South American Journal of Logic
Vol. 1, n. 1, pp. 299–319, 2015
ISSN: 2446-6719

A Logical Model for Metabolic Networks with Inhibition

Robert Demolombe, Luis Fariñas del Cerro and Naji Obeid

Abstract

Metabolic networks formed by long sequences of biochemical reactions have been widely investigated to determine the catalytic role of genomes and how they interfere in the process. Many tumors have been reported to be the result of a pathology in the cell's pathway. Knowing that the complexity of the imbrication of such networks is beyond human reasoning, the use of artificial intelligence to help scientists in their experiments might seem adapted. This paper aims to present a logical model for metabolic pathways capable of describing both positive and negative reactions (activations and inhibitions) based on a fragment of first order logic. We also present an efficient automated deduction method allowing us to predict results by deduction and infer reactions and proteins states by abductive reasoning.

Keywords: metabolic pathways, logical model, inhibition, deduction, abduction.

Introduction

Cells in general and human body cells in particular incorporates a large series of intracellular and extracellular signalings, notably protein activations and inhibitions, that specify how they should carry out their functions. Networks formed by such biochemical reactions, often referred as *pathways*, are at the center of a cell's existence and they range from simple and chain reactions and counter reactions to simple and multiple regulations and auto regulations. Cancer, for example, can appear as a result of a pathology in the cell's pathway, thus, the study of signalization events appears to be an important factor in biological, pharmaceutical and medical researches. However, the complexity of the imbrication of such processes makes the use of a physical model as a representation seem complicated.

In the last couple of decades, scientists that used artificial intelligence to model cell pathways [8, 7, 17, 18, 4, 22, 16] faced many problems especially because information about biological networks contained in knowledge bases is generally incomplete and sometimes uncertain and contradictory. To deal with such issues, abduction [1] as theory completion [12] is used to revise the state of existing nodes and add new nodes and arcs to express new observations. Languages that were used to model such networks had usually limited expressivity, were specific to special pathways or were limited to general basic functionalities. We, in this work, present a fragment of first order logic [20] capable of representing node states and actions in term of positive and negative relation between said nodes. Then an efficient proof theory for these fragments is proposed. This method can be extended to define an abduction procedure which has been implemented in SOLAR [13], an automated deduction system for consequence finding.

The rest of this paper is organized as follows. Section 1 introduces the problem from a biological point of view. Section 2 presents a basic language and axiomatic capable of describing general pathways, ans shows how it is possible to extends this language and axiomatic to accommodate the requirements of section 1. Section 3 defines a translation procedure capable of eliminating first order variables and equality predicates and shows how it can be applied to derive new axiomatic that can be used in the automated deduction process in SOLAR. Section 4 provide some case studies, and finally section 5 gives a summary and discusses future works.

1 Biological Background

Cancer has been at the center of countless biological researches trying to figure out what was causing the strange cells behaviors. Many treatments and cures have been developed and successfully administered, but in many other cases, therapeutic responses are limited and tumors relapse or fail to respond in a large fraction of patients. There is currently no way to predict how the tumor's will respond to the treatment. One approach is to investigate the molecular determinants of tumor response. These molecular parameters include the cell cycle checkpoint, DNA repair and programmed cell apoptosis pathways [15, 10, 5, 11, 14]. When DNA damage occurs, cell cycle checkpoints are activated and can rapidly kill the cell by apoptosis or arrest the cell cycle progression to allow DNA repair before cellular reproduction or division. Two important checkpoint that appear to function when parallel transduction cascades from DNA damage to the cell cycle checkpoint effectors are the ATM-Chk2 and the ATR-Chk1 pathways [15].

Intracellular signalization is actively studied and subject to many experiments because there are many unknown reactions that lead to checkpoints and ones that come after which cannot be proved or described. The goal of these experiments is to try to understand why in some cases cell treatment fails and cells do not go through these checkpoints when DNA damage occurs. As a result, scientist have been building networks showing, in human readable form, the cell cycle checkpoints pathways, that are constantly updated as new interaction are discovered. Figure 1 defines a list of symbols used in the molecular interaction map of ATM-Chk2 shown in Figure 2.

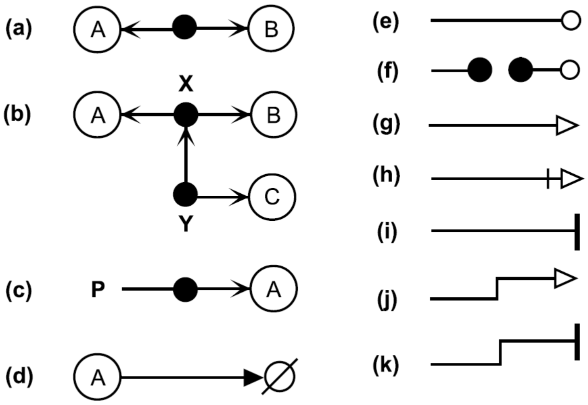

Figure 1: Symbol definitions and map conventions.

(*a*) Proteins A and B can bind to each other. The node placed on the line represents the A:B complex. (*b*) Multimolecular complexes: x is A:B and y is(A:B):C. (*c*) Covalent modification of protein A. (*d*) Degradation of protein A. (*e*) Enzymatic stimulation of a reaction. (*f*) Enzymatic stimulation in trans. (*g*) General symbol for stimulation. (*h*) A bar behind the arrowhead signifies necessity. (*i*) General symbol for inhibition. (*j*) Shorthand symbol for transcriptional activation. (*k*) Shorthand symbol for transcriptional inhibition.

