IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 19, 2018, accepted January 21, 2019, date of publication February 7, 2019, date of current version February 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2896947

Soft Set Theory for Decision Making in Computational Biology Under Incomplete Information

BEATRIZ SANTOS-BUITRAGO[1], ADRIÁN RIESCO[2], MERRILL KNAPP[3], JOSÉ CARLOS R. ALCANTUD[4,5], GUSTAVO SANTOS-GARCÍA[4], (Member, IEEE), AND CAROLYN TALCOTT[6]

[1]Bio and Health Informatics Laboratory, Seoul National University, Seoul 08826, South Korea
[2]Universidad Complutense de Madrid, 28040 Madrid, Spain
[3]Biosciences Division, Stanford Research Institute International, Menlo Park, CA 94025, USA
[4]IME, University of Salamanca, 37007 Salamanca, Spain
[5]BORDA Research Unit, University of Salamanca, 37007 Salamanca, Spain
[6]Computer Science Laboratory, Stanford Research Institute International, Menlo Park, CA 94025, USA

Corresponding author: Gustavo Santos-García (santos@usal.es)

Pathway Logic development has been funded in part by NIH BISTI R21/R33 grant (GM068146-01), NIH/NCI P50 grant (CA112970-01), and NSF grant IIS-0513857. This work was partially supported by NSF grant CNS-1318848. Research was also supported by Spanish projects Strongsoft TIN2012-39391-C04-04, TRACES TIN2015-67522-C3-3-R, and Comunidad de Madrid project N-Greens Software-CM (S2013/ICE-2731).

ABSTRACT The study of biological systems is complex and of great importance. There exist numerous approaches to signal transduction processes, including symbolic modeling of cellular adaptation. The use of formal methods for computational biological systems eases the analysis of cellular models and the establishment of the causes and consequences of certain cellular situations associated with diseases. In this paper, we define an application of logic modeling with rewriting logic and soft set theory. Our approach to decision-making with soft sets offers a novel strategy that complements the standard strategies. We implement a metalevel strategy to control and guide the rewriting process of the Maude rewriting engine. In particular, we adopt the mathematical methods to capture imprecision, vagueness, and uncertainty in the available data. Using this new strategy, we propose an extension in the biological symbolic models of pathway logic. Our ultimate aim is to automatically determine the rules that are most appropriate and adjusted to reality in dynamic systems using decision-making with incomplete soft sets.

INDEX TERMS Biological system modeling, decision making, rewriting logic, rewriting strategies, soft set, symbolic systems biology.

I. INTRODUCTION

Systems biology is an emergent field that facilitates understanding biological systems by describing their structure, dynamics, and control methods. The growth of genomic sequence information combined with technological advances in the analysis of global gene expression has revolutionized the research in biology and biomedicine [1]. Investigation of mammalian signaling processes, the molecular pathways by which cells detect, convert, and internally transmit information from their environment to intracellular targets such as the genome, would greatly benefit from the availability of predictive models [2], [3]. Various models for computational analysis of cellular signaling networks have been proposed to simulate responses to specific stimuli [4]. However, in many cases complex cell signaling pathways have to be treated with other more qualitative modeling approaches, like logic modeling.

Symbolic models allow researchers to represent partial information and to model and analyze systems at multiple levels of detail, depending on the information available and the questions to be studied. Such models are based on formalisms that provide a language for representing system states and mechanisms of change such as reactions, while the corresponding analysis tools are based on computational or logical inference. Symbolic models can be used for simulation of system behavior.

The associate editor coordinating the review of this manuscript and approving it for publication was Shubhajit Roy Chowdhury.

Pathway Logic is a symbolic systems biology approach for modeling and analyzing biological processes, such as signal transduction. Pathway Logic models are represented and analyzed using Maude, a formal system based on rewriting logic [5].

Rewriting logic is a logic of change where non-deterministic systems can be easily specified and analyzed. Rewriting logic extends equational logic by means of *rewrite rules*. While the equational part is used to define the static nature of a system, allowing the user to define sorts, constructors, and functions on these sorts; rewrite rules stand for state transitions in these systems.

With the use of Pathway Logic, we have the problem of not being able to determine the rewriting rules that are more appropriate or more probable among all the possible ones. A metalevel strategy language would be required to control the rewriting process. Our aim is to automatically determine the rules that are most appropriate and adjusted to reality in dynamic systems using decision making with incomplete soft sets.

Many real life problems require the use of imprecise or uncertain data. Their analysis must involve the application of mathematical principles capable of capturing these features. Fuzzy set theory meant a paradigmatic change in Mathematics by allowing partial membership. We are especially interested in a generalization of fuzzy sets: the application of soft sets theory and their extensions to decision making problems (cf., Molodtsov [6] for a definition and arguments about its applicability to several fields).

In this paper, we propose a novel metalevel rewriting-logic approach to guide the executions of rewrite rules in biological dynamic systems using decision making with incomplete soft sets. The use of the theory of soft sets allows us a valid execution even in cases of incomplete information. We include basic examples of the different phases of the implementation, with the execution on some pathways of Pathway Logic.

In the following sections of this introduction, we define the purpose of our work and its significance, and review the current state-of-art in the related research fields. This interdisciplinary work covers three very different areas: soft set theory under incomplete information, rewriting logic, and computational biology and Pathway Logic. The rest of the paper is organized as follows. Section II shows the proposed design of the language extension and the relevant details of the implementation. In Section III, we expose and describe its applications in a biological symbolic system through the Pathway Logic environment. Finally, conclusions are drawn in Section IV.

A. SOFT SET UNDER INCOMPLETE INFORMATION

One of the great difficulties in designing intelligent systems consists in the absence of a knowledge base that collects all the evidence. Scientists must resort to mathematical methods that capture imprecision, vagueness, or uncertainty in the available data. In an attempt to overcome this problem, Molodtsov introduced the theory of soft sets in 1999 [6]. He described the theoretical foundations of this general mathematical tool for dealing with uncertainty and discussed its application to several areas.

Research on soft set theory has progressed rapidly in many different lines. Concerning the theoretical study of soft sets, Maji *et al.* [7] defined some basic algebraic operations on soft sets; Akta and Ça man [8] initiated soft groups as an algebraic structure for soft sets; and Alcantud [9] established formal relationships among notions arising from the theories of soft sets and fuzzy sets.

We recall the notion of soft sets introduced by Molodtsov, and some other useful definitions on soft sets [7], [10]. Throughout the paper, let U be the universe of alternatives, let E be the universal set of parameters, and let A be a subset of relevant parameters. $\mathcal{P}(U)$ denotes the power set of U.

Definition 1 (cf. Molodtsov [6]): The pair (F, A) is a *soft set* over U, if and only if $F : A \longrightarrow \mathcal{P}(U)$ and $A \subseteq E$.

Definition 2 (cf. Maji et al. *[7]):* The intersection of the soft sets (F_1, A) and (F_2, B) is a soft set $(F_1, A) \wedge (F_2, B)$, defined as $(G, A \times B)$ where $G(a, b) = F_1(a) \cap F_2(b)$ for each element (a, b) in $A \times B$.

In other words, a soft set over U is a parameterized family of subsets of U. For $e \in A$, $F(e)$ may be considered as the set of e-approximate elements of (F, A), or the subset of U approximated by e [11].

