

A Mathematical Basis for Medication Prescriptions and Adherence

by

Simon Diemert

B.SEng., University of Victoria, 2015

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

© Simon Diemert, 2017
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.

A Mathematical Basis for Medication Prescriptions and Adherence

by

Simon Diemert

B.SEng., University of Victoria, 2015

Supervisory Committee

Dr. Jens Weber, Co-supervisor
(Department of Computer Science)

Dr. Morgan Price, Co-supervisor
(Department of Computer Science)

Supervisory Committee

Dr. Jens Weber, Co-supervisor
(Department of Computer Science)

Dr. Morgan Price, Co-supervisor
(Department of Computer Science)

ABSTRACT

Medication prescriptions constitute an important type of clinical intervention. Medication adherence is the degree to which a patient consumes their medication as agreed upon with a prescriber. Despite many years of research, medication non-adherence continues to be a problem of note, partially due to its multi-faceted in nature. Numerous interventions have attempted to improve adherence but none have emerged as definitive. A significant sub-problem is the lack of consensus regarding definitions and measurement of adherence. Several recent reviews indicate that discrepancies in definitions, measurement techniques, and study methodologies make it impossible to draw strong conclusions via meta-analyses of the literature.

Technological interventions aimed at improving adherence have been the subject of ongoing research. Due to the increasing prevalence of the Internet of Things, technology can be used to provide a continuous stream of data regarding a patient's behaviour. To date, several researchers have proposed interventions that leverage data from the Internet of Things, however none have established an acceptable means of analyzing and acting upon this wealth of data.

This thesis introduces a computational definition for adherence that can be used to support continued development of technological adherence interventions. A central part of the proposed definition is a formal language for specifying prescriptions that uses fuzzy set theory to accommodate imprecise concepts commonly found in natural language medication prescriptions. A prescription specified in this language can be transformed into an evaluation function which can be used to score the adherence

of a given medication taking behaviour. Additionally, the evaluator function is applied to the problem of scheduling medication administrations. A compiler for the proposed language was implemented and had its breadth of expression and clinical accuracy evaluated. The results indicate that the proposed computational definition of adherence is acceptable as a proof of concept and merits further works.

Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	v
List of Tables	vii
List of Figures	viii
Acknowledgements	xi
1 Introduction	1
1.1 Non-Adherence: A Complex Problem	1
1.2 Proposed Solutions to Non-Adherence	8
1.3 Objectives and Outline	11
2 Foundations	14
2.1 Formal Languages	14
2.2 Fuzzy Set Theory	24
3 Concept Formulation	29
3.1 Working Example	29
3.2 Prescriptions and Compliance	30
3.3 Compliance in Context	33
3.4 Towards a Formal Definition of Adherence	37
3.5 Scheduling Medication Administration	46
3.6 Chapter Summary	48
4 Formalizing Adherence	50

4.1	Traces	50
4.2	Prescriptions	52
4.3	Semantics	57
4.4	Measurement	66
4.5	Scheduling	70
4.6	Implementation	71
4.7	Example	75
4.8	Chapter Summary	81
5	Evaluation	83
5.1	Requirements Verification	83
5.2	Language Breadth of Expression	85
5.3	Clinical Evaluation	87
5.4	Chapter Summary	96
6	Discussion	97
6.1	Comments on Evaluation	99
6.2	Limitations	104
6.3	Data for Prescription Interpretation	107
6.4	Comparison with Prior Work	108
7	Conclusions and Future Work	109
7.1	Future Work	110
A	Extended Language Definition	114
B	Language Breadth Evaluation	118
C	Verification Data	122
	Bibliography	128

List of Tables

Table 4.1	Subset of grammar for prescription specification language. . . .	54
Table 5.1	Prescription Interpretation Requirements	84
Table 5.2	Observer Device Requirements	84
Table 5.3	Evaluator Device Requirements	85
Table 5.4	Authored Prescriptions (APs) provided by domain expert	86
Table 5.5	Scenarios posed to participants of survey	88
Table 5.6	Participant Professional Experience	92
Table 5.7	Participant Practice Types	92
Table 5.8	Survey Timeline Responses	92
Table 5.9	RMSE of Use Cases	94
Table 5.10	Expert Review of Generated Compliance Functions	96
Table A.1	Prescription Specification Language Grammar Extension	114

List of Figures

Figure 2.1	Sample graph grammar production rule	16
Figure 2.2	Sample production rules.	19
Figure 2.3	Derivation of the bit string graph “011”.	20
Figure 2.4	Parse tree for binary number 10	23
Figure 2.5	Visualizing crisp and fuzzy sets as Venn diagrams, adapted from [59].	25
Figure 2.6	Membership functions for wavelengths of light which are considered “blue”.	26
Figure 2.7	Visual representation of an alpha-cut	27
Figure 2.8	Visual representation of fuzzy set operations.	28
Figure 3.1	Two traces of Ms. Smith’s behaviour w.r.t the prescription: <i>600 mg of ibuprofen three times per day as needed with food</i> ; pill icons represent administrations, fork/knife icons represent the start of meals.	31
Figure 3.2	Depiction of AIM [62]	34
Figure 3.3	Three dimensional space exhibiting adherence evolving over time.	35
Figure 3.4	Three dimensional plot showing change in Ms. Smith’s behaviour in time.	37
Figure 3.5	Diagram describing the high level architecture of the proposed approach to formalizing compliance.	41
Figure 3.6	Control loop architecture for scheduling medication administration.	47
Figure 4.1	Sample trace showing observable factors of time and dose.	52
Figure 4.2	A non-compliant trace.	67
Figure 4.3	Non-compliance curves for trace in Figure 4.2	68
Figure 4.4	Cumulative non-compliance curve, $\alpha = 0.95$, $\beta = 0.05$	69
Figure 4.5	Scheduling output for Ms. Smith’s next dose of ibuprofen.	71
Figure 4.6	Package and class/module diagram of implementation	72

Figure 4.7 Sequence diagram for handling request to evaluate a trace's compliance	73
Figure 4.8 Invalid frame nesting	74
Figure 4.9 Graphical representation of Ms. Smith's morphine prescription.	75
Figure 4.10 Fuzzy functions for Ms. Smith's morphine prescription.	77
Figure 4.11 Ms. Smith's behaviour	79
Figure 4.12 Non-compliance curves for Ms. Smith's behaviour shown in Figure 4.11, $\alpha = 0.95$, $\beta = 0.05$	80
Figure 4.13 Scheduler output for Ms. Smith's next dose of morphine.	81
Figure 5.1 Sample IP for AP number 1 from Table 5.4	86
Figure 5.2 Timeline with sliders for collecting quantitative data.	89
Figure 5.3 Experimental compliance function for use case 9 (<i>Penicillin V 500 mg three times a day</i>).	92
Figure 5.4 Interpreted prescription for use case 9 (<i>Penicillin V 500 mg three times a day</i>).	93
Figure 5.5 Generated and Experimental Compliance Functions for Use Case 9 (RMSE=0.070)	94
Figure B.1 Breadth Evaluation Prescription 1: <i>keflex 500 mg four times daily for 10 days</i>	118
Figure B.2 Breadth Evaluation Prescription 2: <i>azithromycin 500 mg once daily for 1 day then 250 mg once daily for 4 days</i>	119
Figure B.3 Breadth Evaluation Prescription 3: <i>ibuprofen 600 mg three times daily as needed</i>	119
Figure B.4 Breadth Evaluation Prescription 4: <i>seroquel 25-50 mg three times daily as needed</i>	119
Figure B.5 Breadth Evaluation Prescription 5: <i>morphine 10 mg three times daily as needed up to 10 per week</i>	120
Figure B.6 Breadth Evaluation Prescription 6: <i>prednisone 50 mg decrease by 10 mg per week until done</i>	120
Figure B.7 Breadth Evaluation Prescription 7: <i>coumadin 7 mg once daily on M/W/F 8 mg once daily otherwise</i>	121
Figure B.8 Breadth Evaluation Prescription 8: <i>hydrochlorothiazide 25 mg once daily in the AM for 30 days</i>	121

Figure C.1 Expert Data and Generated Compliance Function for Use Case 1 <i>Nitrofurantoin 100 mg twice daily</i> with previous dose at: 08:00.	123
Figure C.2 Expert Data and Generated Compliance Function for Use Case 2 <i>Ramipril 2.5 mg once daily</i> with previous dose at: 10:00.	123
Figure C.3 Expert Data and Generated Compliance Function for Use Case 3 <i>Hydromorphone 4 mg every 4 hours as needed</i> with previous doses at: 06:00, 10:00, and 14:00.	124
Figure C.4 Expert Data and Generated Compliance Function for Use Case 4 <i>Marvelon 1 tablet once daily</i> with previous dose at: 09:00.	124
Figure C.5 Expert Data and Generated Compliance Function for Use Case 5 <i>Glargine 10 units at bedtime</i> with previous dose at: 22:00.	125
Figure C.6 Expert Data and Generated Compliance Function for Use Case 6 <i>Moxifloxacin 400 mg every twenty four hours</i> with previous dose at: 12:00.	125
Figure C.7 Expert Data and Generated Compliance Function for Use Case 7 <i>Cephalexin 500 mg four times a day</i> with previous doses at: 08:00 and 12:00.	126
Figure C.8 Expert Data and Generated Compliance Function for Use Case 8 <i>Warfarin 7 mg once daily</i> with previous dose at: 09:00.	126
Figure C.9 Expert Data and Generated Compliance Function for Use Case 9 <i>Penicillin V 500 mg three times a day</i> with previous dose at: 09:00 and 17:00.	127
Figure C.10 Expert Data and Generated Compliance Function for Use Case 10 <i>Hydromorphone Contin 9 mg twice daily and Hydromorphone 4 mg every 4 hours as needed</i> with previous doses at: 08:00 (9 mg) and 14:00 (4 mg).	127

ACKNOWLEDGEMENTS

I would like to thank:

Drs. Jens Weber and Morgan Price for providing an opportunity to grow, and for on-going guidance and support through most of my time at the University of Victoria. Both of you have been instrumental in shaping my thoughts about computer science, software engineering, and health care. I look forward to working with you in the future.

Colleagues who participated in this work either directly or indirectly. In particular: Fieran Mason-Blakely for many thoughtful conversations; Jordan Banman for joining us as the third investigator in the online survey of practitioners; and many other members of the LEAD Lab and Computer Science Department.

NSERC and the University of Victoria for providing financial support throughout my Master of Science degree through fellowships and teaching opportunities.

Family and Friends for continued and unconditional support throughout both of my degree programs. In particular, India Wiebe for putting up with my seemingly confused and disjoint ideas, providing insightful comments about this work, and for being there day after day.

Our basic tool for writing specifications is mathematics. Mathematics is nature's way of letting you know how sloppy your writing is. It's hard to be precise in an imprecise language like English or Chinese. In engineering, imprecision can lead to errors. To avoid errors, science and engineering have adopted mathematics as their language

- Leslie Lamport

Chapter 1

Introduction

Medication prescriptions are a common and important form of medical intervention used in modern clinical settings. In 2014 29.4 billion dollars were spent on prescription drugs in Canada, roughly 12.7% of total health care spending and approximately 1.6% of the total GDP [1]; this is a non-negligible amount, it is desirable to ensure that the intended effect of these medications are achieved. However, for prescription medications to be effective, they must be correctly managed and administered - a task that is often left to the patients themselves. Indeed, amongst populations of developed countries less than 50% of medications are administered as intended [2], and failure to administer medications as prescribed has been associated with a negative health outcomes, including: exacerbation of existing conditions [2-4], adverse drug reactions [5, 6], and increased hospital admissions [4, 7, 8]. In their report from 2003, the World Health Organization (WHO) indicated that improving adherence to prescribed medication regimes may be a more effective use of resources than focusing on the creation of new therapies [2]. Adherence is defined by the WHO as:

“the extent to which a person’s behaviour – taking medication, following a diet, and/or executing lifestyle changes, corresponds with agreed recommendations from a health care provider” [2].

1.1 Non-Adherence: A Complex Problem

Non-adherence is a seemingly simple problem - patients need only take their medications as agreed upon, and in an ideal situation the problem would be solved. However reality is rarely simple where humans are concerned. Non-adherence is a

multifaceted problem, which is influenced by many factors including but not limited to:

- forgetfulness and prescriptions not fitting into lifestyles [9–11],
- decline of cognitive function in older adults [9, 12],
- regime complexity, polypharmacy, and administration frequency [9, 11–13],
- cost of prescription drugs paid by patients [6, 14–16],
- adverse drug events and side effects due to medication consumption [5, 6],
- health literacy and education [6, 9, 11],
- age, with the largest effects reported in elderly populations [11, 12],
- social supports, including family and peer support [11],
- race and ethnicity [11, 17],
- and stigma around disease [16].

Evidently, the problem is complex and involves patients, providers, health care organizations, and governments [18]. The problem is worsened by a lack consensus regarding how to study and measure adherence. A Cochrane Review from 2014 of 182 studies concluded that before any meaningful results can be synthesized from the current body of research improvements must be made to definitions, study protocols, and adherence measurement techniques [19].

1.1.1 Terminology and Frameworks for Adherence

Though a general definition of adherence has been accepted by the research community, researchers continue to identify new aspects of adherence and develop new adherence frameworks. This section discusses select frameworks for adherence and associated terminology from the literature.

Cramer *et al.* conducted a review of the literature to establish guidelines for the use of the terms *compliance* and *persistence* [20]. The term *compliance* was defined as the “act of conforming to the recommendations made by the provider with respect to timing, dosage, and frequency of medication taking” [20]. The term *persistence* was

defined as “conforming to a recommendation of continuing treatment for the prescribed length of time” [20]. Cramer *et al.* note that compliance is often used as a synonym for adherence; however, based on the WHO’s definition (above), adherence subsumes compliance and also includes agreement. Further, their definition of persistence implies compliance continuing over time, also an important part of adherence. Cramer *et al.*’s conclusion, which is based on a literature review, points to terminology confusion in the adherence literature. Indeed, confusion in the literature regarding terminology is an ongoing problem [16, 19, 21, 22].

For example, Vrijens *et al.* propose a taxonomy for the study of adherence consisting of: 1) *adherence* - “the process by which patients take medications as prescribed”; 2) *adherence management* - “monitoring and supporting patient’s adherence”; and 3) *adherence related sciences* - “disciplines that seek understanding of the causes or consequences” of non-adherence” [23]. While their effort to impose structure in a field burdened by terminology confusion is not without merit - distinguishing between the process of patient adherence and the study of adherence is useful - their definitions are restricted to compliance and ignore patient agreement entirely.

As another example, Raebel *et al.* proposed a framework for defining adherence based on pharmacy dispensing records and insurance claims data [21]. Their framework distinguishes between primary and secondary adherence; the former being “a discrete event that assesses whether or not the patient received the first prescription” the later being “an ongoing process that measures whether or not the patient received dispensings or refills as prescribed” [21]. This distinction is important, failing to initially fill prescriptions is a common mode of non-adherence [24, 25], however Raebel *et al.*’s definitions do not consider patient agreement.

More recently, another term has emerged to describe medication taking behaviour. *Concordance* which extends adherence to add the notion of “consensual agreement about treatment taking established between patient and practitioner” [26]. While adherence implies agreement between patient and provider, it does not imply how said agreement was reached; the concern being that the agreement is one-sided. Concordance indicates that the agreement was arrived at by considering more than the immediate medical concern(s) through continuing engagement with the patient [18, 27]. For example, a patient may enjoy playing a sport but be prevented from doing so by a medication; concordance would imply that the health care provider and patient discussed how the prescription could be adjusted such that the patient could continue to play a sport [27]. Due to scope, this thesis focuses primarily on adherence,

though some results may be generalizable to concordance.

Drawing on the mental health literature, Gearing *et al.* proposed a dynamic six phase model which acknowledges that adherence patterns change as a patient progresses through treatment, acquires news information, and has new experiences [28]. The underlying assumption is that medication taking behaviours are based on a series of rational decisions made collaboratively by the patient and providers. Gearing *et al.*'s model has the following phases: 1) Treatment Initiation, 2) Trial Treatment, 3) Partial Acceptance, 4) Intermittent Adoption, 5) Premature Discontinuation, and 6) Adherence [28]. Their model clearly acknowledges that one's adherence to a treatment regime is not static; one does not *achieve* adherence and stop, it requires an ongoing effort from both the patient and the provider.

Bailey *et al.* examined adherence through a health literacy lens [29]. They created a six phase model, similar to Gearing *et al.*'s model, which considers the patient's knowledge of medications and their overall health. The model's phases are: 1) *Fill* the prescription, 2) *Understand* how to administer to medication; 3) *Organize* and plan daily schedules, 4) *Take* the medication as prescribed, 5) *Monitor* health and potential side effects, and 6) *Sustain* safe and appropriate use [29]. They describe the impact of health literacy on adherence at each stage in the model. For example, an important part of the *Monitor* phase is improving literacy related to potential side effects and determining when to contact care providers for assistance [29]. A study by McGinnis *et al.* supports Bailey *et al.*'s focus on health literacy; McGinnis *et al.*'s results indicate that non-adherence is more likely to be exhibited by patients with lower health literacy [6].

1.1.2 Measurement of Adherence

Vrijens *et al.* point out that existing definitions and models of adherence require that methods of measurement exist for the concepts of compliance and agreement, with compliance being much easier to quantify than agreement [23]. Indeed, many techniques exist for measuring compliance, but to my knowledge, there is no widely accepted technique for measuring agreement. Nonetheless, existing measurement techniques for compliance provide a starting point for the work presented in this thesis, and are discussed below.

Measurement of compliance has proven a significant challenge, various reasons exist for this, including: 1) failure of existing measures to capture the intricacies

of human behaviour [10]; 2) the cost of accurate data collection methods at scale [10]; and 3) variation of definitions in literature [21]. Similarly to the confusion regarding terminology, discussed above, a lack of precision in discussions regarding compliance measurement exists; several works report on measures of compliance but fail to correctly categorize the techniques [10, 18, 19, 21]. The primary issue is a failure to distinguish between techniques of *data collection* and techniques for *computing metrics*. A single data source could be used to compute different compliance metrics or vice versa.

Data Collection Techniques

Techniques for data collection regarding compliance vary in their complexity, accuracy, and cost. The most common techniques reported in the literature are presented, these include: patient self-report, pharmacy and insurance records, pill counting, biological testing, and electronic devices. Here the discussion of data collection techniques is limited to an out-patient setting, though some techniques may be applicable in other settings.

Patient self-reporting, which includes patient diaries, surveys, questionnaires, and discussions with providers, is the most commonly used technique to gather data regarding compliance [2, 19, 30]. These methods are simple and inexpensive to implement at scale, in both clinical and research settings. However, the accuracy of these techniques relies upon patients to accurately report their adherence [2, 19], and for providers to correctly interpret their patients' responses [10]. Interestingly, of all of the data collection techniques discussed here, self-reporting is the most promising for determining a level of agreement between patients and providers, however, the literature only describes the use of self-reporting for evaluation of compliance.

Pharmacy and insurance claims databases contain records of a patient's refilling patterns; this information can be used to calculate numerous compliance metrics [10, 31]. Refill datasets are often electronic, therefore they are easily accessed and analyzed, this makes them ideal for studying long term population level trends [10, 19, 31]. However, this data collection mechanism does not reveal detailed administration information such as the time of day the medications are administered. Further, refill datasets are only suitable for medications that must be refilled regularly and are not applicable for "as needed" medications or short-course antibiotics. Refill datasets have been used as a data source by many studies, interestingly, there is tremendous

variation in the calculation of compliance metrics based on the data [31].

Pill counting is a practice whereby an individual (patient, health care provider, or peer) periodically counts the number of doses remaining in the patient’s supply [10]. This technique produces similar data to refill datasets and has a higher resolution as counts can occur several times per refill. However, the approach suffers from a lack of detailed timing and dosing [2, 10]. Additionally, pill counting requires a person to count the pills which reduces the scalability of the approach, especially in cases where counting cannot be conducted by the patient themselves.

Testing a patient’s blood or urine for the prescribed substance, metabolites, or biological markers may be used to determine whether medications have been consumed [10]. This approach is generally expensive and invasive for the patient; further, variations in patient metabolism of substances can effect results [10]. This type of data collection has been called *direct measurement* in the literature [10]; however, that is a misnomer, these measurements are logically outcome measures. A patient should be considered compliant with a prescription if they have administered the medications as prescribed, since biological levels can be affected by additional factors [2]. This is not to say that biological levels are not important to clinical medicine, in fact the opposite is true, but calling such measures “direct” may lead one to draw false conclusions about the patient’s compliance.

Electronic devices embedded in medication containers can capture administration information by recording the time the patient interacted with the device. Such devices are becoming increasingly popular since they capture higher resolution data and are considered by many to be one of the most reliable data sources for measuring compliance [2, 10, 19, 32]. Since these devices are electronic the data they produce can be easily aggregated and automatically analyzed. Two drawbacks to electronic data collection exist: 1) at this time, it is not possible to determine if the patient consumed the medication after it was dispensed, and 2) individual devices may be quite expensive making large scale use difficult.

Calculating Compliance Metrics

Given one or more data sources one can calculate a compliance metric, a numerical score representing the patient’s compliance. Numerical values permit quantitative reasoning, which is critical to understanding compliance in both research and clinical settings. However, there are no standardized compliance metrics [21, 31] which has led to issues

during meta-analyses of the literature aimed at establishing scientific consensus [19]. The remainder of this section presents the most most common (according to several published reviews [19, 21, 31]) metrics for compliance.

The Morisky Medication Adherence Scale (MMAS) is an eight item scale that uses patient self-reported data [33, 34]. It is a commonly used metric when self-report is used as a data collection method [18]. The main advantage of the MMAS is its simplicity and ease of use in a clinical environment [34]; additionally, the scale has been validated by clinical studies [33, 34]. However, concerns regarding the accuracy of self-reported adherence remain an issue for this metric.

The Mean Possession Ratio (MPR) is a common metric for compliance that is generally calculated as the ratio of the number of days of medication supplied to the number of days in an observation window. The metric is usually computed using data derived from pharmacy refill records, insurance claims, or pill counting [21, 31]. Since this metric is a ratio it naturally corresponds to the notion of compliance as a non-binary concept. Interestingly, several variations of the MPR metric have been used in the literature [21]. Furthermore, as Raebel *et al.* point out, the MPR metric, by its reliance on refill datasets, is not able to capture patients who never fill or refill their prescriptions [21]. Finally, the MPR is a coarse metric as it does not account for moment to moment behaviour. Related metrics, which use similar formulae, include: Proportion of Days Covered (PDC), Medication Refill Adherence (MRA), and Continuous Multiple Interval Measure of Oversupply (CMOS) [21].

Many studies reportedly use electronic devices as sources of data for measuring compliance [19, 30, 35, 36]. Both Varshney and Bosworth identified two metrics: 1) the percentage of days the device was accessed, and 2) the percentage of inter-dose intervals that were executed as prescribed [18, 37]. While the later metric provides a clearer picture of behaviour, both metrics fail to take full advantage of the data provided by the electronic device. Other studies have used the percentage of doses taken at the “correct time” of day over the observation timeframe, however it is not clear how “correct time” of day was defined or how doses taken outside of the recommended time were assessed [35, 36].

Section Summary

In summary, adherence is a complex multifaceted problem. Imprecise terminology and conceptual models have led to confusion in the literature at a conceptual level.

A common problem is the use of the term “adherence” when “compliance” (which does not imply patient agreement) is the subject of discussion. Determining accurate terminology and conceptual models of adherence remains a problem for researchers, until an agreement is reached by the research and clinical communities, confusion will likely persist.

Imprecision has affected the area of adherence measurement with confusion being caused by failure to distinguish between data sources and metrics for compliance. Furthermore, there is no known metric for patient-provider agreement with respect to prescriptions. Several data capture techniques exist, each with advantages and disadvantages: electronic devices, while expensive, provide a high-resolution data source; pharmacy refill data permits studying larger populations of patients; and self-reporting is easily used by front-line health care providers and can provide valuable qualitative information. Combing data collection methods may yield a richer picture of a patient’s compliance. Many metrics for compliance have been used in the literature. The Morisky scale and Mean Possession Ratio (MPR) are commonly used; however, several reviews have found variation in the calculation of these metrics which has led to an inability to compare results across studies. This ultimately makes it difficult to determine how best to support patients and clinicians seeking to improve compliance.

1.2 Proposed Solutions to Non-Adherence

Keeping with terminology used in medical research, enacted solutions to a patient’s non-adherence are called *interventions*. Many studies have explored the impact of various interventions with mixed results [19, 26]. Commonly reported interventions include [18, 26]:

- strategies to improve patient education,
- reducing regime frequency/complexity,
- peer/mentor support and engagement,
- professional counseling before and during treatment,
- individual and group goal setting,
- improving access to and support from pharmacists,
- reminder systems,
- and specially designed pill containers.

Given the multifaceted nature of the problem, it is not surprising that the most successful interventions are combinations of the items listed above [18]. Unfortunately, several reviews of interventions have concluded that, due to a lack of methodological rigor, it is not possible to determine, in general, the best intervention for improving adherence [19, 30, 38, 39]. Nonetheless, authors have indicated that the most promising interventions include: patient education, on-going patient provider engagement, and regular reminders and feedback for patients. Though, additional high quality studies are required to reach a strong conclusion [38, 40].

1.2.1 Role of Technology

Technology has become a key part of non-adherence interventions and recent reviews have reported that technology, when used in combination with other intervention types, can reduce non-adherence [26, 30, 38, 41]. Technological interventions for improving adherence include [30, 40, 42]:

- SMS message reminders,
- telephone (voice only) reminders and counseling,
- video-conference counseling,
- automated reminders from mobile (or other electronic) devices,
- pager systems,
- email reminders and counseling,
- interactive computer programs and mobile applications,
- pill containers with built in reminder functions,
- and electronic and personal health records (EHRs and PHRs).

Increasingly, humans are immersed in technologically rich environments that are capable of observing behaviours and providing feedback. The Internet of Things (IoT) can be leveraged to incorporate data from numerous devices, including: mobile telephones, wearable technology, kitchen appliances, and motion sensors placed in a patient's home. Non-adherence interventions that consider patient lifestyle factors, such as daily routines, and provide meaningful feedback and real-time decision support hold tremendous promise [38, 40, 43]. A conceptual architecture for such a system has been proposed by Varsheny wherein patients interact with an electronic pill dispenser which is connected to a health care provider's electronic record [37]. This architecture permits analysis of compliance data in the context of other information available in

the provider’s record; for example, correlating blood pressure with compliance rates over time [37].

1.2.2 A Computational Definition of Adherence

As discussed above, several conceptual frameworks for adherence have been proposed, and while they may be useful from a clinical point of view, they do not explicitly consider the role of technology. Developing frameworks that can support engineers and scientists who are creating new technologies will hasten the advancement of effective technological interventions. Varshney’s architecture provides some guidance, but his description of compliance fails to account for the fact that human behaviour is not easily described by “crisp” intervals or boolean concepts [37].

Additionally, Varshney’s architecture illustrates how technology can be used as a significant part of an intervention to improve adherence, but does not indicate how to transform medication prescriptions stored in an electronic database into a representation that is readily consumable by adherence management technology. Computing technology, for all of its complexity, is based upon relatively simple mathematical constructs. Expressing prescriptions in such a form is a critical prerequisite to achieving the full benefit of technology. To this end, a computationally precise definition of adherence is required.

An important part of a computational definition of adherence is a means of expressing and communicating prescriptions between devices: a language for describing prescriptions must exist. Since such a language must communicate with computing technology, it would be beneficial if the language had its syntax and semantics formally defined. A formal language would remove ambiguity regarding adherence and make comparing measures of adherence between different devices possible. Additionally, by formally defining a semantics for such a language, the precise meaning of a prescription is understood. A major challenge of creating a computational definition of adherence is ensuring the semantics of the language correctly capture the intent of the prescriber and that they consider factors beyond the medication of interest.

Once a computational definition of adherence has been established there are numerous possible applications. From a research perspective, such a definition can be used to measure adherence (compliance and/or agreement). A computational definition can also be used for patient decision support, i.e. helping patients understand when to take their next dose of medication. Finally, by providing a clear definition of adherence

that is compatible with technology, a computational definition can support creators of new technological interventions aimed at improving adherence.

Many methods of describing prescriptions exist; electronic systems that handle prescriptions must have a data model for describing them. Arguably these are prescription languages, some of which may have formal underpinnings. An interesting prescription description language was developed by Yeh *et al.* and is called APAMAT [44]. APAMAT is notable for its formal syntactic description, the breadth of prescribing concepts covered by the language, and its use as input to a drug-drug interaction engine [44], all of which are important for a language to underpin a computational definition of adherence. However, APAMAT does not have, to my knowledge, a formalized semantics and does not consider factors beyond medications, e.g. meal times. Indeed, these qualities cannot be found amongst any published languages or data models.

Section Summary

In summary, many interventions aimed at improving adherence have been proposed. Technology has and will continue to be an important part of a multifaceted approach to solving non-adherence. As computing technology becomes more pervasive in our daily lives as part of the Internet of Things, interventions can leverage the rich data produced by a patient's environment. To facilitate the use of technological driven interventions a definition of adherence that is consumable by computing technology must be created. An important part of a computational definition of adherence is the language that is used to communicate prescriptions between devices, such a language would benefit from a formal syntax and semantics and must capture a breadth of concepts related to the prescription.

1.3 Objectives and Outline

Based on the discussion above, this thesis seeks to provide a suitable computational definition of adherence. More specifically, the objectives of this work are as follows.

- Create a conceptual framework for understanding adherence that explicitly includes the notion of patient agreement and can be used to inform the development of technological adherence interventions.

- Propose a precise means of measuring adherence (compliance and agreement) that considers both medications and external factors such as patient lifestyles and is tolerant of the imprecision inherent to human behaviours.
- Create a concrete implementation of the proposed method for *compliance* measurement that could be integrated with an existing or future health information technologies.
- Apply the proposed measurement method to the problem of scheduling medication administrations.

1.3.1 Summary of Approach and Outline

To meet these objectives, this thesis first introduces the Adherence Interaction Model (AIM), a conceptual framework describing adherence and the relationship between three concepts: compliance, persistence, and agreement. AIM is used to guide the creation of a language for specifying medication prescriptions, the syntax and semantics of which are formally defined. A prescription expressed in this language can be used to derive a compliance evaluation function which is capable of evaluating a patient's historical behaviour and producing a compliance measure. The generated compliance function can be subjected to an optimization algorithm to determine the best time to administer the next dose of medication. The resulting computational definition and language is evaluated for breadth of expression and clinical accuracy. The remainder of this thesis is structured as follows.

- Chapter 2 provides foundational material, including an overview of formal language theory and fuzzy set theory.
- Chapter 3 outlines the approach proposed by this thesis, including: a description of AIM, requirements for a prescription specification language, an approach for measuring compliance, and for scheduling medication administration.
- Chapter 4 provides a formal description of the language's syntax and semantics and gives an example of the approach proposed for scheduling medication administration.
- Chapter 5 evaluates both the language's breadth of expressiveness and the approach for measuring compliance.

- Chapter 6 discusses the results of the evaluation and known limitations of the proposed approach.
- Chapter 7 outlines directions for future work and makes concluding remarks.

Chapter 2

Foundations

The computational definition of adherence developed by this thesis uses concepts from theoretical Computer Science. This chapter gives an overview of the requisite theories and concepts. First, the formalization of languages and associated techniques are discussed, with particular emphasis on graph grammars, attribute grammars, and denotation semantics as these are heavily used in later chapters. Second, fuzzy set theory and fuzzy logic are introduced as an extension of classical set theory that permits reasoning with degrees of uncertainty or imprecision.

2.1 Formal Languages

Fundamentally, languages are systems for communication between entities [45]. The study of languages may be considered from the perspectives of *syntax*, *semantics*, and *pragmatics* [45–47]. The syntax of a language defines the structure of the language, semantics provide a mapping from the syntactic forms to concepts in a universe of discourse, and pragmatics capture how syntax and semantics are used to communicate [45].

Many languages are informal, they have established syntactic, semantic, and pragmatic conventions that may naturally change and are subject to ambiguity in their interpretations [45, 48]. An interesting, somewhat contemporary, example of this is the English word “phone”, which, has changed from meaning an analog device used for voice communication between two humans, to describing a pocket-sized digital computer which is capable of transmitting multiple different types of communications to numerous other devices. This is an example of both the pragmatics and semantics

of the word “phone” changing due to contextual change. Given that at least two different meanings of “phone” exist, entities communicating using English may make assumptions regarding what type of “phone” is being discussed.

Conversely, formally defined languages have a precisely defined syntax and semantics. Accordingly, a “sentence” in a formal language can only have one interpretation. Often mathematical techniques are used to formalize a language’s syntax and semantics. Due to the nature of computing technology, computer programming languages tend to be more formal than human natural languages. Here, the qualifier “more” is required since not all computer languages are entirely formalized; most languages have a standardized formal syntax and a subset of their semantics formally defined, but different semantic definitions may exist. The language presented in this thesis has a formally defined syntax and semantics, accordingly, several mathematical techniques for formalization are presented below.

2.1.1 Syntax

The syntax of a language is typically formalized by means of a *grammar*. Formally, a grammar consists of:

- Σ a set terminal symbols.
- Δ a set of non-terminal symbols.
- P a set of production rules $X \rightarrow Y$ where both X and Y are strings from the set $\Sigma \cup \Delta$.
- $S \in \Delta$, a start symbol.

Any “sentence” consisting exclusively of symbols from Σ is considered to be part of the grammar’s language if it can be generated by successive application of production rules from P to the start symbol S . A grammar is called *context-free* if the left-hand side of all production rules consists of a single non-terminal symbol. The Backus-Naur Form (BNF) of a grammar requires that production rules be both context-free and expressed using a particular notation. For example, a grammar for binary numbers may have terminal symbols $\Sigma = \{0, 1\}$ and non-terminal symbols $\Delta = \{B, D\}$ and production rules expressed in a BNF [47]:

$$B ::= DB \mid D$$

$$D ::= 0 \mid 1$$

The first rule replaces a binary number with another binary number and/or a digit. In the second rule, a digit is replaced by either zero or one. Then, all binary numbers with at least one digit are part of the language described by this grammar.