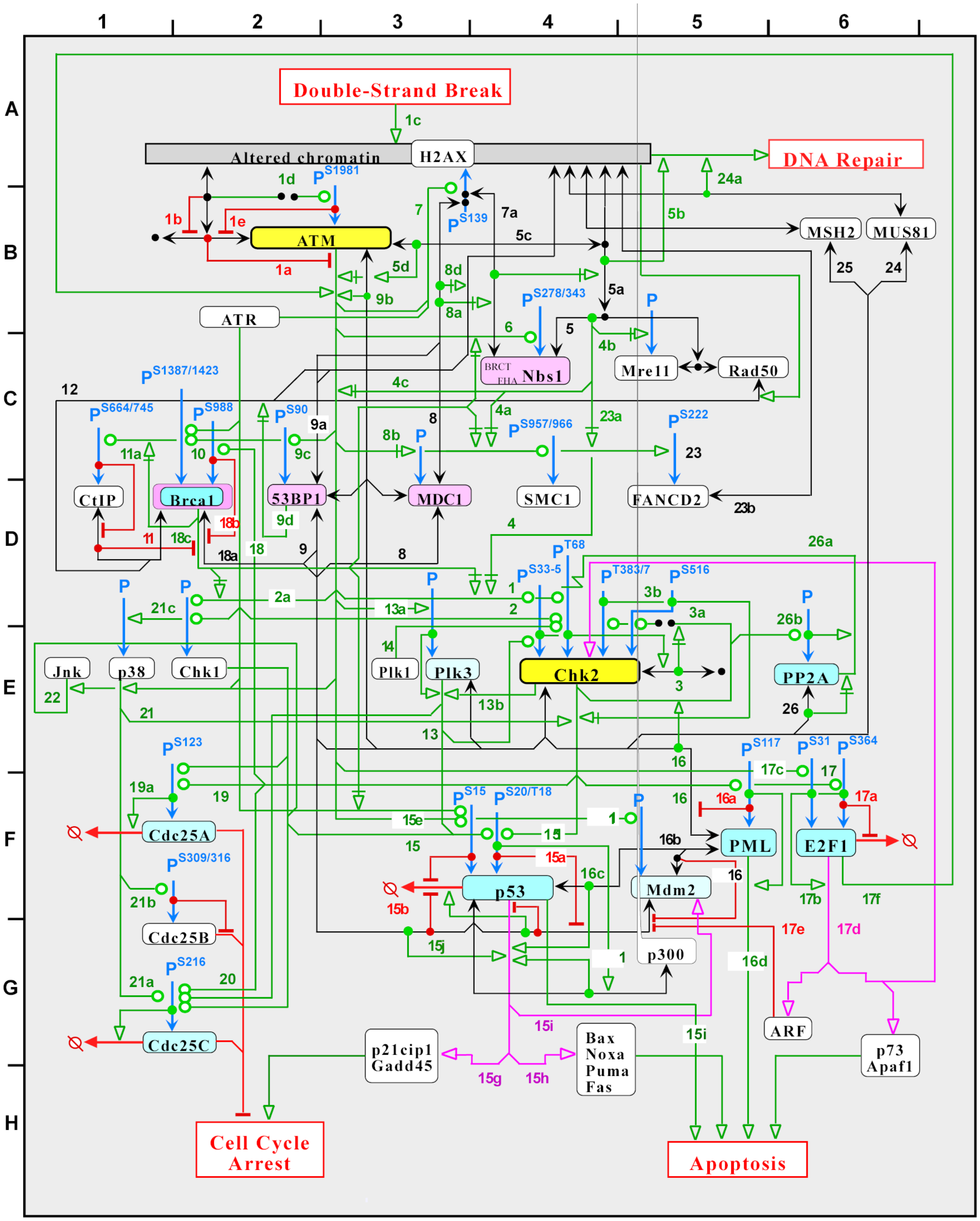

Figure 2: ATM-Chk2 molecular interaction map.

2 Logical Model

In this section we will present a basic language capable of modeling some basic positive and negative interaction between two or more proteins in some pathway. We will first focus on the stimulation and inhibition actions, points (g) and (i) of Figure 1, and then show how this language can be modified to express the different other actions described in the same figure.

2.1 Formal Language

Let's consider a fragment of first order logic with some basic predicates, boolean connectives $(\wedge)$ and, $(\vee)$ or, $(\neg)$ negation, $(\rightarrow)$ implication, $(\leftrightarrow)$ equivalence, $(\exists)$ existential and $(\forall)$ universal quantifiers, and $(=)$ equality.

The basic state predicates are:

- $P(x)$: with intended meaning that the protein x is *Present*.
- $A(x)$: with intended meaning that the protein x is *Active*.
- $I(x)$: with intended meaning that the protein x is *Inhibited*.

The basic state axioms are:

- $\forall x(P(x) \leftrightarrow A(x) \vee I(x))$

  Indicates that a certain *Present* protein x can be either *Active* or *Inhibited*, and that an *Active* or and *Inhibited* protein is considered *Present* in the cell.

- $\neg\exists x(A(x) \wedge I(x))$

  Indicates that a certain protein x can never be in both *Active* and *Inhibited* states at the same time.

An interaction between two or more different proteins is expressed by a predicate of the form *Action*$(protein_1, ..., protein_n)$. In our case we are interested by the simple *Activation* and *Inhibition* actions that are defined by the following predicates:

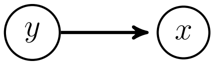

Figure 3: $CAP(y, x)$.

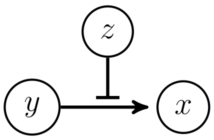

Figure 4: $CICAP(z, y, x)$.

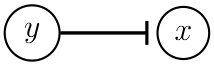

Figure 5: $CIP(y,x)$.

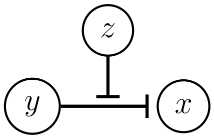

Figure 6: $CICIP(z,y,x)$.

