Scala 上に実現した生物の代謝パスウェイ解析用のドメイン特化言語について

宋 剛秀　馬場 知哉

生体内の細胞は多くの化学反応の連鎖によって活動を維持している．システム生物学において，そのような化学反応の相互作用は代謝パスウェイと呼ばれるネットワークによって表現され，その解析を行うことは重要な課題の一つとなっている．本発表では，オブジェクト指向言語と関数型言語が融合された Scala 上に実現した代謝パスウェイ解析用のドメイン特化言語 Scarabio の説明をを行う．

Cells in living organism maintain their activity by chains of chemical reactions. In Systems Biology, such interactions of chemical reactions are represented in a network called metabolic pathway. Nowadays, the analysis of metabolic pathways is an important research topic in Systems Biology. In this paper, we describe a Domain-Specific Language in Scala for the Analysis of Metabolic Pathways.

1 はじめに

生体内の細胞は多くの化学反応の連鎖によって活動を維持している．システム生物学において，そのような化学反応の相互作用は**代謝パスウェイ**と呼ばれるネットワークによって表現される．代謝パスウェイの数学的なモデルを構築し解析することで遺伝子欠損が細胞に与える影響の予測が可能となり，創薬に必要なアミノ酸などの代謝物を効率よく合成する微生物の設計への応用などを期待できる [26].

これまで小規模の代謝パスウェイに対しては微分方程式系を用いる数学的モデルが一般的であったが，大きな代謝パスウェイに対しては方程式中のパラメータの決定が困難になるため適用が少なかった．それに対して近年では細胞規模の巨大な代謝パスウェイに対して制約を用いた数学的モデル (**制約モデル**) が研究されている [6]．例えば米科学誌セルに 2012 年に発表され話題を集めたマイコプラズマ菌の細胞解析にも制約モデルが採用されるなど [17]，制約モデルを用いた解析の注目度は高い．このため制約モデルを用いた代謝パスウェイの解析ツールが活発に研究・開発されている [30] [36] [5] [16] [18] [5].

このような代謝パスウェイ解析ツールの開発では以下の点が課題になると考えられる．

- 解析ツールの利用者は生物学（システム生物学)、創薬・代謝工学、計算機科学分野の研究者と想定される．利用者が幅広く背景知識も異なるため，誰でも簡単に代謝パスウェイを記述できる必要がある．
- 代謝パスウェイの制約モデルの代表的なものとして Elementary Mode (EM) 解析 [29] [28] [27] があるが，さらに高度な解析を行うためには文献 [2] [13] [37] [32] [21] で研究されているような遺伝子欠損の影響を考慮する等の様々な拡張を容易に行える必要がある．
- 標準的な計算機上で動作し，高速に解析結果を返す必要がある．

これまで提案されてきたツール [36] [5] [16] [18] は，与えられたパスウェイの EM を求めるために Double

---

A Domain-Specific Language in Scala for the Analysis of Metabolic Pathways.

Takehide Soh, 神戸大学 情報基盤センター, Information Science and Technology Center, Kobe University.

Tomoya Baba, 情報・システム研究機構 新領域融合研究センター, Transdisciplinary Research Integration Center, Research Organization of Information and Systems.

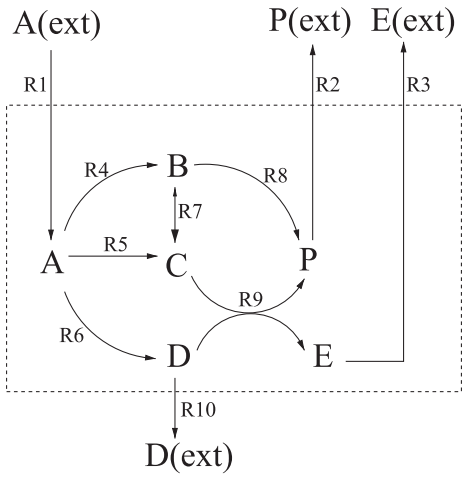

図 1　代謝パスウェイの例 (文献 [6])

```
 1: import jp.kobe_u.scarabio._
 2:
 3: val p = Pathway()
 4:
 5: p.add(CSum() ==> 'A as "R1")
 6: p.add('P ==> CSum() as "R2")
 7: p.add('E ==> CSum() as "R3")
 8: p.add('A ==> 'B as "R4")
 9: p.add('A ==> 'C as "R5")
10: p.add('A ==> 'D as "R6")
11: p.add('B <=> 'C as "R7")
12: p.add('B ==> 'P as "R8")
13: p.add('C + 'D ==> 'E + 'P as "R9")
14: p.add('D ==> CSum() as "R10")
15:
16: val esol = EmSolver(p)
17: esol.findEM
```