Graph Grammars

Graph grammars are a generalization of the previously described string grammars in which graph nodes and edges replace traditional string-only symbols [49]. It follows that a graph grammar defines a language of graphs which may be generated by application of its production rules. In a graph grammar, production rules have a left-hand side graph (LHS) and right-hand graph (RHS). To apply a production, a sub-graph that matches LHS of the rule must be found in a host graph, the changes specified by the RHS are then applied to the matched sub-graph resulting in a modified host graph; changes may include additions or deletions of edges and nodes.

Figure 2.1 shows a sample production rule definition and application to a host graph. The rule (top two graphs) creates three new nodes and connects them with the matched node in the host graph.

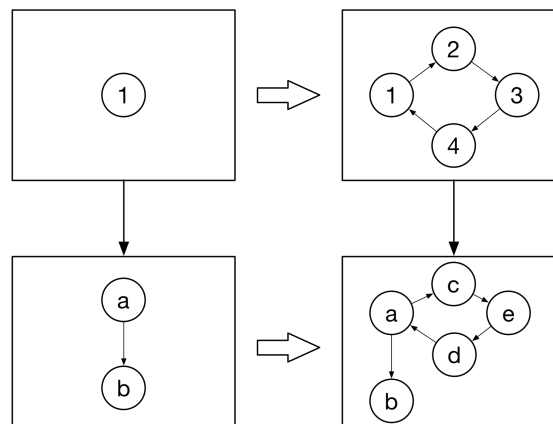


Figure 2.1: Sample graph grammar production rule

There are several formulations of graph grammars, they differ in their mathematical foundations and the details for rule application [50]. For example, while both the Double Pushout and Single Pushout approaches are grounded in category theory, they

differ in how they handle “dangling” edges, i.e. those edges that may be left after a node is deleted by a rule application; the former prohibits the rule application while the latter deletes the dangling edges [51].

Graph grammars have an extensive theoretical background and are suitable for many tasks beyond the definition of formal languages [50]. This thesis does not attempt to provide a full account of graph grammars, instead it focuses on Node Replacement graph grammars [52] which are used to define the syntactic structure of a prescription language.

Node Replacement Graph Grammars In a node replacement graph grammar a production rule’s LHS consists of a single typed node that may be replaced by a sub-graph. Rules are augmented with embedding relations which determine how to connect the new sub-graph to the neighbourhood of the replaced LHS node [52]. Node replacement graph grammars are inherently context-free, this makes them useful for defining graphical languages. Several types of node replacement graph grammars exist that vary in the structure of the embedding relation [52]. The syntactic definition in this thesis uses *edge-directed Neighborhood Controlled Embedded* (edNCE) node replacement graph grammars which are described in detail below.

Allow a graph to be defined as $H = (V, E, \lambda)$ where:

- V is the set of nodes with unique identifiers.
- $E = \{(u, v, \gamma) : u, v \in V, \gamma \in \Gamma\}$, is the set of directed edges connecting two nodes u and v with an edge labeled by γ .
- $\lambda : V \rightarrow \Sigma$ is a node labeling relation which assigns labels to each node.

Then an edNCE grammars is a tuple $GG = (\Sigma, \Delta, \Gamma, P, S)$ where Σ is the set of node labels, Δ is the set of non-terminal node labels, Γ is a set of edge labels, and S is a starting node label, and P is a set of productions which have form $X \rightarrow (D, C)$ where [52]:

- $X \in \Delta$ is the LHS of the production and is a single (non-terminal) node label.
- $D = (V_D, E_D, \lambda_D)$ is the RHS of the production and is a graph such that: V_D is a set of nodes, $E_D \in V_D \times V_D \times \Gamma$, and $\lambda_D : V_D \rightarrow \Sigma$

- $C = \{(x, a, b, y, d) : x \in \Sigma, y \in V_D, a, b \in \Gamma, d \in \{\text{in}, \text{out}\}\}$ is a set of tuples which define how the RHS graph D is embedded into the neighborhood of the X labeled LHS node. Informally, a node with label x that was previously adjacent to X via an d -directional a -labeled edge will be connected to y via a d -directional b -labeled edge.

Application of a production $p : X \rightarrow (D, C)$ to a host graph $H = (V_H, E_H, \lambda_H)$ to produce a new graph $H' = (V_{H'}, E_{H'}, \lambda_{H'})$ proceeds as follows [52]:

1. Find a node $v \in V_H$ such that $\lambda_H(v) = X$.
2. $V_{H'} = V_D \cup (V_H - \{v\})$.
3. $E_{H'} = E_D \cup (E_H - \{(p, q, \gamma) \in E_H : v = p \vee v = q\}) \cup E_{DH'_{in}} \cup E_{DH'_{out}}$
4. $\lambda_{H'} = (\lambda_H - (v, X)) \cup \lambda_D$

Where $E_{DH'_{out}}$ and $E_{DH'_{in}}$ are defined as:

$$E_{DH'_{out}} = \bigcup_{(x,a,b,y,d) \in C} \{(y, q, b) : y \in V_D, q \in V_H, \lambda_H(q) = x, \wedge (v, q, a) \in E_H, d = \text{out}\}$$

$$E_{DH'_{in}} = \bigcup_{(x,a,b,y,d) \in C} \{(y, q, b) : y \in V_D, q \in V_H, \lambda_H(q) = x, (q, v, a) \in E_H, d = \text{in}\}$$

For example, consider an edNCE graph grammar with $\Sigma = \{0, 1, B, D\}$, $\Delta = \{B, D\}$, $\Gamma = \{\text{next}\}$, $S = B$, and the set of production rules shown below in Figure 2.2. Note that in Figure 2.2 each production rule is shown graphically with the non-terminal shown on the LHS and the replacement graph D shown on the RHS; connection relations are provided below in Equation 2.1.

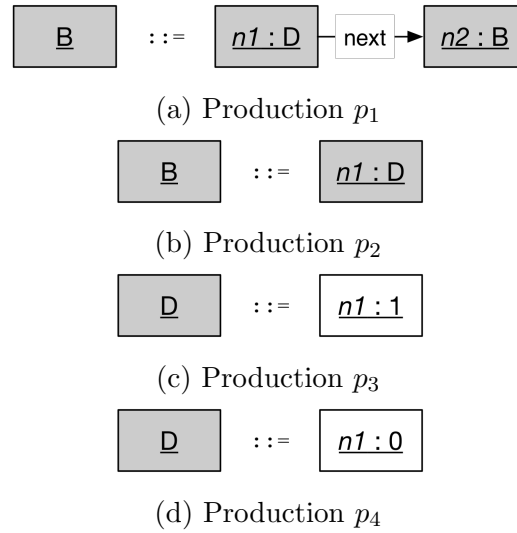


Figure 2.2: Sample production rules.

$$\begin{aligned}
 C_1 = C_2 &= \{(D, \text{next}, \text{next}, n_1, \text{in})\} \\
 C_3 = C_4 &= \{(D, \text{next}, \text{next}, n_1, \text{in}), \\
 &\quad (D, \text{next}, \text{next}, n_1, \text{out}), \\
 &\quad (0, \text{next}, \text{next}, n_1, \text{out}), \\
 &\quad (0, \text{next}, \text{next}, n_1, \text{out}), \\
 &\quad (1, \text{next}, \text{next}, n_1, \text{in}), \\
 &\quad (1, \text{next}, \text{next}, n_1, \text{out}), \\
 &\quad (B, \text{next}, \text{next}, n_1, \text{out})\}
 \end{aligned} \tag{2.1}$$

This grammar generates graphs that represent binary bit strings and is similar to the string grammar presented above. Figure 2.3 shows the derivation of the graph “011” via the sequence of productions: $p_1, p_1, p_2, p_3, p_4, p_4$.

2.1.2 Semantics

The semantics of a language, which associates meaning with syntactic forms, may be formalized by use of: attribute grammars [53]; defining an axiomatic semantics; defining an operational semantics; and/or defining a denotational semantics [47, 54, 55]. Each of these is described in turn.

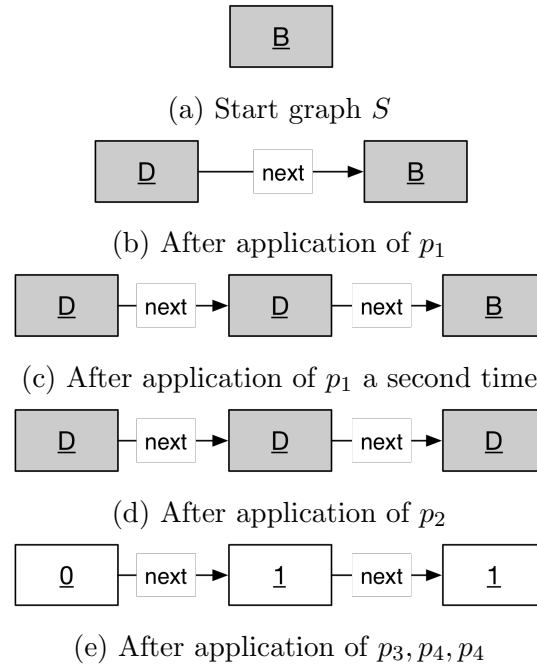


Figure 2.3: Derivation of the bit string graph “011”.

Attribute Grammars Attribute grammars were first introduced by Knuth as a method of assigning meaning to a sentence expressed in a language with a syntax defined by a context-free grammar [53]. Attributes are associated with non-terminal symbols in the language and are either *inherited* or *synthesized*; their value is determined by semantic rule which are, in essence, functions associated with the production rules of a context-free grammar [53]. The value of an inherited attribute is based on the value of the node’s ancestors in the parse tree, while the value of a synthesized attribute is based on the value of the node’s descendants in the parse tree.

Consider an example adapted from [53]. The base-ten value of a string of binary bits may be defined by associating semantic rules with the grammar provided in section 2.1.1 above. The grammar and associated semantic rules are (note - B1 and B2 are both B symbols, numbers are used to distinguish between them in the semantic rules):

$$\begin{array}{ll}
 \text{B1} ::= \text{B2 D} & v(\text{B1}) = v(\text{D})2^{l(\text{B1})} + v(\text{B2}) \quad l(\text{B1}) = l(\text{B2}) + 1 \\
 \text{B} ::= \text{D} & v(\text{B}) = v(\text{D}) \quad l(\text{B}) = 0 \\
 \text{D} ::= 0 & v(\text{D}) = 0 \\
 \text{D} ::= 1 & v(\text{D}) = 1
 \end{array}$$

Here, each non-terminal symbol has a value attribute v which gives the base-ten value of the non-terminal. The attribute l gives the level in the parse tree of a bit string B and corresponds to the bit string's least significant bit position. Note that in this example both attributes v and l are synthesized as they depend only on the value of descendants in the parse tree.

Axiomatic Semantics Hoare introduced an axiomatic semantics for a number of widely used computer programming language elements such as variable assignment, conditionals, and loops [56]. In Hoare's formulation the effect of each statement on a set of assertions is formally expressed. Assertions are typically Boolean functions over the current state of the program's execution. Each statement, S , if executed from a state where the assertion P is satisfied will (if the statement terminates) result in a state where the assertion Q is satisfied; P and Q are typically called the *precondition* and *postcondition* respectively. It may be said that: *the statement S establishes the condition Q from condition P .* Defining an axiomatic semantics of a language requires establishing appropriate assertions for each type of a statement in a language [55].

Operational Semantics Operational semantics define the meaning of syntactic structures by showing their effect on a real or abstract machine [55]. Compilers for conventional computers often define an operational semantics by mapping the syntax of the programming language (e.g. C) into discrete instructions that affect the state of real physical device(s) (e.g. CPU and memory). Alternatively, an abstract machine, such as a Turing Machine, may be used; syntactic structures of the programming language are mapped to discrete transformations of the abstract machine's state [54].

Denotational Semantics Of the methods for defining a language's semantics, the denotational approach is the most mathematically rigorous [55]. A denotational semantics definition has three components [47]:

1. A context free grammar that describes the syntax of the language.
2. A mathematical description of the domain(s) of interest, referred to as a *semantic algebra*; this includes a definition of the domain itself and operations that are permitted on elements of the domain.
3. A set of *valuation functions* which map the the grammar's production rules to

the elements described by the semantic algebras using the recursive functions (e.g. the lambda calculus).

The denotational approach is used in this thesis to formally describe the semantics of a language for specifying prescriptions. Given its importance to this thesis, a descriptive example (adapted from [47]) is provided.

Consider the context free grammar for a binary string provided above, repeated here for convenience:

$$\mathbf{B} ::= \mathbf{DB} \mid \mathbf{D}$$

$$\mathbf{D} ::= 0 \mid 1$$

Where \mathbf{B} is a **binary-number** and \mathbf{D} is a **binary-digit**. To define a denotational semantics for this language of binary strings a semantic algebra is required; there is only one domain of interest:

I. Natural Numbers

Domain: \mathbb{N}

Operations:

zero, one, two, ... : \mathbb{N}

plus, multiply : $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

Where *zero, one, two, ...* are constant values and *plus* and *multiply* have the definitions afforded by conventional arithmetic. The valuation functions that map from the grammar's production rules to the domain are as follows, note that functions are invoked using a prefix notation:

\mathbf{B} : **binary-number** $\rightarrow \mathbb{N}$

$\mathbf{B}[[\mathbf{BD}]] = (\mathbf{plus} (\mathbf{multiply} \mathbf{B}[[\mathbf{B}]] \mathit{two}) \mathbf{D}[[\mathbf{D}]])$

$\mathbf{B}[[\mathbf{D}]] = \mathbf{D}[[\mathbf{D}]]$

\mathbf{D} : **binary-digit** $\rightarrow \mathbb{N}$

$\mathbf{D}[[0]] = \mathit{zero}$

$\mathbf{B}[[1]] = \mathit{one}$

The left-hand side of each production rule is associated with function, **B** or **D** above, with the right-hand side providing the argument (shown in double square braces). Each function-argument pair corresponds to an operation whose result is returned after recursively evaluating all internal valuation functions. Given a parse tree for a sentence in the language, one may derive an expression which when evaluated will yield a result within the domain of interest. For example, the binary number 10 would have the parse tree as seen in Figure 2.4, the following expression would be obtained by application of the valuation functions: (**plus (multiply one two) zero**).

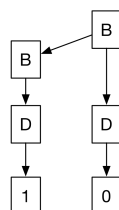


Figure 2.4: Parse tree for binary number 10

Which, when evaluated, yields a result of *two* as wanted for the binary number 10. In addition to the notation described above, denotational definitions use conditional expression: $C \rightarrow A[]B$ which is interpreted as: *if the Boolean condition C evaluates to true, then evaluate A, otherwise evaluate B* [47].

Lambda Expressions In future chapters lambda expressions are used to describe functions and are used in conjunction with the denotational approach to formalize the semantics of a language. The concise notation of lambda expressions is ideal for describing higher-order functions and are easily mapped into functional programming languages, both of which are extensively used in this thesis. A short description of the notation, based on the description in [57] is given here.

Lambda expressions represent unary functions, and are structured as:

$$\lambda\langle\text{arg}\rangle.\langle\text{operation}\rangle$$

Where $\langle\text{arg}\rangle$ is a symbol that represents the argument or input of the function, and $\langle\text{operation}\rangle$ is a prefix notation expression whose value is the returned value of the expression. The expression is anonymous in the sense that it does not have a name assigned to it, in contrast to the more conventional notation $f(\langle\text{arg}\rangle) = \langle\text{operation}\rangle$ which names the function as f . As an example, a lambda expression describing a function which adds 5 to the input is:

$$\lambda x.(+ 5 x)$$

Functions with multiple arguments are described by composing several lambda expressions:

$$\lambda x.\lambda y.(+ x y)$$

This expression is actually two functions; evaluating the outer function results in a new function which adds a constant to the input. Functions are invoked by surrounding them with round braces and providing the arguments directly after the function name, for example:

$$(\lambda x.\lambda y.(+ x y) 5)$$

This results in the expression: $\lambda y.(+ 5 y)$ which can then be applied to another argument. The outer function (using argument x) is an example of a higher-order function, i.e. a function whose result is another function; higher-order functions may also accept another function as an argument.

2.2 Fuzzy Set Theory

Fuzzy sets were first introduced by Zadeh in 1965 as a means of describing classes of objects which have varying degrees of membership in a set [58]. This is in contrast to classical sets which are “crisp” in the sense that objects are either in a set or not in a set. As a result fuzzy logic permits reasoning about imprecise statements which would be challenging to analyze using only classical “crisp” logic.

Fuzzy set theory, and more generally fuzzy systems, have many applications from natural language processing to robotics control [59]. This thesis proposes yet another application of fuzzy set theory to medication prescriptions and adherence. A short introduction is provided here, and a comprehensive description of the theory and contemporary applications is provided in [59], upon which the following notes are based.

2.2.1 Fuzzy Sets

A classical “crisp” set is a collection of objects, A , from a universe of discourse, U , which satisfy some Boolean criteria, $\mu : U \rightarrow \mathbb{B}$, often this is expressed as

$A = \{x \in U : \mu_A(x)\}$. Importantly, μ is a membership criteria or function, and U is itself a set to which all objects of interest belong. For example, consider a membership function $\mu_A(x) = x < 5$ and $U = \mathbb{N}$, the resulting set is $A = \{1, 2, 3, 4\}$. Objects and set membership may be visually expressed using a Venn diagram, as in Figure 2.5a.

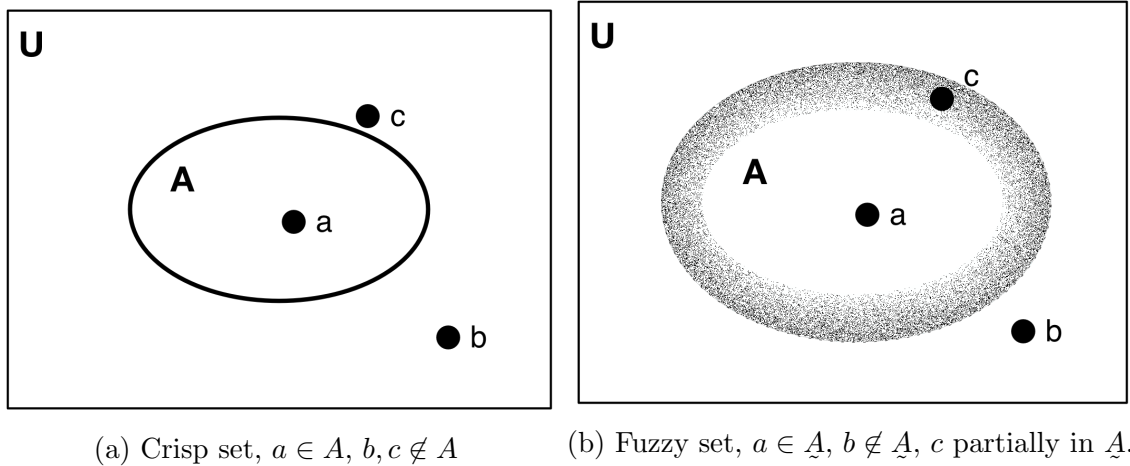


Figure 2.5: Visualizing crisp and fuzzy sets as Venn diagrams, adapted from [59].

When modeling reality using mathematics, as is often done in science and engineering, one must abstract away details and make assumptions. In some cases these assumptions are appropriate, in others they are limiting. Consider the set, R , of cars which are painted the color “blue”, the membership function μ_R is seemingly crisp, cars are either painted blue or they are not. For some purposes, μ_R may be sufficient. However, in an application with a more subtle sense of color gradients (such as computer vision), it may not be acceptable to say a car is blue or not blue, the visible color of the car is composed of a combination wavelengths of light with higher intensity around wavelength 470 nm. The following questions may arise, *what wavelengths of light are sufficiently close to 470 nm to be considered “blue”?* and *what relative intensity of wavelengths is required to produce colors which appear “blue”?*

2.2.2 Fuzzy Membership Functions

Fuzzy set theory proposes that objects in the universe of discourse are members of a set to some *degree*, from 0 (not in the set) to 1 (in the set). It follows that a fuzzy set’s membership function maps from the universe of discourse, U , to the real interval between 0 and 1 (inclusive), i.e. $\mu : U \rightarrow [0, 1]$ [59]. The Venn diagram representation

may be modified, as seen in Figure 2.5b, to describe a “fuzzy” boundary of a set, the shape/character of the fuzzy boundary depends on the character of the fuzzy membership function.

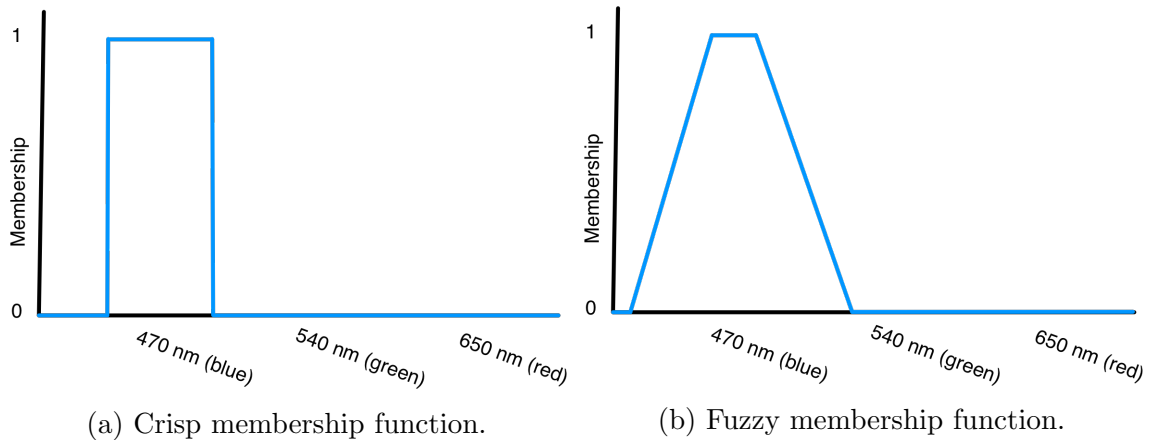


Figure 2.6: Membership functions for wavelengths of light which are considered “blue”.

A common shape for a membership function is a trapezoid, shown in Figure 2.6b for wavelengths of light which are “near” 470 nm. Wavelengths that are sufficiently close to 470 nm are considered blue, longer and shorter wavelengths are either blue to some degree or not blue at all. This is in contrast to the crisp rectangular membership function in Figure 2.6a which does not consider some wavelengths to be “close” to blue. Of course, other membership functions which may capture a particular intent are permitted.

De-fuzzing Fuzzy Sets

In some cases, it may be useful to recover a crisp set from a fuzzy set, this is accomplished via an *alpha-cut* operation [59]. An alpha-cut returns a new set $A_\alpha = \{x \in U : \mu_{\tilde{A}}(x) \geq \alpha\}$ where $\mu_{\tilde{A}}$ is a fuzzy membership function, i.e. all of the items which are part of the set A to at least degree *alpha*. The alpha-cut operation is depicted in Figure 2.7.

The preceding paragraphs use a number of adjectives (e.g. “close”, “near”, “short”, “long”) that have relative and somewhat ambiguous meanings to describe wavelengths of light. This is an excellent example of the ambiguity and imprecision that exists in natural language and more generally in human behaviour. Careful selection of fuzzy membership allows one to take advantage of human “tolerance for imprecision” [60]

when designing systems which humans will interact with.

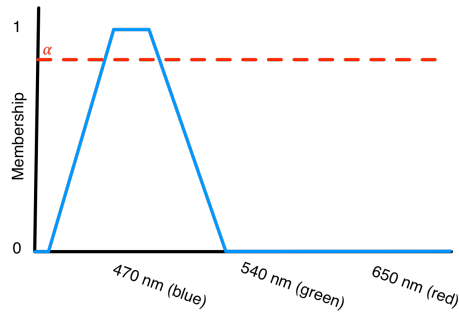


Figure 2.7: Visual representation of an alpha-cut

2.2.3 Fuzzy Set Operations

Commonly used classical set theory operations include: union ($A \cup B$), intersection ($A \cap B$), and complement (\bar{A}). Analogous operations exist for fuzzy sets and are defined as operations on the membership functions [59]:

$$\mu_{\underline{A} \cup \underline{B}} = \max(\mu_{\underline{A}}, \mu_{\underline{B}})$$

$$\mu_{\underline{A} \cap \underline{B}} = \min(\mu_{\underline{A}}, \mu_{\underline{B}})$$

$$\mu_{\bar{\underline{A}}} = 1 - \mu_{\underline{A}}$$

These operations are described visually in Figure 2.8, the shaded area shows the membership function which is produced by performing the operation on the given membership functions.

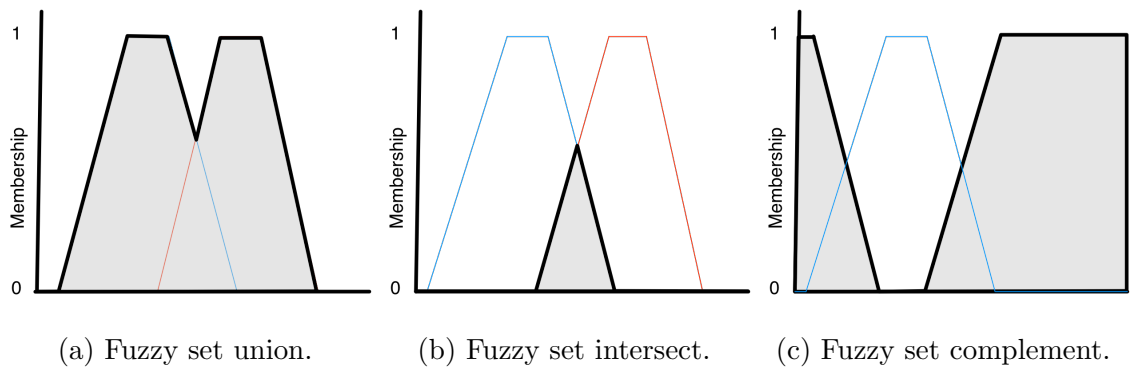


Figure 2.8: Visual representation of fuzzy set operations.

Chapter 3

Concept Formulation

Engineering is the application of concepts from mathematics and the sciences to real-world problems. However, without appropriate conceptual guidance and context these tools and their application inevitably result in a confusing mix of formulae and source code. To avoid confusion, this chapter seeks to clearly define the approach employed in this thesis at a conceptual level without diving into mathematical detail. For the mathematically inclined reader Chapter 4 formalizes the approach.

3.1 Working Example

It is instructive to have an example to refer to and to illustrate concepts. The following persona and set of prescriptions will be used throughout the remainder of this thesis.

Ms. Smith is a 57 year old school teacher at a local elementary school. She used to lead an active lifestyle and was on her feet for most of the day at work. Recently, she was severely injured in a motor vehicle accident and has ongoing pain in her left leg and lower back. Additionally, she has recently been diagnosed with deep vein thrombosis. She is often sore and tired at the end of the day, and frequently goes to sleep when she gets home from work, before dinner.

Ms. Smith's physician has prescribed *6 mg of warfarin to be taken orally once daily* to address her deep vein thrombosis. Additionally, she was prescribed *15 mg of long-acting morphine orally twice daily* and *600 mg of ibuprofen to be taken three times a day as needed with food* to help manage her pain.

3.2 Prescriptions and Compliance

The notions of a prescription and compliance are intimately connected. A prescription can be considered as a set of instructions for the administration of a (medication) treatment, and compliance is the degree to which a patient's behaviour corresponds with a given prescription. To describe the link between prescriptions and compliance first one must understand how to describe patient behaviours.

3.2.1 Behavioural Traces

Consider a hypothetical device that can perfectly observe specific factors of an individual's behaviour. When considering prescriptions and compliance factors of interest may include: timing of administration, amount of substance administered, or food consumption. The only restriction placed on these factors is that they must be observable by the device, as such they are referred to as *observable factors*; however, since the device is hypothetical it stands to reason that one could observe any facet of an individual's behaviour. Then the device could construct a historical account of an individual's behaviour describing the occurrence of each event, such a history is referred to as a *trace*.

For example, consider Ms. Smith's prescription for ibuprofen, the following observable factors might be of interest: time of administration, substance administered, amount of substance administered, time of food consumption, and route of administration. Figure 3.1 shows two possible traces of Ms. Smith's behaviour over a 24-hour period. For simplicity only two observable factors are shown (time of administration and time of food consumption).

Leveraging the Internet of Things

While true omniscient observer devices are likely impossible, it is feasible to create reasonable approximations via an Internet of Things (IoT) ecosystem. The IoT proposes an ecosystem of networked devices which may collectively be able to observe the factors of an individual's behaviour which are relevant to a given prescription. Though not without challenges [61], architectures and devices for an IoT ecosystem for monitoring medication behaviours are the subject of ongoing work [32, 43].

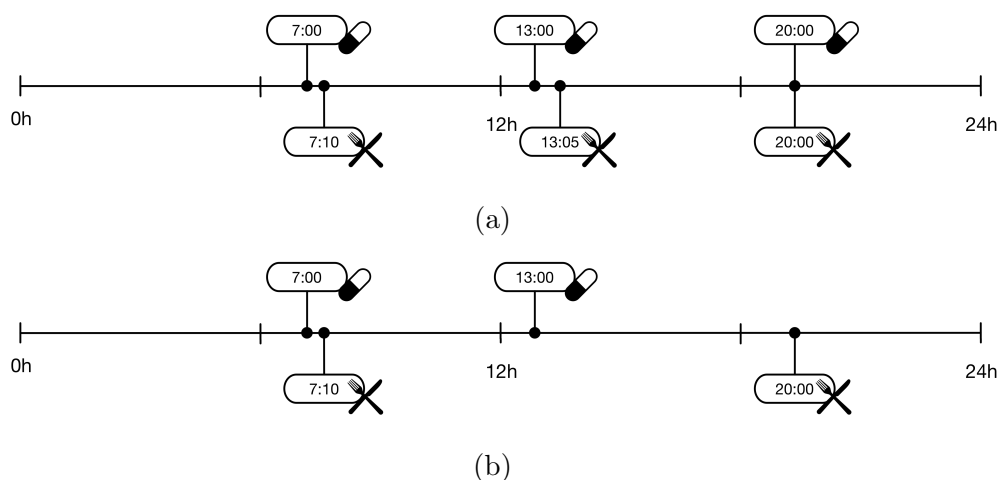


Figure 3.1: Two traces of Ms. Smith’s behaviour w.r.t the prescription: *600 mg of ibuprofen three times per day as needed with food*; pill icons represent administrations, fork/knife icons represent the start of meals.

3.2.2 Compliance

Imagine another hypothetical device, which specializes in the evaluation of traces. Given a prescription and a trace, the device will determine whether the trace satisfies the instructions outlined in the prescription. In essence, the evaluation device is a boolean function, a predicate, that determines whether the trace complies with the prescription. In this sense, a prescription is a specification that is interpreted by the device and that the trace must satisfy to be considered compliant.

The hypothetical device could analyze the trace presented in Figure 3.1a and determine whether it satisfies the associated prescription, the likely answer is “yes - the trace satisfies the prescription”. Similarly, analysis of the trace in Figure 3.1b would likely yield the answer “no - the trace does not satisfy the prescription” because a dose of ibuprofen was administered without food.

3.2.3 The Meaning of a Prescription

It is clear that the notion of a prescription is a prerequisite to the notion of compliance, i.e. determining compliance depends on having a prescription. However, the concepts are interdependent: understanding a prescription depends on the ability to determine compliance.

A linguistic lens is required to understand the dependency prescriptions have on

compliance. When a prescriber decides on a prescription, they have an intent or a desired outcome they (and, hopefully the patient) wish to achieve. However, to be communicated to the patient, a prescription must be expressed in a language, therefore a syntax for describing the prescription must exist; more specifically, a syntax for describing specifications related to medication administration must exist.

Languages are also a means of communicating about some reality; in this context reality is modeled as a trace. Then the meaning of the prescription, the semantics of the language, is a mapping of its syntax to one or more traces. However, there are many possible mappings between a prescription and traces; only some (perhaps only one) correctly capture the intent of the prescriber. These correct mappings indicate which traces are compliant. Evidently, an understanding of compliance is required to correctly capture the semantics of a prescription. Therefore the concepts of compliance and a prescription are interdependent.

Consider Ms. Smith's prescription for warfarin. The prescriber's intent was likely to reduce Ms. Smith's the existing blood clot in her leg. Ms. Smith may want to plan her week, she may read the prescription and attempt to determine a schedule for administering warfarin. She should only consider schedules (which are possible future traces) that comply with the prescription specification, e.g. "once per day". As a human-being Ms. Smith has an understanding of the meaning of these (natural language) constraints and is thus able to select schedules that are likely to satisfy the prescriber's original intent and accommodate her own lifestyle. One reasonable interpretation would be to administer warfarin "in the morning with her breakfast" which is at approximately the same time each day and is a meal she will not miss due being tired from her day of work.

From this perspective, the hypothetical evaluator device described above, which takes as input a prescription and a trace and returns a boolean answer, encodes the semantics of a prescription language. The device must have an internal mapping of what constitutes compliance and then must evaluate a trace based on that mapping.

3.2.4 Degrees of Compliance

Thus far, compliance has been considered as a boolean concept. A trace either satisfies a prescription or it does not. This is useful for making philosophical arguments about the relation between compliance and prescriptions, but in reality it is somewhat limiting. If an individual *usually* follows the prescription with occasional deviations it

would be more accurate to say they are “mostly compliant”, or compliant to some degree. To this end, the previous definition of compliance must be amended to include the idea of a *degree* of compliance.