- $CAP(y,x)$: CAP or the *Capacity of Activation* expresses that the protein y has the capacity to activate the protein x. (Figure 3)
- $CICAP(z,y,x)$: $CICAP$ or the *Capacity to Inhibit the Capacity of Activation* expresses that the protein z has the capacity to inhibit the capacity of the activation of x by y. (Figure 4)
- $CIP(y,x)$: CIP or the *Capacity to Inhibit a Protein* expresses that the protein y has the capacity to inhibit the protein x. (Figure 5)
- $CICIP(z,y,x)$: $CICIP$ or the *Capacity to Inhibit the Capacity of Inhibition of a Protein* expresses that the protein z has the capacity to inhibit the capacity of inhibition of x by y. (Figure 6)

In the next section we will define the needed axioms that will be used to model the *Activation* and *Inhibition* actions.

2.2 Action axioms

Giving the fact that a node can acquire the state active or inhibited depending on different followed pathways, one of the issues answered by abduction is to know which set of proteins is required to be active of inhibited for our target protein be active or inhibited.

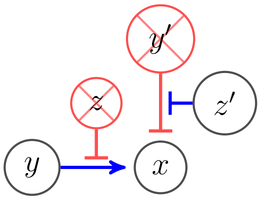

Figure 7: Activation

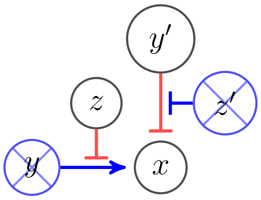

Figure 8: Inhibition

A protein x is active if there exists at least one *active* protein y such as $CAP(y,x)$ that has the capacity to activate x, **and** for every protein z that has the capacity to inhibit the capacity of activation of x by y, such as $CICAP(z,y,x)$, z is *not active*. **And** for every protein y' such as $CIP(y',x)$ that has the capacity to inhibit x, y' is *not active*, **or** there exist at least one *active* protein z' such as $CICIP(z', y', x)$ that has the capacity to inhibit the capacity of inhibition of x by y'. (Figure 7)

A protein x is inhibited if there exists at least one *active* protein y' such as $CIP(y', x)$ that has the capacity to inhibit x, **and** for every protein z' that has the capacity to inhibit the capacity of inhibition of x by y', such as $CICIP(z', y', x)$, z' is *not active*. **And** for every protein y such as $CAP(y, x)$ that has the capacity to activate x, y is *not active*, **or** there exist at least one *active* protein z such as $CICAP(z, y, x)$ that has the capacity to inhibit the capacity of activation of x by y. (Figure 8)

More formally, a protein x is active iff the activation conditions $activ(x)$ are satisfied, and x is not inhibited. And a protein x is inhibited iff the inhibition conditions $inhib(x)$ are satisfied, and x in not active.

Formally we have:

- $\forall x(A(x) \leftrightarrow activ(x) \wedge \neg I(x))$
- $\forall x(I(x) \leftrightarrow inhib(x) \wedge \neg A(x))$

We can deduce in classical logic:

- (A1) $\forall x(activ(x) \wedge \neg inhib(x) \rightarrow A(x))$
- (A2) $\forall x(\neg activ(x) \rightarrow \neg A(x))$
- (I1) $\forall x(inhib(x) \wedge \neg activ(x) \rightarrow I(x))$
- (I2) $\forall x(\neg inhib(x) \rightarrow \neg I(x))$

The activation and inhibition conditions are defined as follow:

- $activ(x) \overset{\text{def}}{=} \exists y(A(y) \wedge CAP(y, x) \wedge \forall z(CICAP(z, y, x) \rightarrow \neg A(z)))$
  In another way, it is sufficient to have $A(y)$ if there exists an activation arc $CAP(y, x)$ going from y to x, and $\neg A(z)$ for all arcs $CICAP(z, y, x)$ that inhibit that arc.
- $inhib(x) \overset{\text{def}}{=} \exists y(A(y) \wedge CIP(y, x) \wedge \forall z(CICIP(z, y, x) \rightarrow \neg A(z)))$
  In another way, it is sufficient to have $A(y)$ if there exists an inhibition arc $CIP(y, x)$ going from y to x, and $\neg A(z)$ for all arcs $CICIP(z, y, x)$ that inhibit that arc.

We can deduce in classical logic:

- $\neg activ(x) \leftrightarrow \forall y(CAP(y,x) \rightarrow (\neg A(y) \vee \exists z(CICAP(z,y,x) \wedge A(z))))$
- $\neg inhib(x) \leftrightarrow \forall y(CIP(y,x) \rightarrow (\neg A(y) \vee \exists z(CICIP(z,y,x) \wedge A(z))))$

Axiomatic of activation

From (A1) and the definitions of *activ* and *inhib*, we have the following activation axiom:

$$\begin{array}{l}\forall x(\exists y(A(y) \wedge CAP(y,x) \wedge \forall z(CICAP(z,y,x) \rightarrow \neg A(z)))\wedge \\ \quad \forall y'(CIP(y',x) \rightarrow (\neg A(y') \vee \exists z'(CICIP(z',y',x) \wedge A(z')))) \rightarrow A(x))\end{array}$$

Axiomatic of inhibition

From (I1) and the definitions of *activ* and *inhib*, we have the following activation axiom:

$$\begin{array}{l}\forall x(\exists y'(A(y') \wedge CIP(y',x) \wedge \forall z'(CICIP(z',y',x) \rightarrow \neg A(z')))\wedge \\ \quad \forall y(CAP(y,x) \rightarrow (\neg A(y) \vee \exists z(CICAP(z,y,x) \wedge A(z)))) \rightarrow I(x))\end{array}$$

2.3 Extension with new states and actions

The basic language defined in 2.1 and 2.2 can be easily extended to express different and more precise node statuses and actions. For example the action of phosphorylation of Chk2 on site S33-5 by ATM can be expressed by the predicate $CP(atm, chk2\_s33\_5, p\_chk2\_s33\_5)$, having $p\_chk2\_s33\_5$ as a result of this phosphorylation.

In a more formal way, the new predicates can be defined as following:

- $CP(z,y,x)$: CP or the *Capacity of Phosphorylation* expresses that the protein z has the capacity to phosphorylate the protein y on a certain site, knowing that x is the result of said phosphorylation.
- $CICP(t,z,y,x)$: $CICP$ or the *Capacity to Inhibit the Capacity of Phosphorylation* expresses that the protein t has the capacity to inhibit the capacity of the phosphorylation of y by z leading to x.