図 2　Scarabio を用いたプログラム例

Description (DD) 法[9] を用いたアルゴリズムを実装しているが，専用のアルゴリズムのため二つ目の課題である制約モデルの拡張を行うのが難しい．また DD 法では，一つだけ解を出力することが出来ず，常に全解列挙を行う．そのためパスウェイが大きくなると計算に時間がかかったり大量のメモリが必要になることがあった．

本論文では，上記の課題を解決することを目的として設計・実装した，生物の代謝パスウェイ解析用のドメイン特化言語 (Domain-specific Language; DSL) である Scarabio について述べる．Scarabio は Scala の内部 DSL として実装され，近年急速に発展している SAT ソルバーを用いた SAT 型制約プログラミングシステム Scarab[33] を制約モデルの記述および計算エンジンとして採用しており，次の特長を持つ．

- Scala の持つ演算子の多重定義や暗黙変換を用いることで代謝パスウェイの構成要素を簡潔に記述することができる．
- Scarab の制約記述用 DSL を用いることで EM の制約モデルを簡潔に記述することができる．Scarab は動的な制約の追加や削除にも対応しており，制約モデルの拡張を容易にすることを期待できる．
- SAT ソルバーの求解能力を利用し，高速に複数の EM を計算することができる．

本論文の構成は以下になる．まず第 2 節で簡単な代謝パスウェイの例を通して Scarabio を使ったプログラムについて説明する．本節の目的は Scarabio の感触を伝えることである．第 3 節で, Scarabio を構成する要素技術である汎用プログラミング言語 Scala と SAT 型制約プログラミングシステム Scarab の説明を行う．次に第 4 節で，Scala 上に実現した Scarabio DSL と Scarab を利用した EM 解析について説明する．最後に第 5 節で本論文をまとめる．

2　代謝パスウェイと Scarabio プログラム例

2.1　代謝パスウェイの例

図 1 は，代謝パスウェイをネットワーク（グラフ）で表したものである．一般的にパスウェイはこのようにラベル付きの有向ハイパーグラフで表されることが多い．頂点は化学化合物 (Chemical Compounds) を表しており，有向辺は化学反応 (Chemical Reaction) を表している．以降，化学化合物，化学反応を単純に化合物，反応と呼ぶ．このパスウェイには A, B, C, D, E, P の 6 個の化合物と R1 から R10 までの 10 個の反応が含まれている．R1, R2, R3, R10 は反応物もしくは生成物を持たない特別な反応であり，パスウェイの入力と出力を表す．R7 は可逆反応であり，双方向の反応が可能である．

2.2 代謝パスウェイ例の Scarabio DSL 表現．

前節のパスウェイを Scarabio DSL で記述したものを図 2 に示す．1 行目はパスウェイに関する DSL クラスをインポートしている．3 行目はパスウェイ p の定義を行っている．5 行目から 14 行目は，上述した 10 個の反応を定義しパスウェイ p へと追加している．このように Scarabio DSL では Scala 言語の特長である暗黙変換，演算子の多重定義，メソッドにおけるピリオドの省略などを用いることで，化学反応式とその名前の紐付けを自然に記述可能である．次に 16 行目は EM 解析を行うための解析ソルバーを定義しており，17 行目はソルバーを使って EM を計算している．

Scarabio では，このプログラムが示すように基本的な機能として代謝パスウェイの記述と EM を用いた解析を行うことができる．また第 4.3 節に記述するように EM を解析する Scarabio のプログラム自体も 40 行ほどで書かれており，利用者は自由に追加の条件を記述してより高度な解析へと応用することができる．

3 Scala と Scarab

本節では Scarabio の説明の準備として，Scarabio を構成する要素技術である汎用プログラミング言語 Scala と SAT 型制約プログラミングシステム Scarab について概説する．なおこれらの詳細については本論文の範囲外である．Scala については文献 [24] を Scarab については文献 [33] を参照されたい．

3.1 Scala の概要と特徴

Scala[†1] は Martin Odersky により設計された比較的新しいプログラミング言語で，Twitter の分散 DB フレームワークに用いられたことなどもあり，近年注目を集めている [24]．言語上の特長としては，関数型言語とオブジェクト指向言語の融合，強力な型推論，型安全性，高階関数，不変コレクションなどが挙げられる．

処理系としては，JVM (Java Virtual Machine) へのコンパイラとインタラクティブな実行環境 (REPL; Read Eval Print Loop) が用意されている．Java との親和性は高く，Java のクラス・ライブラリをそのまま利用できる．

さらに Scala は，関数型言語およびオブジェクト指向言語としての高度な記述能力，リスト，マップ，集合等の豊富なコレクションフレームワーク，演算子の多重定義や柔軟な構文，オブジェクトのメソッドのインポート機能など，埋込みの **ドメイン特化言語** (**DSL**; Domain-specific Language) を実装するために適した機能を数多く備えている [20]．特に Scarabio DSL ではケースクラス，暗黙変換，シングルトンオブジェクトおよびそのオブジェクトからのインポートなどの機能を利用しているが，これらの詳細については文献 [24] を参照されたい．