Then, the hypothetical evaluation device discussed above, given a prescription and a trace, must produce a real number between 0 and 1 that captures the extent of compliance. Indeed, this notion of a “degree” is used directly in the currently accepted definition of adherence from the WHO [2].

From a set theoretic perspective, one may consider a “compliance set” which contains traces that are compliant to a given prescription. Using a boolean definition of compliance the boundary of the set is “crisp”, a trace is either in the set or it is not. However, as described above, considering a degree of compliance is more useful. Therefore one can consider the set as a fuzzy set where traces are within the compliance set to some degree. Of course, the original crisp definition may be reconstructed by making an alpha-cut (described in Section 2.2) of the fuzzy set at some arbitrary degree of choice.

Section Summary

In summary, this section gave informal definitions for the concepts of a trace, a prescription, and compliance; two abstract machines, an observer and an evaluator were introduced to motivate these definitions. Figure 3.5 below shows the role of the observer and evaluator devices in the context of a larger system architecture. The interdependence between prescriptions and compliance was discussed; in essence compliance requires the specification of the prescription to evaluate the behaviour, and compliance is required to formally understand the meaning of a prescription. Further, fuzzy sets were introduced as a means of describing a degree of compliance, a practical improvement over a strictly Boolean formulation.

3.3 Compliance in Context

The discussion of prescription and compliance in the previous section neglects the relationship between physicians and patients, and it does not leave any room for patients to exercise autonomy with respect to their medication-taking behaviour. Current definitions of *adherence* emphasize both compliance and patient agreement with the prescription [2]. Thus, it is important to embed the concepts of prescriptions,

traces, and compliance within a framework that permits discussions about agreement. To this end, this section describes the *Adherence Interaction Model* (AIM) [62], and its relation to the concepts described above. AIM is used to focus the formal/technical work described in subsequent sections and chapters.

3.3.1 Adherence Interaction Model

The Adherence Interaction Model (AIM) describes three aspects of adherence: *compliance*, *persistence*, and *agreement* as measures of distance between a *prescription*, a *conscription*, and a *description* [62].

A *prescription* is as described above, i.e. a plan for medication administration. However, AIM restricts a prescription to the health care provider’s recommendations, and does not imply patient agreement.

A *conscription* is the plan that the patient adopts. This may or may not be the same as the prescription, the patient may: entirely adopt the prescription, partially adopt the prescription, or disregard it entirely [28].

Finally, a *description* is a trace of the patient’s enacted behaviour. This may be thought of as a structure that perfectly describes a behavioural trace.

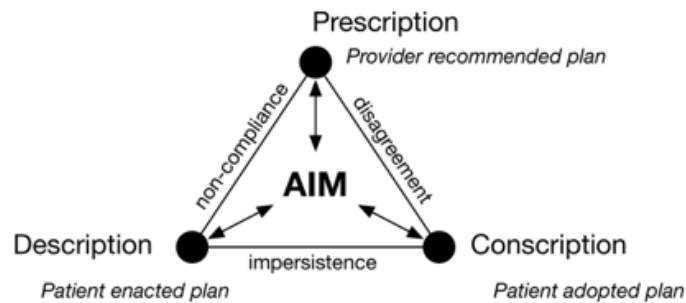


Figure 3.2: Depiction of AIM [62]

Given a prescription, conscription, and description, it is possible to measure “distances” between each pair. The types and names for the corresponding distance measures are shown in Figure 3.2.

The distance between a prescription and conscription is defined as *disagreement*. Since the prescription is the provider’s recommended plan and the conscription is the adopted plan, disagreement naturally represents the distance between the two plans.

Cramer *et al.* define persistence as the “act of continuing the treatment for the prescribed duration” [20]. If the notion of a prescription is replaced by conscription

in Cramer *et al.*'s definition then *impersistence*, the opposite of persistence, is a natural measure for the distance between the conscription, patient adopted plan, and description, the behavioural trace.

Finally, the distance between the the description, behavioural trace, and the prescription is measured by *non-compliance*. The definition of non-compliance as a distance measure follows directly from the discussion in the preceding section.

3.3.2 Measuring Distances

It is worth noting that the distance measures used in AIM are negatives (**non-compliance**, **impersistence**, and **disagreement**). Using this formulation means that the distance between a pair of plans/trace types (e.g. prescription and conscription) will approach zero as the plans/traces become “aligned”. Alignment between plans/traces differs depending on the distance measure chosen.

Viewing the distance measures as axes in a three-dimensional space permits the notion of adherence changing in time. A patient’s compliance, persistence, or agreement may change through time and it may be useful to visualize the evolution. Consider the three-dimensional plot in Figure 3.3, here the measures of adherence form the axes of the three dimensional space and points are plotted showing the change in adherence over time. The optimum state is one where disagreement, impersistence, and non-compliance are all zero, thus the origin of the three-dimensional space becomes the ideal state for a patient, from an adherence perspective.

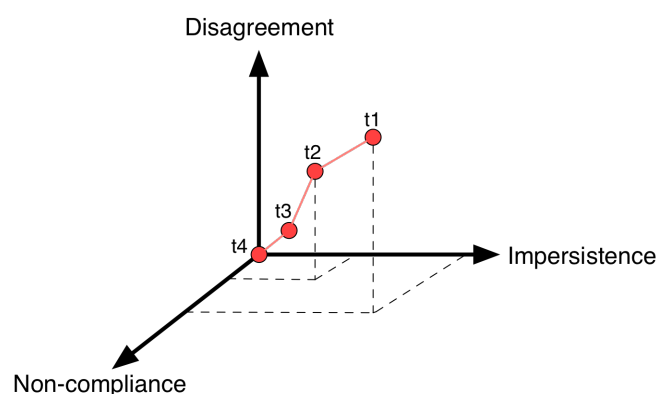


Figure 3.3: Three dimensional space exhibiting adherence evolving over time.

Both impersistence and non-compliance may be considered as degree measures (from 0 to 1) as per the discussion of compliance in section 3.2.4; maximal non-

compliance and impersistence occur when this degree measure goes to zero. A formal approach for determining a degree of impersistence or non-compliance is addressed in subsequent sections.

Disagreement, as with impersistence and non-compliance, may also be considered as a distance measure; however the notion of refinement also becomes relevant. In general, given two specifications, A and B , A may be said to refine B if and only if all behaviours/traces that satisfy A also satisfy B [63]. Then since both prescriptions and conscriptions are specifications, a conscription is a refinement of a prescription if all traces that are compliant with the conscription are also compliant with the prescription. Further, if a conscription is a refinement of a prescription, then there is zero disagreement between the prescription and the conscription.

3.3.3 Example

Consider Ms. Smith and her prescription for long-acting morphine described above. The prescription indicates she should *take 15 mg of long-acting morphine twice daily*, this is Ms. Smith's physician's recommendation. However, Ms. Smith is an individual who has autonomy and may decide to add additional constraints and refine the prescription. Perhaps she finds morphine affects her digestion unless she takes it with a snack or a meal, the conscription would be *take 15 mg of long-acting morphine twice daily with food*. In this case the conscription clearly refines the prescription, thus disagreement is zero.

Knowing this, consider a three day description for Ms. Smith. On the first day, she takes her second dose of morphine in the evening without food as she is too tired to prepare a meal; her *non-compliance* score is 0 - perfect, she is compliant with the *prescription*. However, her *impersistence* is non-zero, she failed to consume food with her medication. On the second day, Ms. Smith fails to administer her second dose of morphine; she is both *impersistent* and *non-compliant*. Finally, on the third day Ms. Smith takes both doses of morphine with food, all measures are zero. This evolution through time is captured in a three dimensional plot shown in Figure 3.4.

Section Summary

In summary, this section has taken the concepts of a prescription and compliance and placed them within an Adherence Interaction Model (AIM) which permits a more holistic definition of medication adherence. AIM relates the concepts of

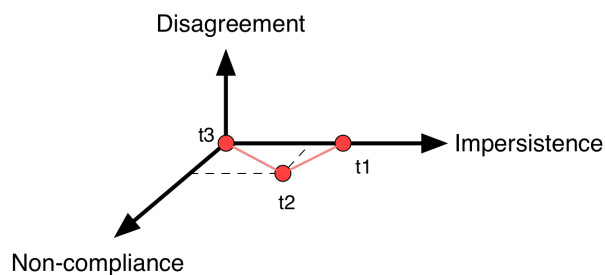


Figure 3.4: Three dimensional plot showing change in Ms. Smith’s behaviour in time.

prescription, conscription, and description via the distance measures of non-compliance, impersistence, and disagreement. Though discussed abstractly, no concrete definition for these measures was provided. Providing a formal definition of these concepts is the subject of the remainder of this chapter and Chapter 4.

3.4 Towards a Formal Definition of Adherence

Until this point, the concepts of compliance, persistence, and agreement have been informally defined and discussed as components of adherence; however, more rigorous (formal) definitions are required. This section begins the process of formalization by outlining a high level approach for formalization and providing specific requirements for aspects of the approach.

3.4.1 Formalizing Compliance

Non-compliance and impersistence are both measures of distance between medication plans, the prescription and conscription respectively, and the patient’s behavioural trace, the description. Given the similarity of these two measures, the same theoretical framework may be used formalize them. Therefore, for the remainder of this thesis, unless otherwise specified, the terminology “prescription” also refers to conscriptions, similarly “non-compliance” is may also refer to impersistence.

Action and Inaction

All medication taking behaviours are embedded in a timed context, thus it is reasonable for time to be a fundamental concept in a formal definition of compliance. Traces, provided by an observer device, have already been established as a means of encoding

past patient behaviours, the events reported in the trace correspond to *actions* in time taken by the patient. Conversely, from the perspective of medication consumption, every instant in time that is not associated with an action represents an *inaction*. One may “execute” an inaction by not performing any activities associated with the medication in question, e.g. not administering a pill. If the observer device were to periodically sample the patient’s behaviour it would be able to capture both actions (occurring within the last sampling period) and inactions (nothing occurred within the last sampling period).

A prescription may be interpreted to determine when to and when not to carry out actions. Each action/inaction may be scored independently for compliance. It follows that evaluation of overall compliance amounts to stepping through time and evaluating the compliance of each action or inaction. It is interesting to note that this approach results in the current action/inaction’s compliance being dependent on the previous actions or inactions that have occurred, but not future actions/inactions - from a systems engineering perspective the evaluation is said to be *causal*.

For example, Ms. Smith may administer 6 mg of wafarin at 11:30, this is an action, and she should only carry this action out if she has not yet taken a dose during the current day. However, once Ms. Smith has administered her warfarin she should not administer further doses, only inaction is permissible until the next day when she is again expected to administer another dose. Here, the degree of compliance of an inaction is predicated on whether an administration action has occurred in the near past. Further, if Ms. Smith gets to the end of the current day and has not yet administered warfarin then any subsequent inactions, such as not administering her medication, may be viewed as non-compliant.

The compliance of each action or inaction in a trace may be determined by an evaluator device, as discussed above in Section 3.2. In addition to accepting a prescription and behavioural trace as input, the evaluator device must also accept an instant in time at which to evaluate the compliance. This idea of an action or an inaction occurring and being evaluated at every instant is fundamental to the approach proposed by this thesis for describing prescriptions and compliance.

An underlying assumption of this action/inaction model, as presented in this thesis, is that all actions are *atomic*. An action, administration action or otherwise, can be associated with a single point in time at which it occurred, i.e. actions cannot be spread out across multiple points in time. Demanding atomicity simplifies the underlying mathematics for compliance measurement and is not unreasonable given

that many medications in a primary care/outpatient setting are administered via methods that are more or less instantaneous.

Measures of Compliance

Given that it is possible to determine the compliance of each action/inaction in a trace, two possible measures of compliance may be considered: *instantaneous compliance* and *cumulative compliance*.

Instantaneous compliance is the compliance of each action/inaction in the trace. It represents the direct output of the evaluator device and may be considered a function of time.

Cumulative compliance, also a function of time, is a measure of compliance up to a point in time. It is calculated as the sum of instantaneous compliance divided by the total time elapsed. Cumulative compliance may be a windowed measure in the sense that it is an aggregate measure starting from some point in time and ending at another point in time, e.g. the compliance for the last month.

The relationship between instantaneous and cumulative compliance is not unlike the relationship between a probability density function (PDF) and cumulative density function (CDF) used in statistics. The PDF represents the probability of a particular value being selected and the instantaneous compliance gives the compliance of a particular action/inaction; the CDF represents the probability of a value less than or equal to a particular value being selected and the cumulative compliance measure represents the compliance of all actions/inactions up to a point in time.

The motivation for two measures of compliance stems for the need to simultaneously understand moment-to-moment compliance (to support adherence decision support applications) and to also understand longer-term trends in a patient's adherence. For example, a decision support application may make recommendations to patients regarding administering the next dose of medication based on the effect of instantaneous compliance, while a patient or care provider may wish to have a "snapshot" of compliance over a period of time represented as a single number or a trend over time that abstracts away the moment-to-moment details.

Interpreting Prescriptions

Until this point, the prescriptions have been discussed as specifications for a patient's behaviour. However, for the purposes of formalization, it is necessary to distinguish

between a prescription that was authored in an informal language or notation (e.g. English), denoted as the *authored prescription* (AP), and a prescription expressed in a formal language.

The author of a prescription may make assumptions about the prescription's interpretation which are not explicitly stated by the AP. For example, Ms. Smith's warfarin prescription, which is expressed in natural language, has an administration frequency of *once per day* which may be interpreted as: *once per calendar day* or *once every 24 hours*; both are valid interpretations in the absence of additional information, such as the substance being consumed. However, the first interpretation is potentially dangerous given the substance is warfarin, a blood thinner. Interpreting once per day as *once per calendar day* would result in a trace with an administration at 23:59 and another administration at 00:01 being considered compliant. The phenomena of readers misinterpreting an ambiguous specification without realizing their mistake has been referred to in requirements engineering literature as *unrecognized disambiguation* [48].

To avoid ambiguity, an AP must be interpreted in the context of additional information to form an *interpreted prescription* (IP) which is expressed in a language with a formal syntax and semantics. The process of creating an IP from an AP is called *Prescription Interpretation*.

In addition to the original AP, the process of Prescription Interpretation may use as input:

- additional information sourced from the prescriber, not initially provided in the prescription,
- additional information sourced from the dispensing pharmacist,
- expert consensus sourced by polling a population of medication experts (physicians, pharmacists, nurses, medical researchers, etc.),
- individual expert opinion sourced by interviewing one expert about the prescription(s) in great detail,
- and medical literature such as research studies and/or drug monographs.

Additional information from these sources may be incorporated into an IP to remove ambiguity and obtain a clinically accurate specification.

Architecture

Combining the elements discussed thus far the approach’s architecture shown in Figure 3.5 emerges. Prescriptions are interpreted using a number of input data sources. The resulting IP is expressed in a language which has a formal syntax and semantics. The IP is provided as input to the evaluator device, described in Section 3.2, which, given a trace provided by an observer device, generates a compliance score.

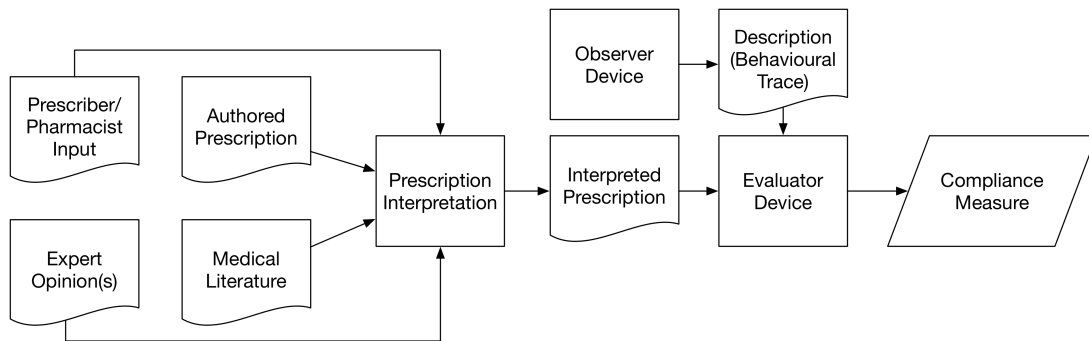


Figure 3.5: Diagram describing the high level architecture of the proposed approach to formalizing compliance.

3.4.2 Formalizing Agreement

The previous section presented a high level approach for formalizing compliance (and persistence); however, as AIM indicates, the concept of agreement is central to the definition of adherence. Disagreement is the “distance” between the prescription, the provider’s plan, and the conscription, the patient’s plan. A technique called *trace enumeration* is used to measure the distance between the two plans.

First, IPs for the prescription and conscription are generated as discussed above. Then the traces that satisfy each specification enumerated. The agreement, a real number between 0 and 1, is computed as:

$$\frac{\text{number of traces that satisfy both the prescription and conscription}}{\text{number of traces that satisfy the conscription}}$$

Disagreement is computed as $1 - \text{Agreement}$. Notice that when a conscription refines a prescription (e.g. conscription *once per day with food* and prescription: *once per day*) the agreement will become 1.0 since the number of traces that satisfy both the conscription and prescription will be equal to the number of traces that satisfy only the conscription.

This definition of agreement also captures the inherent power relation between a provider and the patient in the sense that agreement is a one-way measure; the definition only considers to what degree the patient's plan agrees with the provider's plan. In the event that the provider disagrees with a prescription, then either: 1) the patient has disagreed with the prescription, which is captured by the definition; or 2) the provider was not restrictive enough when crafting the prescription. In the second case, the provider must consider revising their prescription to be more restrictive. Of course, this is not to say that providers and patients may not iterate on a prescription to find an agreeable arrangement, indeed this is desirable in cases where there is initially disagreement. The idea of iteration and improving agreement follows from the previous discussion AIM in Section 3.3 wherein disagreement may be a quantity that varies with time.

3.4.3 Requirements

The architecture presented in Figure 3.5 has three core components/processes: 1) the process of Prescription Interpretation, 2) the Observer Device, and 3) the Evaluator Device; in this section requirements for each of these are presented. These requirements are derived from the discussion of AIM in Section 3.3, the high level approach for formalizing compliance presented directly above, and ten sample prescriptions provided by a domain expert.

Prescription Interpretation Requirements

Prescription Interpretation is the task of using additional contextual information to reduce ambiguity and address implicit assumptions in the original AP, the result is an IP with a more precise meaning. The resulting IP specification is expressed in a language with a formally defined syntax and semantics. Some of the requirements below pertain to the process of prescription interpretation, while others pertain to the specification language used to express IPs.

R1-1 Language Concepts The specification language must be capable of describing a breadth of concepts regarding prescriptions, the nature of the requirements may vary depending on the application. This thesis imposes the following breadth requirements which were derived by examining ten prescriptions provided by a domain expert.

1. The language shall be capable of specifying the substance to be consumed.
2. The language shall be capable of specifying the dose of substance to be consumed, doses should be expressed as ranges with units: a minimum amount, maximum amount, and a unit, e.g. *10 - 20 mg*.
3. The language shall be capable of specifying the duration of the prescription.
4. The language shall be capable of specifying the timing of administration, including:
 - (a) specifying a point in time, to a suitable degree of precision, to administer the medication, e.g. *at 12:05 on Monday*,
 - (b) specifying a PRN (as needed) administration regime,
 - (c) specifying a minimum and maximum number administrations per period of time, e.g. *3 - 6 times per day*,
 - (d) and specifying a minimum and maximum interval between subsequent administrations, e.g. *20 - 24 hours between doses*.
5. The language shall be capable of specifying a minimum and maximum time an administration should occur from another event type, e.g. *take within 30 minutes of a meal*.
6. The language shall be capable of expressing prescriptions that have increasing or decreasing dosing in time, e.g. *take 5 mg once daily, increase by 5 mg per week for 10 weeks*.

R1-2 Language Fuzziness The specification language shall be capable of specifying “fuzzy” inputs or degrees of compliance for aspects of the prescription such as timing or dose amount.

R1-3 Language Formality The specification language shall have a formally described syntax and semantics such that a “sentence” expressed in the language has a precise meaning.

R1-4 Language Consumability The specification language shall be in a format that permits consumption by a computing device, e.g. a reasonable implementation of the hypothetical Evaluator Device.

R1-5 Primary Input An authored prescription shall be the primary input to the specification synthesis process.

R1-6 Additional Inputs The specification synthesis process shall be capable of including additional sources of information to provide evidence for assumptions that may have been implicit in the prescription but are made explicit in the resulting prescription specification.

Observer Device Requirements

As per the high-level approach architecture in Figure 3.5, the Observer Device provides a behavioural trace to the Evaluator Device. The following requirements pertain to the behavioural trace language used to communicate between the Observer Device and the Evaluator Device.

R2-1 Observable Factors The behavioural trace language must be capable of describing the observable factors required by the application in question. For the purpose of this thesis, the behavioural trace language must be capable of describing:

1. medication administration actions, which include the following observable factors: substance, dose amount and unit, and time of administration,
2. food consumption actions describing the time a meal was consumed,
3. awake actions describing the time the patient wakes up,
4. and sleep actions describe the time the patient goes to sleep.

R2-2 Language Consumability The behavioural trace language must be in a format that permits consumption by a computing device, i.e. a reasonable implementation of the Evaluator Device.

Evaluator Device Requirements

Given a prescription and a trace specified in appropriate languages, the degree of compliance may be evaluated. The following describe requirements for the Evaluator described in section 3.2.2 and in Figure 3.5.

R3-1 Evaluation Given a prescription and a trace, the device shall determine a compliance score for the trace.

R3-2 Degree of Compliance The compliance determined by the device shall be a real number between 0.0 and 1.0 representing the degree of membership in a fuzzy “compliance set”.

R3-3 Platform Independent The Evaluator Device’s logic shall be defined in a manner that is independent of programming languages and a specific platform; this will ensure the implementation of the Evaluator Device is not restricted based on the features of a single language/platform.

R3-4 Implementation Simplicity An implementation of the Evaluator Device (as a computer program) shall not use any non-generic programming language features or libraries, and remain as true as possible to the original mathematical definition.

R3-5 Implementation Accessibility An implementation of the Evaluator Device shall be done in a manner that could permit access to the formalism from other platforms; i.e. the implementation can be integrated into an Internet of Things type system.

R3-6 Clinically Representative The result of behavioural evaluation produced by the formalism shall be representative of those produced by a human clinician.

Notes Regarding Requirements

A few requirements are intentionally excluded by the listing above. First, the usability of the languages is not directly considered. While usability is a key factor and must be considered for clinical use, the language’s expressiveness and formality is the focus of this work.

Second, route of administration is intentionally not considered; while route has clinical significance, its likely to be uniform for a prescription and thus is less relevant than concepts like timing, which may vary dramatically.

3.5 Scheduling Medication Administration

Given an understanding of how to measure adherence, technological interventions have the potential to improve compliance [41, 43]. As described above, an Internet of Things (IoT) ecosystem has the potential to observe a patient’s behaviour; a reasonable next step is to use that ecosystem to affect patient behaviour by providing context-aware reminders and patient decision support. This section describes the application of the approach to measuring compliance to the problem the problem scheduling medication administration.

3.5.1 Scheduling Problem

The following scenario is based upon the architectures proposed in [32, 37, 43, 64]. Suppose Ms. Smith lives within a well connected IoT environment that is capable of determining:

1. when she consumes her medication,
2. when she consumes food,
3. and when she goes to sleep and wakes up.

These are determined by combining data from several sensors which collectively make up the Observer component of the architecture depicted in Figure 3.5. Ms. Smith has a “Smart Pill Bottle” device that records when she takes her medication. Further, Ms. Smith regularly uses a mobile device (e.g. smart phone) and a connected wearable device (e.g. smart watch) that is usually on her person. Assume Ms. Smith’s original authored prescription information, from her physician’s and pharmacist’s information systems, is accessible via a secured channel.

Ms. Smith may have trouble remembering to administer her medication and could benefit from support provided by her IoT environment. The problem is stated as:

given a prescription and a behavioural trace, determine the next time to administer the medication to achieve optimal compliance.

Importantly, the proposed administration time should account for observable factors relevant to the prescription, e.g. meal times for Ms. Smith’s ibuprofen prescription.

3.5.2 Approach to Scheduling

Applying the approach for measuring compliance described in Section 3.4.1 leads to the control loop architecture seen in Figure 3.6.

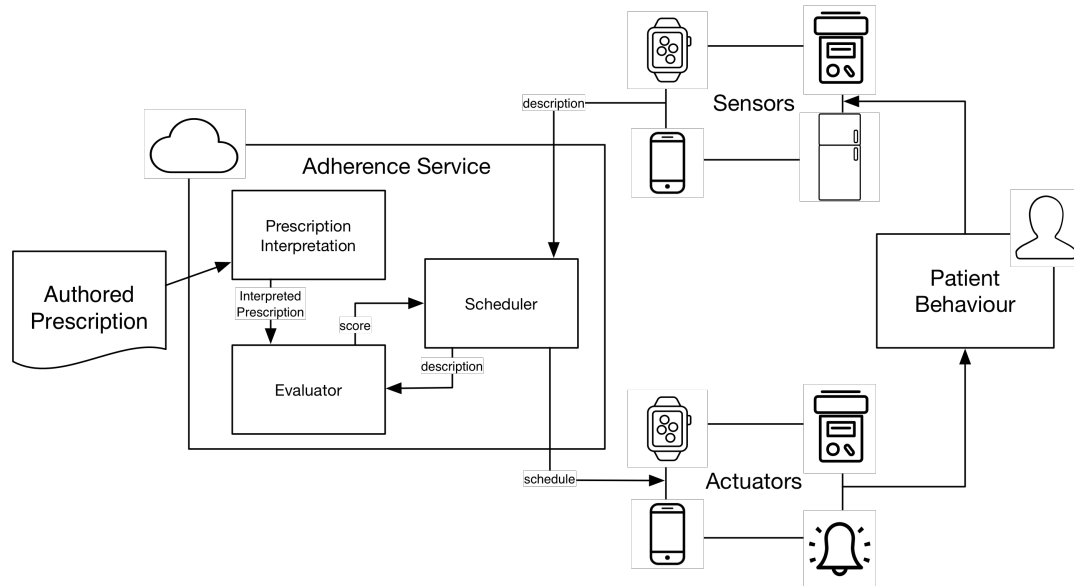


Figure 3.6: Control loop architecture for scheduling medication administration.

An Adherence Service, possibly hosted in the “cloud”, attempts to actuate Ms. Smith’s compliance by means of reminders and decision support delivered via her wearable devices, mobile phone, and pill bottle. Ms. Smith’s behaviour is sensed by the same devices (and her connected refrigerator for meals), the description is provided to the Adherence Service. The description, as a stream of actions from sensing devices, is provided to a Scheduler component of the Adherence Service which uses an optimization algorithm and the Evaluator to find the optimal next administration time. As above, the Interpreted Prescription, which is the input specification for the Evaluator, is interpreted based on the Authored Prescription and additional information such as expert opinion (not shown in Figure 3.6).

3.5.3 Reminders and Decision Support

The approach above can be used to generate a schedule that, if followed by the patient, will result in an optimal compliance score. Such a schedule can be used to generate system *states* which may be coupled with appropriate actuator actions (e.g. reminders, messages, etc.) delivered through the IoT ecosystem, the states are:

- **Green** - Administer medication, compliance is optimal.
- **Yellow** - Administer medication, compliance is non-optimal.
- **Red** - Do not administer the medication.

One may imagine an indicator light on a pill bottle device or wearable device used as a reference to quickly ascertain the state. This three state model has two obvious advantages: it is simple, and it parallels a traffic light model many users are familiar with. Indeed, Varshney has suggested a similar state machine [37].

While the **Yellow** state indicates it is acceptable to administer the medication, it does not provide additional context such as whether it is “too early” or “too late”, i.e. should Ms. Smith wait to administer the dose or immediately administer dose? Additional information regarding the administration timing is required and may be communicated through the IoT ecosystem textually, such as on a screen on Ms. Smith’s pill bottle device, for example:

1. *Take your next dose with dinner.*
2. *Take your next dose in 2 hours.*
3. *You are late, take your next dose immediately.*

The first two items describe future actions, i.e. actions which have been scheduled and whose time has not yet passed; these are easily accomplished by the approach for scheduling presented in this thesis. The last item in the list above requires that system to react to a situation in near real-time, which is more complicated as it requires the specification of “miss-behaviours”; at this time, this is beyond the scope of this thesis, which is restricted to measurement of compliance and generation of optimal schedules.

3.6 Chapter Summary

In this chapter compliance was introduced as the degree to which a patient’s behavioural trace corresponds with a prescription. Formalization of the relationship between prescriptions, traces, and compliance provides formal definition of the language in which prescription specifications are expressed. Additionally, these concepts were placed within the context of an Adherence Interaction Model that describes how prescriptions - provider plans, conscriptions - patient plans, and descriptions - patient

behaviour are related. In doing so, the definition of adherence, which is a concept that entails agreement, persistence, and compliance can be recovered.

A high-level approach to formalizing adherence was presented in which the original authored prescription (a informal set of instructions) is interpreted in the context of additional information to produce a formalized prescription specification for a patient's behaviour. The specification is then used by an Evaluator Device to evaluate a behavioural trace provided by an Observer Device. As a first step towards formalization, requirements for each of these system components/processes were provided. Finally, a system for generating administration schedules and for supporting compliance was presented.

Chapter 4 formalizes the concepts presented in this chapter and describes an implementation of the approach for measuring compliance and scheduling medication administration.

Chapter 4

Formalizing Adherence

This chapter provides formal definitions of the approach described in the preceding chapter. This includes: a formal definition of a behavioural trace; a language for expressing prescription specifications; and the definition of a compiler for converting a prescription specification into an evaluator function. Then, an approach for measuring non-compliance of a trace and scheduling the next administration of a medication are presented. The chapter is concluded by an illustrative example demonstrating the application of the formalism to a hypothetical patient scenario.

4.1 Traces

4.1.1 Observable Factors

First, recall the definition of an observable factor: an aspect of behaviour that may be tracked by a hypothetical observer/device. Allow β to denote an observable factor, further β is a set containing all possible values that the observable factor could take. For example, if time was an observable factor and time was measured at intervals of one second then $\beta = \{0, 1, 2, \dots\}$ for some arbitrary starting time, perhaps the Unix Epoch, 00:00 on Jan 1st 1970.

4.1.2 Actions and Inactions

Actions and inactions, in the sense discussed in Chapter 3, are described as a collection of observations or instances of observable factors. Formally, for factors $\beta_1, \beta_2, \dots, \beta_n$, an action/inaction is a tuple (b_1, b_2, \dots, b_n) where each value $b_i \in \beta_i$. An action/inaction,

e , is then an element of the set of all possible tuples, i.e. $e \in \beta_1 \times \beta_2 \times \dots \times \beta_n$.

For convenience, names are given to elements of the tuple, and given an action/inaction, accessed using a dot notation. This is formalized by using a function to map indices of the tuple to names. For example, if time and dose amount are observable factors then a mapping function would be: $f = \{(\text{“time”}, 0), (\text{“dose”}, 1)\}$, then for an event $e = (1000000, 10)$ $e.\text{time}$ would access the zeroth element of the tuple and return 1000000.

In the case of an inaction, some observable factors will have null values to indicate that there was no recorded value, this is denoted as \perp . Some observable factors, such as the time of inaction, will always have values associated with them. Formally, allow $\perp \in \beta$ if a null value is permissible for that observable factor. For example, an inaction tuple, where observable factors are time and dose amount, might be $e_i = (1000000, \perp)$.

4.1.3 Traces

A trace is a tuple, $\tau = (S, R)$ containing: a set of actions and inactions, S , and a total ordering relation, R . A sequence of actions and inactions, τ , is defined by applying the ordering relation to the elements of the set. For notational convenience allow e_i to denote the i^{th} element of a trace τ .

Though any set of observable factors may tracked; however in practice, time is almost always a factor of interest. From here on time is assumed to be a factor for all traces, the ordering relation, for all e_1, e_2 in the set S , is:

$$R(e_1, e_2) = e_1.\text{time} \leq e_2.\text{time}$$

Sampling

Traces are generated by the observer device which samples the patient’s behaviour with an arbitrary sampling period T ; consequently, a trace may be viewed as a sequence of samples. The observer device will report an action at sample k if the time the action occurred at is between the time of sample $k - 1$ and k , i.e. $(k - 1)T < t_e \leq kT$, where t_e is the true recorded time of the action. If no action was observed, the observer will report an inaction at sample k .

For convenience, a trace may be represented only by actions that were reported by the observer and the existence of inactions are inferred. If actions exist at sample k

and sample $k - 2$ and nothing at $k - 1$ is reported, an inaction tuple, with appropriate null values, may be inferred at sample $k - 1$.

Sample Trace

As an example, consider the trace shown in Figure 3.1 which was produced by an observer device with sampling period $T = 3600$ seconds (1 hour). This trace expresses two observable factors, time and dose - other factors such as substance are omitted for clarity; furthermore, the trace does not show inactions.

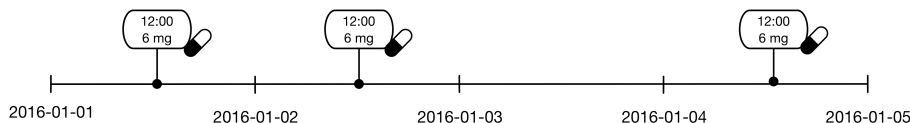


Figure 4.1: Sample trace showing observable factors of time and dose.

Each action is a tuple with time as the first element and dose as the second. The corresponding set of actions is $S = \{(1451649600, \text{“6 mg”}), (1451736000, \text{“6 mg”}), (1451908800, \text{“6 mg”})\}$ where time is given in seconds since the epoch - January 1st 1970. Ordering the set of actions S using the relation R , given above, yields the sequence:

$$(1451649600, \text{“6 mg”}), (1451736000, \text{“6 mg”}), (1451908800, \text{“6 mg”})$$

4.2 Prescriptions

A prescription is a specification for a medication taking behaviour; to precisely communicate a prescription a formal language is required. This section formally describes the form of such a language. In the approach architecture discussed in Chapter 3, this is the output language of the Prescription Interpretation process and input language of the Evaluator Device.