With the previous phosphorylation predicates we can now modify the *activ* property to the following:

$$\begin{aligned} phos(x) \overset{\text{def}}{=} & \exists y1, y2(A(y1) \wedge A(y2) \wedge CP(y1, y2, x) \wedge \\ & \forall z(CICP(z, y1, y2, x) \rightarrow \neg A(z))) \end{aligned}$$

And respectively (A1), (A2), and (I1) to the following:

- (P1) $\forall x(phos(x) \wedge \neg inhib(x) \rightarrow A(x))$
- (P2) $\forall x(\neg phos(x) \rightarrow \neg A(x))$
- (IP1) $\forall x(inhib(x) \wedge \neg phos(x) \rightarrow I(x))$

We can now define the new phosphorylation axiom as:

$$\begin{aligned} & \forall x(\exists y1, y2(A(y1) \wedge A(y2) \wedge CP(y1, y2, x) \wedge \forall z(CICP(z, y1, y2, x) \rightarrow \neg A(z))) \wedge \\ & \quad \forall y'(CIP(y', x) \rightarrow (\neg A(y') \vee \exists z'(CICIP(z', y', x) \wedge A(z')))) \rightarrow A(x)) \end{aligned}$$

$Auto-phosphorylation$, *Dephosphorylation*, *Binding*, *Dissociation* etc. actions that can be found in Figure 2, and some of the newly discovered ones such as *Methylation* and *Ubiquitination* [14, 5] can formalized in a similar fashion.

3 Automated Deduction Method

In this section we define a fragment of first order logic with equality capable of supporting the language of states and actions defined in 2. The properties of this fragment allow us to define a procedure capable of eliminating the quantifiers in this fragment, in other words to transform the first order formulas in formulas without variables, in order to obtain an efficient automated deduction procedure with these fragments.

Definition 3.1 *Restricted formulas are formulas without free variables defined by the following grammar:*

$$\delta \ ::= \ \forall x_1, ..., x_n \ (\varphi \rightarrow \psi) \mid \exists x_1, ..., x_n \ (\varphi \wedge \psi)$$

Where φ is an atomic formula, called domain formula. and ψ is either a restricted formula or a formula without quantifiers, and every variable appearing in a restricted formula must appear in a domain formula.

Examples of this kind of formulas are:

- $\forall x(P(x) \to Q(x))$
- $\forall x(P(x) \to \exists y(Q(y) \wedge R(x,y)))$

Definition 3.2 *A completion formula is a formula of the following form:*

$$\forall x_1, ..., x_n \ (P(x_1, ..., x_n, c_1, ...c_p) \leftrightarrow ((x_1 = a_{1_1} \wedge ... \wedge x_n = a_{1_n}) \vee ... \vee \\ (x_1 = a_{m_1} \wedge ... \wedge x_n = a_{m_n})))$$

Where P is a predicate symbol of arity $n+p$.
Completion formulas are similar to the completion axioms defined by Reiter in [19] where the implication is substituted by an equivalence.

Note : for notation purpose, we will sometimes represent $x_1, ..., x_n$ by $\overline{x}$, and $c_1, ..., c_p$ by $\overline{c}$.

Definition 3.3 *Given a restricted formula φ and a set of completion for φ noted $C(\varphi)$, we say that $C(\varphi)$ saturates φ, if and only if, for each domain formula in φ, there is a unique completion formula in C.*

Definition 3.4 *Given an atomic formula φ and a set $C(\varphi)$ of φ, we define the domain of the variables of φ with respect to $C(\varphi)$, denoted $D(\mathcal{V}(\varphi), C(\varphi))$, where $\mathcal{V}(\varphi)$ represents the variables of φ, as follows:*

if φ is of the form $P(x_1, ..., x_n, c_1, ..., c_p)$ and if in $C(\varphi)$[1] we have a formula of the form:

$$\forall x_1, ..., x_n \ (P(x_1, ..., x_n, c_1, ..., c_p) \leftrightarrow ((x_1 = a_{1_1} \wedge ... \wedge x_n = a_{1_n}) \vee ... \vee \\ (x_1 = a_{m_1} \wedge ... \wedge x_n = a_{m_n})))$$

then

$$D(\mathcal{V}(\varphi), C(\varphi)) = \{< a_{1_1}, ..., a_{1_n} >, ..., < a_{m_1}, ..., a_{m_n} >\}$$

[1] If there isn't a completion formula for a certain predicate P, that means that the extension of that predicate is empty. Formally this is represented by $\forall x_1, ..., x_m(P(x_1, ..., x_n, c_1, ..., c_p) \leftrightarrow false$.

Quantification elimination procedure

Let φ be a restricted formula of the following forms: $\forall\overline{x}(\varphi_1(\overline{x}) \rightarrow \varphi_2(\overline{x}))$ or $\exists\overline{x}(\varphi_1(\overline{x}) \wedge \varphi_2(\overline{x}))$, let $C(\varphi_1(\overline{x}))$ a set of completion formulas for φ_1, then we define recursively a translation T, allowing to replace universal (existential) quantifiers by conjunction (disjunction) of formulas where quantified variables are substituted by constants as follows:

- if $D(\mathcal{V}(\varphi_1), C(\varphi_1)) = \{< \overline{c_1} >, ..., < \overline{c_n} >\}$ with $n > 0$:
  - $T(\forall\overline{x}(\varphi_1(\overline{x}) \rightarrow \varphi_2(\overline{x})), C(\varphi)) = T(\varphi_2(\overline{c_1}), C(\varphi_2)) \wedge ... \wedge T(\varphi_2(\overline{c_n}), C(\varphi_2))$
  - $T(\exists\overline{x}(\varphi_1(\overline{x}) \wedge \varphi_2(\overline{x})), C(\varphi)) = T(\varphi_2(\overline{c_1}), C(\varphi_2)) \vee ... \vee T(\varphi_2(\overline{c_n}), C(\varphi_2))$
- if $D(\mathcal{V}(\varphi_1), C(\varphi_1)) = \varnothing$:
  - $T(\forall\overline{x}\ (\varphi_1(\overline{x}) \rightarrow \varphi_2(\overline{x}))\ ,\ C(\varphi)) = True$
  - $T(\exists\overline{x}\ (\varphi_1(\overline{x}) \wedge \varphi_2(\overline{x}))\ ,\ C(\varphi)) = False$

Note : for a given formula φ, it will be noted that the translation T of φ allows to eliminate a set of quantifiers, in other words the set of variables symbols in φ. This procedure can be considered as kind of compilation to first order logic without variables and without equality.