In practical applications both U and A are usually finite. Soft sets can be represented using a two-dimensional table, where rows are attached with objects in U, and columns are attached with parameters in A [12]. If $u_i \in F(e)$ then $t_{ij} = 1$, otherwise $t_{ij} = 0$, where t_{ij} are the entries of the table. These representations are binary (i.e., all cells are either 0 or 1). The incomplete soft set notion leads to a more general scenario. In incomplete soft sets, according to Definition 3, one can proceed similarly but the possible values for cells are 0, 1, or $*$, the latter standing for the lack of information case. In formal terms one has the following definition:

Definition 3 (cf. Han et al. *[10]):* A pair (F, A) is an *incomplete soft set* over U if and only if $A \subseteq E$ and $F : A \longrightarrow \{0, 1, *\}^U$, where $\{0, 1, *\}^U$ is the set of all functions from U to $\{0, 1, *\}$.

In the analysis of soft sets, the evaluations always take the value 1 (if u belongs to the set of e-approximate elements of the soft set) or 0 (otherwise). However, when we deal with incomplete data in soft sets, we cannot assure that such entry is either 0 or 1. In that case, we use the $*$ symbol to indicate lack of information. That is, $F(e)(u) = *$ means that we do not know whether u belongs to the subset of U approximated by e or not. Obviously, every soft set can be considered an incomplete soft set in a trivial manner.

Concerning standard soft set based decision making, the fundamental reference is Maji *et al.* [13]. When a soft set (F, A) is represented in matrix form through the $k \times l$ matrix $T = (t_{ij})$, where k and l are the cardinals of U and A, respectively, then the *choice value* of an object $u_i \in U$ is $c_i = \sum_j t_{ij}$. A suitable choice is made when the selected object u_k verifies $c_k = \max_i c_i$: objects that maximize the choice value

TABLE 1. Tabular representation C of the incomplete soft set (F, E) in the Example 1.

C matrix				
	e_1	e_2	e_3	e_4
u_1	$*$	1	0	0
u_2	1	$*$	1	0
u_3	1	0	0	1

TABLE 2. The four completed tables C_i for the incomplete soft set (F, E) in the Example 1 with corresponding choice values c_i. (a) C_1 matrix. (b) C_2 matrix. (c) C_3 matrix. (d) C_4 matrix.

(a)

	e_1	e_2	e_3	e_4	$\mathbf{c_i}$
u_1	0	1	0	0	1
u_2	1	0	1	0	2
u_3	1	0	0	1	2

(b)

	e_1	e_2	e_3	e_4	$\mathbf{c_i}$
u_1	1	1	0	0	2
u_2	1	0	1	0	2
u_3	1	0	0	1	2

(c)

	e_1	e_2	e_3	e_4	$\mathbf{c_i}$
u_1	0	1	0	0	1
u_2	1	1	1	0	3
u_3	1	0	0	1	2

(d)

	e_1	e_2	e_3	e_4	$\mathbf{c_i}$
u_1	1	1	0	0	2
u_2	1	1	1	0	3
u_3	1	0	0	1	2

are satisfactory outcomes of the problem. Example 1 below shows a matrix representation for an incomplete soft set.

Example 1: Let U be a nonempty finite set of objects and let E be a nonempty finite set of attributes. Suppose that $U = \{u_1, u_2, u_3\}$ and $E = \{e_1, e_2, e_3, e_4\}$. Define an incomplete soft set (F, E) as follows:

1) $u_1 \in F(e_2), u_1 \notin F(e_3) \cup F(e_4)$. It is unknown whether $u_1 \in F(e_1)$ or not.
2) $u_2 \in F(e_1) \cap F(e_3), u_1 \notin F(e_4)$. It is unknown whether $u_2 \in F(e_2)$ or not.
3) $u_3 \in F(e_1) \cap F(e_4), u_3 \notin F(e_2) \cup F(e_3)$.

Table 1 captures the information defining a soft set (F, E). We have two unknown values (which we denote by the parameter $w = 2$). As a result, the soft set (F, E) becomes an incomplete soft set. We can enumerate the two cells with value $*$ as $((1, 1), (2, 2))$. For each $v \in \{0, 1\}^w$, one feasible completed table arises, that is, we can associate respective values $\{v_1 = (0, 0), v_2 = (0, 1), v_3 = (1, 0), v_4 = (1, 1)\}$ to these two cells in the enumeration. The four completed tables that are produced appear in Table 2, together with the corresponding choice values of the objects. Note that u_1 reaches the highest choice value in C_2 only, u_2 reaches the highest choice value c_i in all these tables, and u_3 reaches the highest choice value exactly in C_1 and C_2.

Soft set theory has potential practical applications in many different fields including decision-making. Decision-making procedures under incomplete information were investigated by Zou and Xiao [14], Han *et al.* [10], Qin *et al.* [15], and Alcantud and Santos-García [16], [18] and Alcantud *et al.* [17]. There are numerous approaches to obtain "decision values". In this sense, Alcantud and Santos-García [18] define a new criterion of choice values extraction for incomplete soft sets based on a combinatorial study of potential associated completed soft sets. It aims at reasoning as in Example 1: the higher the proportion of completed tables where an option gets the top choice value, the better position it occupies in the final ranking.

B. REWRITING LOGIC AND MAUDE

Rewriting logic [19] is a logic of change where non-deterministic systems can be easily specified and analyzed. Rewriting logic extends equational logic by means of *rewrite rules*. While the equational part is used to define the static nature of a system (e.g. data structures), allowing the user to define sorts, constructors, and functions on these sorts, rules stand for state transitions. Following these ideas, the rewriting logic approach has been successfully applied to many models of concurrency [20], [21].

Rewriting logic is implemented in Maude [22], a high-performance logical framework with support for equational and rewriting logic computation. Since rewriting logic is parameterized by an equational theory Maude implements *membership equational logic* [23] that, in addition to sorts, constructors, and functions, allows users to state membership axioms stating the members of a sort.

A Maude system module is defined with syntax `mod NAME is... endm`. Maude types are defined with the keyword `sort`; in turn, subtypes are defined by means of `subsort`. For example, we can define the sort `Marking` that stands for a multiset of items and coins. This sort has `Coin` and `Item` as subsorts, which stand for coins and items, respectively:

```
sorts Coin Item Marking.
subsorts Coin Item < Marking.
```

Functions are defined with the keyword `op`, given the sorts of the arguments and the sort of the result. In particular, constants are defined with an empty lists of sorts as arity. For example, quarters (`q`), dollars (`$`), apples (`a`), and cakes (`c`) are defined as follows:

```
ops q $: -> Coin [ctor format (r! o)].
ops a c: -> Item [ctor format (b! o)].
```

where the attribute `ctor` indicates these functions are constructors and the `format` attribute specifies the pretty printing options (red bold font in the first case and blue bold font in the second one). Likewise, we define sets by specifying the empty set (`null`) and the composition, defined in this case with empty syntax:

```
op null: -> Marking.
op _ _: Marking Marking -> Marking
        [ctor assoc comm id: null].
```

where the attribute `id` indicates that `null` is the identity element for `Marking` and `assoc` and `comm` indicate that the operator is associative and commutative, respectively. Using this syntax, a `Marking` with two quarters, one dollar, and two apples could be written as `q q $ a a`, but it would be equivalent to `a a q q $` and `a q $ q a`, among all others including all these elements.