In this section, the syntax of the language is given as a Neighborhood Controlled Embedding (NCE) graph grammar. For brevity, only the core of the language is presented; an extended grammar may be found in Appendix A. The semantics (meaning) of the language are defined via a denotational definition in section 4.3. The notation used here was introduced in Section 2.1.

4.2.1 Syntax

The grammar of the language is a system $G = (\Sigma, \Delta, P, S)$ where:

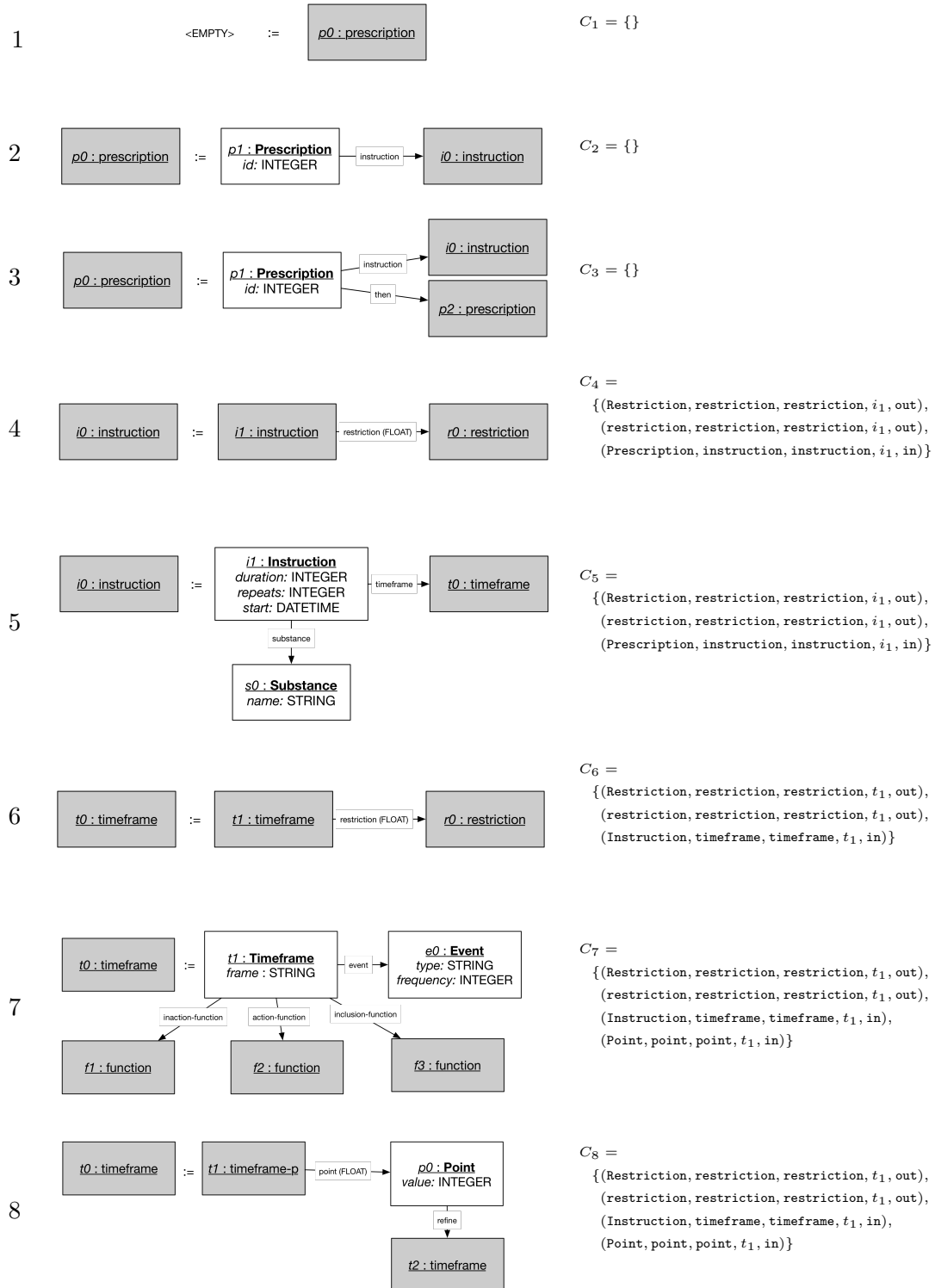
- $\Delta = \{\text{prescription, instruction, timeframe, timeframe-p, point, function, restriction}\}$ - the set of non-terminal node labels.
- $\Sigma = \{\text{Prescription, Instruction, Timeframe, Point, Restriction, Function, Event}\}$ - the set of terminal node labels.
- $\Gamma = \{\text{instruction, then, restriction, timeframe, event, action-function, inaction-function, inclusion-function, point, function, substance}\}$
- P - the set of graph production rules shown below.
- $S = \text{EMPTY}$ - the start graph.

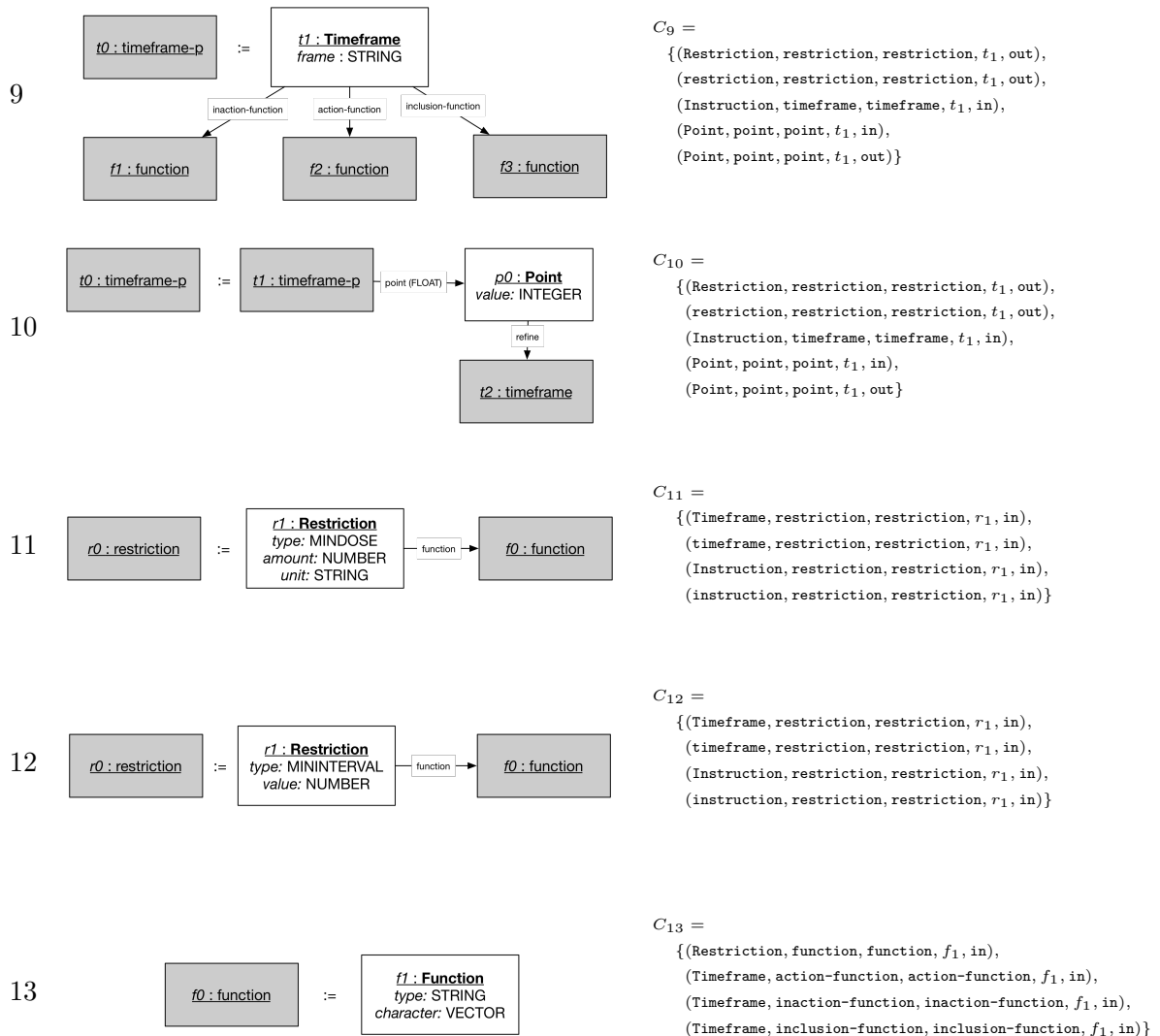
Each production rule in P is a tuple $p_i = X_i \rightarrow (C_i, D_i)$ where:

- $X_i \in \Delta$ (a non-terminal) - the left-hand side of the production rule.
- D_i is right-hand side of the production rule.
- C_i is a connection relation describing how the nodes in D_i should be embedded within the context of X_i .

A subset of the production rules for the grammar is given below. In the rule definitions non-terminal nodes are denoted as shaded boxes and terminal nodes have a white background with bolded labels.

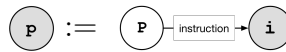
Table 4.1: Subset of grammar for prescription specification language.





The grammar given above takes some liberties with notation. The directed edges between nodes that have both a label and value, as in the right hand side of production Rule 10, should strictly be represented by a labeled edge, connecting to an intermediary node which contains the value, which connects to the desired child node. However, such notation becomes cumbersome and would make the grammar more difficult to understand.

For convenience, each production rule may be described using a condensed notation. Nodes are identified by their type and an additional numerical identifier where required and attributes are dropped. This notation will be used for the definition of the language's semantics. For example, Rule 2 above could be expressed as:



4.2.2 Attributes

In addition to the strict form of the language, there are attributes of interest, some of which have values that may be derived from the structure of a graph generated by the grammar presented above, others are defined as the production rules are applied; *attribute grammars* are used to formally describe these attributes of interest. Two complementary techniques are used below, both are adapted from the approach given by Knuth in [53].

Node Attributes

Most terminal nodes in the grammar have attributes of some sort, for example, **Instruction** nodes contain an integer value that represents the duration. To formalize the existence of these attributes, first assume that as nodes are generated by the grammar they are given a unique global identifier n_i , for the i^{th} node generated, denote the set of all node identifiers N .

Then, for each possible node attribute, a , define an attribute relation named: r_a ; $r_a \subseteq N \times \text{Type}(a)$, where $\text{Type}(a)$ is the type of attribute as denoted in the grammar, e.g. Integer, String, Datetime, etc. Call the set of all attribute relations in the grammar R . Initialize all relations in R as empty.

As nodes are generated by the application of graph production rules, values chosen for attributes of the generated nodes, perform the following operation:

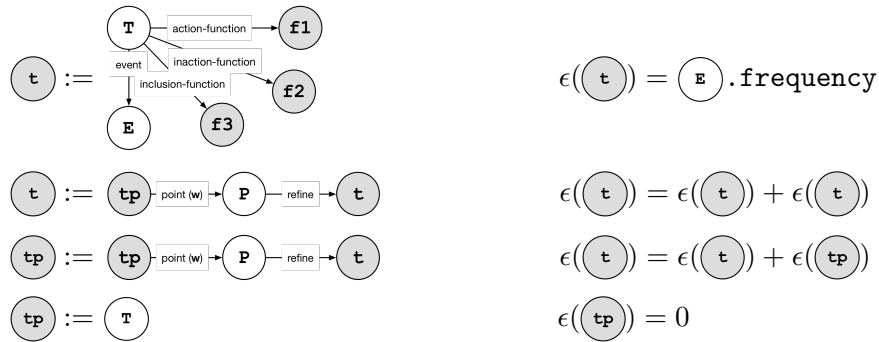
$$R' = \bigcup_{r \in R} (r \cup (n_i, v))$$

That is, create a new set R' with each relation from R having a new tuple added relating the new node identifier, n_i , to the value chosen for that attribute, v . Note, if the attribute is not defined for node n_i , then v is denoted as undefined, $v = \perp$.

In fact, these relations may be considered functions defined over all nodes in N , for each input value they contain exactly one output. Then, given an attribute, a , and a node identifier, n_i , the value of the attribute may be determined via the function invocation: $r_a(n_i)$. For convenience, and in keeping with the conventional notation used in many programming languages, allow the node-dot-attribute to be used, i.e. $n_i.a \equiv r_a(n_i)$.

Expected Action Counts

Given a Timeframe node it is useful to know the number of actions expected to occur within the timeframe, call this value ϵ . Note that due to the recursive nature of the graph, counting events requires recursively counting events for sub-timeframes. The following *synthesized* attribute grammar is used to determine the expected number of events:



Above, the notation (tp) is used to refer to a non-terminal node labeled `timeframe-p` to indicate a timeframe that has been refined by a point.

4.3 Semantics

A syntax for prescription specification language, which is the input for a compliance evaluator device, was given above. This section presents a denotational semantics for the language; an introduction to denotational semantics is provided in Section 2.1.

4.3.1 Semantic Algebras

Semantic algebras for the denotational definition are provided. These define domains, e.g. observable factors, actions, and traces, that represent concrete concepts or entities. In essence, these ground the entire mathematical formulation in reality.

First, Booleans and real numbers are given the usual definitions. Several utility operations are also defined over the real numbers.

I. Booleans

Domain: $\{\text{true}, \text{false}\} = \mathbb{B}$

Operations:

$$\neg : \mathbb{B} \rightarrow \mathbb{B}$$

$$\wedge, \vee : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

II. Numbers

Domain: \mathbb{R}

Operations:

$$\geq, \leq, >, < : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B}$$

$$*, +, -, / : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$$

$$\mathbf{saturnate} : \mathbb{R} \rightarrow \{x \in \mathbb{R} : x \geq 0.0 \wedge x \leq 1.0\}$$

$$\mathbf{saturnate} \doteq \lambda x. (x \geq 1.0 \rightarrow 1.0 \sqcap x \leq 0.0 \rightarrow 0.0 \sqcap x)$$

$$\mathbf{ramp}, \mathbf{n-ramp} : \mathbb{R} \rightarrow \mathbb{R} \rightarrow \mathbb{R} \rightarrow \mathbb{R}$$

$$\mathbf{ramp} \doteq \lambda x_1. \lambda x_2. \lambda x. ((\leq x x_1) \rightarrow 0 \sqcap (\geq x x_2) \rightarrow 1.0 \sqcap (* x (/ 1 (- x_2 x_1))))$$

$$\mathbf{n-ramp} \doteq \lambda x_1. \lambda x_2. \lambda x. ((\leq x x_1) \rightarrow 1.0 \sqcap (\geq x x_2) \rightarrow 0.0 \sqcap (* x (/ 1 (- x_1 x_2))))$$

$$\mathbf{step} : \mathbb{R} \rightarrow \mathbb{R} \rightarrow \mathbb{R}$$

$$\mathbf{step} \doteq \lambda x_0. \lambda x. ((\geq (- x x_0) 0) \rightarrow 1.0 \sqcap 0.0)$$

Next, the conceptual definitions of observable factors and events (actions or inactions) are formalized. Here **time**, **dose**, **substance**, and **action type** are used. There are no operations directly associated with observable factors. Events are defined as a combination of the given observable factors. The “dot” operator is defined for convenience.

III. Observable Factors

Domains:

$$\mathbf{time} = \beta_0 \doteq \mathbb{N}$$

$$\mathbf{dose} = \beta_1 \doteq \{\mathbb{R}, \perp\}$$

$$\mathbf{substance} = \beta_2 \doteq \{\text{“aspirin”}, \text{“warfarin”}, \text{“penicillin”}, \dots, \perp\}$$

$$\mathbf{type} = \beta_3 \doteq \{\text{“admin”}, \text{“food”}, \text{“awake”}, \text{“asleep”}, \perp\}$$

Operations:

—

IV. Events

$$\text{Domain: } e \in \mathbf{Event} \doteq (\beta_0 \times \beta_1 \times \beta_2 \times \beta_3) = (\mathbf{time} \times \mathbf{dose} \times \mathbf{substance} \times \mathbf{type})$$

Operations:

(“dot”) \cdot : $Event \rightarrow (\beta_0 \cup \beta_1 \cup \beta_2 \cup \beta_4 = \mathbf{time} \cup \mathbf{dose} \cup \mathbf{substance} \cup \mathbf{type})$

(“dot”) $\cdot \doteq \lambda\beta_i.\lambda e.e(i)$ (returns the i^{th} element of event tuple e , expressed as: $e.b_i$)

Dose and substance apply to “admin” type events, but not to other types such as “awake” events; for these event types the values of dose and substance are set to \perp . Similarly, \perp is used to denote an inaction type.

Below, a semantic algebra for traces, which are sets of events coupled with an ordering relation, is given; the ordering relation orders events based on their time. A single trace is denoted as τ . Additionally, utility operations are defined, each of which has a short description below its definition.

V. Trace

Domain: $Traces \doteq (S, R)$ where:

$S \doteq \mathbb{P}(Events)$

$R : Events \rightarrow Events \rightarrow \mathbb{B} = \lambda x.\lambda y.(\geq x.time\ y.time)$

Operations:

action? : $\mathbf{time} \rightarrow Traces \rightarrow Events \cup \{\perp\}$

determine if an action occurred in the trace T at time t .

action? $\doteq \lambda t.\lambda\tau.\left(\right.$
 let $s \doteq S_R$ in (for S_R the ordered sequence of events in S using relation R)
 $\left.((= |s| 0) \rightarrow \perp \ [] (= s(0).time\ t) \rightarrow s(0) \ [] ((\mathbf{action?}\ t)\ (S - s(0), R)))\right)$

admin? : $\mathbf{time} \rightarrow Traces \rightarrow \mathbb{B}$

determine if the event at time t is an “admin” type event.

admin? $\doteq \lambda t.\lambda\tau.\left(\right.$
 let $e \doteq ((\mathbf{action?}\ t)\ \tau)$ in
 $\left.((= e\ \perp) \rightarrow \mathbf{false} \ [] (= e.type\ \mathbf{“admin”}))\right)$

next : $\mathbf{time} \rightarrow Traces \rightarrow Events \cup \{\perp\}$

finds the event occurring after time t in trace τ , if no event is found returns \perp .

next $\doteq \lambda t_0.\lambda\tau.\left(\right.$
 let

$s \doteq S_R$ in (for S_R the ordered sequence of events in S using relation R)

$t_{end} \doteq e_{end} \cdot \mathbf{time}$ where e_{end} is the last action in the trace τ .

$e \doteq ((\mathbf{action?} t_0) \tau)$

in

$((> t_0 t_{end}) \rightarrow \perp \parallel (= e \perp) \rightarrow (\mathbf{next}(+ 1 t_0) \tau \parallel e))$

nil? : $Events \cup \{\perp\} \rightarrow \mathbb{B}$

determines whether the provided Event is has a null/nil value.

nil? $\doteq \lambda e. ((= e \perp) \rightarrow \mathbf{true} \parallel \mathbf{false})$

filter : $(Events \rightarrow \mathbb{B}) \rightarrow \tau \rightarrow \tau$

returns a sub-trace with events that satisfy the function.

filter $\doteq \lambda f. \lambda \tau. (\{e \in S : (\mathbf{f} e)\}, R)$

Finally, an algebra for frames (units of time) is provided. This representation considers seconds as the smallest meaningful unit of time for managing medication prescriptions, certainly a smaller unit could be used but given the context it is not clear what value this might add. Operations for frames are provided such that the **seconds** operation, which given a frame, returns the number of seconds in that frame.

VI. Frames

Domain: $Frames = \{\text{"second"}, \text{"minute"}, \text{"hour"}, \text{"day"}, \text{"week"}, \text{"month"}, \text{"year"}\}$

Operations:

child : $Frames \rightarrow Frames \cup \{\perp\}$

returns the next smallest frame type.

child $\doteq \lambda x. ($

$(= x \text{"second"}) \rightarrow \perp$

$\parallel (= x \text{"minute"}) \rightarrow \text{"second"}$

$\parallel (= x \text{"hour"}) \rightarrow \text{"minute"}$

...

seconds : $Frames \rightarrow \mathbb{N}$

returns the number of seconds in the given Frame.

seconds $\doteq \lambda x. ($

$(= x \text{ “second”}) \rightarrow 1$
 $\square (= x \text{ “minute”}) \rightarrow (* 60 (\mathbf{seconds} \text{ “second”}))$
 $\square (= x \text{ “hour”}) \rightarrow (* 60 (\mathbf{seconds} \text{ “minute”}))$
 \dots
 $\square (= x \text{ “year”}) \rightarrow (* 12 (\mathbf{seconds} \text{ “month”}))$

upper, lower : **time** \rightarrow **time** \rightarrow \mathbb{N} \rightarrow *Frames* \rightarrow **time**

returns the upper time bound and lower time bounds, respectively, of the v^{th} frame of type f after time t_1 .

upper $\doteq \lambda t_2. \lambda t_1. \lambda v. \lambda f. (- (+ t_1 (* (\mathbf{seconds} f) (+ v 1))) 1)$

lower $\doteq \lambda t_2. \lambda t_1. \lambda v. \lambda f. (+ t_1 (* (\mathbf{seconds} f) v))$

4.3.2 Valuation Functions

Functions, which utilize the semantic algebras above, may be associated with each of the grammar production rules; doing so gives a precise meaning to each of the nodes/structures production rules. Subsequent application of grammar production rules will build up a complete function that may be used to evaluate the compliance of a trace.

The first valuation function is associated with the grammar rule that generates a prescription graph node. It returns a function that takes as arguments a trace, τ and a time to evaluate, t . The function defers evaluation of the trace to the child instruction node. In cases where there is an additional Prescription attached via a **then** edge, the total compliance score from the instruction and additional Prescription are added together.

p : $\textcircled{\mathbf{p}} \rightarrow \mathbf{time} \rightarrow \textit{Traces}$

p[[$\textcircled{\mathbf{p}} \xrightarrow{\text{instruction}} \textcircled{\mathbf{i}}$]] = $\lambda t. \lambda \tau. ((\mathbf{i}[[\textcircled{\mathbf{i}}]]) \tau) t$

p[[$\textcircled{\mathbf{p}} \xrightarrow{\text{instruction}} \textcircled{\mathbf{i}} \xrightarrow{\text{then}} \textcircled{\mathbf{p}}$]] = $\lambda t. \lambda \tau. (+ ((\mathbf{p}[[\textcircled{\mathbf{p}}]]) \tau) t) (\mathbf{i}[[\textcircled{\mathbf{i}}]]) \tau) t$

Instructions

The instruction node, depending on the subsequent grammar production rules that were applied, conducts two types of computations. First, for any restrictions that are direct children of the instruction, the compliance of the trace at time t with respect to the restriction is evaluated; this is achieved by deferring to the valuation function of the restriction itself, the result is multiplied by the given weighting value, \mathbf{w} (indicating the importance of the restriction w.r.t to the overall prescription) and combined with the compliance value computed for other aspects of the prescription. This may be seen in the first valuation function for an instruction directly below.

Second, the effect of the action/inaction in the trace τ at the current time t is evaluated over all timeframes within the prescription. Here the “top level” timeframe, which is the direct child of the instruction, is used. This function defers to the valuation function of its child timeframe. The function may be seen in the second valuation function for an instruction directly below.

$$\begin{aligned}
 \mathbf{i} &: \textcircled{\mathbf{i}} \rightarrow \mathbf{time} \rightarrow \mathit{Traces} \\
 \mathbf{i}[[\textcircled{\mathbf{i}} \xrightarrow{\text{restriction } \mathbf{w}} \textcircled{\mathbf{r}}]] &= \lambda t. \lambda \tau. (+ ((\mathbf{i}[[\textcircled{\mathbf{r}}]] \tau) t) (* \mathbf{w} ((\mathbf{r}[[\textcircled{\mathbf{r}}]] \tau) t))) \\
 \mathbf{i}[[\textcircled{\mathbf{I}} \xrightarrow{\text{timeframe}} \textcircled{\mathbf{t}} \xrightarrow{\text{substance}} \textcircled{\mathbf{S}}]] &= \\
 \text{let} & \\
 \quad \mathit{size} &\doteq (\mathbf{seconds} \textcircled{\mathbf{t}}.\mathbf{frame}) \\
 \quad \mathit{n} &\doteq (/ (- t \textcircled{\mathbf{I}}.\mathbf{start}) \mathit{size}) \\
 \quad \mathit{end} &\doteq (+ \textcircled{\mathbf{I}}.\mathbf{start} (+ (* \mathit{n} \mathit{size}) (* \textcircled{\mathbf{I}}.\mathbf{repeats} (* \mathit{n} \mathit{size})))) \\
 \text{in} & \\
 \lambda t. \lambda \tau. \sum_{x=0}^{n-1} &(((\mathbf{t}[[\textcircled{\mathbf{t}}]] \\
 & (+ \textcircled{\mathbf{I}}.\mathbf{start} (* \mathit{size} (+ x 1)))) \\
 & (+ \textcircled{\mathbf{I}}.\mathbf{start} (* \mathit{size} x))) \tau) t
 \end{aligned}$$

Timeframes

There are two kinds of timeframe nodes in the grammar; this is required to distinguish between timeframes that have been refined by a point (and subsequently sub-

timeframes) and those that are “terminal” timeframes that are not refined. However, the valuation functions are almost identical, thus only the valuation function for a non-refined timeframe, \textcircled{t} , is given. The full set of valuation functions is provided in Appendix A.

As with instructions, timeframes first evaluate the satisfaction of child restrictions by deferring to the restriction’s valuation function and weighting the result accordingly. This may be seen in the first timeframe valuation function below.

Evaluation of the compliance of the action/inaction at time t in the trace is given in the second valuation function below, it may be considered in three parts:

1. The number of medication administration actions within the timeframe, up to and including the current time t , are counted. This is done by summing the degree of inclusion (via the timeframe’s inclusion function: $\textcircled{f3}$) of each event in the trace with the current timeframe. Thus, in cases where an action occurs just outside of a timeframe boundary (as determined by an inclusion function) a fractional value for the number of administration actions within the timeframe may be computed.
2. If an action occurred at time t in τ , then the compliance of the action is evaluated by combining the “magnitude” of non-compliance with “degree” of impact the action had on non-compliance within the timeframe. Here, magnitude is computed by comparing the actual number of administration actions that occurred, a in the definition below, with the expected number for that timeframe, \textcircled{t} . ϵ . Degree of impact is given by evaluating the action function, $\textcircled{f1}$, at time t .
3. If an inaction occurred at time t in the trace τ , then the inaction non-compliance, similarly to action compliance, is computed as the magnitude of non-compliance and the degree of impact the inaction had on the non-compliance of the current timeframe. Magnitude is calculated as above. Degree of impact is found by evaluating the inaction function $\textcircled{f2}$ at the current time t .

Finally, the timeframe may be refined by a point and sub-timeframe; this is captured in the third valuation function listed below. Here, the effect of the sub-timeframe is combined with the current timeframe by weighting the compliance score of the sub-timeframe. A point, \textcircled{p} , is used to determine which sub-timeframe is to be

considered, e.g. a point with value 12 within a timeframe with frame “DAY” would result in the sub-timeframe from 12 to 12:59 being evaluated for compliance.

$$\begin{aligned}
\mathbf{t} : \textcircled{\mathbf{t}} &\rightarrow \text{time} \times \text{time} \times \text{time} \times \text{Traces} \\
\mathbf{t} [[\textcircled{\mathbf{t}} \xrightarrow{\text{restriction (w)}} \textcircled{\mathbf{r}}]] &= \\
&\lambda t_2. \lambda t_1. \lambda t. \lambda \tau. (+ (((\mathbf{t} [[\textcircled{\mathbf{t}}]]) t_2) t_1) t) \tau) (* \mathbf{w} (((\mathbf{r} [[\textcircled{\mathbf{r}}]]) t_2) t_1) t) \tau) \\
\mathbf{t} [[\textcircled{\mathbf{T}} \xrightarrow{\text{event}} \textcircled{\mathbf{E}}, \textcircled{\mathbf{T}} \xrightarrow{\text{action-function}} \textcircled{\mathbf{f1}}, \textcircled{\mathbf{T}} \xrightarrow{\text{inaction-function}} \textcircled{\mathbf{f2}}, \textcircled{\mathbf{T}} \xrightarrow{\text{inclusion-function}} \textcircled{\mathbf{f3}}]] &= \\
\text{let} & \\
a \doteq \sum_{e \in \tau} & (* (((\mathbf{f} [[\textcircled{\mathbf{f3}}]]) t_2) t_1) e.\text{time}) ((\mathbf{step} t) (* -1 e.\text{time})) \\
\text{in} & \\
\lambda t_2. \lambda t_1. \lambda t. \lambda \tau. ((\mathbf{action?} t) \tau) \rightarrow & \\
(* (((\mathbf{ramp} \textcircled{\mathbf{T}}).\epsilon) (+ 1 \textcircled{\mathbf{T}}).\epsilon) a) (((\mathbf{f} [[\textcircled{\mathbf{f1}}]]) t_2) t_1) t) & \\
[] (* (* (((\mathbf{n-ramp} 0) \textcircled{\mathbf{T}}).\epsilon) a) (((\mathbf{f} [[\textcircled{\mathbf{f2}}]]) t_2) t_1) t) & \\
(\mathbf{nil?} ((\mathbf{next} t_2) \tau)) \rightarrow 1 [] ((\mathbf{step} ((\mathbf{next} t_2) \tau)) (* -1 t)) & \\
\mathbf{t} [[\textcircled{\mathbf{tp}} \xrightarrow{\text{point (w)}} \textcircled{\mathbf{p}} \xrightarrow{\text{refine}} \textcircled{\mathbf{t}}]] &= \\
\lambda t_2. \lambda t_1. \lambda t. \lambda \tau. (+ & \\
(((\mathbf{tp} [[\textcircled{\mathbf{tp}}]]) t_2) t_1) t) \tau) & \\
(* \mathbf{w} (((\mathbf{t} [[\textcircled{\mathbf{t}}]]) & \\
(((\mathbf{upper} t_2) t_1) \textcircled{\mathbf{p}}.\text{value}) (\mathbf{child} \textcircled{\mathbf{tp}}.\text{frame})) & \\
(((\mathbf{lower} t_2) t_1) \textcircled{\mathbf{p}}.\text{value}) (\mathbf{child} \textcircled{\mathbf{tp}}.\text{frame})) t) \tau) &
\end{aligned}$$

Restrictions

There are several types of restrictions that may be used to specify various behaviours. Here, the minimum dosing and minimum interval restrictions are presented. For the semantic definition of the remaining restrictions see Appendix A.

The minimum dose restriction, first valuation function below, evaluates the action at time t in the trace τ . If the action was an administration action then evaluation of the dose administered is deferred to a child function, $\textcircled{\mathbf{f}}$. Ideally, the function will be crafted such that a dose below the minimum amount is scored as non-compliant.

In the case where there is no administration action at time t non-compliance (due to dosing) is evaluated to zero.

The minimum interval restriction, the second valuation function below, evaluates intervals between doses within the range of time t_1 to t_2 . Specifically, if there is an action at time t in the trace τ then and there is a previous action within the time range, then the distance between current time and the previous administration time is passed a function which will determine the non-compliance of the interval. Naturally, if the interval between the two doses is small then the function will return some non-compliance score. If there is no action at time t then non-compliance, for this restriction, is zero.

$$\begin{aligned}
\mathbf{r} : \textcircled{\mathbf{r}} &\rightarrow \text{time} \rightarrow \text{time} \rightarrow \text{time} \rightarrow \text{Traces} \\
\mathbf{r}[[\textcircled{\text{Mid}} \xrightarrow{\text{function}} \textcircled{\mathbf{f}}]] &= \lambda t_2. \lambda t_1. \lambda t. \lambda \tau. \\
&((\mathbf{admin? } t) \tau) \rightarrow ((\mathbf{f}[[\textcircled{\mathbf{f}}]] \textcircled{\text{Mid}}.\text{amount}) ((\mathbf{event } t) \tau).\text{dose}) \square 0 \\
\mathbf{r}[[\textcircled{\text{MidI}} \xrightarrow{\text{function}} \textcircled{\mathbf{f}}]] &= \lambda t_2. \lambda t_1. \lambda t. \lambda \tau. \\
&(\mathbf{not } (\mathbf{nil? } (((\mathbf{previous } t) \tau) t_1) t_2))) \rightarrow \\
&(* ((\mathbf{action? } t) \tau) (\mathbf{f}[[\textcircled{\mathbf{f}}]] (- t (((\mathbf{previous } t) \tau) t_1) t_2))) \\
&\square 0
\end{aligned}$$

Here, it is important to point out that restriction valuation functions are only evaluated for times between t_1 and t_2 . This means that a restriction applied to a timeframe will only examine events that occurred within the timeframe. This approach means that restrictions may be applied to “scopes” within prescription specification and provides more flexibility when specifying prescriptions with complex timing. The consequence of this is that to achieve comparison across timeframes requires a parent node (timeframe or instruction) to have a restriction.

Functions

Functions have been utilized heavily by the previous valuation functions, often by deferring the evaluation of compliance to a child function node. In essence, the function nodes provide the concrete meaning of non-compliance by describing how the compliance of particular behaviours (e.g. a late administration) is measured. These functions need not be “crisp” in nature, the approach permits “fuzzy” functions which

naturally leads to the notion of fuzzy compliance discussed in Chapter 3. Indeed, one could specify a wide range of prescriptions simply by adjusting the relevant function characteristics.