Then in the theory $\mathcal{T}$ in which we have the axioms of equality and axioms of the form $\neg(a = b)$ for each constant a and b representing different objects, which are called unique name axioms by Reiter in [19], we have the following main theorem:

Theorem 3.5 *Let φ be a restricted formula, and $C(\varphi)$ a saturated completion set of formulas of the domain formulas of φ, then:*

$$\mathcal{T},\ C(\varphi) \vdash \varphi \leftrightarrow T(\varphi, C(\varphi))$$

We will now present two examples of translation from first order logic formulas composed of action and state axioms to variable free formulas:

Example 3.6 *In the following example we apply the translation procedure to the axioms defined in section 2, we consider a certain protein a with the following completion axioms:*

$\forall y(CAP(y, a) \leftrightarrow y = b_1 \vee ... \vee y = b_n)$
If there is no arc $CAP(b_i, a)$:

$\forall y(CAP(y,a) \leftrightarrow true)$

For each b_i *we have a completion axiom of the form:*
$\forall z(CICAP(z,b_i,a) \leftrightarrow z = c_{i,1} \vee ... \vee z = c_{i,n_i})$
If there is no arc $CICAP(c_{i,j}, b_i, a)$*:*
$\forall z(CICAP(z,b_i,a) \leftrightarrow false)$

We also have:
$\forall y(CIP(y,a) \leftrightarrow y = d_1 \vee ... \vee y = d_m)$
If there is no arc $CIP(d_i, a)$*:*
$\forall y(CIP(y,a) \leftrightarrow true)$

For each d_i *we have a completion axiom of the form:*
$\forall z(CICIP(z,d_i,a) \leftrightarrow z = e_{i,1} \vee ... \vee z = e_{i,n_i})$
If there is no arc $CICIP(e_{i,j}, d_i, a)$*:*
$\forall z(CICIP(z,d_i,a) \leftrightarrow false)$

From the above and the definitions of activ and inhib in 2.2, we can deduce the following:

$$\begin{aligned} activ(a) \leftrightarrow (A(b_1) \wedge \forall z(CICAP(z,b_1,a) \rightarrow \neg A(z))) \vee ...\vee \\ (A(b_n) \wedge \forall z(CICAP(z,b_n,a) \rightarrow \neg A(z))) \end{aligned}$$

and

$$\begin{aligned} activ(a) \leftrightarrow (A(b_1) \wedge \neg A(c_{1,1}) \wedge ... \wedge \neg A(c_{1,n_1})) \vee ...\vee \\ (A(b_n) \wedge \neg A(c_{n,1}) \wedge ... \wedge \neg A(c_{n,n_n})) \end{aligned}$$

Following the same reasoning we also have:

$$\begin{aligned} \neg activ(a) \leftrightarrow (\neg A(b_1) \vee \exists z(CICAP(z,b_1,a) \wedge A(z))) \wedge ...\wedge \\ (\neg A(b_n) \vee \exists z(CICAP(z,b_n,a) \wedge A(z))) \end{aligned}$$

and

$$\begin{aligned} \neg activ(a) \leftrightarrow (\neg A(b_1) \vee A(c_{1,1}) \vee ... \vee A(c_{1,n_1})) \wedge ...\wedge \\ (\neg A(b_n) \vee A(c_{n,1}) \vee ... \vee A(c_{n,n_n})) \end{aligned}$$

We can also define inhib by:

$$\begin{gathered} inhib(a) \leftrightarrow (A(d_1) \wedge \forall z(CICIP(z, d_1, a) \rightarrow \neg A(z))) \vee ... \vee \\ (A(d_n) \wedge \forall z(CICIP(z, d_n, a) \rightarrow \neg A(z))) \end{gathered}$$

and

$$\begin{gathered} inhib(a) \leftrightarrow (A(d_1) \wedge \neg A(e_{1,1}) \wedge ... \wedge \neg A(e_{1,m_1})) \vee ... \vee \\ (A(d_m) \wedge \neg A(e_{m,1}) \wedge ... \wedge \neg A(e_{m,m_m})) \end{gathered}$$

And finally:

$$\begin{gathered} \neg inhib(a) \leftrightarrow (\neg A(d_1) \vee \exists z(CICIP(z, d_1, a) \wedge A(z))) \wedge ... \wedge \\ (\neg A(d_m) \vee \exists z(CICIP(z, d_m, a) \wedge A(z))) \end{gathered}$$

and

$$\begin{gathered} \neg inhib(a) \leftrightarrow (\neg A(d_1) \vee A(e_{1,1}) \vee ... \vee A(e_{1,m_1})) \wedge ... \wedge \\ (\neg A(d_m) \vee A(e_{m,1}) \vee ... \vee A(e_{m,m_m})) \end{gathered}$$

Then these axioms can be used in the previously defined (A1), (A2), (I1) and (I2).

