
<td></td>
<td>proband</td>
<td>![tresp2, 'fetal macrocephaly', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge27</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge27</td>
<td></td>
<td>proband</td>
<td>![gm, 'FGFR3', =, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge28</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>experience</td>
<td>ge28</td>
<td></td>
<td>father</td>
<td>![gf1, 'FGFR3', =, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge33</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>false</td>
<td>ge33</td>
<td></td>
<td>proband</td>
<td>![sm, 'Achondroplasia symptoms', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge34</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>false</td>
<td>ge34</td>
<td></td>
<td>proband</td>
<td>![gm, 'FGFR3', =, 1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge37</td>
<td>posttest</td>
<td>posttest</td>
<td></td>
<td></td>
<td>continuing</td>
<td>false</td>
<td>ge37</td>
<td></td>
<td>father</td>
<td>![sf, 'Achondroplasia symptoms', [], true]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-3 (continued).

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge38</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge39</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge40</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
</tbody>
</table>

### Table A-4: Generic events in the Achondroplasia KB.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge1</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge2</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge3</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge4</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge5</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge6</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge7</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge11</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge12</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge13</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge16</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge17</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge18</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge21</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge22</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge23</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
</tbody>
</table>
Table A-4 (continued).

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>Duration</td>
</tr>
<tr>
<td>ge25</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge26</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge29</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge30</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge31</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge32</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge33</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge35</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge36</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge41</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge42</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge43</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge44</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
<tr>
<td>ge45</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
</tr>
</tbody>
</table>

Table A-5: Non-generic events in the Familial Hypercholesterolemia KB.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time</td>
<td>Duration</td>
</tr>
<tr>
<td>e1</td>
<td>pretest</td>
<td>pretest</td>
<td>referral</td>
</tr>
<tr>
<td>e2</td>
<td>pretest</td>
<td>pretest</td>
<td>diagnosis</td>
</tr>
<tr>
<td>e2_1</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>e3_1</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>e3_2</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>Event ID</td>
<td>Graph</td>
<td>Modality</td>
<td>Time</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>e4</td>
<td>pretest</td>
<td>pretest</td>
<td>presume</td>
</tr>
<tr>
<td>e5_1</td>
<td>posttest</td>
<td>posttest</td>
<td>test</td>
</tr>
<tr>
<td>e5_2</td>
<td>posttest</td>
<td>infinitive</td>
<td>knowif</td>
</tr>
<tr>
<td>e7</td>
<td>posttest</td>
<td>posttest</td>
<td>produce</td>
</tr>
<tr>
<td>e10</td>
<td>posttest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>e11</td>
<td>posttest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge1</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge8</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge11</td>
<td>pretest</td>
<td>pretest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge14</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge15</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge19</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge20</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge21</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge22</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge23</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge24</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge28</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge29</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge32</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
<tr>
<td>ge33</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
</tr>
</tbody>
</table>
### Table A-6: Generic events in the Familial Hypercholesterolemia KB.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Time</th>
<th>Duration</th>
<th>Action</th>
<th>Agent</th>
<th>Enabler</th>
<th>Object</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge2</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[genotype, 'LDLR', =, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge3</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[biochemistry, 'elevated LDL',=[], abnormal]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge4</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
<td>ge2</td>
<td>ge3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge5</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[behavior, 'mild obesity',[], true]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge6</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[symptom, 'myocardial infarction',[], true]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge7</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
<td>ge3</td>
<td>ge5</td>
<td>ge6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge9</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[behavior, smoker,[], true]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge10</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
<td>ge3</td>
<td>ge9</td>
<td>ge6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge11</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[behavior, 'physical inactivity',[], true]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge13</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
<td>ge3</td>
<td>ge12</td>
<td>ge6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge16</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[test, 'gene test',[], done]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge17</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[test_result, 'mutation in 1 allele',[], true]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge18</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
<td>ge2</td>
<td>ge16</td>
<td>ge17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge25</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[test, 'LDL test',[], done]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge26</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[test_result, 'LDL result',[], true]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge27</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
<td>ge3</td>
<td>ge25</td>
<td>ge26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge30</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
<td>generic</td>
<td>[genotype, 'LDLR', =, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge31</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
<td>ge2</td>
<td>ge30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table A-7: Non-generic events in the Phenylketonuria KB.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Time</th>
<th>Duration</th>
<th>Polarity</th>
<th>Action</th>
<th>Agent</th>
<th>Effect</th>
<th>Experiencer</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>e10</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>proband</td>
<td>[gp, 'phenylalanine gene', =, 2]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e12_1</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>mother</td>
<td>[gm, 'phenylalanine gene', =, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e12_2</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
<td>father</td>
<td>[gf, 'phenylalanine gene', =, 1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-7 (continued).

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Semantics</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge1</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge2</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge9</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>FALSE</td>
</tr>
<tr>
<td>ge10</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>FALSE</td>
</tr>
<tr>
<td>ge13</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge14</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge18</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge19</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>FALSE</td>
</tr>
<tr>
<td>ge20</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>FALSE</td>
</tr>
<tr>
<td>ge21</td>
<td>posttest</td>
<td>posttest</td>
<td>continuing</td>
<td>experience</td>
</tr>
</tbody>
</table>

### Table A-8: Generic events in the Phenylketonuria KB.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Graph</th>
<th>Modality</th>
<th>Semantics</th>
<th>Generic Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge3</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge4</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge5</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge6</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge7</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge8</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>enable</td>
</tr>
<tr>
<td>ge11</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge12</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>responsible</td>
</tr>
<tr>
<td>ge15</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>Event ID</td>
<td>Graph</td>
<td>Modality</td>
<td>Time</td>
<td>Duration</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>ge16</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>experience</td>
</tr>
<tr>
<td>ge17</td>
<td>generic</td>
<td>generic</td>
<td>continuing</td>
<td>joint_responsible</td>
</tr>
</tbody>
</table>