To find a balance between usability and flexibility of the approach, a library of common functions has been created - each of these functions may be accessed by name. Currently, there are a number of standard functions, including: step, ramp, trapezoid, sigmoid, and default action and inaction functions; however, provided a desired function has an appropriate signature, any function is permissible. A condensed version of the valuation function for a function node may be seen directly below, these utilize operations that were defined by the semantic algebras given above.

$$\begin{aligned}
 \mathbf{f} : (\mathbf{f}) &\rightarrow \mathbb{R} \rightarrow \mathbb{R} \rightarrow \mathbb{R} \rightarrow \mathbb{R} \\
 \mathbf{f} [[(\mathbf{F})]] &= \\
 & \quad (= (\mathbf{F}).\text{type} \text{ "RAMP" }) \rightarrow \mathbf{ramp} \\
 & \quad [] (= (\mathbf{F}).\text{type} \text{ "NEGATIVE_RAMP" }) \rightarrow \mathbf{n-ramp} \\
 & \quad [] \dots \\
 & \quad [] \lambda x.\lambda y.\lambda t.0
 \end{aligned}$$

4.4 Measurement

Thus far, a formal language for specifying prescriptions has been described. The denotational semantics of the language, when applied to a prescription specification result in a function which is capable of evaluating non-compliance. If the above denotational semantics were implemented as a computer program, the implementation would constitute a compiler which produces an evaluation function from an input “sentence” in the language. Given an evaluation function, it is pertinent to discuss how it may be used to measure compliance (and/or persistence) and agreement. This section uses the formalism provided thus far to formalize the concepts from AIM in Chapter 3.

4.4.1 Compliance

The function, $c : \mathbf{time} \rightarrow Traces \rightarrow \mathbb{R}$, produced by applying the denotational semantics described above, determines a degree of non-compliance of the action or inaction taken at any instant in time; accordingly, it is called the *instantaneous* non-compliance function. This function may be further broken down into functions of non-compliance due to actions, $c_a : \mathbf{time} \rightarrow Traces \rightarrow \mathbb{R}$, and functions of non-compliance due to inactions, $c_i : \mathbf{time} \rightarrow Traces \rightarrow \mathbb{R}$; separation may be achieved in a number of ways, an easy approach is to define c_a and c_i as:

$$c_a(t, \tau) = \begin{cases} c(t, \tau) & \text{if } ((\mathbf{action? } t) \tau) \\ 0 & \text{otherwise} \end{cases} \quad c_i(t, \tau) = \begin{cases} 0 & \text{if } ((\mathbf{action? } t) \tau) \\ c(t, \tau) & \text{otherwise} \end{cases}$$

Where the **action?** operator is as defined in the denotational definition given above, t is the time or interest, and τ is the trace to evaluate.

As an example, consider Ms. Smith’s warfarin prescription from the previous chapter: *6 mg of warfarin once per day*. Suppose Ms. Smith’s behaviour is observed and the trace in Figure 4.2 is generated.

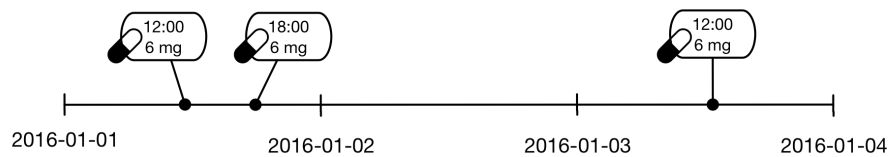


Figure 4.2: A non-compliant trace.

Ms. Smith’s behaviour contains a non-compliant action - the extra dose on day 2 - and a series of non-compliant inactions - she missed a dose on day 3. The two non-compliance curves are shown in Figure 4.3. In Figure 4.3a, Ms. Smith’s additional dose is shown as a single peak of action non-compliance, i.e. the action of administering the extra dose that was determined to be non-compliant. In Figure 4.3b inactions are shown to increase in non-compliance after the end of 2016-01-03, an administration action is expected to satisfy the *once per day* frequency of the prescription, thus continuing inactions are deemed non-compliant. Eventually, the dose is counted as “missed and skipped” and inactions are no longer non-compliant, this is shown via the negative sloped ramp around 10:00 on 2016-01-03. To achieve a different “missed” dynamic the character of the ramp (or whether it decreases at all) may be adjusted

via the inaction function in the prescription specification.

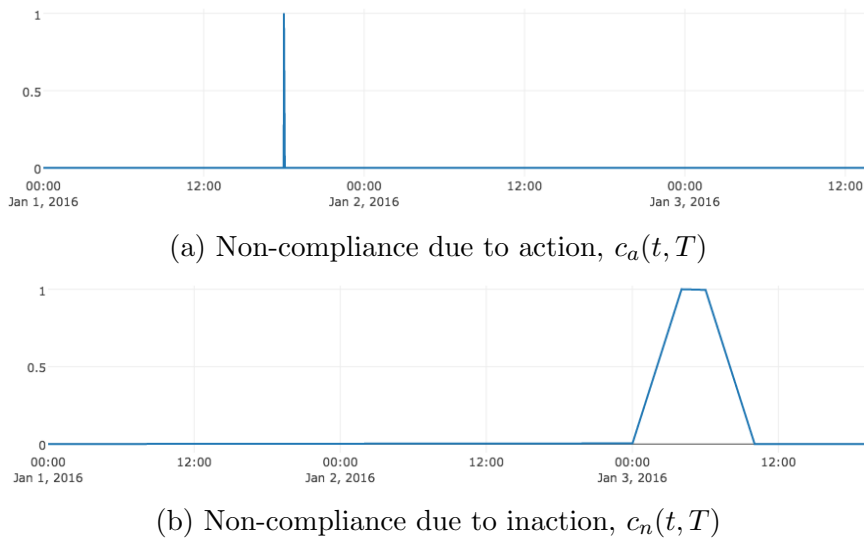


Figure 4.3: Non-compliance curves for trace in Figure 4.2

While these curves are telling of Ms. Smith's behaviour, they are potentially non-intuitive, often non-compliance is thought of as cumulative measure over time. For this reason, a *cumulative* non-compliance measure is also defined. The cumulative non-compliance, $C(t, \tau)$, is defined as:

$$C(t, \tau) = \int_{t_1}^t \frac{1}{x - t_1} (\alpha c_a(x, \tau) + \beta c_i(x, \tau)) dx$$

Here, the coefficients α and β are weighting factors between 0.0 and 1.0 with the property that $\alpha + \beta = 1$. They are required to equalize the relationship between action and inaction. Without them, the integral of the inaction curve would dominate the cumulative non-compliance measure and non-compliant actions would be barely registered since inactions occur more considerably more frequently than actions. Careful selection of α and β will result in a balanced cumulative non-compliance measure.

Consider the plot of cumulative non-compliance in Figure 4.4 which was generated based on the instantaneous non-compliance curves in Figure 4.3. Here, non-compliant actions result in spikes which then fall off as time without non-compliance progresses. In Figure 4.4, the cumulative non-compliance scale is notably small, on the order of 10^{-3} ; this is due to the term $\frac{1}{t-t_1}$ in the cumulative formula. The result may be interpreted as: at time t , 0.1 % of all actions/inactions that were taken by Ms. Smith

thus far (since t_1) were non-compliant.

The “sharp rise” and subsequent “falling off” of the cumulative function is also of interest. The sharp rise is due to (relatively) rapid growth in the instantaneous function and is as expected. The gradual decrease is a result of an increasing number of compliant actions/inactions accumulating over time. This cumulative function’s behaviour, while counter-intuitive at first glance, is in fact quite representative when discussed in the context of actions and inactions.

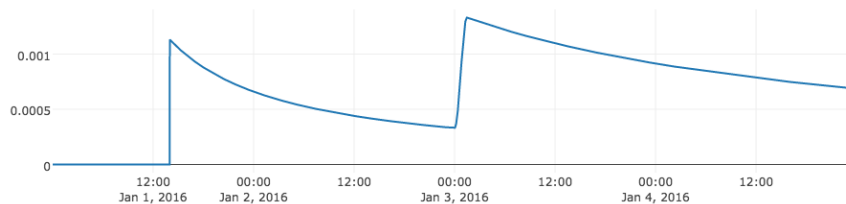


Figure 4.4: Cumulative non-compliance curve, $\alpha = 0.95$, $\beta = 0.05$

4.4.2 Agreement

In AIM, introduced in Chapter 3, disagreement was defined as the distance between two specifications, the *prescription* (provider recommended plan) and the *conscription* (patient adopted plan). The non-compliance function, $c(t, \tau)$ described above, may be used to develop a formal notation of agreement.

For simplicity, consider strict or crisp non-compliance, i.e. a trace is either compliant or non-compliant for all time points of interest. Mathematically a trace, τ , is compliant only if $\forall t, t_1 \geq t \leq t_2 : c(t, \tau) \leq \alpha$ for a cut off point, $\alpha \in [0, 1]$, that describes non-compliance and a time range t_1 to t_2 .

First, define τ_p to be the set of all traces that are (crisply) compliant with the prescription, and τ_c be the set of all traces that are (crisply) compliant with the conscription. Agreement between the prescription and conscription is measured as:

$$A = \frac{|\tau_p \cap \tau_c|}{|\tau_c|} \quad (4.1)$$

Notice that in the case where the conscription is a *refinement* of the prescription ($\tau_c \cap \tau_p = \tau_c$) then agreement becomes 1.0. If the patient strongly disagrees with the prescription, and opts to not administer the medication then τ_c contains only the “empty trace” ($|\tau_c| = 1$) and $|\tau_p \cap \tau_c| = 0$ and agreement becomes 0. Finally, it

is possible that the patient adopts a plan that has no overlapping traces with the prescription, in this case $T_p \cap T_c = \{\}$ and the agreement would again be 0.

4.5 Scheduling

Chapter 3 discussed the scheduling problem in the abstract and introduced a control-loop approach wherein the patient’s medication taking behaviour was the process under control. The proposed scheduling approach combines the Evaluator device (modeled mathematically as the non-compliance function $c(t, \tau)$) and a Scheduler component to determine a schedule for the next administration actions. The task of scheduling may be viewed as an optimization problem: *given the history as a trace, find the next time of an administration action such that non-compliance is minimized.*

When scheduling doses of medication, timing is often of primary interest; other factors such as dose may be considered, or even multiple factors simultaneously. However, for simplicity, here only timing is considered; the approach outlined is easily extended to multiple factors/dimensions.

For the purpose of this thesis, the Scheduler component uses a naive numerical optimization approach for determining the optimal next administration time. Given a trace and prescription specification, a hypothetical action and inaction are advanced through time and the compliance of each evaluated. The following algorithm describes the approach in detail:

Data: trace τ , time to start scheduling t_1 , time to end scheduling t_2
for each time t between some t_1 and t_2 **do**
 generate a fake administration event, e at time t ;
 evaluate the non-compliance due an action: $a = c_a(t, \tau \cup \{e\})$;
 compute the non-compliance at time t due to inaction: $i = c_i(t, \tau)$;
 store the maximum of a and i as the non-compliance score for time t ;
end

Find the time(s) that give the minimal non-compliance; select the first (earliest) minimum value as the next time for administration.

4.5.1 Scheduling Example

Consider Ms. Smith’s ibuprofen prescription: *600 mg of ibuprofen three times per day as needed with food.* If Ms. Smith has already taken two doses of ibuprofen (6:00 and

12:00) and knows that her evening meal will be at 18:00 then a scheduling algorithm may suggest a range of times that: 1) ensure her third dose of the day is not too close to the previous dose - assuming the prescription specification has accounted for minimum intervals between doses, perhaps five hours; and 2) her dose is taken sufficiently close, two hours, to dinner time at 18:00. Figure 4.5 shows the non-compliance of the third dose for a range of future times at which it could be administered. Administration between 17:00 - 20:00 indicates the lowest non-compliance, this range is far enough from the previous dose at 12:00 and close enough to dinner time.

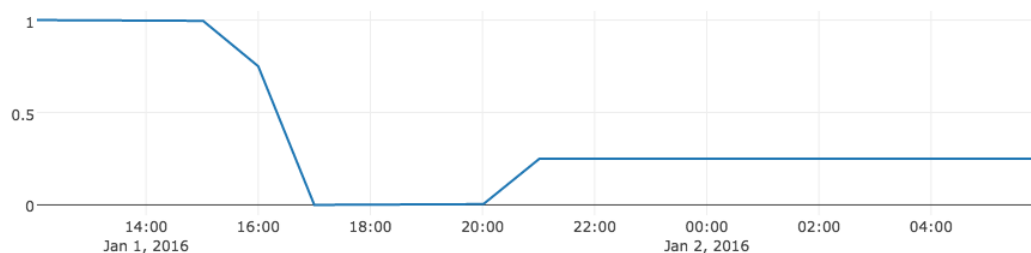


Figure 4.5: Scheduling output for Ms. Smith's next dose of ibuprofen.

As suggested in Chapter 3, an IoT ecosystem might indicate the optimal time to take the dose using a combination of notifications. A light on the IoT device indicates whether she should administer a dose using a color scheme: green light - optimal time (between 17:00 - 20:00), yellow light - OK time (after 20:00), and red light - do not administer (before 16:00).

4.6 Implementation

The formalism described in this chapter has been implemented as a prototype stand-alone package in the Clojure programming language¹. Clojure is a functional programming language whose core syntax and semantics is very similar to the lambda calculus that was used to define the semantics of the prescription language in Section 4.3, thus the core logic for the formalism is mapped cleanly into a core set of Clojure modules in the implementation. An effort was made to use basic language functions and to minimize the dependency on external libraries/packages which may have behaviour that is not directly represented in the denotational semantic definitions.

Several additional features were implemented, including: scheduling (as above),

¹<https://clojure.org/>

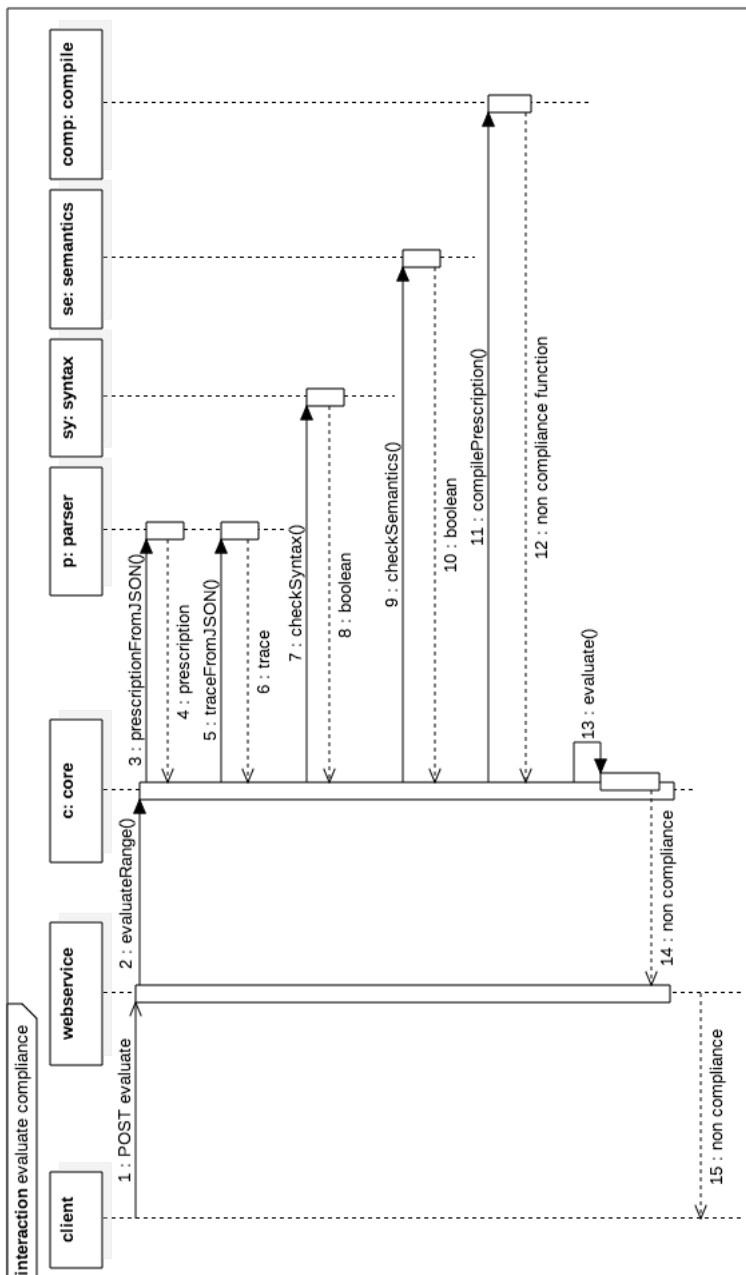


Figure 4.7: Sequence diagram for handling request to evaluate a trace's compliance

4.6.2 Static Semantic Checks

As in [64], static semantic checks were created to (partially) verify the static semantics of incoming prescription specifications. These checks ensure that the prescription, which may be syntactically correct, does not contain any obvious contradictions or internal conflicts. In total seven static semantic checks were implemented, while not an exhaustive set, these cover easily made errors. For example, one check ensures that timeframes are nested appropriately, one frame cannot be refined by another that has a larger timeframe; Figure 4.8 shows a portion of a prescription graph that would violate this constraint.

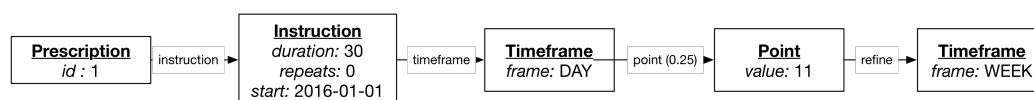


Figure 4.8: Invalid frame nesting

4.6.3 Web Interface

Packaging the formalism implementation as a Clojure library permits interoperability with other JVM languages/platforms. However, it is possible that non-local and/or non-JVM systems would like to evaluate compliance of a patient behaviour. To facilitate this, a RESTful web service that maps HTTP(S) requests to operations in the `adherence.formalism.core` module. If this access to a compliance measurement service, running utilizing this or a similar package, was provided over the Internet then there are a few foreseeable uses:

- Providing a consistent compliance measurement service to Internet of Things (IoT) devices.
- Off-loading the extensive processing required for measurement and scheduling to a powerful server and allowing IoT devices to consume less power, thus extending their battery lifetime.
- Evaluation of compliance by Health Information Systems in clinical settings.
- Standardizing compliance measures across different use cases, e.g. IoT devices, clinical settings, and research studies.

4.7 Example

In this section Ms. Smith’s prescription for morphine and a subsequent administration behaviour is presented as a detailed example. The example first demonstrates the process of prescription interpretation to produce an interpreted prescription (IP) from the original author prescription (AP). Then a compliance analysis of a hypothetical trace is conducted. Finally, the scheduling approach is employed to determine the next administration time for Ms. Smith.

4.7.1 Prescription Interpretation

Recall, Ms. Smith’s AP for morphine from Chapter 3: *10 mg of morphine twice daily with food*; the IP is shown in Figure 4.9. Some of the nodes in the IP follow directly from the natural language in the AP, however there are several implicit assumptions made by the AP and explicitly stated in the IP whose correspondence may not be immediately obvious, these are discussed below.

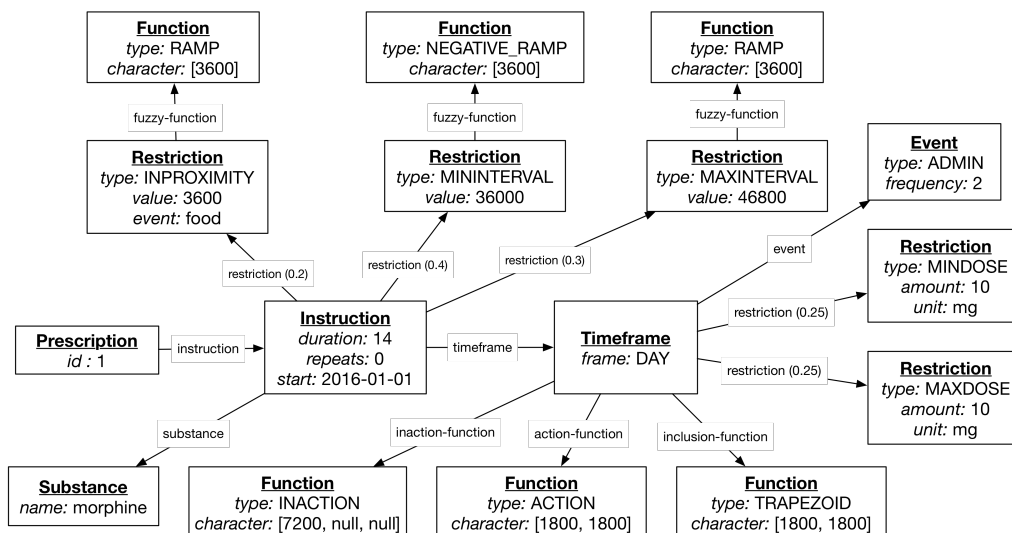


Figure 4.9: Graphical representation of Ms. Smith’s morphine prescription.

Implicit Time Intervals

The AP indicates that morphine should be administered *twice daily*. Taken literally, such a specification would permit any behavioural trace wherein two doses are taken within each calendar day. Indeed, the prescription language does permit such a

specification, the formal semantics associated with the **frequency** field of an Event node indicate as much.

However, in reality *twice daily* implies that the doses should be reasonably spaced in time. This additional implication must be explicitly captured in the IP, thus an interval between administrations must be specified. The AP could be amended to: *10 mg morphine twice daily 12 hours apart*. When specifying prescriptions in general it may be desirable to specify a minimum and maximum interval between administrations independently, thus for additional flexibility, two types of restriction which have been employed in the specification in Figure 4.11.

In most cases, it would be impractical to demand that doses are administered precisely 12 hours apart. Instead, the example IP in Figure 4.9 uses a maximum interval of 13 hours (46800 seconds) and a minimum interval of 11 hours (36000 + 3600 seconds). Further, a fuzzy function is used to evaluate non-compliance of time between administrations, in Figure 4.9 these are shown as Function nodes which describe functions as seen in Figures 4.10c and 4.10d.

It should be noted that in a real prescription interpretation process the character of these functions can be determined based on input from a number of sources as discussed in Chapter 3. For the purpose of this example fictitious functions are used. The evaluation in Chapter 5 demonstrates the addition of expert opinion data to the prescription interpretation process.

Proximity to Food

The AP indicates that doses of morphine should be administered *with food*. As with timing/intervals, if taken literally, such a prescription would require that food and the medication be consumed simultaneously, while this could be the intent of the prescription, it may also be more accurate for the prescription to read *within one hour of food*, or similar. To this end, the IP in Figure 4.9 uses an In Proximity Restriction to indicate that the medication should be administered within one hour of food consumption. Further, the restriction has a Function node as a direct child which represents the fuzzy function shown in Figure 4.10e that indicates a degree of non-compliance based on the interval between a dose being administered and food being consumed.

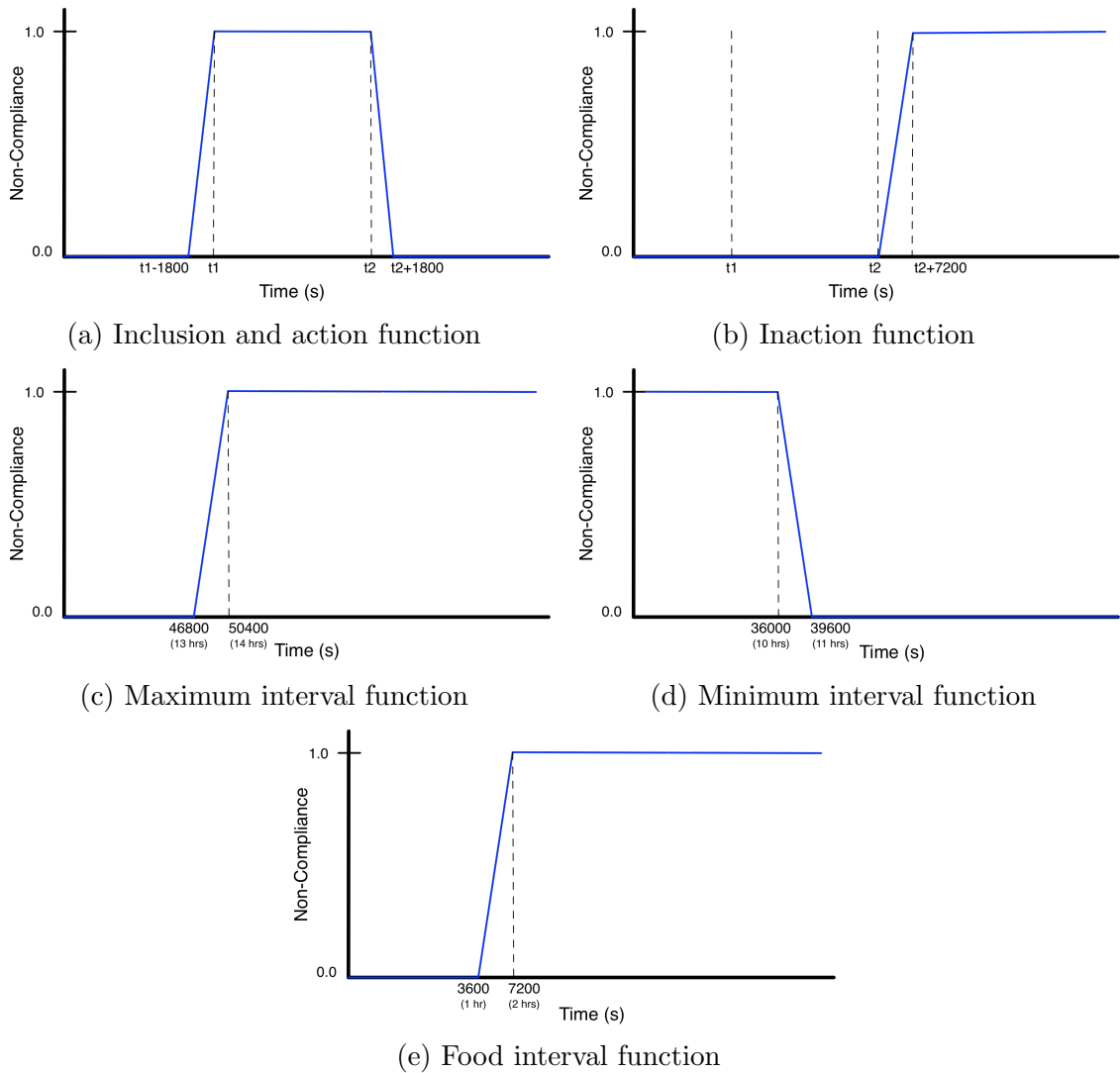


Figure 4.10: Fuzzy functions for Ms. Smith's morphine prescription.

Inaction Functions and Miss Behaviours

Ms. Smith's morphine prescription fails to provide any information about how to score the non-compliance of a missed or late dose of medication; a dose is considered "missed" if the timeframe in which the dose was expected expires without an administration event being recorded. The semantics of the specification language permit at least three possible behaviours:

1. Ignore the missed dose and do not consider it non-compliant; this is equivalent to the notion of a *pro re nata* (as needed) instruction.

2. Count subsequent inactions as non-compliant until a dose of the expected medication is administered, continue this indefinitely.
3. Count subsequent inactions as non-compliant until either a dose of the expected medication is administered or a predefined amount of time has elapsed.

These different behaviours are captured in the IP by selection of the appropriate inaction function. The purpose of an inaction function is to evaluate the non-compliance of inactions occurring in or around a timeframe of interest. For Ms. Smith's morphine prescription, the second option - take the dose as soon as possible - is used, the corresponding inaction function may be seen in Figure 4.10b.

Inclusion and Action Functions

The IP includes two Function nodes that are used to describe the treatment of actions/administration events within a timeframe: an action function and an inclusion function. The action function is used to determine the non-compliance of an action occurring within (or close to) a timeframe of interest. The inclusion function is used to determine to what degree an action/event is within a timeframe. For example, an event occurring at 00:01 on January 2nd is fully within a day long timeframe starting at 00:00 on January 2nd, however, for the purposes of medication administration, it may also be considered to be “close enough” to January 1st to be counted as part of that timeframe as well. In the case of Ms. Smith's morphine prescription, the action and inclusion functions have the same shape and may be seen in Figure 4.10a.

4.7.2 Evaluating Non-Compliance

Consider the following scenario describing Ms. Smith's behaviour with respect to her morphine administration, the corresponding behavioural trace may be found in Figure 4.11:

- On January 1st and 2nd Ms. Smith administers her morphine in the morning with breakfast and in the evening around dinner time.
- On January 3rd she takes her morning dose, but then has a challenging day and falls asleep before dinner and misses her evening dose and meal.

- At 04:00 in the morning on January 4th Ms. Smith wakes up in pain (possibly due to her missed dose from the night before) and immediately administers a dose of morphine - she goes back to sleep.
- She then eats breakfast at 07:00 and does not take another dose, she already took one earlier that morning.
- On the evening of January 4th she has dinner, but forgets to take her evening dose until 23:45, right before she goes to sleep.
- She wakes up at 05:00 on January 5th and takes her morning dose with breakfast.
- The scenario ends at 07:30 on January 5th, Ms. Smith plans on eating her evening meal at 18:00.

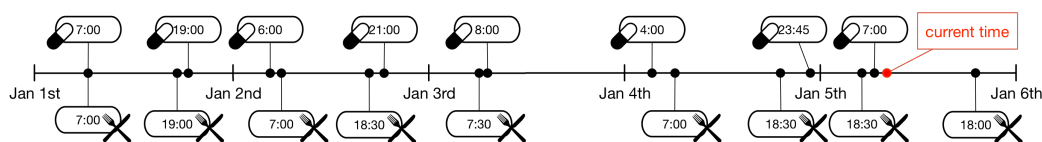


Figure 4.11: Ms. Smith's behaviour

Compiling the IP generates a non-compliance function which may be used to evaluate the Ms. Smith's behaviour. The instantaneous and cumulative non-compliance curves are shown in Figure 4.12.

In the instantaneous curve there are four non-zero sections, these may be interpreted as follows (from left to right):

1. Ms. Smith's evening dose on January 2nd, was a bit late. The maximal interval permitted by the prescription was 13 hours between doses. As a result, inactions after 19:00 on January 2nd are counted as increasingly non-compliant. when she finally administers the dose the administration action is non-compliant in the sense that it was: 1) late, and 2) not within one hour of a meal; this is seen as a spike at 21:00.
2. Ms. Smith missed her evening dose on January 3rd. Thus, after the 13-hour interval (since her last dose) expires, her inactions are increasingly non-compliant. Further, after 00:00 on January 4th, it is determined that she has not administered her second dose, thus her non-compliance increases again. Finally,

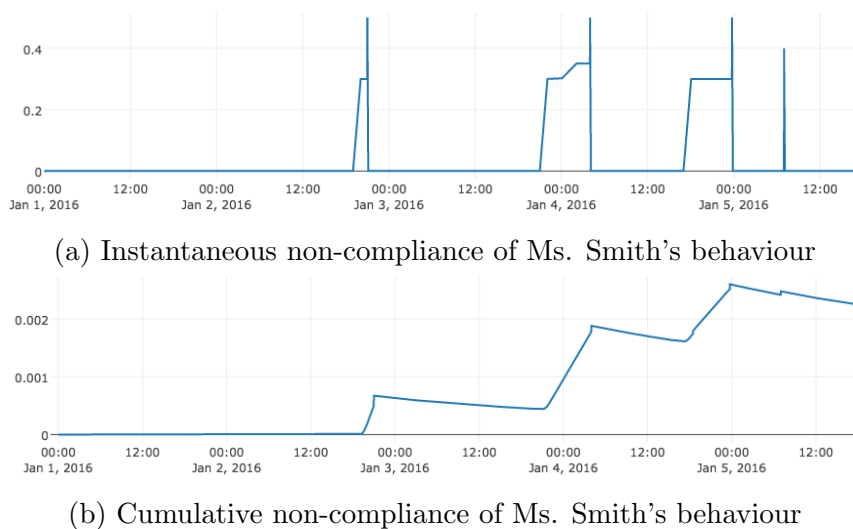


Figure 4.12: Non-compliance curves for Ms. Smith's behaviour shown in Figure 4.11, $\alpha = 0.95$, $\beta = 0.05$

she takes her dose, but again, it is not proximal to any food and results in a spike in non-compliance.

3. Ms. Smith's evening dose on January 4th is late, thus inactions are increasingly non-compliant, when she finally does administer the dose it is not proximal to food.
4. The time interval between Ms. Smith's evening dose on January 4th at 23:45 and the morning dose on January 5th at 07:00 violates the minimum interval restriction of 11 hours, thus this action is scored as non-compliant.

The cumulative non-compliance function is a direct result of integrating the instantaneous non-compliance function in the manner described in Section 4.4.

4.7.3 Scheduling

The presented scenario ends at 07:30 on January 5th. However, the timing of the next dose of morphine may be determined based on the prescription specification and Ms. Smith's behaviour thus far. Further, Ms. Smith's evening meal on January 5th is planned for 18:00, this information is provided as part of the trace to the scheduling algorithm. Applying the process described in Section 4.5 results in the non-compliance curve shown in Figure 4.13.

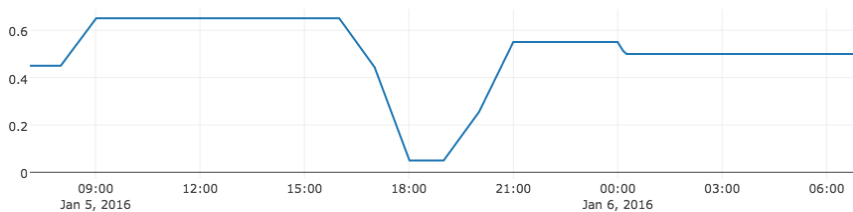


Figure 4.13: Scheduler output for Ms. Smith’s next dose of morphine.

According to the output, the best time to administer the next dose is between 18:00 and 19:00 on January 5th. Times before 18:00 are too close to the previous dose (violation of minimum interval restriction), and times after 19:00 become too far from the previous dose (violation of the maximum interval restriction).