Example 3.7 *Let's consider another example where a protein b has the capacity to activate another protein a, and that two other proteins c_1 and c_2 have the capacity to inhibit the capacity of activation of a by b. This proposition can be expressed by the following completion axioms:*

- $\forall y(CAP(y, a) \leftrightarrow y = b)$*: Expresses that b is the only protein that has the capacity to activate a.*
- $\forall z(CICAP(z, b, a) \leftrightarrow z = c_1 \vee z = c_2)$*: Expresses that c_1 and c_2 are the only proteins that have the capacity to inhibit the capacity of activation of a by b.*

From the definition of activ, we can deduce:

$$activ(a) \leftrightarrow (A(b) \wedge \forall z(CICAP(z, b, a) \rightarrow \neg A(z)))$$

then

$$activ(a) \leftrightarrow (A(b) \wedge \neg A(c_1) \wedge \neg A(c_2))$$

Which means that the property $activ(a)$ is satisfied iff the protein b is active and both proteins c_1 and c_2 are not active.

We can also deduce using the same reasoning:

$$\neg activ(a) \leftrightarrow (\neg A(b) \vee A(c_1) \vee A(c_2))$$

Which means that the property $\neg activ(a)$ is satisfied iff the protein b is not active or one proteins c_1 and c_2 is active.

Let's also consider that a protein d has the capacity to inhibit the protein a and that there is no proteins capable of inhibiting the capacity of inhibition of a by d. This proposition can be expressed by the following completion axioms:

- $\forall y(CIP(y,a) \leftrightarrow y = d)$*: Expresses that d is the only protein that has the capacity to inhibit a.*
- $\forall z(CICIP(z,d,a) \leftrightarrow false)$*: Expresses that there are no proteins capable of inhibiting the capacity of inhibition of a by d.*

From the definition of inhib, we can deduce:

$$inhib(a) \leftrightarrow (A(d) \wedge \forall z(CICIP(z,b,a) \rightarrow \neg A(z)))$$

then

$$inhib(a) \leftrightarrow A(d)$$

Which means that the property $inhib(a)$ is satisfied iff the protein d is active.

We can also deduce using the same reasoning:

$$\neg inhib(a) \leftrightarrow (\neg A(d))$$

Which means that the property $\neg inhib(a)$ is satisfied iff the protein b is not active.

Using the axioms (A1) and (I1) defined in 2.2:

- *(A1)* $\forall x(activ(x) \wedge \neg inhib(x) \rightarrow A(x))$
- *(I1)* $\forall x(inhib(x) \wedge \neg activ(x) \rightarrow I(x))$

We can finally deduce:

$$(A(b) \wedge \neg A(c_1) \wedge \neg A(c_2) \wedge \neg A(d)) \rightarrow A(a)$$

Which means that the protein a is active if the protein b is active and the proteins c_1, c_2, d are not active.

And

$$(A(d) \wedge (\neg A(b) \vee A(c_1) \vee A(c_2))) \rightarrow I(a)$$

Which means that the protein a is inhibited if the protein d is active and either b is not active or c_1 or c_2 are active.

4 Queries and Results

From what we defined in sections 2 and 3, we can now model metabolic pathways using the *activ* and *inhib* properties and the translation mechanism defined in 3. The resulting axioms are of the following type *conditions* → *results*, and can be chained together to create a series of reactions forming our pathway. Then questions of two different types can be answered using *deduction* or *abduction* reasoning.

Questions answered by deduction request all entities that satisfy a given property. In our case, we may have some information about states and actions of certain proteins in some knowledge base (**KB**). A question can be of the following form: *What is the result of reactions formed by the proteins of **KB***, or in other means, *what is the state (active or inhibited) of the proteins that result from the reactions formed by proteins of **KB***.

And questions answered by abduction looks for minimal assumptions that must be added to **KB** to derive that a certain fact is true. For instance, we may have some informations about actions of certain proteins in **KB**. A question can be of the following form: *What are the reactions that are needed to deduce that a certain protein is active or inhibited*, in other means, *what are the proteins and their respective states (active or inhibited) that should be present in order to derive that a certain protein is active or inhibited.*

Both types of questions can be addressed in SOLAR (SOL for Advanced Reasoning) [13] a first-order clausal consequence finding system based on SOL (Skip Ordered Linear) tableau calculus [6, 21].

In the following we are going to show an example based on figure 9, demonstrating both deduction and abduction type queries.

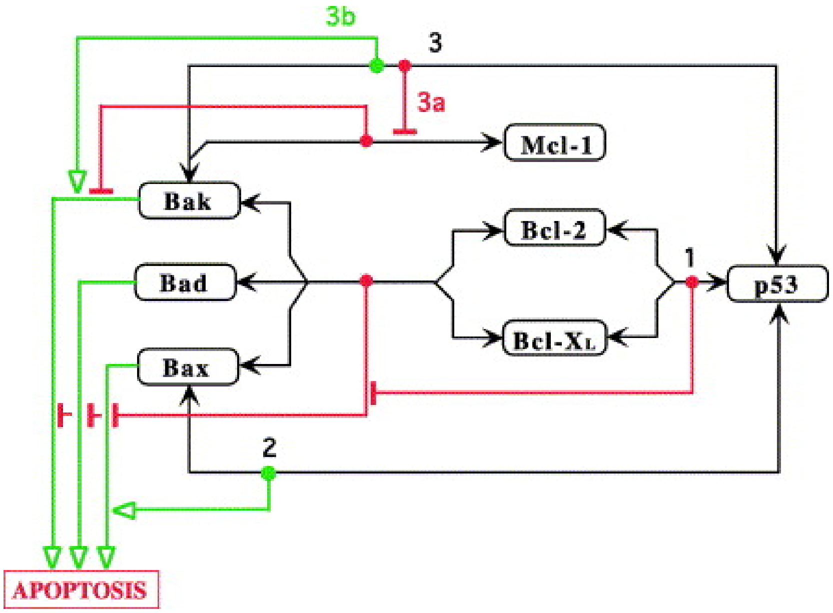

Figure 9: Mitochondrial apoptosis induced by p53 independently of transcription

In Figure 9 the metabolic network shows how p53 can induce mitochondrial apoptosis independently of transcription. Three coherent pathways have been found [10]:

1. p53 can bind directly to Bcl-2 and Bcl-XL, and block their interaction with pro-apoptotic Bcl-2 proteins (Bak, Bad, and Bax).
2. p53 can bind and activate Bax oligomerization.
3. p53 can bind to Bak, block its interaction with the anti-apoptotic Bcl-2 protein Mcl-1 (3a), and promote Bak oligomerization and induction of apoptosis (3b).