The static behavior of the system is defined by means of (possibly conditional) equations. However, in this work we are only interested in the dynamic behavior, which in Maude is defined by means of rewrite rules (`rl` and `crl` for conditional rules). For example, the rules `buy-apply` and `change` indicate that we can get an apple and a quarter from a dollar and that it is possible to obtain a dollar from four quarters, respectively:

```
rl [buy-apply]: $       => a q.
rl [change]   : q q q q => $.
```

Now, we illustrate how Maude works with a simple example from [22]. There, the authors define a vending machine where it is possible to buy an apple (`a`) from a dollar (`$`) and receive a quarter (`q`) back, as shown by rule `buy-apple`; or buy a cake (`c`), receiving no change, as specified by rule `buy-cake`. It is also possible to get a dollar from 4 quarters, as shown by rule `change`. All the elements in this example form a set of sort `Marking`:

```
mod VENDING-MACHINE is
 sorts Coin Item Marking.
 subsorts Coin Item < Marking.
 op _ _: Marking Marking -> Marking
          [assoc comm id: null].
 op null: -> Marking.
 ops q $: -> Coin [format (r! o)].
 ops a c: -> Item [format (b! o)].
 rl [buy-apply]: $       => a q.
 rl [buy-cake] : $       => c.
 rl [change]   : q q q q => $.
endm
```

C. COMPUTATIONAL BIOLOGY AND PATHWAY LOGIC

Substantial progresses over the past four decades in biochemistry, molecular biology, and cell physiology, coupled with emerging high throughput techniques for detecting protein-protein interaction, have ushered in a new era in signal transduction research [24]. Given the size and complexity of the cellular signaling networks, it has become necessary to develop predictive mathematical models to understand the system behavior of these networks, and to predict higher order functions that can be validated by experiments [2].

Various models for the computational analysis of cellular signaling networks have been proposed to simulate responses to specific stimuli [3], [25]–[27]. The use of differential equations to represent changes in the concentrations from the input to the output is an adequate approach when, for a given pathway or sub-pathway, there is a large amount of quantitative information and a small number of reactions to be modeled [28]. However, in many cases complex cell signaling pathways have to be treated with other more qualitative modeling approaches, like logic modeling [29].

Symbolic models are based on formalisms that provide a language to represent the states of a system; mechanisms to model their changes (such as reactions); and tools for analysis based on computational or logical inference. A variety of formalisms have been used to develop symbolic models of biological systems, including Petri nets; ambient calculi; statecharts; live sequence charts; and rule-based systems [30]. The key feature of rule-based modeling that makes this approach suitable for studying the site dynamics of biomolecular networks is the simplifying idea of representing biomolecular interactions in terms of local rule [31].

Pathway Logic is a symbolic systems biology approach for modeling and analyzing biological processes, such as signal transduction. Pathway Logic models are represented and analyzed using Maude, a formal system based on rewriting logic [5]. Some capabilities of Pathway Logic include the ability to build and analyze models with multiple levels of detail, define new sorts of data and properties, execute queries on dynamically generated pathways using search, and model-checking. Pathway Logic allows researchers to develop abstract qualitative models, even quantitative and probabilistic models [32], of signaling processes that can be used as the basis for analysis by powerful tools to study a wide range of questions. The Pathway Logic system, its documentation, a collection of examples, and related papers are available at http://pl.csl.sri.com.

D. MODELING IN PATHWAY LOGIC: DISHES AND REWRITE RULES

We briefly present STM7, a model of intracellular signal transduction, to illustrate how Pathway logic can deal with signaling pathways. A formal knowledge base contains information about the changes that occur in the proteins inside a cell in response to exposure to receptor ligands, chemicals, or various stresses. In our case study, we will focus on models of response to *transforming growth factor beta 1* (TGF-β1) stimulation. TGF-β1 is the prototypic member of a large family of structurally related pleiotropic-secreted cytokines. The TGF-β1 signaling pathway is involved in many cellular processes such as growth, proliferation, differentiation, and apoptosis. TGF-β1 has been investigated for association with risk of breast cancer [33].

Fig. 1 shows a general view of Pathway Logic Assistant, which is a Java software that implements the Pathway Logic vision. It shows the Petri net representation of interleukin 6 signaling pathway. Rectangles are transitions (biochemical reactions) and ovals are occurrences (biological entities) in which the initial occurrences are darker. The reactants of a rule are the occurrences connected to the rule by arrows from the occurrence to the rule. The products of a rule are the occurrences connected to the rule by arrows from the rule to the occurrence. Dashed arrows indicate an occurrence that is both input and output. For example, the reaction/rule `1229c` appears in Fig. 1. In this rule, `Jak1` protein (in the cytoplasm) and `Gp130` transmembrane protein (at `GP130C` location) intervene as reactants. The result of this reaction is that `Gp130` protein is unchanged and `Jak1` moves from cytoplasm to `GP130C` location.

Pathway Logic models are structured in four layers: sorts and operations, components, rules, and queries. The *sorts* and

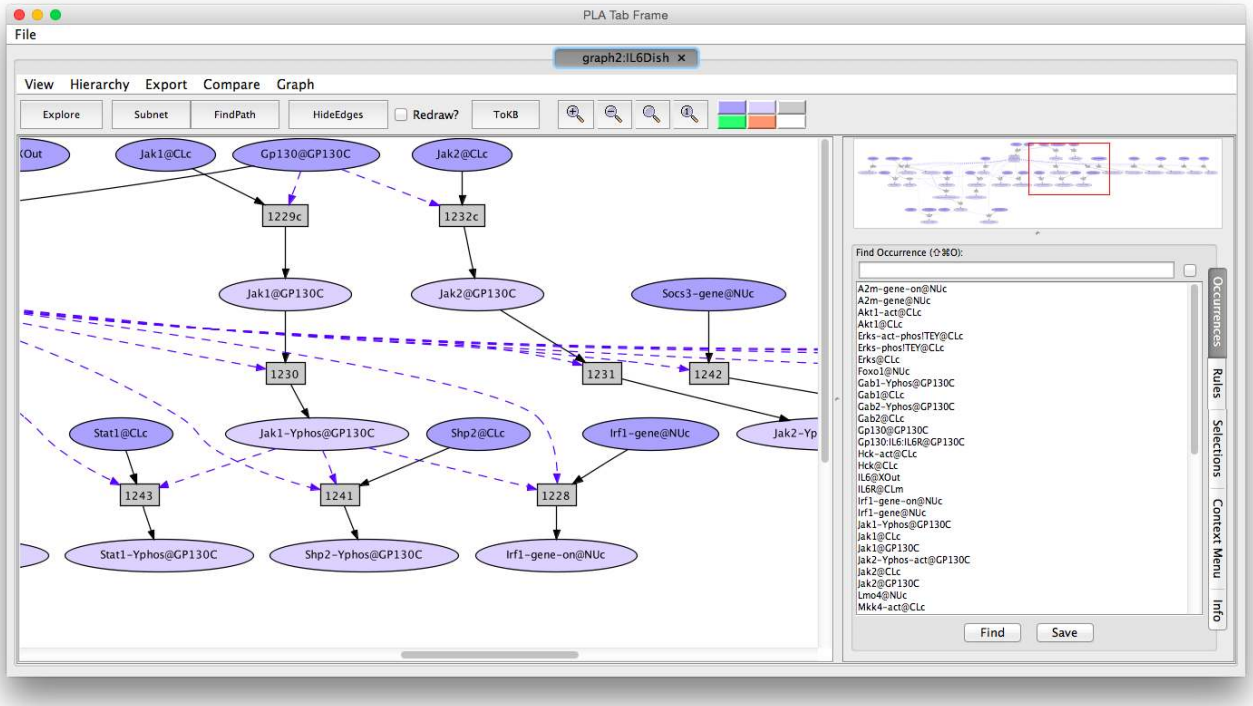

FIGURE 1. A general view of a signaling pathway using Pathway Logic Assistant (cf. Santos-García [4]).

operations layer declares the main sorts and subsort relations, the logical analogue to ontology. The sorts of entities include `Chemical`, `Protein`, `Complex`, `Location` (cellular compartments), and `Cell`. These are all subsorts of the `Soup` sort that represents unordered multisets of entities. The sort `Modification` is used to represent post-translational protein modifications (e.g., activation, binding, phosphorylating). Modifications are applied using the operator `[_-_]`. For example, the term `[Rac1 - GDP]` indicates that Ras-related C3 botulinum toxin substrate 1 (`Rac1`) is binding to guanosine diphosphate (`GDP`).