It is interesting to note that the “best” range for the next dose, though minimal, has a non-zero non-compliance. This is due to the dose at 23:45 on January 4th being included within the January 5th timeframe, this results in a count of 2.5 doses being administered on January 5th, which exceeds the prescribed number, and results in a non-zero non-compliance score for the next dose.

Example Summary

This example was intended to provide the reader with a clear intuition for how the formalism presented in this chapter may be used to: 1) formulate a formal specification of a natural language prescription; 2) evaluate the compliance of a patient’s behaviour; and 3) use the non-compliance evaluation function to schedule the next dose of medication. To achieve this, some liberties were taken with the interpretation of the original authored prescription, in particular the relative importance (weighting) of the restrictions. Thus, this example is not representative of a clinical scenario and may require “tuning” to achieve a clinically useful specification.

4.8 Chapter Summary

This chapter has formally defined the concepts of a prescription, behavioural trace, and non-compliance of the trace to the prescription. The syntax of a prescription specification language was formalized via a graph grammar. Then a denotational semantics of the language was provided by associating each grammar production rule with a valuation function. Thus the precise meaning of prescription specified using

this language may be determined.

The valuation functions were crafted to produce a non-compliance evaluation function that is capable of, given a trace and a point in time, determining a non-compliance score for the behaviour encoded in the trace at that time. Using this, the non-compliance of a patient's longitudinal behaviour to a prescription may be measured and then future administration actions may be scheduled.

An example containing a hypothetical prescription and patient behaviour was provided to demonstrate the formalism. This example, while internally consistent, is likely not representative of reality. A more rigorous evaluation is presented in Chapter 5 using data obtained from a survey of practicing physicians.

Chapter 5

Evaluation

This chapter evaluates the language and approach for measuring compliance presented thus far. The evaluation is structured in three parts: 1) verifying that the requirements given in Chapter 3 are satisfied; 2) evaluating the breadth of expressibility of the language; and 3) evaluating the language and approach for clinical accuracy.

5.1 Requirements Verification

Chapter 3 provided requirements for the approach and language, these were categorized as: 1) Prescription Interpretation Requirements, 2) Observer Device Requirements, and 3) Evaluator Device Requirements. Tables 5.1, 5.2, and 5.3 show each requirement and its status as either: *satisfied*, *partially satisfied*, *not satisfied*. The status of each requirement was determined via inspection.

All of the Prescription Interpretation Requirements (Table 5.1) are satisfied with the exception of *R1-5* and *R1-6*. These indicate that both the AP and additional data should be used as inputs to the process of prescription interpretation. As it currently stands, prescription interpretation is a manual process requiring a human to translate the AP into the language from Chapter 4 and to augment the specification with additional data. Given the scenario for scheduling Ms. Smith's medication administrations, which embedded this approach within an automated Internet of Things environment, it is desirable to (a large extent) automate prescription interpretation. Therefore *R1-5* and *R1-6* are marked as *Partially Satisfied*.

The Observer Device Requirements (Table 5.2) are entirely satisfied by the approach and formalism as presented. The trace language is capable of expressing the required

Table 5.1: Prescription Interpretation Requirements

Id	Name	Status	Comment
R1-1	Language Concepts	Satisfied	-
R1-1.1	Substance Specification	Satisfied	-
R1-1.2	Dose Specification	Satisfied	-
R1-1.3	Duration Specification	Satisfied	-
R1-1.4	Timing Specification	Satisfied	-
R1-1.4.a	Time Points	Satisfied	-
R1-1.4.b	PRN Timing	Satisfied	-
R1-1.4.c	Administrations Per Time Period	Satisfied	-
R1-1.4.d	Intervals Between Doses	Satisfied	-
R1-1.5	Proximity to Events	Satisfied	-
R1-1.6	Changing Dosing	Satisfied	-
R1-2	Fuzziness	Satisfied	-
R1-3	Language Formality	Satisfied	-
R1-4	Language Consumability	Satisfied	-
R1-5	Authored Prescription Input	Partially Satisfied	Not fully automated
R1-6	Additional Inputs	Partially Satisfied	Not fully automated

concepts, and though it is described using set-theoretic notation, may be represented as a simple list of objects by a computing device.

Table 5.2: Observer Device Requirements

Id	Name	Status
R2-1	Observable Factors	Satisfied
R2-1.1	Medication Administrations	Satisfied
R2-1.2	Food Consumption	Satisfied
R2-1.3	Awake Actions	Satisfied
R2-1.4	Asleep Actions	Satisfied
R1-2	Trace Language Consumability	Satisfied

The current evaluator device implementation is capable of measuring the compliance of a trace against a given prescription and returns a degree of compliance. However, the implementation is not entirely platform independent; Clojure source code is capable of running within the Java Virtual Machine (JVM) environment. In the absence of additional cross-compilation tools, the implementation is limited to devices capable of running the JVM which prevents the current evaluator implementation from being deployed on devices such as micro-controllers.

Additionally, requirement R3-6 requires that the evaluator produce results that are representative of those produced by a human clinician; this requirement is only partially satisfied. An infinite number of combinations of APs and patient behaviours mean

that it is not possible to validate that every measurement produced by the evaluator is clinically representative. However, some confidence for a sample of prescriptions and behaviours is achieved by the clinical evaluation which is described in Section 5.3.

Table 5.3: Evaluator Device Requirements

Id	Name	Status	Comment
R3-1	Evaluation	Satisfied	-
R3-2	Degrees of Compliance	Satisfied	-
R3-3	Platform Independent	Partially Satisfied	Requires JVM
R3-4	Implementation Simplicity	Satisfied	-
R3-5	Implementation Accessibility	Satisfied	-
R3-6	Clinically Representative	Partially Satisfied	Scope restriction

5.2 Language Breadth of Expression

As per the previous section, the language for specifying prescriptions is capable of expressing a range of prescribing concepts independently (requirement R1-1), however the ability of the language to complete prescriptions that use several concepts concurrently is important for its applicability to real-world settings.

To evaluate expressiveness, ten prescriptions, which span a range of prescribing concepts that are typical of a primary care setting, were expressed in the language. The prescriptions were provided as natural language APs by a domain expert (physician); they were interpreted without additional data sources to generate the interpreted prescription (IP) specifications. Since the purpose of this evaluation was determine if expression of the given APs was possible, no further evaluation of the clinical accuracy of the IP was conducted. An evaluation of the clinical accuracy is presented below in Section 5.3. The prescriptions and whether it could be expressed in the language (*yes*, *partial*, or *no*) are shown in Table 5.4.

A sample of the IP expressed in the language is shown in Figure 5.1, the remainder of the APs from Table 5.4 have their respective IPs shown in Appendix B.

Eight of the ten given APs were fully expressed in the specification language. However, APs 9 and 10 were only partially expressed in the language as it is presented in Chapter 4. AP 9 requires the addition of an *administration indication* (e.g. insomnia) which is not currently part of the language; however the remainder of the AP is expressible. It is conceivable that a proximity type restriction for a medical indication, e.g. insomnia, could be added to the language similarly to the proximity restrictions

Table 5.4: Authored Prescriptions (APs) provided by domain expert

No.	Authored Prescription	Status	Comment
1	keflex 500 mg four times daily for 10 days	yes	-
2	azithromycin 500 mg once daily for 1 day then 250 mg once daily for 4 days	yes	-
3	ibuprofen 600 mg three times daily as needed	yes	-
4	seroquel 25 - 50 mg three times daily as needed	yes	-
5	morphine 10 mg three times daily as needed up to 10 per week	yes	-
6	prednisone 50 mg decrease by 10 mg per week until done	yes	-
7	coumadin 7 mg once daily on M/W/F 8 mg once daily otherwise	yes	-
8	hydrochlorothiazide 25 mg once daily in the AM for 30 days	yes	-
9	zopiclone 7.5 mg once daily at bedtime as needed for insomnia	partial	no explicit indication
10	ethinyl estradiol with norethindrone 1 tablet once daily for 21 days starting on first day of menses and then restart on day 29	partial	floating start time

that exist for food events. As with AP 9, AP 10 requires an additional administration indication concept, though it also requires a “floating” start time that is dependent on the occurrence of the first day of menses; the current language description requires a fixed starting date. However, the remainder of AP 10 is expressible in the language.

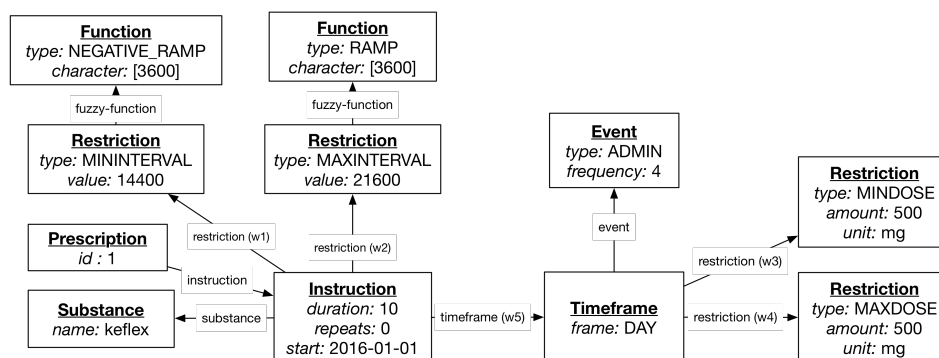


Figure 5.1: Sample IP for AP number 1 from Table 5.4

5.3 Clinical Evaluation

Requirement **R4-5** states that: *The result of behavioural evaluation produced by the formalism shall be representative of those produced by a human clinician.*

Evaluation of this requirement demands that the results produced be compared against the opinion of practicing clinicians.

From a clinical perspective, a tangible aspect of the approach is the scheduling of the next administration event. The scheduling approach presented in Chapters 3 and 4, though naive, is a faithful representation of the formalism's semantics. Thus, comparing the scheduling output of the implementation with clinician expectations for when the next administration event should occur provides a reasonable proxy evaluation of the approach and formalism. The remainder of this section describes the method used to conduct the clinical evaluation and presents results.

5.3.1 Method

The clinical evaluation had five phases:

1. Use Case Selection
2. Data Collection
3. Prescription Interpretation
4. Verification
5. Validation

First, ten prescriptions and traces were chosen as the use cases for the evaluation, then expert consensus data was collected via an online survey. The expert consensus data was used to inform prescription interpretation for each of the ten use cases. The prescriptions and traces were provided to a scheduler and the output was verified to ensure that an expected result was produced; the scheduling output was also validated by three domain experts.

Use Case Selection

Ten prescriptions and traces were selected as use cases for the clinical evaluation. The use cases were selected by a domain expert such that a range of medication

types/classes and administration timings typical of primary care were addressed. Table 5.5 shows the ten use cases.

No.	Specification	Partial Trace	Medication Class
1	Nitrofurantoin 100 mg twice daily	8:00	Antibiotic
2	Ramipril 2.5 mg once daily	10:00	ACE Inhibitor
3	Hydromorphone 4 mg every 4 hours as needed	6:00, 10:00, 14:00	Opioid
4	Marvelon 1 tablet once daily	9:00	Birth Control
5	Glargine 10 units at bedtime	22:00	Insulin
6	Moxifloxacin 400 mg every twenty four hours	12:00	Antibiotic
7	Cephalexin 500 mg four times a day	8:00, 12:00	Antibiotic
8	Warfarin 7 mg once daily	9:00	Anti-coagulant
9	Penicillin V 500 mg three times a day	9:00, 17:00	Antibiotic
10	Hydromorphone Contin 9 mg twice daily and Hydromorphone 4 mg every 4 hours as needed	8:00 (9 mg), 14:00 (4 mg)	Opioid

Table 5.5: Scenarios posed to participants of survey

Data Collection

An online survey was conducted to obtain expert consensus data. The survey included qualitative and quantitative questions regarding adherence and compliance; for the purpose of this thesis only the quantitative data was used.

Recruitment The survey was intended to be completed by practicing Canadian physicians. Recruitment was conducted via existing professional networks by two co-investigators (Jordan Banman MD and Morgan Price MD PhD). Messages to (private) community social network boards and (private) email lists were used as the primary means of recruitment; word of mouth was also acceptable. Participants were asked to follow a URL to the survey website where their consent to participate was obtained via a letter of information for informed consent.

Survey The survey web page was available at all times during the study period. Participation was anonymous. The survey had two main phases, the first captured a mix of quantitative and qualitative data including relevant information about the participants, this provided a context for the other data collected in the study; the

second section collected the quantitative data by posing “adherence scenarios” to participants.

The contextual questions were intended to understand the study participant demographic; two questions that were relevant to this thesis were:

1. “How long have you been practicing medicine” (*resident physician, less than 5 years, between 5 and 10 years, or greater than 10 years*).
2. “What best describes your area of practice” (*family medicine in office, family medicine in hospital, specialist in office, specialist in hospital*).

In the second phase of the survey, participants were presented with the ten use case scenarios in which a short patient history, a natural language AP, and a historical patient behaviour were given. For each use case the participants were asked to:

indicate a range of times in which the next dose of medication could be administered while still remaining adherent to the prescription.

Participants indicated the adherent range for the next administration using sliders (green - lower bound of adherent range, red - upper bound of adherent range) on a time line (intervals of 15 minutes) which showed the previous patient behaviour (black markers), a sample is shown in Figure 5.2.

Here the term “adherence” was used as it is the currently widely accepted terminology; however, according to the definitions used by this thesis, participants were in fact considering compliance of patient behaviours.

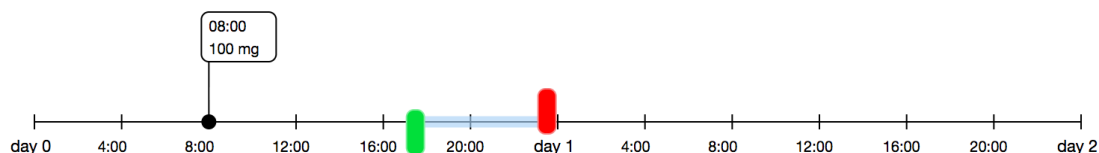


Figure 5.2: Timeline with sliders for collecting quantitative data.

Data Analysis Each participant’s response for each use case was represented as an ordered pair, (t_l, t_u) indicating a lower and upper bound of a compliant range for the next administration event. Responses were excluded if they satisfied at least one of the following criteria:

- the response's range included the one of the previous administration times in the trace, implying that a double dose is would be compliant, and
- the response's range did not include (within a margin of 10%) the expected next administration time, implying that a missed dose would be compliant.

The remaining responses from participants were combined to generate an experimental non-compliance function for each use case using the following algorithm:

Data: S the set of ordered pairs (t_l, t_u) for a given use case and $c_e(t)$ the experimental compliance function determined by the data

for each time t (15 minute intervals) from the use case timeline **do**

 Determine the responses which indicated a dose at t was compliant:

$$R = \{(t_l, t_u) \in S : t \geq t_l \wedge t \leq t_u\}$$

 Calculate the experimental non-compliance measure for time t as

$$c_e(t) = 1 - \frac{|R|}{|S|}$$

end

Return the experimentally determined function $c_e(t)$.

The resulting function, $c_e(t)$ may be interpreted as: *if a an administration event occurs at time t then $c_e(t) \times 100\%$ of clinicians would consider it non-compliant.*

Prescription Interpretation

For each use case the task of prescription interpretation was conducted manually. Interpretation involved using the original AP to create a basic IP in the specification language and then using the expert consensus function, $c_e(t)$ to determine the character of the action, inaction, and interval restriction functions. Function characteristics were selected such that the generated scheduler compliance function, $c_g(t)$, would closely match the experimental compliance function, $c_e(t)$.

Verification

To verify that the language's semantics and scheduling algorithm implementation perform as expected, the IP and associated trace were used as input to the scheduler to generate a new compliance function, $c_g(t)$. The experimental and generated functions were compared over the use case's time line (t_1 to t_2) using a Root Mean Square Error (RMSE) comparison:

$$RMSE = \sqrt{\frac{\sum_{t=t_1}^{t_2} (C_g(t) - C_o(t))^2}{n}}$$

Where n is the number of data points between (inclusive) t_1 and t_2 .

Validation

To validate that the generated compliance functions were medically accurate, i.e. that the generated schedule is acceptable for use in a clinical context, a panel consisting of two experienced physicians and one experienced pharmacist reviewed the generated compliance function for each use case. The panelists responses were recorded on the following scale:

1. *unacceptable*
2. *acceptable with major revisions required*
3. *acceptable with minor revisions required*
4. *acceptable*

For each use case the panelist's responses were averaged. Detailed notes were taken regarding the panelist's qualitative comments.

5.3.2 Results

In this section, results from the data collection, prescription interpretation, verification, and validation phases of the clinical evaluation are presented.

Survey Results

In total 73 individuals participated in the web survey, Tables 5.6 and 5.7 show the demographics of the participants. However, not all participants completed the timeline portion of the survey, the number of responses for each timeline question are shown in Table 5.8. For each use case, the data collected were used to create an experimental compliance function and were plotted; Figure 5.3 shows the data for prescription number 9 circles along the horizontal axis indicate when the previous doses were administered. Plots for other use cases are available in Appendix C.

Table 5.6: Participant Professional Experience

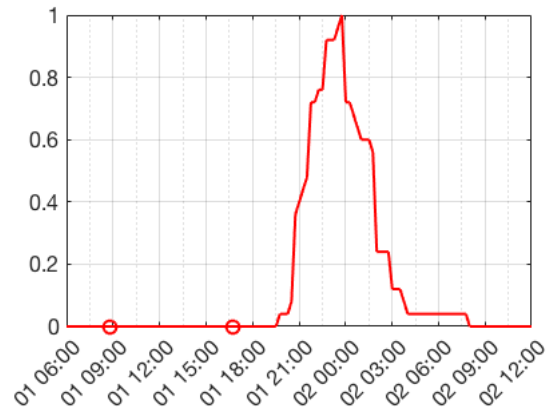
Experience Level	Number	Percentage
Resident Physician	66	90.4%
Less Than 5 Years	3	4.1%
5 to 10 Years	2	2.7%
Greater Than 10 Years	2	2.7%

Table 5.7: Participant Practice Types

Practice Type	Number	Percentage
Family Office	59	80.8%
Specialist Office	4	5.4%
Family Hospital	7	9.6%
Specialist Hospital	3	4.1%

Table 5.8: Survey Timeline Responses

No.	Total	Excluded	Valid
1	39	7	32
2	39	4	35
3	38	3	35
4	38	10	28
5	35	16	19
6	33	8	25
7	33	7	26
8	33	4	29
9	33	10	23
10	33	12	21

Figure 5.3: Experimental compliance function for use case 9 (*Penicillin V 500 mg three times a day*).

Prescription Interpretation Results

In addition to the original AP, the experimental compliance function from the survey was used to create an IP specification expressed in the language described in Chapter 4. The resulting IP from use case 9 is shown in Figure 5.4; Appendix C shows the IPs for all of the use cases. Prescription interpretation was not completed for use case number 10. Upon review of the data it became clear that the compound nature of the prescription (two interacting sets of administration instructions) was too complex for the analysis method used in this evaluation.

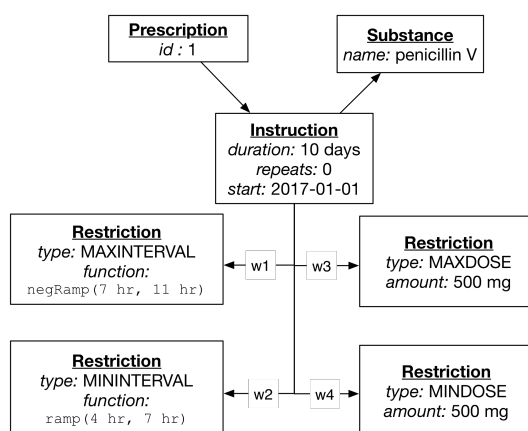


Figure 5.4: Interpreted prescription for use case 9 (*Penicillin V 500 mg three times a day*).

Verification Results

To verify the results, the generated and experimental compliance functions were compared using a RMSE measurement, all errors were less than 0.1; Table 5.9 shows the error for each use case. Figure 5.5 displays the difference between the generated and experimental compliance functions for use case 9.

Table 5.9: RMSE of Use Cases

No.	RMSE
1	0.077
2	0.028
3	0.053
4	0.031
5	0.060
6	0.054
7	0.048
8	0.043
9	0.070
10	-

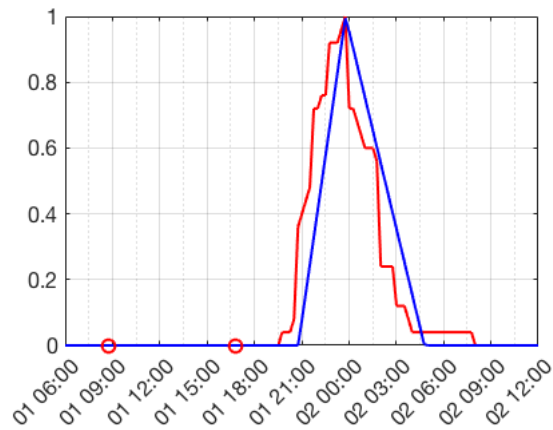


Figure 5.5: Generated and Experimental Compliance Functions for Use Case 9 (RMSE=0.070)

Validation Results

Validation of the generated compliance functions was completed by a panel of three experts consisting of an experienced pharmacist and two experienced physicians. The scores for each use case are shown in Table 5.10.

The following comments/observations were made by one or more of the panelists during the review:

- The approach does not explicitly consider pharmacokinetics, i.e. how the substance behaves in the human body.
- When considering the therapeutic window of the substance, there is a difference between the “ramp-up” period and “steady-state” period, compliance is more critical during “ramp-up”.
- In use case 3 (*Hydromorphone 4 mg every 4 hours as needed*), the peak in both the generated and experimental compliance functions appears counter-intuitive, this is a result of the expert consensus data used for prescription interpretation. For a PRN medication, ideally, the function would grow and then level off, not dip back down.
- In several use cases, a “baseline” or “plateau” was present in the experimental and generated compliance functions, these do not make sense in all cases.
- It is important to distinguish between the optimal schedule based on the generated compliance function and synthesizing recommendations to recover from a missed administration; this is apparent in use case 4 (birth control) where recommended miss behaviour is more complicated than simply skipping or deferring the administration.
- The slope of the functions for some of the use cases is slightly too shallow given the half-life of the substances in question.

Table 5.10: Expert Review of Generated Compliance Functions

No.	Pharm.	Phys. 1	Phys. 2	Avg.
1	4	4	4	4.0
2	4	4	4	4.0
3	3	3	3	3.0
4	3	4	4	3.6
5	3	4	4	3.6
6	4	4	4	4.0
7	2	4	4	3.3
8	3	4	4	3.6
9	4	4	4	4.0

5.4 Chapter Summary

This chapter has presented an evaluation of the approach and formalism described in Chapters 3 and 4. Three evaluations were conducted. First, the requirements for the approach and formalism were reviewed; almost all of the requirements were satisfied with a few exceptions due to the scope of this thesis. Second, the breadth of the formal language was evaluated by expressing ten natural language authored prescriptions in the language; all but one of the authored prescriptions were successfully expressed in the language. Finally, a clinical evaluation consisting of a verification and validation of the output of the scheduler, a proxy for the approach and formalism’s semantics, was presented; the results of the evaluation were acceptable. A detailed discussion of these results is provided in Chapter 6.

Chapter 6

Discussion

This thesis has presented and applied a computational definition of adherence. First, a conceptual foundation was established by the Adherence Interaction Model (AIM) which identified three components of adherence: non-compliance, impersistence, and disagreement as measures of distance between a prescription (prescriber's plan), a conscription (patient's plan), and a description (patient's behaviour); AIM was used to contextualize the approach for measuring adherence. An important part of the approach is a language for specifying prescriptions; the syntax and semantics of which were formalized using graph grammars, attribute grammars, and denotational semantics respectively. Since the concepts of a prescription and adherence are interdependent, formalizing the notion of a prescription, via specification language, resulted in a computational definition of adherence that can form the foundation of technological interventions aimed at improving adherence.

The proposed approach to measuring compliance is based on a system architecture consisting of an Observer Device, which is capable of observing a patient's medication behaviours, and an Evaluator Device, which given a prescription expressed in the specification language, is capable of evaluating the patient's behaviour to produce a compliance score. It is important to ensure that the prescription provided to the Evaluator Device is clinically accurate, thus the original authored prescription (AP) from the health care provider must be interpreted in the context of additional information; this process is called Prescription Interpretation, with the resulting prescription (expressed in the specification language) being called an Interpreted Prescription (IP). The language and approach to compliance measurement were evaluated, the results of which are discussed below.

The proposed approach to compliance measurement was applied to the problem

of scheduling the next medication administration time. Scheduling is treated as an optimization problem wherein the next administration time that produces the best compliance measurement is selected. Being able to determine schedules for medication administration, based on previous and future events, including lifestyle factors, is an important step in developing context-aware compliance improving interventions.

A notable feature of both the specification language and this approach is the use of fuzzy set theory to account for the imprecision inherent to human behaviours. The resulting compliance score, which is a value between 0.0 and 1.0, can be considered the degree to which a patient's behaviour is part of the set of compliant behaviours. The specification language allows one to provide fuzzy functions which indicate how to score patient behaviours that are not strictly compliant but have also not deviated too far from the intent of the prescriber. To date, while many measures of adherence exist, none have used fuzzy set theory to permit imprecise reasoning about human behaviours.

An important aspect of adherence is agreement between the patient and the provider regarding the prescription. AIM explicitly accounts for disagreement as the distance between the provider's plan and the patient's plan, which would ideally be measured as zero. Formalizing a language for expressing prescriptions also permits a measurement of agreement; to accomplish this a technique called trace enumeration was proposed, agreement is calculated as the ratio of the number of traces that satisfy both the provider's and patient's plans to the number of traces that satisfy the patient's plan. Admittedly, the treatment of agreement by this thesis is somewhat lacking relative to the treatment of compliance; the provided definition of agreement has not been evaluated and has remained strictly theoretical. This is in part a matter this thesis' scope but due to the current body of adherence research, which has focused primarily on compliance measures. Indeed at this point compliance, rather than adherence, is much easier to objectively study given the breadth of data sources available.

The remainder of this discussion reviews the results of the evaluation presented in Chapter 5. Then theoretical and implementation limitations of the language and computational approach are discussed. Finally, additional data sources for the process of prescription interpretation are discussed.

6.1 Comments on Evaluation

Evaluation of the approach to measuring compliance was conducted in three parts, results from each are discussed in turn.

6.1.1 Requirements Verification

In Chapter 3, 28 requirements were outlined for the approach to measuring compliance, several of these are related to the prescription specification language. Twenty-four of the 28 were fully satisfied; the remaining four requirements were only partially satisfied, largely due to matters of scope.

Two of the partially satisfied requirements were related to the automation of the process of prescription interpretation, which, at this point, is a manual process. Fully automating this process to produce meaningful prescription specifications is undoubtedly challenging as it requires one to combine insights from multiple data sources and assign priorities to different prescribing concepts. However, for the purpose of demonstrating the efficacy of the overall approach to measuring compliance, automated prescription interpretation is not required. This thesis has contributed a framework in which automated prescription interpretation can occur.

Another of the partially satisfied requirements is related to the platform chosen for the implementation. Clojure, which is a Java Virtual Machine (JVM) language, is platform independent with respect to operating systems, but cannot run on hardware that cannot run the JVM (e.g. microprocessors). This limitation is mitigated by 1) defining an IoT system architecture that ensures the evaluator implementation is deployed on a JVM capable device; or 2) re-implementing the evaluator (whose logic is described in Chapter 4 independently of any programming language) in a different programming language. It should be noted that just about any implementation would have inevitably resulted some platform restrictions, using a JVM language was a reasonable choice given that it can be used on many modern operating systems.

The last partially satisfied requirement is related to the clinical accuracy of the resulting compliance measures. There are many possible combinations of prescriptions and behavioural traces, and each could produce a unique compliance score. At best, confidence in the accuracy of the approach for a set of use cases can be established by conducting a series of tests; to generalize that claim to all prescriptions and behaviours would be dubious. This limitation regarding performance in real-world scenarios is similar to the accepted fact that one cannot prove correctness of a non-trivial software

system by testing alone.

Since 24 of 28 requirements were satisfied and the remaining four were partially satisfied, with suitable reasoning, it can be concluded that the requirements from Chapter 3 are acceptably satisfied by the current computational approach.

6.1.2 Language Expressiveness

The language for expressing prescriptions was evaluated for its breadth of expressiveness by attempting to interpret ten APs. To be useful in a clinical context the language must be able to express a breadth of prescription concepts. The ten prescriptions were provided by a practicing family physician with domain expertise, the prescriptions were chosen to cover a range of prescribing concepts that might be found in a primary care setting. Each prescription was marked as “yes” (expressible), “partial” (some concepts expressed), or “no” (not expressible). Of the ten prescriptions, eight were fully expressed, the remaining two were partially expressed.

Prescription number 9, *zopiclone 7.5 mg once daily at bedtime as needed for insomnia*, could not be fully expressed because it contains a reference to a symptom the patient might experience, namely insomnia. Currently, the language uses restrictions to account for external action types such as “bedtime”; a new restriction is easily added to account for some types symptoms or indications such as insomnia.

Prescription number 10, *ethinyl estradiol with norethindrone 1 tablet once daily for 21 days starting on first day of menses and then restart on day 29*, could not be fully expressed because of the changing start date of the prescription. This prescription reveals two underlying assumption of the language: 1) the prescription has one start date, and 2) that the patient may regularly administer the medication after the start date. Conversely, prescription number ten has two start dates: 1) the date to begin monitoring for menses, and 2) the date to begin administering the medication and counting the elapsed days. This is different from the idea of an *as needed* medication to be administered regularly when some condition is met, e.g. insomnia, which has a single start date from which elapsed days may be counted.

Prescription number 10 could be expressed if a “terminate” concept was added to the language; then, one could create a prescription for inactions, i.e. do not administer the medication, to be terminated upon menses and then replaced by the regular prescription on a 29 day cycle. Indeed, the idea of a “terminate” concept would be useful for prescribers to indicate to stop the prescription if a particular symptom or

clinical sign is exhibited. Given the current language, prescription number 10 can be expressed without the additional start date and would require one to assume it is started on the first day of menses.

It should be noted that, as with the requirements verification above, this evaluation cannot be exhaustive and it is likely that some, if not many, prescriptions cannot be fully expressed in this language. However, many prescriptions use a relatively simple set of concepts; therefore, based on the findings of this phase of evaluation, it is reasonable to conclude that the breadth of concepts expressible in the language is acceptable for a primary care setting.

6.1.3 Clinical Evaluation

A third phase of evaluation was conducted to determine the clinical accuracy of the compliance measurement approach. Ten natural language APs typical of a primary care environment were chosen as use cases for this evaluation. This evaluation phase also involved conducting prescription interpretation in earnest to produce IPs. Expert consensus data, collected via an online survey, was used to interpret the original APs. Interestingly, this type survey for collecting compliance information from domain experts is the first of its kind.

Prescription interpretation was successfully completed for nine of the ten original APs, data collected for use case number 10 was too diverse to provide input to prescription interpretation. The diversity of the data is likely due to use case number 10 containing a joint prescription for different formulations of an opioid, one prescription for regular pain management and another as needed for pain. Some confusion with this relatively complicated prescription was expected given that this is the first use of this data collection method.

It should be noted that no rigorous statistical analysis was conducted on the data. The number of valid responses ranged from $n = 19$ to $n = 35$, these numbers may or may not be large enough to make a strong conclusion about the generalizability of these results as a true measurement of compliance. However, the purpose of this evaluation is not to make a general claim about compliance measures for the chosen use cases, but rather to demonstrate that this proof of concept approach to measuring compliance is acceptable. Additionally, it is not immediately clear how to conduct a statistical analysis on the collected data, further work in this area is required.

The evaluation was conducted analyzing the output of the scheduler. This proxy

evaluation was required because there is not a “gold standard” to compare the proposed measure of compliance against, at least not for measurements of this resolution. Additionally, the method used to collect data for this evaluation asked participants to “schedule” a patient’s next dose; this question was easier to ask participants than one regarding compliance scores. Since the scheduling algorithm is relatively naive and does not make any decisions regarding administration other than finding a minimum, evaluation by proxy for this proof of concept is acceptable.

Verification

The IPs were used to schedule the next medication administration time based on a short history. The scheduler’s output, a function showing the instantaneous compliance for a range of hypothetical administration times, was compared against the expert consensus data. The root mean squared error between the two functions was measured; errors from all nine of the IPs were well below 10%. Most of the error was related to non-linearities in the expert consensus data that could have been “smoothed out” by further prescription interpretation but would result in over-fitting of the IPs to the data. Manual inspection of the data and generated function confirms that the overall trends were accurately captured. These results verify that the implemented language’s semantics performed as expected and do not contain any unreasonable assumptions.

Validation

A panel of three experienced domain experts, two physicians and one pharmacist, independently reviewed the functions generated by the scheduler. They were asked to indicate whether the function represented a suitable compliance measurement. Overall, the experts agreed with the majority of the generated functions with mostly minor changes requested. The lowest scoring use case was number 3.

For use case number 3, *hydromorphone 4 mg every 4 hours as needed*, the panelists unanimously disagreed with the interpretation of “as needed”. The panelists indicated that after the minimum interval between administrations had elapsed, the compliance of any future dose should be 1.0, i.e. there is not a maximum interval between “as needed” administrations. In contrast, the generated function, which was based on the data collected in the survey, indicated that a partial compliance could be attained without a maximum administration interval but that a maximum interval was required for full compliance. This discrepancy is seemingly a difference of professional opinion

between experienced clinicians and the group of physicians that answered the survey and could be a result of: 1) the inexperience of survey respondents, which were predominantly resident physicians with between 1 and 5 years of practical experience; or 2) given the nature of online surveys, it is not clear whether participants fully understood the assigned task. Furthermore, logically, the panelists appear to be correct, a take “as needed” medication should not be penalized if a maximum interval between administrations is not observed. As a result, the panelist’s interpretation of the prescription is likely more clinically accurate.