Following section 2.3 we can define new predicates to suit the needs of the pathway:

- $CB(z, y, x)$: CB or the *Capacity of Binding* expresses that the protein z has the capacity to bind with the protein y, knowing that x is the result of said binding.
- $CICB(t, z, y, x)$: $CICB$ or the *Capacity to Inhibit the Capacity of Binding* expresses that the protein t has the capacity to inhibit the capacity of the binding of z and y leading to x.

For example, these new predicates can be used to model the binding between p53 and Bak using the predicate $CB(p53, bak, p53\_bak)$ where p53_bak is the complex formed by such binding. In a similar fashion we can define the other needed predicates, like the binding predicate that gives us the possibility to bind three proteins together, such as p53 binding to Bcl-2 and Bcl-XL, and the binding predicate that gives us the possibility to bind 5 proteins together, such as Bcl-2, Bcl-XL, Bax, Bad, and Bak.

With these new predicates, new axiomatic can be defined that would enrich the descriptive capacities of the old axiomatic, as seen in 2.3. Then the translation procedure can be applied to these axioms and to the completion axiomatic that defines actions between proteins. And finally deduction and abduction can be applied to the resulting clauses to answer queries as shown above.

Applying the translation procedure of section 3 the axioms of Figure 9 can be of the following form:

1. $A(p53) \wedge A(bak) \rightarrow A(bak\_p53)$
   bak_p53 is the result of the binding between p53 and Bak.

2. $A(bak\_p53) \rightarrow I(bak\_mcl)$
   bak_mcl is the result of binding between Bak and Mcl-1.

3. $A(bak\_p53) \wedge \neg A(b\_complex) \wedge \neg A(bak\_mcl) \rightarrow A(apoptosis)$
   $b\_complex$ is result of the binding between Bcl-2, Bcl-XL, Bak, Bad, and Bax.

4. $A(bak) \wedge \neg A(b\_complex) \wedge \neg A(bak\_mcl) \rightarrow A(apoptosis)$

5. $A(p53) \wedge A(bcl) \rightarrow A(p53\_bb\_complex)$
   bcl represents Bcl-2 and Bcl-XL.
   $p53\_bb\_complex$ is the result of binding between p53, Bcl-2 and Bcl-XL.

6. $A(p53\_bb\_complex) \rightarrow I(b\_complex)$

7. $A(bax) \wedge \neg A(b\_complex) \rightarrow A(apoptosis)$

8. $A(p53) \wedge A(bax) \wedge \neg A(b\_complex) \rightarrow A(apoptosis)$

9. $A(bad) \wedge \neg A(b\_complex) \rightarrow A(apoptosis)$

If we want to know what are the proteins and their respective states that should be present in order to derive that the cell reached apoptosis, the answer is given by applying abduction over the previous set of transformed clauses. In the set of consequences returned by SOLAR we can find the following:

- $A(p53) \wedge A(bcl) \wedge A(bak)$: is a plausible answer, because p53 can bind to Bcl giving the $p53\_bb\_complex$, which can in return inhibit the $b\_complex$ that is responsible of inhibiting the capacity of Bak to activate the cell's apoptosis. That is why it is sufficient to for this case to have p53, Bcl, and Bak in an active state to reach apoptosis.

- Another interpretation of the previous answer is that p53 can also bind to Bak giving the *bak_p*53 protein, which can in return inhibit the *bak_mcl* responsible of inhibiting the capacity of Bak to activate the cell's apoptosis. *bak_p*53 can also stimulate Bak to reach apoptosis. Without forgetting that *p*53*_bb_complex* should be inhibiting *b_complex*.

Now if we already know that the proteins p53, Bcl, and Bax are present and active in the cell, we can ask if the cell can reach apoptosis with such conditions. The answer is given by deduction over the previous set of transformed clauses plus the following observations $A(p53)$, $A(bcl)$, and $A(bax)$. SOLAR returns two found consequences, which means that there are two different possible pathways that can be followed by having those condition that enables the cell to reach apoptosis. Figure 9 shows:

- p53 can bind to Bcl giving the *p*53*_bb_complex*, which can in return inhibit the *b_complex* that is responsible of inhibiting the capacity of Bax to activate the cell's apoptosis. That is why it is sufficient to for this case to have p53, Bcl, and Bak in an active state to reach apoptosis.

- The second interpretation suggests that p53 can bind to Bax, stimulating the Bax activation the cell's apoptosis. Taking into consideration that *b_complex* should be inhibited as shown above.

5 Conclusion

A new language has been defined in this paper capable of modeling both positive and negative causal effects between proteins in a metabolic pathway. We showed how this basic language can be extended to include more specific actions that describes different relations between proteins. These extensions are important in this context, because there is always the possibility that new types of actions are discovered through biological experiments [14, 5]. We later showed how the axioms defined in such languages can be compiled against background knowledge, in order to form a new quantifier free axioms that could be used in either deduction or abduction reasoning. Although the first order axioms can be also well used to answer queries by deduction or abduction methods, the main advantage of translated axioms is their low computation time needed in order to derive consequences.

Future works can focus on useful methods for introducing the notion of time and quantities in the former model. Trying to get as precise as possible in describing such pathways can help biologists discover contradictory informations and guide them during experiments knowing how huge the cells metabolic networks have become (Figure 2). One of the constraints that can also be introduced is the notion of *Aboutness* [3] that can limit and focus search results to what seems relevant to a single or a group of entities (proteins).