An initial state or *dish* (called `Tgfb1Dish`) with several locations and elements is defined:

- the outside (location tag `XOut`) which contains the transforming growth factor beta 1 (`Tgfb1`);
- the `Tgfb1RC` location which contains the transforming growth factor beta receptor I and II (`TgfbR1` and `TgfbR2`);
- the `CLo` location, which contains the elements stuck to the outside of the plasma membrane, is empty;
- the membrane (location tag `CLm`) is empty as well;
- the inside of the membrane (location tag `CLi`) contains three proteins bound to GDP: `Cdc42`, `Hras`, and `Rac1`;
- the cytoplasm (location tag `CLc`) contains proteins `Abl1`, `Akt1`, `Atf2`, `Erks`, etc.; and
- the nucleus (location tag `NUc`) contains several genes (e.g., `Smad7`, `Tgfb1`, `Cst6`, etc.) and proteins (e.g., `Ctdsp1`, `Ets1`, and so on).

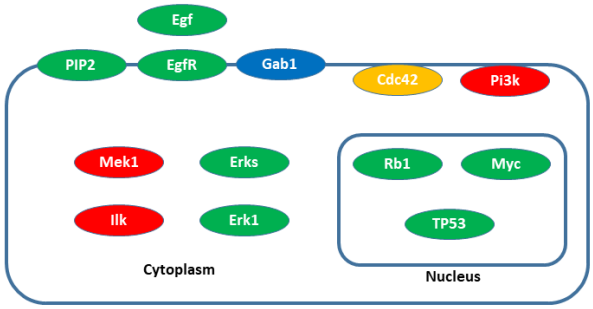

FIGURE 2. Schematic representation of a cell. The activated proteins are marked in red, those phosphorylated in blue, and those bound to GDP in yellow. Proteins that have no modifications are shown in green.

In Maude syntax, this *dish* (called `Tgfb1Dish`) is expressed by the following equation:

```
eq Tgfb1Dish =
PD({XOut | Tgfb1}  {Tgfb1RC | TgfbR1 TgfbR2}
  {CLo | empty}  {CLm | empty}
  {CLi | [Cdc42 - GDP] [Hras - GDP] [Rac1 - GDP]}
  {CLc | Abl1 Akt1 Atf2 Erks Fak1 Jnks Mekk1 Mlk3
    P38s Pak2 Pml Smad2 Smad3 Smad4 Smurf1 Smurf2
    Tab1 Tab2 Tab3 Tak1 Traf6 Zfyve16}
  {NUc | Ctdsp1 Ets1 Smad7 Cdc6-gene Cdkn1a-gene
    Cdkn2b-gene Col1a1-gene Col3a1-gene Ctgf-gene
    Fn1-gene Mmp2-gene Pai1-gene Smad6-gene
    Smad7-gene Tgfb1-gene Timp1-gene Cst6-gene
    Dst-gene Mmp9-gene Mylk-gene Pthlh-gene
    Gfi1-gene Csrp2-gene RoRc-gene}).
```

Figure 2 shows a schematic representation of a cell. Different elements appear in different parts or locations of the

cell: outside the cell (XOut), in/across the cell membrane (CLm), attached to the inside of the cell membrane (CLi), in the cytoplasm (CLc), and in the nucleus (NUc). Some proteins are represented: epidermal growth factor (Egf), PI3 kinase (Pi3k), ERK activator kinase 1 (Mek1), and so on. Some components appear with different modifiers: activation (act), phosphorylation on tyrosine (Yphos), and binding to GDP (GDP). This cell would be represented in Maude with the following SmallDish:

```
eq SmallDish =
  PD( {XOut | Egf}  {CLi | Pi3k [Cdc42 - GDP]}
    {NUc | Rb1 Myc Tp53}
    {CLc | [Mek1 - act] [Ilk - act] Erks Erk1}
    {CLm | EgfR PIP2 [Gab1 - Yphos]}).
```

Rewrite rules detail the behavior of cell components depending on biological contexts and modification states. Each rule represents an action in a biological process such as intra/inter cellular signaling reactions or metabolic reactions. For example, we can say that if, in the location CLc (that corresponds to the cytoplasm), a protein Ikke is found in an activated form and a protein of the family Akts is found, then the protein Akts will be phosphorylated on S473 and T308:

```
rl[1598c.Akts.by.Ikke]:
  {CLc | clc [Ikke - act] Akts}
=> {CLc | clc [Ikke - act]
     [Akts - phos(S 473) phos(T 308)]}.
```

where the variable clc stands for any other element that might appear in the corresponding location (see Figure 3 for a schematic representation of this rule).

FIGURE 3. Schematic representation of the rule 1598c.Akts.by.Ikke. Blue color of the Akts protein indicates that it is phosphorylated at the next state.

The TGFB1 Pathway Logic model contains a total of 57 rules and 968 datums. The experimental evidence for each rule is supplied in datum form. Each datum represents a result from an experiment published in a refereed journal. The rules and evidence can also be found as part of the STM7 model downloadable from the Pathway Logic website (http://pl.csl.sri.com) under the Software link.

1) REWRITE RULE 931.TgfbR1.TgfbR2.by.Tgfb1

Pathway Logic contains a set of transition rules, derived from curated experimental findings. They provide an explanation of how a signal propagates in response to an TGF-β1 stimulus. Here we describe rule 931, directly sourced from the literature. Nakao *et al.* [34] determine that TGF-β signals from the membrane to the nucleus through serine/threonine kinase receptors and their downstream effectors, termed SMAD proteins.

Our rewrite rule 931 establishes: In the presence of transforming growth factor beta receptor I Tgfb1 in the outside of the cell (XOut), the receptors TgfbR1 and TgfbR2 get activated (TgfbR1-act and TgfbR2-act) and bound together and to Tgfb1 ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1). In Maude syntax, this signaling process is expressed by the following rewrite rule:

```
rl[931.TgfbR1.TgfbR2.by.Tgfb1]:
   {XOut | xout Tgfb1 }
   {Tgfb1RC | tgfb1rc TgfbR1 TgfbR2 }
=> {XOut | xout }
   {Tgfb1RC | tgfb1rc
   ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1) }.
```

2) REWRITE RULE 915.Akt1.irt.Tgfb1

Here we describe rule 915, directly sourced from the literature [35]. Our rewrite rule 915 establishes: In the presence of transforming growth factor beta receptor I activated TgfbR1-act, bound to transforming growth factor beta receptor II activated TgfbR2-act, bound to transforming growth factor beta I Tgfb1 and protein Akt1 in the cytoplasm CLc, Akt1 gets phosphorylated at sites S473 and T308—[Akt1 - phos(S 473) phos(T 308)]. In Maude syntax, this signaling process is expressed by the following rewrite rule:

```
rl[915.Akt1.irt.Tgfb1]:
   {Tgfb1RC | tgfb1rc
   ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1) }
   {CLc | clc  Akt1 }
=> {Tgfb1RC | tgfb1rc
   ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1) }
   {CLc | clc  [Akt1 - phos(S 473)phos(T 308)] }.
```

Figures 4 and 5 show the aforementioned rules using the Pathway Logic Assistant. An oval represents a component (e.g., gene, protein, etc.) participating in a reaction. A rectangle illustrates a reaction rule with a label which represents its shortened identifier in the knowledge base. A solid arrow from an occurrence oval to a rule indicates that the occurrence is a reactant. A solid arrow from a rule to an occurrence oval indicates that the occurrence is a product. A dashed arrow from an occurrence oval to a rule indicates that the occurrence is a control (required for the reaction to fire, but not changed).