Many of the generated functions contain “plateaus”, i.e. time ranges where the compliance score is quite low (less than 0.2) but is constant or gently sloping. The panelists had minor concerns with the presence of these plateaus, which are present in use cases: 1, 2, 4, 5, 7, and 9. Upon further investigation, the plateaus appear to be a result of a small number of survey responses which indicated a significantly wider time range. These plateaus are an artifact of the prescription interpretation process; the IPs, upon which the generated functions were based, were interpreted with the survey data as the only data source. In some ways, the presence of these plateaus in the generated compliance function is also a result of naive curve over-fitting on the part of the prescription interpreter (myself). These plateaus illustrate the need for: 1) multiple data sources to be used as input to prescription interpretation; 2) more rigorous statistical analysis to be carried out on expert consensus data; and 3) an individual with knowledge of pharmacology to participate in the prescription interpretation process.

Finally, two of the panelists observed that while a generated compliance function can be used to create schedules for future administrations, the function cannot be used to synthesize real-time recommendations when an administration is missed. Essentially, there is a difference between recommendations of the form: *take your next dose at time A*, and those of form: *you missed your time A dose, take your next dose at time B*. As it currently stands, the approach in this thesis is limited to measuring compliance of past behaviours and scheduling future medication administration. This limitation was also discussed in Chapter 3.

Overall, the results of the clinical validation were positive. Only minor flaws were observed with the generated compliance functions which were largely artifacts of this specific prescription interpretation and do not reflect flaws with the general approach to compliance measurement. Therefore, it is reasonable to conclude that this thesis’ approach to measuring compliance produces clinically acceptable results. Additionally,

the panelists identified a theoretical limitation regarding real-time recommendations which represents a starting point for future work.

6.2 Limitations

In addition to those identified by the evaluation discussed above, this approach to measuring compliance, in its current form, has several known limitations; these are discussed below.

Atomic Actions Assumption

One of the major limitations of this approach to measuring compliance is the underlying assumption that all actions are atomic, i.e. that they occur at a single point in time. This assumption underlies the notion that a patient's medication taking behaviour is composed of a sequence of actions and inactions. Arguably, one could say that no medication administration action occurs at single point in time since a smaller unit of time could be selected that results in an interval; alternatively, a very strict criteria for what constitutes administration could be selected, e.g. the moment a pill touches the mouth of the patient. However, these arguments are seemingly inconsequential when one considers that the effects of many medications are measurable in minutes, hours, or days; and that the administration of many outpatient medications takes on the order of seconds, e.g. swallowing a pill.

A more practical consequence of this theoretical assumption is that it is not clear how the approach could be generalized to medications that are delivered over a period of time, such as an intravenous drip medication in a hospital. One could indicate that each unit of time elapsed, perhaps minutes, constitutes an action in which a particular dose of substance was administered, which is calculated based on the drip rate and concentration of substance. However, such considerations are beyond the scope of this thesis, which is primarily focused on outpatient medications which are commonly prepared as easily consumed formulations such as pills, liquids for oral consumption or injection. An interesting case is medication delivered transdermally (via skin), such as creams or time-release patches, conventions for the specific medications could be determined prior to compliance measurement.

Lifestyle Actions

A novel aspect of the scheduling approach proposed by this thesis is the ability to directly consider lifestyle activities, such as meal times, when determining future medication administration times. For lifestyle actions that were recorded in the past this does not pose any concern, they are simply accounted for by the scheduling algorithm. It is slightly more difficult to schedule medication administration around lifestyle actions that have not occurred yet. It is unrealistic to assume that a patient's supper occurs at 19:00 every day. Using fuzzy reasoning mitigates this problem to some extent, appropriate fuzzy functions can be chosen to capture the idea that "supper time is around 19:00". However, not all patients have such regular lifestyles, for example shift workers may have varied meal times depending on when they work. Determining how to schedule administrations around varied lifestyles remains a problem, though, this thesis does help by providing a foundation for future research.

Measurement and Scheduling Algorithms

Admittedly, the algorithms used to measure compliance and schedule the optimal time for the next medication administration time are quite naive and computationally inefficient. This thesis has focused on developing an accurate approach to adherence measurement and demonstrating its applicability to the problem of scheduling, not on efficiency of the approach. Nonetheless, a more efficient scheduling algorithm would be desirable, especially if the work is to be deployed in a real-world environment. The current approach, for both compliance measurement and scheduling, is to step through the desired time range and evaluate compliance at a point. Given that the semantics of the approach are encoded as functions, it is conceivable that an analytical approach could be developed. Additionally, for scheduling, more advanced optimization techniques could be applied.

A challenge of improving efficiency will be accounting for the causal nature of compliance measurement. As discussed in Chapter 3, the compliance of an action/inaction is determined by the patient's previous actions/inactions. Thus, one cannot consider action/inaction as independent from those that occurred previously. Consequently, one cannot evaluate the compliance function at time t_0 without examining the history $t \leq t_0$. With that said, optimizations are certainly possible; they may be included as theoretical changes, i.e. the language's denotational semantics is optimized, or simply included in an implementation.

Interpreting Compliance Scores

Compliance has been established by AIM as a distance measure between a prescription and a description (trace). It was also defined as the *degree* to which the trace belongs in the (fuzzy) set of compliant traces. A trace's membership in this set is based on a formal description of the prescription, more specifically on the functions that were selected during the process prescription interpretation. However, beyond membership in a set, it is interesting to ask if there is a real-world meaning to a particular compliance score. Many measurements in medicine are based on physical quantities, e.g. volume, mass, or concentration; having physical references is important as they can be used to compare between patients, establish generalizable trends, and understand the implications of particular actions.

Since the degree of membership in the compliance set is based on fuzzy membership functions, the physical meaning of the score is not actually established by the formalism itself, but the process of prescription interpretation and the data used to inform that process. For example, the clinical evaluation presented in this thesis used expert consensus data, since this was the only data provided for prescription interpretation, any resulting compliance scores represent *the percentage of survey participants that would denote the patient's behaviour as compliant*. With appropriate statistical powers, this could be generalized to *the percentage of experts that would denote the patient's behaviour as compliant*. Other physical meanings could be established based on other data, e.g. known half-lives of substances in the human body. This flexibility represents a strength of the approach; it provides a framework for reasoning and leaves the precise definitions to experts.

There are three obvious drawbacks to this approach. First, prescription interpretation, when conducted manually, is subject to bias by the interpreter; this bias could be mitigated by developing rigorous protocols for prescription interpretation and by automation. Second, it is not immediately clear how to assign meaning to compliance scores that are based on prescriptions that were interpreted in the context of multiple data sources. Finally, unless shared standards for prescription interpretation are established, comparison between different prescriptions and studies may be difficult, although not impossible if all of the data was available to analysts.

When considering the context of the patient, assigning a physical quantity to a compliance score may not provide as much value as it would for researchers. Patients and their health care providers may only be interested in trends in compliance scores,

rather than the absolute number itself. Furthermore, if one considers the patient's compliance from a control theoretic perspective, as was done in Chapter 3, the physical meaning of the score is largely irrelevant if the goal is simply to maintain a compliance score of 1.0.

6.3 Data for Prescription Interpretation

As per the discussion above, the process of prescription interpretation is critical to the accuracy of this approach to measuring compliance. Accordingly, ensuring that the data used during prescription interpretation is of high quality is of considerable importance. This thesis has used primarily expert consensus data to interpret prescriptions, however Chapter 3 listed several possible data sources. This section discusses possible methods for cultivating said data. The data sources are divided between primary and secondary sources according to the proximity of the source to the individual prescription in question.

Primary Data Sources

Primary sources are those that produce data specific to the prescription in question. These sources have intimate knowledge of the patient's context, the prescription, and the indication for the prescription. For example: the prescribing physician, the dispensing pharmacist, a nurse working with the patient, or the patient themselves.

Gathering data from these sources is likely a challenging task. Numerous techniques could be employed. Interviews are a candidate technique, but would likely be too time consuming to be used at scale given the number of patients and time constraints placed on health care providers. However, patient-provider engagement is an important part of interventions aimed at improving adherence [26], interviews may be an effective means of achieving both increased engagement and data collection.

Computer Provider Order Entry (CPOE) systems, as part of Electronic Medical Records (EMRs), may be another means of gathering primary data. Currently, CPOE systems collect data to produce what is effectively an AP via user interfaces consisting predominantly of form style inputs. Designing user interfaces that collect additional data that supports prescription interpretation could be beneficial. As with interviews, use by front line clinicians may be time consuming. Creating usable and safe CPOE user interfaces continues to be an active area of research [68].

Secondary Data Sources

Secondary data sources are those that produce generalized data that are not patient specific. Examples of such sources include: expert consensus obtained via surveys, expert opinions obtained through interviews, and medical literature. The data sources used in the clinical evaluation presented in Chapter 5 constitute secondary sources. One advantage of these sources is that data can be shared between multiple patients/providers. Inevitably, more general data means that the data may not fit the patient's specific context.

Using multiple secondary data sources to create a library of prescription archetypes may be beneficial. Many prescriptions, though unique in the context of a patient, share patterns that could be exploited to create such a library. For example, a family of related antibiotics may have similar characteristics with respect to the frequency of administration; an archetype for this family could be created through consultation with experts and used to inform interpretation of the respective APs. Such an approach would be useful to standardize adherence interventions for research studies and could provide an important step towards automating prescription interpretation.

6.4 Comparison with Prior Work

The language presented by this thesis is not the first language to formally describe prescriptions. Yeh *et al.* have created APAMAT, a language for describing prescriptions in an in patient setting. Importantly, APAMAT was created for the purpose of checking for drug-drug interactions; in this respect it differs from the main objective of the language presented by this thesis, which was to express prescriptions for the purpose adherence measurement and administration scheduling. Further, while the APAMAT does have a formally defined syntax, unlike the language presented in this thesis, it does not have a formally defined semantics. Therefore, while APAMAT does permit some understanding of medication prescriptions, the language presented by this thesis is in fact more rigorously defined and thus facilitates a more clear understanding of the prescription being expressed.

Chapter 7

Conclusions and Future Work

Despite many years of research, medication non-adherence continues to be a problem with no clear solution, the multi-faceted nature of which demands a multi-faceted solution. Technology holds much promise as part of an intervention to improve adherence. However, research in this area has been plagued by inconsistent definitions of adherence, a large number of dissimilar measurements, and a lack of studies with sound methodologies. To complicate matters, patients, clinicians, and researchers are increasingly immersed in high resolution data generated by health information technologies and a growing Internet of Things. Providing methods for analyzing this data and deriving useful information is becoming increasingly important.

To address these needs, this thesis proposed a computational definition of adherence, i.e. one that is precise enough to be used within computing devices aimed at measuring and improving adherence. More specifically, the main contributions of this thesis are:

- the Adherence Interaction Model, a conceptual framework for defining adherence,
- a language for specifying prescriptions was presented, the syntax and semantics of which were formalized; a prescription expressed in this language can be used to derive a compliance evaluation function,
- the prescription specification language is capable of using a fuzzy set theory and functions to account for variation and imprecision in human behaviours,
- prescription interpretation, in the context of additional data sources, was identified as an important part of a process accurately expressing natural language authored prescriptions in the specification language,

- an approach for measuring compliance using the evaluation function derived from an interpreted prescription,
- a compiler, which given a prescription expressed in the specification language, will produce an evaluation function, and
- an application of the compliance measurement approach to scheduling medication administrations.

The proposed computational definition can be used to support adherence research, clinical practice, and engineers and scientists developing technological adherence improving interventions. Thus far, the proposed approach and its implementation represent a proof of concept, and as expected, significant work remains to bring this work to the point where it can be used in practice. Nonetheless, the evaluation presented in this thesis, while not definitive, suggests the theoretical and practical aspects of this computational definition are acceptable and merit future consideration.

7.1 Future Work

This thesis is generative in nature, new ideas have been introduced, implemented, and tested; however, much work remains to be done. The following are potential avenues for future work, though presented in distinct sections, these are not mutually exclusive tasks.

Miss Behaviours and Recommendations

The approach proposed by this thesis is limited to: 1) measuring compliance for past behaviours, and 2) generating schedules for medication administration. Notably, the current approach cannot be used to provide real-time decision support to patients who have missed an administration time. This limitation was discussed in detail in Chapters 3 and 6. Being able to provide decision support to patients when they miss a dose is one of the ways in which technology can potentially improve adherence, as such, it is important that this limitation is addressed. A potential solution might be to include a “miss behaviour” concept in the language. Unfortunately, this may pose challenges regarding compliance measurement, namely, how does one score the compliance of a behaviour that missed an administration but also undertook the appropriate miss behaviour? Indeed, there may be multiple conceptual questions

entangled here which will require further theoretical as well as practical work to address.

Language Extension

Due to matters of scope, the current prescription specification language contains a limited number of concepts. Some concepts were intentionally excluded, like administration routes, while others were identified during the evaluation. Extending the language to include additional concepts would help improve the expressiveness of the language for use in a clinical or research environment. In particular, the following concepts could be added:

- administration routes,
- miss recovery behaviours,
- medical indication restrictions, e.g. insomnia, and
- prescription termination conditions.

Additionally, future work on the prescription specification language itself may investigate the creation of a “front-end” for the language that is usable by prescribers and pharmacists. Since the semantics of the current language have been formalized, the semantics of this new “programming language” could be established by identifying a correspondence between the two languages. Such a language could prioritize usability and safety and if adopted could have a significant impact on computerized order entry systems.

Algorithm Efficiency

As discussed in Chapter 6, the algorithms used to evaluate compliance and schedule medication administrations are in need of dramatic efficiency improvements. Indeed, as it currently stands, these algorithms may hinder integration with real-world systems as they would be quite slow if run on large traces. Improvements could be made via several techniques. Determining a means of analytically evaluating a trace’s compliance would be ideal but potentially challenging. Smaller improvements could also be made such as evaluation of relevant time windows rather than the entire trace. For scheduling, which is essentially an optimization problem, the application of linear

programming may be useful, though it is not clear how applicable these techniques would be since the compliance function may not be strictly linear.

Prescription Interpretation

Prescription interpretation is a critical aspect of the proposed approach, at the moment manual interpretation is required; however, in an ideal situation, this would be fully automated. A first step towards automation of prescription interpretation would be to identify common patterns found in prescriptions and to build an archetype library based upon data from secondary sources. An automated interpreter could then access the archetype library and build up the prescription specification by combining the relevant archetypes - LEGO for prescriptions.

Additionally, determining how existing statistical methods can be applied to data obtained from experts would be beneficial. One of the difficulties encountered during evaluation was finding a suitable statistical method to determine appropriate sample sizes such that generalizable conclusions could be drawn. Work in this area would complement the use of archetypes suggested directly above.

Finally, primary data sources, such as a prescriber or patient, have a wealth of knowledge regarding the prescription, clinical indications, and the patient's context. Developing user interfaces to elicit information needed for prescription interpretation from these individuals could be beneficial.

Continued Evaluation

The evaluation conducted by this thesis amounts to a set of test cases, these were useful in establishing limitations determining the efficacy of a proof of concept for the computational approach. However, continued rigorous evaluation is required to draw conclusions about the generalizability of the approach to a wider range of clinical scenarios. Additionally, a method for measuring agreement was proposed but was not evaluated; further evaluation of this proposed metric is required.

Real-World Integration

This work was first inspired by the SmartMed pill bottle project. Rather than continue with the practical engineering aspects, this thesis has attempted to address a number of theoretical questions that emerged from the SmartMed project. While many theoretical questions remain, it is also important to translate the theoretical

contributions into practical technologies. Additionally, translating theory to practice will inevitably reveal additional questions and limitations which in turn will lead to continued theoretical work.

The compiler implemented as part of this thesis has been created in such a way that it can be easily deployed on a web server and accessed over the Internet. Foreseeable implementation challenges include: 1) automating the process of prescription interpretation, 2) integration with existing health information technologies and systems, and 3) creating user interfaces and experiences that efficiently elicit and convey relevant information from/to patients and providers.

Appendix A

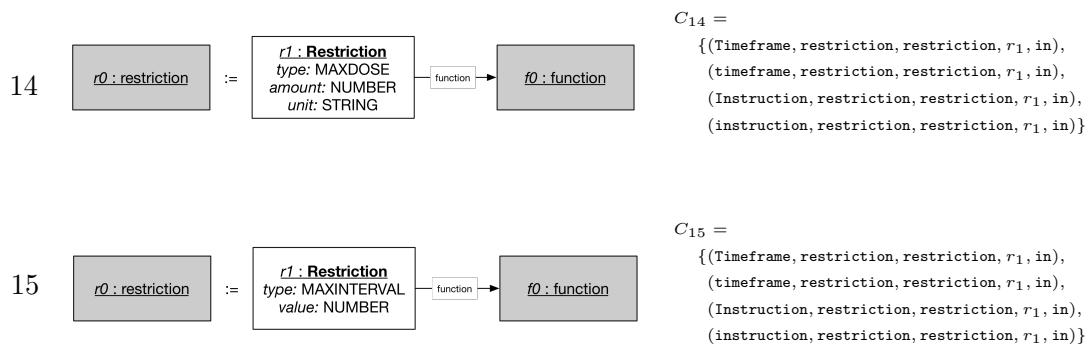
Extended Language Definition

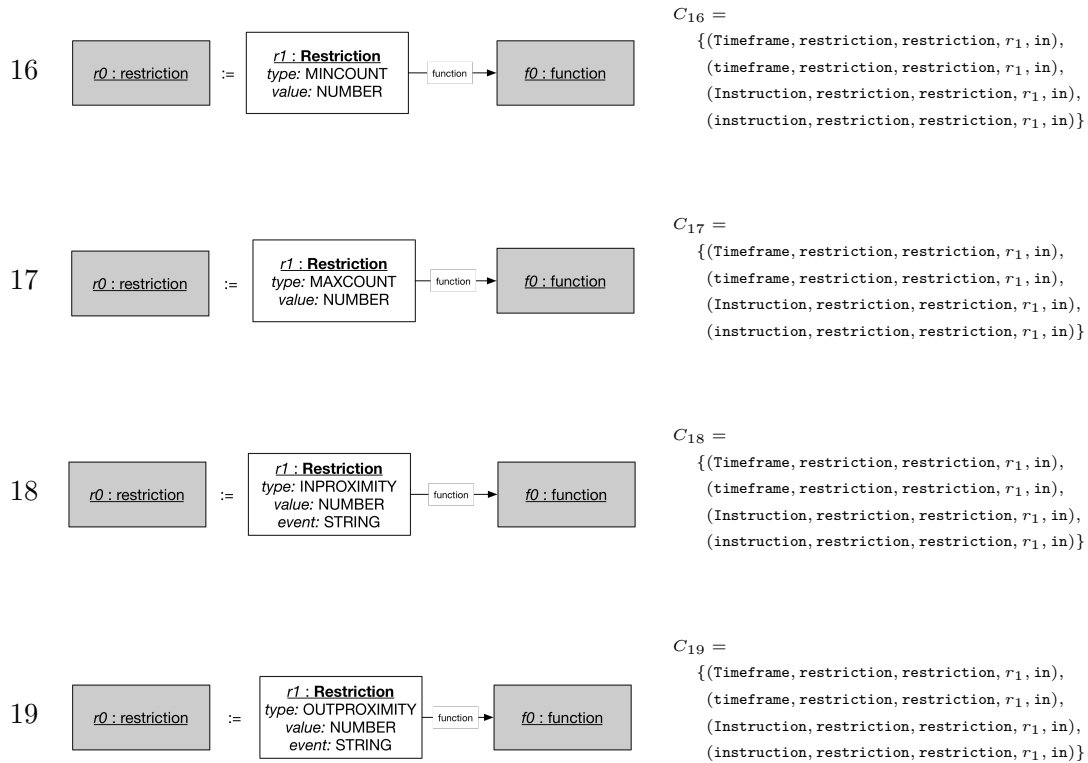
This appendix completes the formal definition of the prescription specification language first presented Chapter 4. First, several additional grammar production rules are given, and then the associated valuation functions are provided. The semantic algebra originally presented in Chapter 4 does not require extension.

Extended Context-Free Grammar

The context-free grammar is extended to contain a number of additional Restrictions. Since these are all Restriction labeled nodes, no changes to the grammar's node labels are required. The additional rules are as follows:

Table A.1: Prescription Specification Language Grammar Extension





Additional Valuation Functions

Valuation functions for the additional grammar production rules are presented with light commentary below. Since all of the additional valuation functions apply to Restrictions nodes, they all have the same signature:

$$\mathbf{r} : \textcircled{r} \rightarrow \text{time} \rightarrow \text{time} \rightarrow \text{time} \rightarrow \text{Traces}$$

Maximum Dosing Restriction

The maximum dosing restriction applies an upper bound on the dose administered. The definition of the upper bound itself is deferred to evaluation of a function.

$$\mathbf{r} [[\textcircled{MaD} \text{function} \textcircled{f}]] = \lambda t_2. \lambda t_1. \lambda t. \lambda \tau.$$

$$((\text{admin? } t) \tau) \rightarrow ((\mathbf{f} [[\textcircled{f}]]) \textcircled{MaD}.amount) ((\text{event } t) \tau).dose) \square 0$$

Maximum Interval Restriction

The maximum interval restriction indicates the maximum time permitted between consecutive administration times. The definition of the maximum interval itself is deferred to the evaluation of a function. Both action and inaction are considered non-compliant after the time interval has elapsed; actions are considered non-compliant because the dose was “late”, and inactions are considered non-compliant because the dose is not being administered.

$$\begin{aligned} \mathbf{r}[[\text{Max}_{\text{function}} \cdot \mathbf{f}]] &= \lambda t_2. \lambda t_1. \lambda t. \lambda \tau. \\ &(\mathbf{not} (\mathbf{nil?} (((\mathbf{previous} t) \tau) t_1) t_2))) \rightarrow \\ &(\mathbf{f}[[\mathbf{f}]] (- t (((\mathbf{previous} t) \tau) t_1) t_2))) \\ &\square 0 \end{aligned}$$

Proximity Restrictions

Proximity restrictions evaluate the degree of compliance with respect to an administration action being sufficiently temporally close (In Proximity Restriction) or sufficiently temporally distant (Out of Proximity Restriction) to another action type (e.g. a meal). Both types of restrictions are defined below, as with other restriction types, the definition of “temporally close” and “temporally distant” is deferred to a function. The **nearest** function is used here, informally, it returns the time of the temporally nearest action a particular type.

$$\begin{aligned} \mathbf{r}[[\text{IP}_{\text{function}} \cdot \mathbf{f}]] &= \lambda t_2. \lambda t_1. \lambda t. \lambda \tau. \\ &(((\mathbf{action?} t) \tau) \rightarrow ((\mathbf{f}[[\mathbf{f}]] (\mathbf{abs}(- t ((\mathbf{nearest} t) \tau) \text{IP}.event)))))) \square 0 \\ \mathbf{r}[[\text{OP}_{\text{function}} \cdot \mathbf{f}]] &= \lambda t_2. \lambda t_1. \lambda t. \lambda \tau. \\ &(((\mathbf{action?} t) \tau) \rightarrow ((\mathbf{f}[[\mathbf{f}]] (\mathbf{abs}(- t ((\mathbf{nearest} t) \tau) \text{OP}.event)))))) \square 0 \end{aligned}$$

Count Restrictions

Count restrictions evaluate to what degree the absolute number of administrations in a time window satisfy an upper or lower bound. A maximum count restriction will evaluate to non-compliant if an action at time t in the trace τ will cause the count of actions between t_1 and t_2 to exceed a maximum value indicated by the associated

function. A minimum count restriction will evaluate inactions as non-compliant if the time window has elapsed ($t > t_2$) and a minimum number of administrations has not been observed. A **count** function is used by this valuation, informally, it counts the number of administration actions in a trace occurring within a window.

$$\mathbf{r}[[\text{Mac} \text{function} \cdot \mathbf{f}]] = \lambda t_2. \lambda t_1. \lambda t. \lambda \tau.$$

$$((\mathbf{action?} \ t) \ \tau) \rightarrow (\mathbf{f}[[\mathbf{f}]]) (((\mathbf{count} \ t_1) \ t_2) \ \tau) \ \square \ 0$$

$$\mathbf{r}[[\text{MiC} \ \text{function} \cdot \mathbf{f}]] = \lambda t_2. \lambda t_1. \lambda t. \lambda \tau.$$

$$(\mathbf{and} \ (\mathbf{not} \ ((\mathbf{action?} \ t) \ \tau)) \ (> \ t \ t_2) \rightarrow (\mathbf{f}[[\mathbf{f}]]) (((\mathbf{count} \ t_1) \ t_2) \ \tau) \ \square \ 0$$

Appendix B

Language Breadth Evaluation

This appendix contains the results of the breadth evaluation phase presented in Chapter 5. Ten prescriptions were provided by a domain expert. Of those, eight were readily expressible in the prescription specification language; these are shown below in Figures B.1 through B.8. For a detailed account of why the remaining two could not be expressed see Chapters 5 and 6.

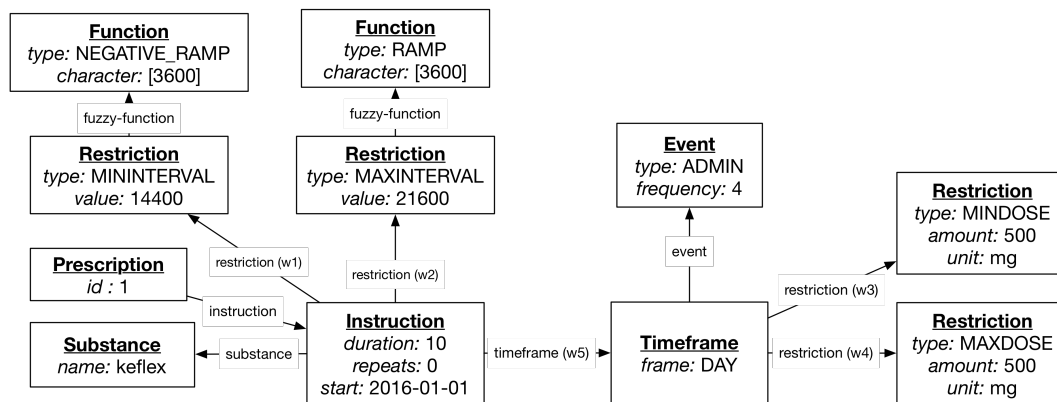


Figure B.1: Breadth Evaluation Prescription 1: *keflex 500 mg four times daily for 10 days.*

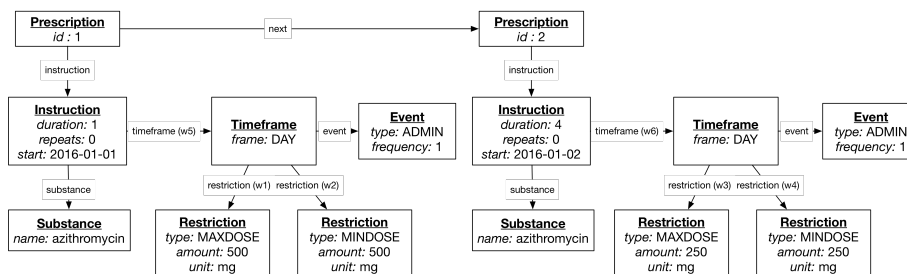


Figure B.2: Breadth Evaluation Prescription 2: *azithromycin 500 mg once daily for 1 day then 250 mg once daily for 4 days.*

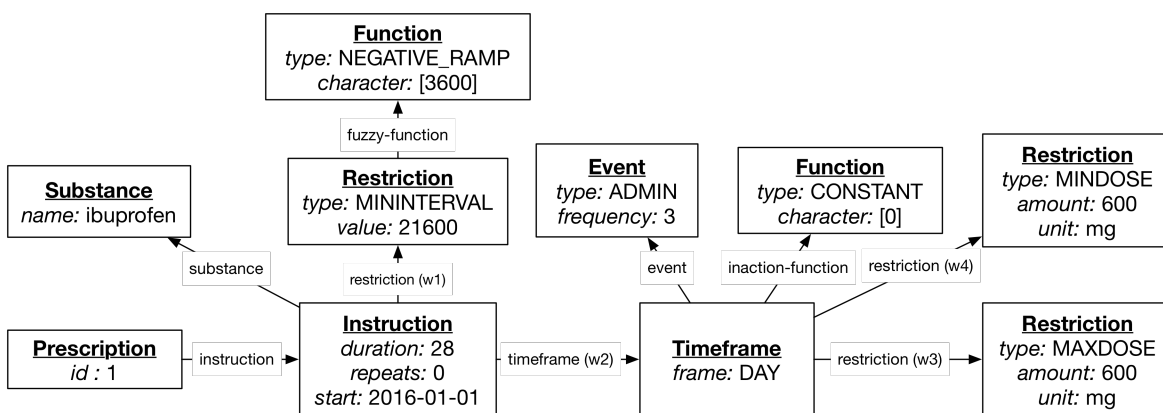


Figure B.3: Breadth Evaluation Prescription 3: *ibuprofen 600 mg three times daily as needed.*

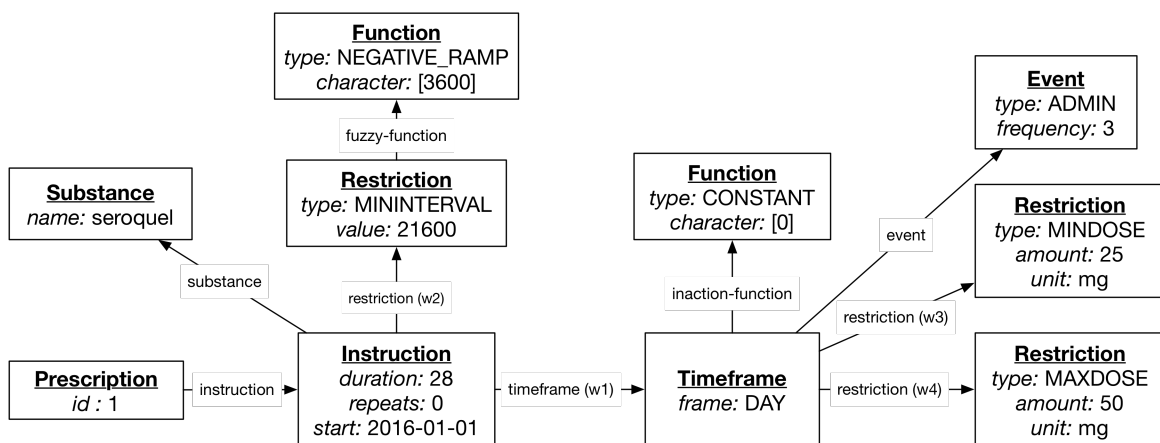


Figure B.4: Breadth Evaluation Prescription 4: *seroquel 25-50 mg three times daily as needed.*

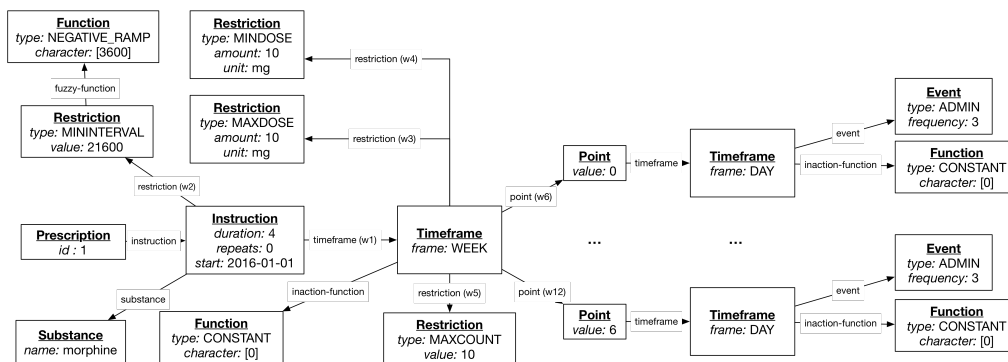


Figure B.5: Breadth Evaluation Prescription 5: *morphine 10 mg three times daily as needed up to 10 per week.*

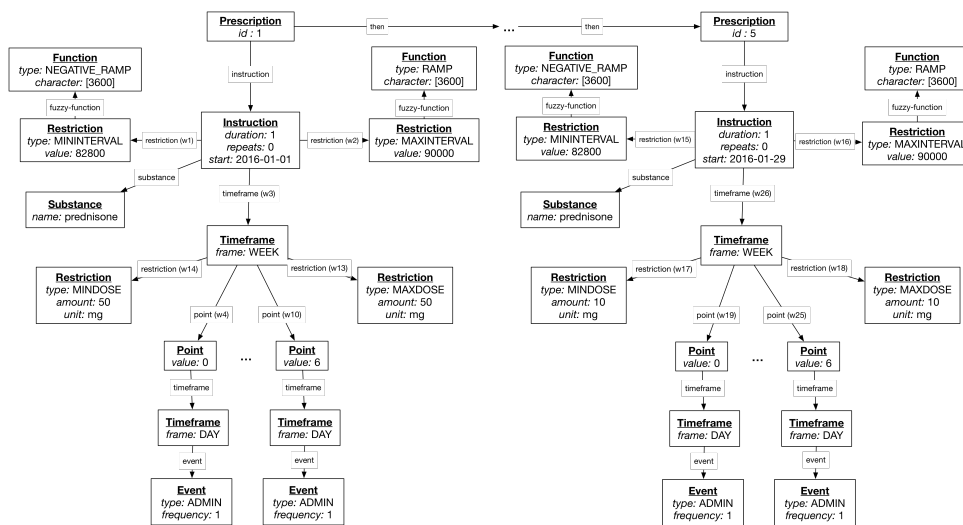


Figure B.6: Breadth Evaluation Prescription 6: *prednisone 50 mg decrease by 10 mg per week until done.*

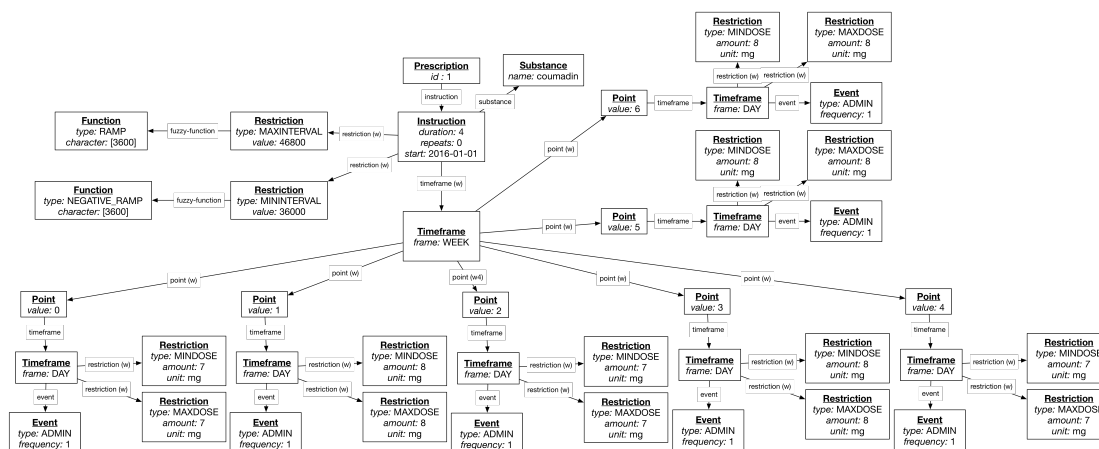


Figure B.7: Breadth Evaluation Prescription 7: *coumadin 7 mg once daily on M/W/F 8 mg once daily otherwise.*

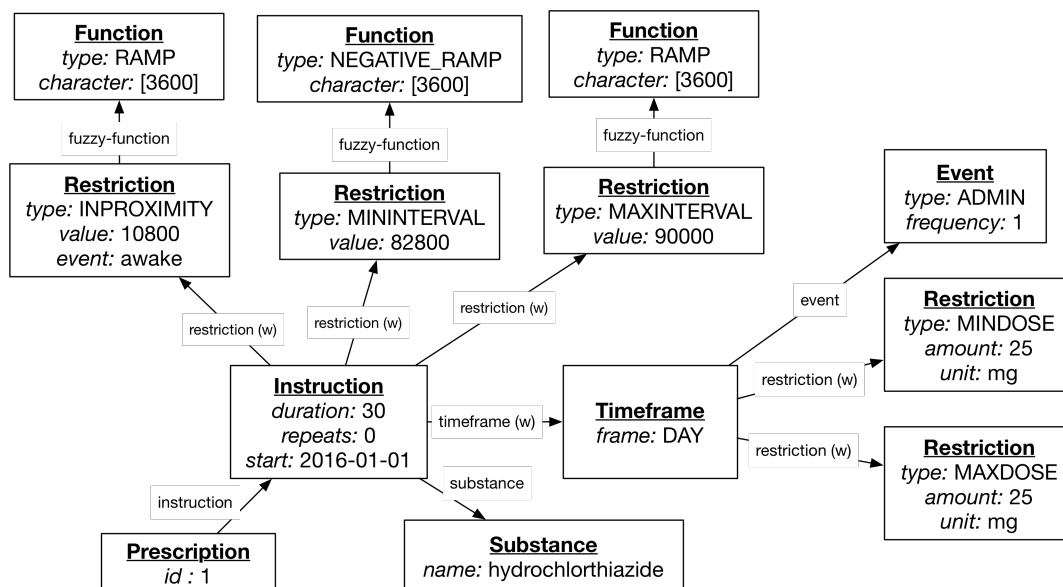


Figure B.8: Breadth Evaluation Prescription 8: *hydrochlorothiazide 25 mg once daily in the AM for 30 days.*

Appendix C

Verification Data

This appendix contains the data collected during the expert consensus survey that was used as input to prescription interpretation for the clinical evaluation. Responses were compiled according to the method described in Chapter 5. Additionally, for each use case considered, the generated compliance function (blue) is shown superimposed on top of the expert data collected from the survey (red).