Acknowledgements

This work is partially supported by the Région Midi-Pyrénées project called CLE, the Lebanese National Council for Scientific Research (LNCSR), and the French-Spanish lab LIRP.

We would like to thank Gilles Favre, Jean-Charles Faye and Olivier Sordet for their precious metabolic network knowledge and comments that were used as a base for this paper. We would also like to thank Katsumi Inoue and Hidetomo Nabeshima for their important help with SOLAR.

References

[1] R. Demolombe and L. Fariñas del Cerro. An Inference Rule for Hypothesis Generation. In *Proc. of International Joint Conference on Artificial Intelligence*, Sydney, 1991.

[2] Robert Demolombe. Syntactical characterization of a subset of domain-independent formulas. *J. ACM*, 39(1):71–94, 1992.

[3] Robert Demolombe and Luis Fariñas del Cerro. Information about a given entity: From semantics towards automated deduction. *J. Log. Comput.*, 20(6):1231–1250, 2010.

[4] Martin Erwig and Eric Walkingshaw. Causal reasoning with neuron diagrams. In *Proceedings of the 2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, VLHCC '10, pages 101–108, Washington, DC, USA, 2010. IEEE Computer Society.

[5] V Glorian, G Maillot, S Poles, J S Iacovoni, G Favre, and S Vagner. Hur-dependent loading of mirna risc to the mrna encoding the ras-related small gtpase rhob controls its translation during uv-induced apoptosis. *Cell Death Differ*, 18(11):1692–70, 2011.

[6] Katsumi Inoue. Linear resolution for consequence finding. *Artificial Intelligence*, 56(23):301 – 353, 1992.

[7] Katsumi Inoue, Andrei Doncescu, and Hidetomo Nabeshima. Hypothesizing about causal networks with positive and negative effects by meta-level abduction. In *Proceedings of the 20th international conference on Inductive logic programming*, ILP'10, pages 114–129, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] Katsumi Inoue, Andrei Doncescu, and Hidetomo Nabeshima. Completing causal networks by meta-level abduction. *Machine Learning*, 91(2):239–277, 2013.

[9] Antonis Kakas, Alireza Tamaddoni Nezhad, Stephen Muggleton, and F Pazos. Modelling inhibition in metabolic pathways through Abduction and Induction. pages 305–322, 2004.

[10] Kurt W Kohn and Yves Pommier. Molecular interaction map of the p53 and mdm2 logic elements, which control the off-on swith of p53 response to dna damage. *Biochem Biophys Res Commun*, 331(3):816–27, 2005.

[11] Won-Jeong Lee, Dong-Uk Kim, Mi-Young Lee, and Kang-Yell Choi. Identification of proteins interacting with the catalytic subunit of pp2a by proteomics. *PROTEOMICS*, 7(2):206–214, 2007.

[12] Stephen Muggleton and Christopher H. Bryant. Theory completion using inverse entailment. In *Proceedings of the 10th International Conference on Inductive Logic Programming*, ILP '00, pages 130–146, London, UK, UK, 2000. Springer-Verlag.

[13] Hidetomo Nabeshima, Koji Iwanuma, Katsumi Inoue, and Oliver Ray. Solar: An automated deduction system for consequence finding. *AI Commun.*, 23(2-3):183–203, April 2010.

[14] Huadong Pei, Lindsey Zhang, Kuntian Luo, Yuxin Qin, Marta Chesi, Frances Fei, P. Leif Bergsagel, Liewei Wang, Zhongsheng You, and Zhenkun Lou. Mmset regulates histone h4k20 methylation and 53bp1 accumulation at dna damage sites. *Nature*, 470(7332):124–128, 2011.

[15] Y. Pommier, O. Sordet, V.A. Rao, H. Zhang, and K.W. Kohn. Targeting chk2 kinase: molecular interaction maps and therapeutic rationale. *Curr Pharm Des*, 11(22):2855–72, 2005.

[16] Oliver Ray. Automated abduction in scientific discovery. In *Model-Based Reasoning in Science and Medicine*, pages 103–116. Springer, June 2007.

[17] Oliver Ray, Ken Whelan, and Ross King. Logic-based steady-state analysis and revision of metabolic networks with inhibition. In *Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems*, CISIS '10, pages 661–666, Washington, DC, USA, 2010. IEEE Computer Society.

[18] Philip G. K. Reiser, Ross D. King, Douglas B. Kell, Stephen H. Muggleton, Christopher H. Bryant, and Stephen G. Oliver. Developing a logical model of yeast metabolism. *Electronic Transactions in Artificial Intelligence*, 5:233–244, 2001.

[19] R. Reiter. Readings in nonmonotonic reasoning. chapter On closed world data bases, pages 300–310. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[20] J.R. Shoenfield. *Mathematical logic.* Addison-Wesley series in logic. Addison-Wesley Pub. Co., 1967.

[21] P. Siegel. *Representation et utilisation de la connaissance en calcul propositionnel.* Thèse d'État, Université d'Aix-Marseille II, Luminy, France, 1987.

[22] Alireza Tamaddoni-Nezhad, Antonis C. Kakas, Stephen Muggleton, and Florencio Pazos. Modelling inhibition in metabolic pathways through abduction and induction. In Rui Camacho, Ross D. King, and Ashwin Srinivasan, editors, *Inductive Logic Programming, 14th International Conference, ILP 2004, Porto, Portugal, September 6-8, 2004, Proceedings*, volume 3194 of *Lecture Notes in Computer Science*, pages 305–322. Springer, 2004.

Robert Demolombe
IRIT-CNRS, Université de Toulouse,
France
E-mail: robert.demolombe@orange.fr

Luis Fariñas del Cerro
IRIT-CNRS, Université de Toulouse,
France
E-mail: luis.farinas@irit.fr

Naji Obeid
IRIT-CNRS, Université de Toulouse,
France
E-mail: naji.obeid@irit.fr