II. INCOMPLETE SOFT SET STRATEGIES FOR DECISION MAKING

We present in this section a brief overview of the different execution strategies available in Maude. Then, we introduce how we have implemented our rewrite strategy for incomplete soft sets. The source code of the tool, examples, and more information is available at https://github.com/ariesco/pathway.

Given a rewriting logic specification and an initial term, Maude [22] provides several executions for transforming this term:

- The rewrite command applies rewrite rules in a non-deterministic way until a final state (a state that cannot be further rewritten) is obtained.
- The frewrite command applies rewrites rules in a non-deterministic and fair way until a final state

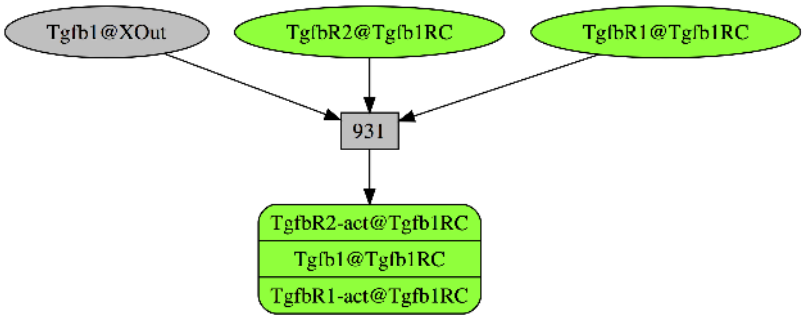

FIGURE 4. **Rule** `931.TgfbR1.TgfbR2.by.Tgfb1` **using Pathway Logic Assistant.**

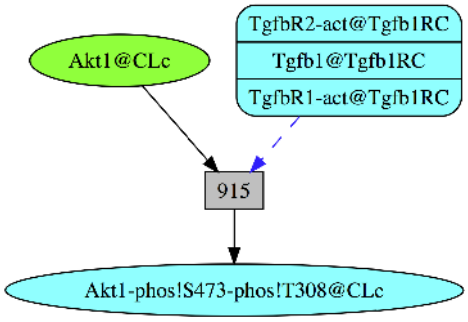

FIGURE 5. **Rule** `915.Akt1.irt.Tgfb1` **using Pathway Logic Assistant.**

is reached. Fair rewriting prevents rules from being applied twice if other rules can be applied to the term.

- The `search` command checks whether it is possible to reach a state matching a given pattern and fulfilling a given condition. Maude follows a breadth-first strategy to generate the state space, so it is possible to find solutions even in systems with infinite states.

Given that we cannot control the execution in the first two cases, analyses usually focus on the `search` command. However, due to its breadth-first strategy, it is unfeasible to analyze highly non-deterministic systems. The execution of these commands can be bounded with an upper bound clause and some other options.

An interesting feature of rewriting logic is that, thanks to its reflective capabilities [23], it is possible to manipulate modules and terms as usual data. Using this feature, it is possible to implement tools for analyze and direct how Maude modules are executed. Following this idea, the Maude strategy language [36] was implemented. This language allows specifiers to define strategies to define which rules should be applied to a specific term and which values should the variables take in a particular step.

However, it is not possible to use the strategy language to execute a specification implementing incomplete soft sets because these strategies focus on the term that must be rewritten, while our approach focuses on optimizing reached state. For this reason, and following the same metalevel approach as the strategy language, we present a rewrite strategy that directs the execution to optimize the value computed from an incomplete soft set. In particular, we implement our rewrite strategy extending the framework in [37]. This framework is implemented in top of Full Maude [22, Part II], an extension of Maude written in Maude itself that provides a richer syntax and features for implementing I/O applications and metatools extending Maude. In this way, our incomplete soft set strategy complements the transformation already integrated in the framework, such as analysis of causes [29] and stepwise reachability analysis [5].

A. IMPLEMENTATION OF AN INCOMPLETE SOFT SET STRATEGY

In order to apply our strategy, we require terms and rules to be extended with an incomplete soft set. In Maude, the `SOFTSET` module defines attributes of incomplete soft set objects as a set of pairs of the form `[Att = V]`, where `Att` is the attribute name and `V` is its value, which can take the values `0`, `1`, and `*`. In this way, different attributes, when put together and separated by commas, produce a term of sort `AttSet` of the form `[Att1 = V1], ..., [Attn = Vn]`. Since sets are defined in Maude as associative and commutative, the order in the set can be modified without modifying its meaning.

Once we have defined attributes of incomplete soft sets, we can extend terms and rules to deal with them. We transform each rewrite rule `rl T => T'.` (respectively, conditional rules `crl T => T' if COND.`, for `COND` a condition), for `T` and `T'` terms of the given system, as `rl T Atts => T' Atts'.` (respectively `crl T Atts => T' Atts' if COND.`), with `Atts` the attributes of the incomplete soft set before applying the rule and `Atts'` the transformed attributes of the incomplete soft set. Note how the values of the attributes in the incomplete soft set can be modified in each rewrite rule. Likewise, initial (ground) terms `t` must now be defined as `t atts`, with `atts` the particular values of the attributes for `t`.

Now we apply the decision-making procedure under incomplete information in [14]. In order to choose the most appropriate rule to be executed we need to extract the incomplete soft set information and manipulate it. Our strategy computes all the reachable terms from the current term, takes the information of the soft set attributes, and places it into a matricial representation of the incomplete soft set `SoftSet`, which is defined as a list of lists of values. In order to process

this matrix, the SS-STRAT module is parameterized by the SOFT-SET-FUN theory, which requires a computeValue to be implemented. This function takes a SoftSet as argument and returns a natural number standing for the row of the matrix that maximizes the choice value. That is, given the matrix representation $T_{k\times l} = (t_{ij})$ of an incomplete soft set, where k is the number of reachable terms and l is the number of attributes, the *choice value* of an object $u_i \in U$ is $c_i = \sum_j t_{ij}$. Objects that maximize the choice value are satisfactory outcomes of this decision making problem. Put differently, a suitable choice is made when the selected object u_k verifies $c_k = \max_i c_i$. Following the ideas in [16], we examine all completed tables arising from the original incomplete table. Their choice values are computed as in [13].

The function computeValue can be implemented in different ways so undefined values $\star$ take different values. Users can define their own strategies but our system also provides three predefined strategies: $\star$ are computed as 0 (implemented through the UndefToZero view), as 1 (UndefToOne view), or as 0.5 (UndefSemi view), according to indicator d_{i-p} proposed by Zou and Xiao [14].