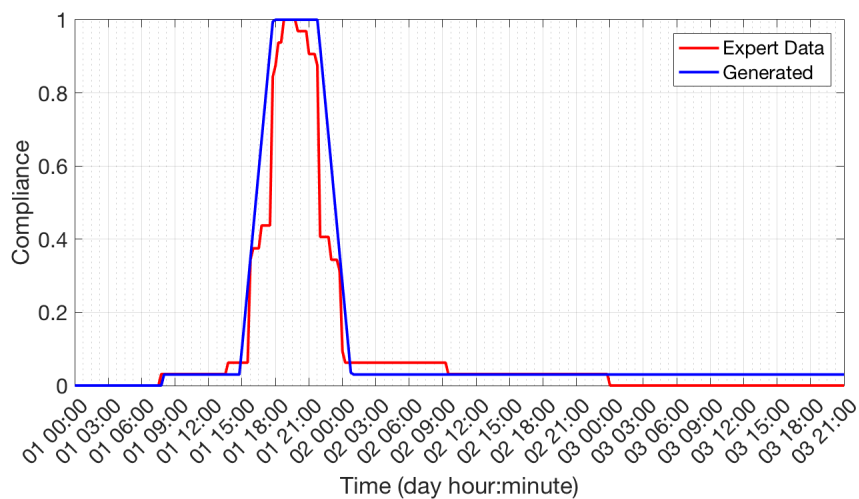


Figure C.1: Expert Data and Generated Compliance Function for Use Case 1 *Nitrofurantoin 100 mg twice daily* with previous dose at: 08:00.

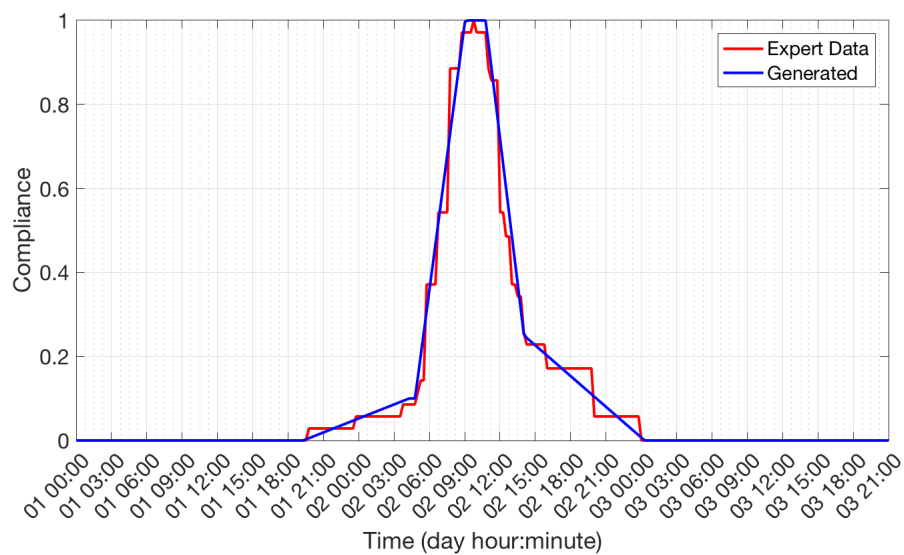


Figure C.2: Expert Data and Generated Compliance Function for Use Case 2 *Ramipril 2.5 mg once daily* with previous dose at: 10:00.

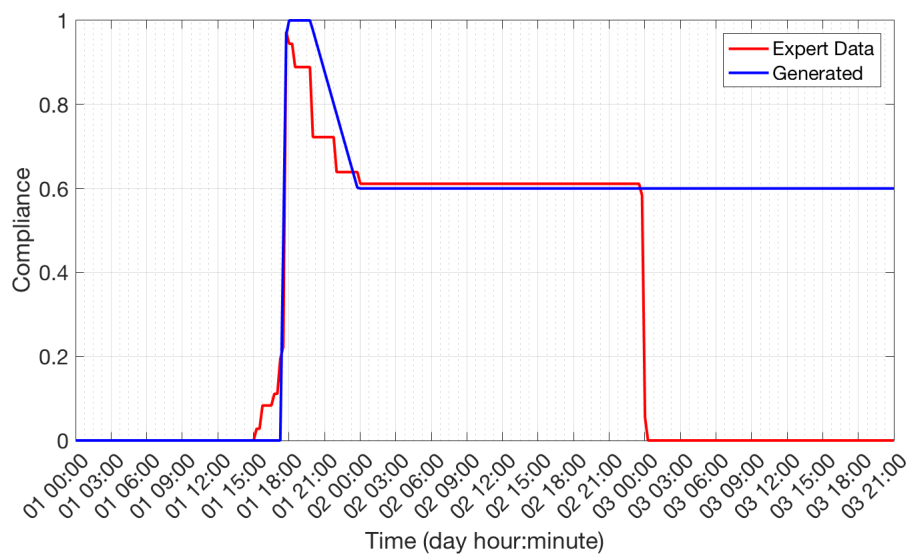


Figure C.3: Expert Data and Generated Compliance Function for Use Case 3 *Hydromorphone 4 mg every 4 hours as needed* with previous doses at: 06:00, 10:00, and 14:00.

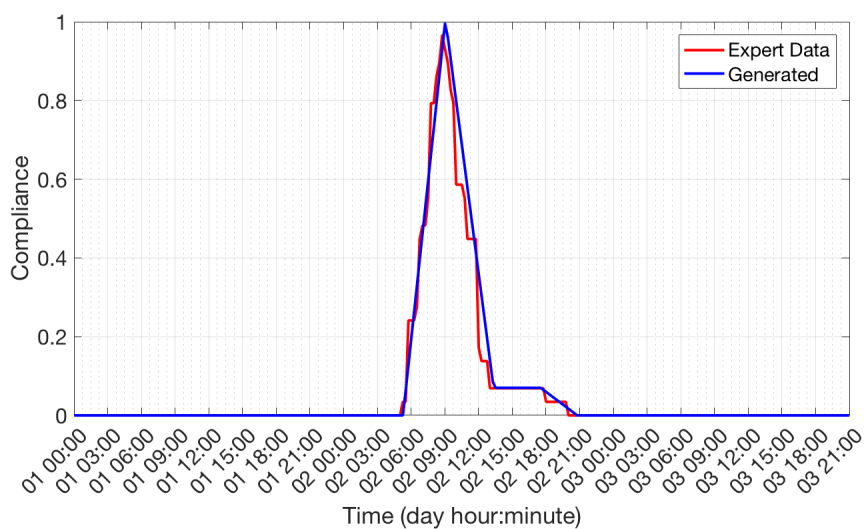


Figure C.4: Expert Data and Generated Compliance Function for Use Case 4 *Marvelon 1 tablet once daily* with previous dose at: 09:00.

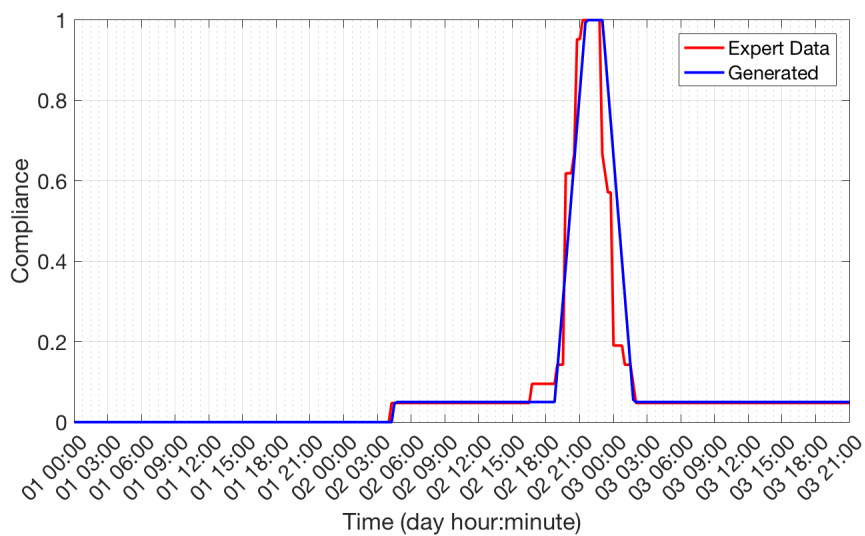


Figure C.5: Expert Data and Generated Compliance Function for Use Case 5 *Glargine 10 units at bedtime* with previous dose at: 22:00.

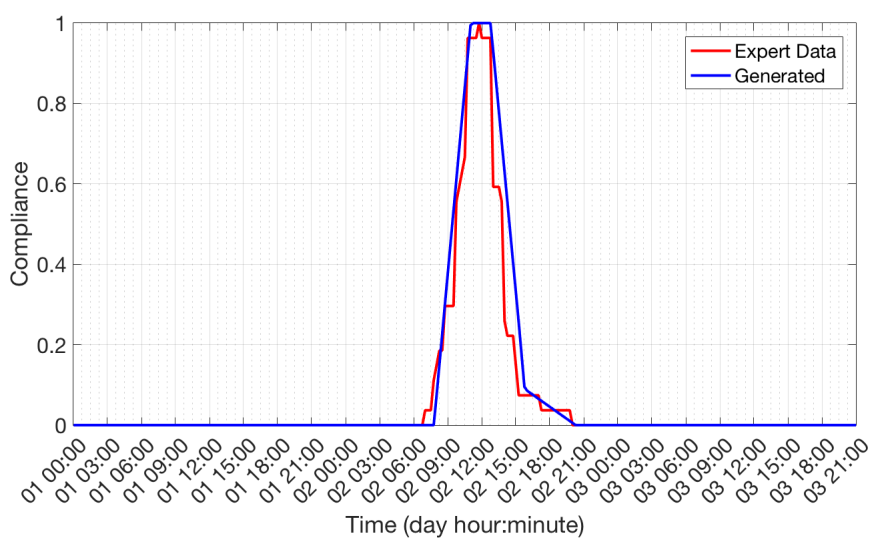


Figure C.6: Expert Data and Generated Compliance Function for Use Case 6 *Moxifloxacin 400 mg every twenty four hours* with previous dose at: 12:00.

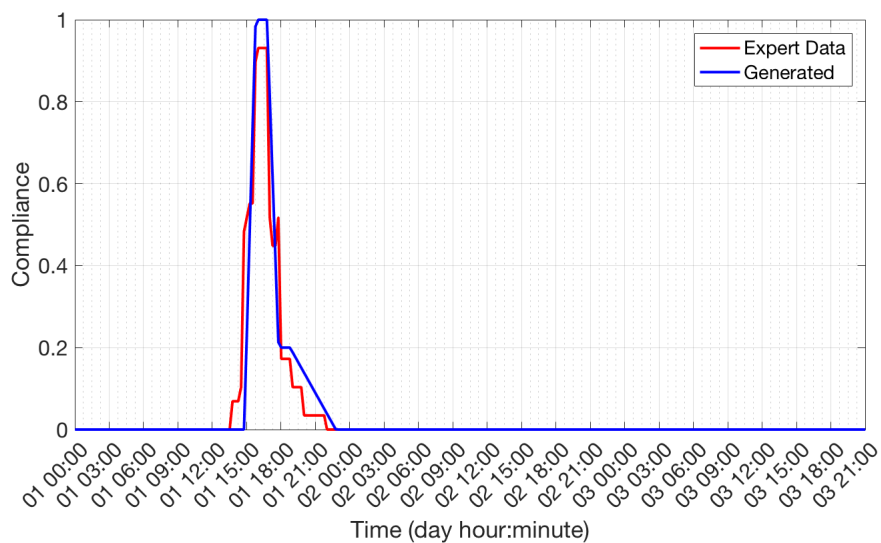


Figure C.7: Expert Data and Generated Compliance Function for Use Case 7 *Cephalexin 500 mg four times a day* with previous doses at: 08:00 and 12:00.

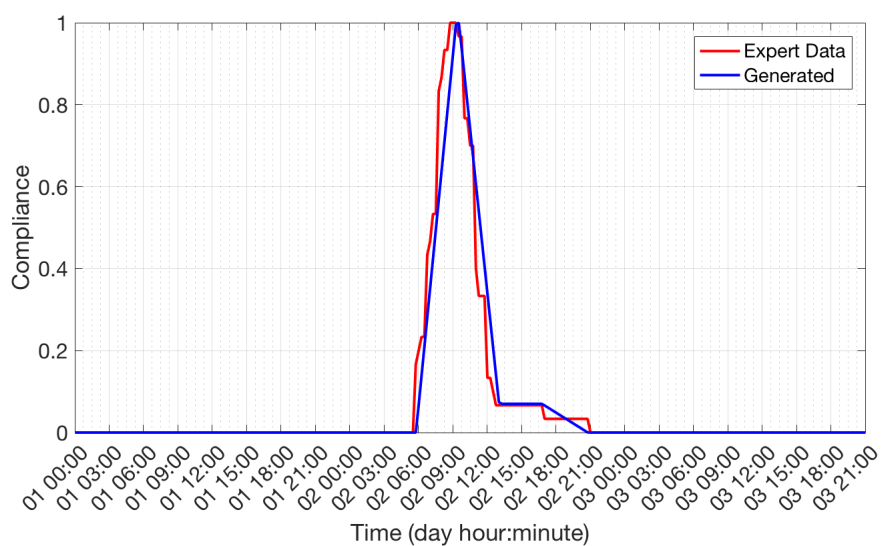


Figure C.8: Expert Data and Generated Compliance Function for Use Case 8 *Warfarin 7 mg once daily* with previous dose at: 09:00.

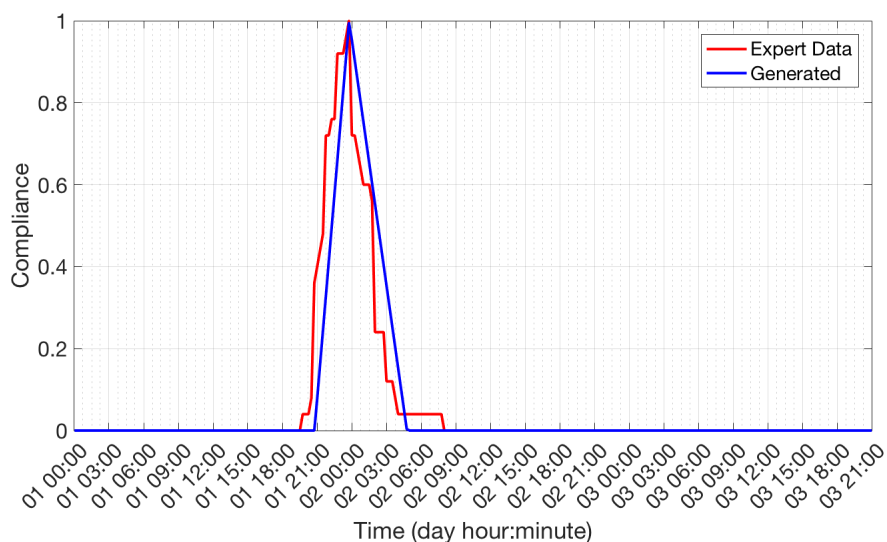


Figure C.9: Expert Data and Generated Compliance Function for Use Case 9 *Penicillin V 500 mg three times a day* with previous dose at: 09:00 and 17:00.

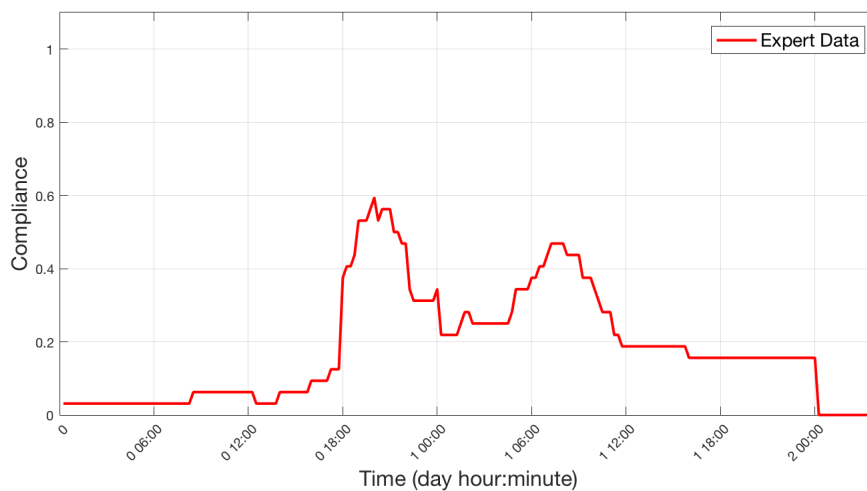


Figure C.10: Expert Data and Generated Compliance Function for Use Case 10 *Hydromorphone Contin 9 mg twice daily and Hydromorphone 4 mg every 4 hours as needed* with previous doses at: 08:00 (9 mg) and 14:00 (4 mg).

Bibliography

- [1] *Prescribed Drug Spending in Canada, 2016 - A Focus on Public Drug Programs*. Tech. rep. Ottawa, Canada: Canadian Institute for Health Information, 2016.
- [2] World Health Organization. *Adherence For Long-term Therapies - Evidence For Action*. Tech. rep. 2003.
- [3] Bruce G Bender and Cynthia Rand. “Medication non-adherence and asthma treatment cost”. In: *Current Opinion in Allergy & Clinical Immunology June 2004* 4.3 (2004), pp. 191–195.
- [4] L Wei et al. “Adherence to statin treatment and readmission of patients after myocardial infarction: a six year follow up study”. In: *Heart* 88.3 (2002), pp. 229–233.
- [5] Christian Leporini, Giovambattista De Sarro, and Emilio Russo. “Adherence to therapy and adverse drug reactions: is there a link?” In: *Expert Opinion on Drug Safety* 13 (2014), pp. 41–55.
- [6] Brandy McGinnis et al. “Dyslipidemia: Factors Related to Adherence to Statin Therapy”. In: *Annals of Pharmacotherapy* 41.11 (2007), pp. 1805–1811.
- [7] N. Col, J. E. Fanale, and P. Kronholm. “The role of medication noncompliance and adverse drug reactions in hospitalizations of the elderly”. In: *Archives of Internal Medicine* 150.4 (1990), pp. 841–845.
- [8] M. Christopher Roebuck et al. “Medication Adherence Leads To Lower Health Care Use And Costs Despite Increased Drug Spending”. In: *Health Affairs* 30.1 (2011), pp. 91–99.
- [9] Jia-Rong Wu et al. “Medication Adherence in Patients Who Have Heart Failure: a Review of the Literature”. In: *Nursing Clinics of North America. Biology, Psychology, and Behavior in Cardiovascular and Pulmonary Disease* 43.1 (2008), pp. 133–153.

- [10] Lars Osterberg and Terrence Blaschke. “Adherence to Medication”. In: *New England Journal of Medicine* 353.5 (2005), pp. 487–497.
- [11] Jon J. Vlasnik, Sherry L. Aliotta, and Bonnie DeLor. “Medication adherence: Factors influencing compliance with prescribed medication plans”. In: *The Case Manager* 16.2 (2005), pp. 47–51.
- [12] Walid F. Gellad, Jerry L. Grenard, and Zachary A. Marcum. “A Systematic Review of Barriers to Medication Adherence in the Elderly: Looking Beyond Cost and Regimen Complexity”. In: *The American Journal of Geriatric Pharmacotherapy* 9.1 (2011), pp. 11–23.
- [13] Craig I Coleman et al. “Dosing frequency and medication adherence in chronic disease.” In: *J Manag Care Pharm* 18.7 (2012), pp. 527–539.
- [14] Augustine Lee and Steve Morgan. “Cost-related nonadherence to prescribed medicines among older Canadians in 2014: a cross-sectional analysis of a telephone survey”. In: *CMAJ Open* 5.1 (2017), E40–E44.
- [15] Michael R. Law et al. “The effect of cost on adherence to prescription medications in Canada”. In: *Canadian Medical Association Journal* 184.3 (2012), pp. 297–302.
- [16] D. Vitalis. “Factors affecting antiretroviral therapy adherence among HIV-positive pregnant and postpartum women: an adapted systematic review”. In: *International Journal of STD & AIDS* 24.6 (2013), pp. 427–432.
- [17] Barbara J. Turner et al. “Barriers to adherence and hypertension control in a racially diverse representative sample of elderly primary care patients”. In: *Pharmacoepidemiology and Drug Safety* 18.8 (2009), pp. 672–681.
- [18] Hayden B Bosworth. *Enhancing Medication Adherence - The Public Health Dilemma*. Springer Healthcare, 2012.
- [19] Robby Nieuwlaat. “Interventions for enhancing medication adherence.” In: *Cochrane Database of Systematic Reviews* 11 (2014).
- [20] Joyce A. Cramer et al. “Medication Compliance and Persistence: Terminology and Definitions”. In: *Value in Health* 11.1 (2008), pp. 44–47.
- [21] Marsha A. Raebel et al. “Standardizing Terminology and Definitions of Medication Adherence and Persistence in Research Employing Electronic Databases”. In: *Medical Care* 51 (2013), S11–S21.

- [22] T Mathes et al. “Adherence-enhancing interventions for highly active antiretroviral therapy in HIV-infected patients – a systematic review”. In: *HIV Medicine* 14.10 (2013), pp. 583–595.
- [23] Bernard Vrijens et al. “A new taxonomy for describing and defining adherence to medications”. In: *British Journal of Clinical Pharmacology* 73.5 (2012), pp. 691–705.
- [24] Michael A. Fischer et al. “Trouble Getting Started: Predictors of Primary Medication Nonadherence”. In: *The American Journal of Medicine* 124.11 (2011).
- [25] Andrew J. Karter et al. “New Prescription Medication Gaps: A Comprehensive Measure of Adherence to New Prescriptions”. In: *Health Services Research* 44.5p1 (2009), pp. 1640–1661.
- [26] R. Brian Haynes. “Interventions for enhancing medication adherence”. In: *Cochrane Database of Systematic Reviews* 4 (2008). Ed. by Elizabeth Ackloo et al.
- [27] K. Khair. “Compliance, concordance and adherence: what are we talking about?” In: *Haemophilia* 20.5 (2014), pp. 601–603.
- [28] Robin E Gearing et al. “Reconceptualizing medication adherence: six phases of dynamic adherence.” In: *Harv Rev Psychiatry* 19.4 (2011), pp. 177–189.
- [29] Stacy C. Bailey, Christine U. Oramasionwu, and Michael S. Wolf. “Rethinking Adherence: A Health Literacy–Informed Model of Medication Self-Management”. In: *Journal of Health Communication* 18 (2013), pp. 20–30.
- [30] Niraj Mistry et al. “Technology-mediated interventions for enhancing medication adherence.” In: *Journal of the American Medical Informatics Association : JAMIA* 22.e1 (2015), e177–193.
- [31] Elisabeth Sattler, Jung Lee, and Matthew Perri. “Medication (Re)fill Adherence Measures Derived from Pharmacy Claims Data in Older Americans: A Review of the Literature”. In: *Drugs & Aging* 30.6 (2013), pp. 383–399.
- [32] Simon Diemert et al. “SmartMed: A Medication Management System to Improve Adherence”. In: *Proc. of Information Technology and Communications in Health (ITCH)*. IOS Press, 2015.

- [33] Donald E. Morisky et al. “Predictive Validity of a Medication Adherence Measure in an Outpatient Setting”. In: *The Journal of Clinical Hypertension* 10.5 (2008), pp. 348–354.
- [34] Donald E. Morisky, Lawrence W. Green, and David M. Levine. “Concurrent and Predictive Validity of a Self-Reported Measure of Medication Adherence”. In: *Medical Care* 24.1 (1986), pp. 67–74.
- [35] Jia-Rong Wu et al. “Effect of a Medication-Taking Behavior Feedback, Theory-Based Intervention on Outcomes in Patients with Heart Failure”. In: *Journal of cardiac failure* 18.1 (2012), pp. 1–9.
- [36] Lora L. Sabin et al. “Using Electronic Drug Monitor Feedback to Improve Adherence to Antiretroviral Therapy Among HIV-Positive Patients in China”. In: *AIDS and Behavior* 14.3 (2010), pp. 580–589.
- [37] Upkar Varshney. “Smart medication management system and multiple interventions for medication adherence”. In: *Decision Support Systems* 55.2 (2013), pp. 538–551.
- [38] Jenny Demonceau et al. “Identification and Assessment of Adherence-Enhancing Interventions in Studies Assessing Medication Adherence Through Electronically Compiled Drug Dosing Histories: A Systematic Literature Review and Meta-Analysis”. In: *Drugs* 73.6 (2013), pp. 545–562.
- [39] Alexander S. Misono et al. “Healthcare information technology interventions to improve cardiovascular and diabetes medication adherence.” In: *The American journal of managed care* 16.12 Suppl HIT (2010).
- [40] Jennifer A. Pellowski and Seth C. Kalichman. “Recent advances (2011-2012) in technology-delivered interventions for people living with HIV.” In: *Current HIV/AIDS reports* 9.4 (2012), pp. 326–334.
- [41] Bradi B Granger and Hayden B Bosworth. “Medication adherence: emerging use of technology.” In: *Current Opinion in Cardiology July 2011* 26.4 (2011), pp. 279–287.
- [42] Linda G. Park, Jill Howie-Esquivel, and Kathleen Dracup. “Electronic Measurement of Medication Adherence”. en. In: *Western Journal of Nursing Research* 37.1 (2015), pp. 28–49.
- [43] G. Gimpel, U. Varshney, and P. Ahluwalia. “Emerging IT for Medication Adherence”. In: *IT Professional* 18.3 (2016), pp. 30–36.

- [44] Han-Chun Yeh et al. “APAMAT: A Prescription Algebra for Medication Authoring Tool”. In: *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06*. Vol. 5. 2006, pp. 4284–4291.
- [45] George Yule. *The Study of Language*. 3rd. Cambridge University Press, 2006.
- [46] H. G. Widdowson. *Linguistics*. Oxford Introduction to Language Study ELT, 1996.
- [47] David A. Schmidt. *Denotational Semantics: A Methodology for Language Development*. Dubuque, Iowa: William C Brown Pub, 1988.
- [48] Daniel M. Berry and Erik Kamsties. “Ambiguity in Requirements Specification”. en. In: (2004), pp. 7–44.
- [49] John L. Pfaltz and Azriel Rosenfeld. “Web Grammars”. In: *Proceedings of the 1st International Joint Conference on Artificial Intelligence. IJCAI'69*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1969, pp. 609–619.
- [50] Grzegorz Rozenberg, ed. *Handbook of Graph Grammars and Computing by Graph Transformation*. World Scientific, 1997.
- [51] H. Ehrig et al. *Fundamentals of Algebraic Graph Transformation*. Monographs in Theoretical Computer Science. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [52] J. Engelfriet and G. Rozenberg. “Node Replacement Graph Grammars”. In: *Handbook of Graph Grammars and Computing by Graph Transformation*. Ed. by Grzegorz Rozenberg. Vol. 1. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1997, pp. 1–94.
- [53] Donald E. Knuth. “Semantics of context-free languages”. en. In: *Mathematical systems theory 2.2* (1968), pp. 127–145.
- [54] Jean-Francois Monin. *Understanding Formal Methods*. English. Trans. by M. G. Hinchey. Softcover reprint of the original 1st ed. 2003 edition. London ; New York: Springer, 2003.
- [55] Robert W. Sebesta. *Concepts of Programming Languages*. 4th. Addison-Wesley, 1999.
- [56] C. A. R. Hoare. “An Axiomatic Basis for Computer Programming”. In: *Commun. ACM* 12 (1969), pp. 576–580.

- [57] Greg Michaelson. *An Introduction to Functional Programming Through Lambda Calculus*. English. Mineola, N.Y: Dover Publications, 2011.
- [58] Lotfi Zadeh. “Fuzzy Sets”. In: *Information and Control* 8 (1965), pp. 338–353.
- [59] Timothy J. Ross. *Fuzzy Logic with Engineering Applications, Third Edition*. English. 3rd Edition. Chichester, U.K: Wiley, 2010.
- [60] L. A. Zadeh. “Outline of a New Approach to the Analysis of Complex Systems and Decision Processes”. In: *IEEE Transactions on Systems, Man, and Cybernetics SMC-3.1* (1973).
- [61] X. Larrucea et al. “Software Engineering for the Internet of Things”. In: *IEEE Software* 34.1 (2017), pp. 24–28.
- [62] Simon Diemert, Jens H Weber, and Morgan Price. “An Engagement Model for Medication Management: From Prescription to Description and Conscription”. In: *Proc. of Information Technology and Communications in Health (ITCH)*. Victoria, BC, Canada: IOS Press, 2017.
- [63] Leslie Lamport. *Specifying systems: the TLA+ language and tools for hardware and software engineers*. Boston: Addison-Wesley, 2003.
- [64] Jens H. Weber, Simon Diemert, and Morgan Price. “Using Graph Transformations for Formalizing Prescriptions and Monitoring Adherence”. In: *Graph Transformation*. Ed. by Francesco Parisi-Presicce and Bernhard Westfechtel. Lecture Notes in Computer Science 9151. Springer International Publishing, 2015, pp. 205–220.
- [65] Stephen C Johnson. *Yacc: Yet another compiler-compiler*. Vol. 32.
- [66] Jens H Weber. “GRAPE – A Graph Rewriting and Persistence Engine”. In: *Graph Transformation*. Springer International Publishing, 2017.
- [67] Arend Rensink. “The GROOVE Simulator: A Tool for State Space Generation”. In: *Applications of Graph Transformations with Industrial Relevance*. Ed. by John L. Pfaltz, Manfred Nagl, and Boris Böhlen. Lecture Notes in Computer Science 3062. Springer Berlin Heidelberg, 2004, pp. 479–485.
- [68] Morgan Price et al. “Evaluation of eHealth System Usability and Safety”. In: *Handbook of eHealth Evaluation: An Evidence-based Approach*. University of Victoria, 2016, pp. 337–350.