Now we continue with the vending machine example shown in Section I-B. For example, if we rewrite three times a term $ q q q, in a possible execution we obtain the term a c as a consequence of applying rules buy-apply, change and buy-cake:

```
Maude> rew~\hbox{[3]} $ q q q.
rewrite~\hbox{[3]} in VENDING-MACHINE: $ q q q.
rewrites: 3 in 0ms cpu (0ms real)
  (3000000 rewrites/second)
result Marking: a c
```

We can add incomplete soft set information by using attributes a, c, and $, indicating whether there are apples, cakes, and dollars, respectively. Since the buy-apple rule introduces an apple in the marking and removes a dollar, we set the value for the attribute a to 1, while the one for $ is $\star$ (we are not sure whether there are more dollars in the rest of the marking, so we cannot set it to 0). We reason analogously for the buy-cake rule. Finally, when the rule change is used the attribute $ is set to 1. Note that when using incomplete soft sets we need to use the complete marking in the rules, using a variable M that matches the rest of the marking:

```
rl [buy-apple]: (M $) ([a = V], [$ = V'], Atts)
  => (M a q) ([a = 1], [$ = *], Atts).
rl [buy-cake] : (M $) ([c = V], [$ = V'], Atts)
  => (M c)   ([c = 1], [$ = *], Atts).
rl [change]   : (M q q q q) ([$ = V], Atts)
  => (M $)   ([$ = 1], Atts).
```

Our rewrite command for incomplete soft sets takes the term init to be reduced and a list of attributes atts of interest as follows:

```
(ssrew init for atts.)
```

In the example above, we can use this command to maximize the apples obtained from two dollars by focusing on the attribute a. The incomplete soft set in our initial term has a and c as 0, because there are no apples or cakes, while $ takes the value 1. In the reached state, the values have changed and a takes the value 1 while $ is $\star$:

```
Maude> (ssrew ($ $) (a = 0, c = 0, $ = 1) for a.)
result SoftMarking: (a a q q)
(a = 1, c = 0, $ = *)
```

III. INCOMPLETE SOFT SET STRATEGIES IN PATHWAY LOGIC

We present in this section the integration between the incomplete soft set strategy outlined in the previous section and the Pathway Logic framework. We first showed a particular pathway to illustrate the process and then we show how it is extended to work with incomplete soft sets.

In our Pathway Logic case study, we will focus on models of response to *transforming growth factor beta 1* (TGF-β1) stimulation. In the previous section the initial state Tgfb1Dish was defined. According to Section II-A, the Maude module SOFTSET defines attributes of an incomplete soft set object as a set of pairs of the form [Att = V]. In the case of our incomplete soft set transformation of the Pathway Logic models, the names of the genes are the attributes and the values correspond to their activation status: the value 0 indicates an inactive gene; the value 1 represents an active gene; and the value $\star$ corresponds to the case of lacking information about its activation. In this way, different attributes, when put together and separated by commas, produce a term of sort AttSet of the form [Att1 = V1], ..., [Attn = Vn]. A concrete example of a term of sort AttSet for the model TGF-β1 is: [cdc6 = 0], [cdkn1a = 0], [cdkn2b = 0], [col1a1 = 1], [col3a1 = 1], [ctgf = *].

Once we have defined these attributes of incomplete soft sets, we can extend terms and rules to deal with them like explained in Section II-A. We illustrate how this system works using the rule 1724.Cdc6-gene.irt.Tgfb1, that activates CDC6 gene under some conditions:

```
rl[1724.Cdc6-gene.irt.Tgfb1]:
   {Tgfb1RC | tgfb1rc
   ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1)}
   {NUc | nuc Cdc6-gene}
=> {Tgfb1RC | tgfb1rc
   ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1)}
   {NUc | nuc [Cdc6-gene - on]}.
```

The rules of Pathway Logic adapted for the new strategy must define the dynamics for the characteristics or attributes. In rule 1724.Cdc6-gene.irt.Tgfb1 we can add incomplete soft set information by using attribute cdc6, indicating whether there is CDC6 gene in an inactivated form. Since this rule activates CDC6 gene ([Cdc6-gene - on]), we set the value for the attribute cdc6 to 1:

```
rl[SS1724.Cdc6-gene.irt.Tgfb1]:
   {Tgfb1RC | tgfb1rc
   ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1) }
   {NUc | nuc Cdc6-gene}
   ([cdc6 = V], Atts)
```

```
=> {Tgfb1RC | tgfb1rc
    ([TgfbR1 - act]: [TgfbR2 - act]: Tgfb1) }
    {NUc | nuc [Cdc6-gene - on]}
    ([cdc6 = 1], Atts).
```

We reason analogously for the remaining Pathway Logic rules.

Likewise, initial (ground) terms `t` must now be defined as `t atts`, with `atts` the particular values of the attributes for the initial term `t`. Then we apply a decision-making procedure under incomplete information proposed by Zou and Xiao [14]. Our rewrite command for incomplete soft sets takes the term `init` to be reduced and a list of attributes `atts` of interest as follows:

```
(ssrew init for atts.)
```

Finally, we apply the previous rule to a specific term `Tgfb1Dish` to achieve the optimal solution. In our Pathway Logic example, we can use this command to obtain the best rule to rewrite. Some genes are considered: `cdc6`, `cdkn1a`, `cdkn2b`, `col1a1`, `col3a1`, and `ctgf`. In the reached state, the values have changed and `cdc6` takes the value `1` while `cdkn1a` is $\star$:

```
Maude> (ssrew init for cdc6 cdkn1a cdkn2b col1a1
  col3a1 ctgf fn1 mmp2 pai1 smad6 smad7 tgfb1
  timp1
  cst6 dst mmp9 mylk pthlh gfi1 csrp2 rorc .)

Result: PD(
  {CLc | Akt1 Atf2 Fak1 Jnks Mekk1 Mlk3 P38s Pml
   Smad2 Smad3 Tab1 Tab2 Tab3 Tak1 Traf6 Zfyve16
   [Abl1 - act][Erks - phos(TEY)]}
  {CLi |([Pak2 - act]:[Rac1 - GTP])[Cdc42 - GTP]
   [Hras - GTP]}
  {CLm | empty}
  {CLo | empty}
  {NUc | Cdkn1a-gene Col1a1-gene Csrp2-gene
   Cst6-gene
   Ctdsp1 Dst-gene Ets1 Gfi1-gene Mmp9-gene
   Mylk-gene
   Pai1-gene Smad4[Cdc6-gene - on][Cdkn2b-gene
   - on]
   [Col3a1-gene - on][Ctgf-gene - on][Fn1-gene
   - on]
   [Mmp2-gene - on][Pthlh-gene - on][RoRc-gene
   - on]
   [Smad6-gene - on][Smad7-gene - on][Tgfb1-gene
   - on]
   [Timp1-gene - on]}
  {Tgfb1RC | Smad7 Smurf1 Smurf2
   Tgfb1: [TgfbR1 - ubiq]:[TgfbR2 - act]}
  {XOut | empty})
   [cdc6 = 1],[cdkn1a = 0],[cdkn2b = 1],[col1a1
   = 0],
   [col3a1 = 1],[csrp2 = 0],[cst6 = 0],[ctgf = 1],
   [dst = 0],[fn1 = 1],[gfi1 = 0],[mmp2 = 1],[mmp9
   = *],
   [mylk = 0],[pai1 = *],[pthlh = 1],[rorc = 1],
   [smad6 = 1],[smad7 = 1],[tgfb1 = 1],[timp1 = 1]
```

Our strategy computes all the reachable terms from the current term, takes the information of the incomplete soft set attributes, and places it into a matricial representation of the incomplete soft set `SoftSet`, which is defined as a list of lists of values. Given the matrix representation $T_{k\times l} = (t_{ij})$ of an incomplete soft set, where k is the number of reachable terms and l is the number of attributes, the *choice value* of an object $u_i \in U$ is $c_i = \sum_j t_{ij}$. Objects that maximize the choice value are satisfactory outcomes of this decision making problem. Put differently, a suitable choice is made when the selected object u_k verifies $c_k = \max_i c_i$. A comprehensive example of this approach is available at https://github.com/ariesco/pathway.

IV. DISCUSSION AND CONCLUSIONS

There are numerous research works and applications that use rewriting logic. Decision making with incomplete soft sets enables our novel strategy that complements the language of standard strategies in rewriting logic. A great problem in any system, which occurs on many occasions, is the management of incomplete information. This paper allows for facing problems from another point of view and with the advantages offered by the theory of soft sets.

Another problem in formal analysis is that situations with a different probability of occurrence are normally considered in the same way. A formal system can analyze elements, options, and cases that in practice do not occur or are highly infrequent, and these cases mask more frequent states. It offers a new approach that allows specifiers to fine tune the dynamics of the biological system. Hence, we can choose the most likely rules among them all, depending on the characteristics of the cells. With the use of decision making techniques, our approach has the advantage of deducting conclusions and/or extracting the most common properties.

With our model, the formalism, the circumstances that participate, and the calculation of the results are established in a natural way. On the other hand, this model can be applied to many and varied types of problems with a common approach. In this paper, we have described a concrete one as an example of its versatility. Another advantage is that anyone could use the existing knowledge bases in Pathway Logic and then modify our *soft* behavior of the desired model.

Our approach based on symbolic modeling offers an alternative view to quantitative methods (such as differential equations). Although the conclusions that we obtain are more theoretical, they are based on evidences contrasted in the literature that constitute a great knowledge base. Thanks to these data, we can face challenges of formal analysis of signaling pathways. Advantages over other symbolic approaches (Kappa, BioNetGen, etc.) lie in the characteristics of the underlying language Maude (simplicity, expressiveness, and performance).

The main contribution of our paper is to offer a framework, based on the theory of soft sets, rewriting logic, and Pathway Logic, which allows programmers to specify and analyze formal biological system when there is vague information. Our second contribution is the extension of the language of rewriting strategies in Pathway Logic that supports a *guided*

execution and analysis, based on a decision making system with incomplete soft sets.

As future works, it would be interesting to develop other fuzzy extensions and with other paradigms of the theory of soft sets. Possible extensions for fuzzy soft sets under incomplete information can be integrated into the Pathway Logic Assistant tool to present visually and graphically these new functionalities. Besides, we intend to explore this method with some signaling pathways and derive concrete results.

ACKNOWLEDGMENT

The authors would like to thank the valuable suggestions of the editor and the anonymous reviewers that allowed us to improve the original manuscript.

REFERENCES

[1] O. G. Vukmirovic and S. M. Tilghman, "Exploring genome space," *Nature*, vol. 405, no. 6788, pp. 820–822, 2000.

[2] R. Donaldson, C. Talcott, M. Knapp, and M. Calder, "Understanding signalling networks as collections of signal transduction pathways," in *Proc. ACM Comput. Methods Syst. Biol. (CMSB)*, P. Quaglia, Ed. 2010, pp. 86–95.

[3] G. Weng, U. S. Bhalla, and R. Iyengar, "Complexity in biological signaling systems," *Science*, vol. 284, no. 5411, pp. 92–96, 1999.

[4] G. Santos-García, C. L. Talcott, and J. De Las Rivas, "Analysis of cellular proliferation and survival signaling by using two ligand/receptor systems modeled by pathway logic," in *Proc. 4th Int. Workshop Hybrid Syst. Biol. (HSB)*, in Lecture Notes in Computer Science, Madrid, Spain, vol. 9271, A. Abate and D. Šafránek, Eds. Cham, Switzerland: Springer, Sep. 2015, pp. 226–245.

[5] G. Santos-García, C. L. Talcott, A. Riesco, B. Santos-Buitrago, and J. De Las Rivas, "Role of nerve growth factor signaling in cancer cell proliferation and survival using a reachability analysis approach," in *Proc. 10th Int. Conf. Practical Appl. Comput. Biol. Bioinf. (PACBB)*, in Advances in Intelligent Systems and Computing, vol. 477, M. S. Mohamad, M. P. Rocha, F. Fdez-Riverola, F. J. D. Mayo, and J. F. De Paz, Eds. Cham, Switzerland: Springer, Jun. 2016, pp. 173–181.

[6] D. Molodtsov, "Soft set theory—First results," *Comput. Math. Appl.*, vol. 37, nos. 4–5, pp. 19–31, 1999.

[7] P. K. Maji, R. Biswas, and A. R. Roy, "Soft set theory," *Comput. Math. Appl.*, vol. 45, nos. 4–5, pp. 555–562, 2003.

[8] H. Akta and N. Ça man, "Soft sets and soft groups," *Inf. Sci.*, vol. 177, pp. 2726–2735, Jul. 2007.

[9] J. C. R. Alcantud, "Some formal relationships among soft sets, fuzzy sets, and their extensions," *Int. J. Approx Reason*, vol. 68, pp. 45–53, Jan. 2016.

[10] B.-H. Han, Y.-M. Li, J. Liu, S.-L. Geng, and H. Li, "Elicitation criterions for restricted intersection of two incomplete soft sets," *Knowl-Based Syst*, vol. 59, pp. 121–131, Mar. 2014.

[11] F. Feng and Y. Li, "Soft subsets and soft product operations," *Inf Sci*, vol. 232, pp. 44–57, May 2013.

[12] Y. Y. Yao, "Relational interpretations of neighborhood operators and rough set approximation operators," *Inf Sci*, vol. 111, nos. 1–4, pp. 239–259, 1998.

[13] P. K. Maji, R. Biswas, and A. R. Roy, "An application of soft sets in a decision making problem," *Comput. Math. Appl.*, vol. 44, nos. 8–9, pp. 1077–1083, 2002.

[14] Y. Zou and Z. Xiao, "Data analysis approaches of soft sets under incomplete information," *Knowl. Based Syst.*, vol. 21, no. 8, pp. 941–945, 2008.

[15] H. Qin, X. Ma, T. Herawan, and J. M. Zain, "Data filling approach of soft sets under incomplete information," in *Intelligent Information and Database Systems* (Lecture Notes in Computer Science), vol. 6592, N. Nguyen, C.-G. Kim, and A. Janiak, Eds. Berlin, Germany: Springer, 2011, pp. 302–311.

[16] J. C. R. Alcantud and G. Santos-García, "Incomplete soft sets: New solutions for decision making problems," in *Decision Economics, in Commemoration of the Birth Centennial of Herbert A. Simon 1916–2016 (Nobel Prize in Economics 1978)* (Advances in Intelligent Systems and Computing), vol. 475, E. Bucciarelli, M. Silvestri, and S. R. González, Eds. Cham, Switzerland: Springer, 2016, pp. 9–17.

[17] J. C. R. Alcantud, G. Santos-García, and E. Hernández-Galilea, "Glaucoma diagnosis: A soft set based decision making procedure," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 9422, J. M. Puerta *et al.*, Eds. Cham, Switzerland: Springer, 2015, pp. 49–60.

[18] J. C. R. Alcantud and G. Santos-García, "A new criterion for soft set based decision making problems under incomplete information," *Int. J. Comput. Int. Syst.*, vol. 10, pp. 394–404, Jan. 2017.

[19] J. Meseguer, "Rewriting as a unified model of concurrency," SRI Int., Comput. Sci. Lab., Menlo Park, CA, USA, Tech. Rep. SRI-CSL-90-02, Jun. 1990.

[20] J. Meseguer, "Conditional rewriting logic as a unified model of concurrency," *Theor. Comput. Sci.*, vol. 96, no. 1, pp. 73–155, 1992.

[21] J. Meseguer, "Twenty years of rewriting logic," *J. Logic Algebraic Program.*, vol. 81, nos. 7–8, pp. 721–781, 2012.

[22] M. Clavel *et al.*, *All About Maude—A High-Performance Logical Framework: How to Specify, Program, and Verify Systems in Rewriting Logic* (Lecture Notes in Computer Science), vol. 4350. Berlin, Germany: Springer, 2007.

[23] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. F. Quesada, "Maude: Specification and programming in rewriting logic," *Theor. Comput. Sci.*, vol. 285, no. 2, pp. 187–243, 2002.

[24] N. J. Eungdamrong and R. Iyengar, "Modeling cell signaling networks," *Biol. Cell*, vol. 96, no. 5, pp. 355–362, Jun. 2004.

[25] P. Smolen, D. A. Baxter, and J. H. Byrne, "Mathematical modeling of gene networks," *Neuron*, vol. 26, no. 3, pp. 567–580, 2000.

[26] R. A. Williams, J. Timmis, and E. E. Qwarnstrom, "Computational models of the NF-κB signalling pathway," *Computation*, vol. 2, no. 4, pp. 131–158, 2014.

[27] A. Mizuta, Q.-W. Ge, and H. Matsuno, "Dependent shrink of transitions for calculating firing frequencies in signaling pathway Petri net model," *Algorithms*, vol. 10, no. 1, p. 4, 2016.

[28] A. Saadatpour and R. Albert, "Discrete dynamic modeling of signal transduction networks," *Methods Mol. Biol.*, vol. 880, pp. 255–272, 2012.

[29] B. Santos-Buitrago, A. Riesco, M. Knapp, G. Santos-García, and C. L. Talcott, "Reverse inference in symbolic systems biology," in *Proc. 11th Int. Conf. Practical Appl. Comput. Biol. Bioinf. (PACBB)*, in Advances in Intelligent Systems and Computing, Porto, Portugal, vol. 616, M. S. Mohamad, M. P. Rocha, J. F. De Paz, and T. Pinto, Eds. Cham, Switzerland: Springer, Jun. 2017, pp. 101–109.

[30] G. Santos-García, J. De Las Rivas, and C. L. Talcott, "A logic computational framework to query dynamics on complex biological pathways," in *Proc. 8th Int. Conf. Practical Appl. Comput. Biol. Bioinf. (PACBB)* in Advances in Intelligent Systems and Computing, vol. 294, Salamanca, Spain, J. Saez-Rodriguez, M. P. Rocha, F. Fdez-Riverola, and J. F. De Paz Santana, Eds. Cham, Switzerland: Springer, Jun. 2014, pp. 207–214.

[31] W. S. Hlavacek, J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana, "Rules for modeling signal-transduction systems," *Sci. STKE*, vol. 2006, no. 344, p. 6, 2006.

[32] A. Abate, Y. Bai, N. Sznajder, C. Talcott, and A. Tiwari, "Quantitative and probabilistic modeling in pathway logic," in *Proc. 7th IEEE Int. Conf. Bioinf. Bioeng (BIBE)*. Boston, MA, USA: IEEE Computer Society, Oct. 2007, pp. 922–929.

[33] M. Kretzschmar, "Transforming growth factor-β and breast cancer: Transforming growth factor-β/SMAD signaling defects and cancer," *Breast Cancer Res.*, vol. 2, no. 2, p. 107, 2000.

[34] A. Nakao *et al.*, "Identification of Smad7, a TGFβ-inducible antagonist of TGF-β signalling," *Nature*, vol. 389, no. 6651, pp. 631–635, 1997.

[35] M. C. Wilkes *et al.*, "Transforming growth factor-β activation of phosphatidylinositol 3-kinase is independent of Smad2 and Smad3 and regulates fibroblast responses via p21-activated kinase-2," *Cancer Res.*, vol. 65, no. 22, pp. 10431–10440, 2005.

[36] S. Eker, N. Martí-Oliet, J. Meseguer, and A. Verdejo, "Deduction, strategies, and rewriting," *Electron. Notes Theor. Comput. Sci.*, vol. 174, no. 11, pp. 3–25, 2007.

[37] A. Riesco, B. Santos-Buitrago, J. De Las Rivas, M. Knapp, G. Santos-García, and C. Talcott, "Epidermal growth factor signaling towards proliferation: Modeling and logic inference using forward and backward search," *BioMed. Res. Int.*, vol. 2017, 2017, Art. no. 1809513. doi: 10.1155/2017/1809513.

BEATRIZ SANTOS-BUITRAGO was born in Salamanca, Spain, in 1993. She received the B.S. degree in computer science from the University of Salamanca, in 2015, and the M.S. degree in computer science and engineering from Seoul National University (SNU), South Korea, in 2017.

Since 2014, she has been with the College of Engineering, SNU, where she was a member of the Bio and Health Informatics Laboratory. She was a Visiting Researcher with the Stanford Research Institute International, Menlo Park, CA, USA, in 2017. Since 2018, she has been a Business Analyst with Amazon, Luxembourg. Her research interests include the role of signaling pathways in breast cancer, the modeling and logic inference in symbolic systems biology, and the soft set theory for decision-making in computational biology.

ADRIÁN RIESCO was born in Madrid, Spain, in 1982. He received the B.S., M.S., and Ph.D. degrees in computer science from the Universidad Complutense de Madrid, Spain, in 2005, 2007, and 2011, respectively.

He was an Interim Professor, from 2011 to 2015, and a Ph.D. Assistant Professor, from 2015 to 2018, with the Department of Software Systems and Computation, Universidad Complutense de Madrid, where he has been an Associate Professor, since 2018. He has authored over 50 articles. His research interests include formal methods, rewriting logic, declarative debugging, testing, and big data analysis.

Dr. Riesco was a recipient of the 2005 Degree Award in computer science of the UCM. He has collaborated as a Program Committee Member and a Reviewer for many international conferences and journals.

MERRILL KNAPP is currently a Computational Biologist with the Stanford Research Institute International. She is also a Cell Biologist with 40 years of experience as an Experimental Biologist (designing and carrying out experiments) and six years developing computational models of signal transduction systems.

JOSÉ CARLOS R. ALCANTUD received the M.Sc. degree in mathematics from the University of Valencia, Spain, in 1991, and the Ph.D. degree in mathematics from the University of Santiago de Compostela, Spain, in 1996. He is currently a Professor with the Department of Economics and Economic History, University of Salamanca, Spain, where he also acts as the Head of the department. His current research interests include fuzzy preference modeling, decision-making, and social choice.

GUSTAVO SANTOS-GARCÍA received the M.Sc. degree in mathematics from the University of Salamanca, Spain, in 1986, and the Ph.D. degree in mathematics from the University of Valladolid, Spain, in 1993. His Ph.D. thesis was developed at the Torres Quevedo Institute, CSIC, Spain.

In 1995, he was an International Research Fellow with the Computer Science Laboratory, Stanford Research Institute International. Since 1988, he has been with the School of Sciences and the School of Economics and Business, University of Salamanca. His research interests include symbolic systems biology, decision-making under uncertainty, and artificial intelligence applied to medicine and mathematical logic.

Dr. Santos has collaborated as a Program Committee Member and a Reviewer for many international conferences and journals.

CAROLYN TALCOTT received the Ph.D. degree in computer science from Stanford University and the Ph.D. degree in chemistry from the University of California at Berkeley.

She is currently the Program Director of the Computer Science Laboratory, Stanford Research Institute International, and the Leader of the Symbolic Systems Technology Group. Her research focuses on formal modeling and analysis, programming languages, security protocols, networks, distributed and networked cyber-physical systems, and symbolic systems biology.