3.2 Scarab の概要と特徴

2000 年以降命題論理の充足可能性判定 (SAT) 問題を解くためのプログラムである **SAT ソルバー**の性能が飛躍的に向上しており [7] [12] [23]，このような性能向上は与えられた問題を SAT 問題に符号化し，SAT ソルバーを用いて解を求める **SAT 型システム**の開発を促進してきた．これまで論理合成，プランニング，スケジューリング，ハードウェア・ソフトウェアの検証などで SAT 型システムが成功をおさめている [12] [4] [35] [22]．

Scarab は制約プログラミングのためのドメイン特化言語, SAT 符号化モジュール, SAT ソルバーへのインターフェースから構成されており，制約プログラミングシステム利用者を対象に，表現性，変更容易性，効率性を備えたワークベンチを提供することを目的としたツールである [33]．Scarab は Scala 言語上にコンパクトに実装されている．バックエンドの SAT ソルバーに Java で実装された Sat4j を用いることにより，インクリメンタル解法等の高度な解法を容易に実現できる点も特長の一つである．

4 代謝パスウェイ解析用 DSL: Scarabio

Scarabio の設計方針は代謝パスウェイ解析を行う研究者・開発者に表現性，効率性，変更性，可搬性を備えたツールを提供することである．このために関数型言語とオブジェクト指向言語が融合され DSL の表

†1 http://www.scala-lang.org/

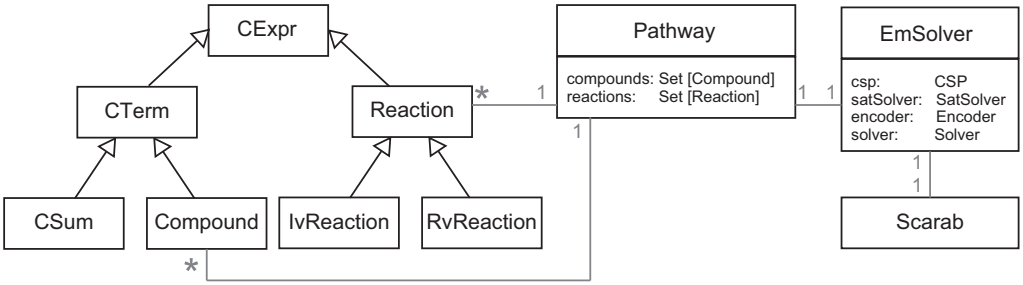

図 3　Scarabio のクラス図

```
1: abstract class CExpr
2: abstract class CTerm extends CExpr
3: case class Compound(name: String, compartment: String = "c") extends CTerm
4: case class CSum(coef: Map[Compound, Int]) extends CTerm
5:
6: abstract class Reaction(l: CSum, r: CSum) extends CExpr
7: case class IvReaction(l: CSum, r: CSum) extends Reaction(l,r)
8: case class RvReaction(l: CSum, r: CSum) extends Reaction(l,r)
```

図 4　化合物と反応に関するクラス定義 (一部)

現力に優れた Scala を記述言語として採用し，また制約記述の簡潔さと SAT ソルバーの求解能力を備えた Scarab を計算エンジンに採用する．

- **表現性:** Scala 言語の全ての機能に加えてパスウェイの記述に Scarabio DSL, 制約モデルの記述に Scarab DSL を用いる．
- **変更性:** Scarab DSL で記述可能な制約を使って自由に制約モデルを拡張・変更可能にする．
- **効率性:** 求解性能が近年飛躍的に向上している SAT ソルバーをパスウェイ解析に用いる．
- **可搬性:** ツール全体を JVM 上で動作させ，可搬性のあるシステムを実現する．

以降では，まず Scarabio の全体構成を説明し，その後パスウェイ記述のための DSL クラス，Scarab を利用した EM 解析のためのソルバークラスについて説明する．

4.1　Scarabio の構成

Scarabio 全体のクラス図を図 3 に示す．図の左側のクラスがパスウェイ (`Pathway` クラス) を記述するための Scarabio DSL のためのクラスであり，反応 (`Reaction` クラス) や化合物 (`Compound` クラス) などで構成される．EM 解析のためのソルバー (`EmSolver` クラス) はパスウェイを入力にとし，SAT 型制約プログラミングシステム Scarab と連携して，EM を計算する．以降の節では，パスウェイを記述するための Scarabio DSL 用のクラス，EM ソルバー用のクラスについて説明する．

4.2　Scarabio DSL 用のクラス

パスウェイにおける化合物や化学反応式を Scala 中で自然に書けるように図 4 に示すようにケースクラスを定義する．なお図ではクラスの宣言のみを抽出して記述しているので注意されたい．1 行目から 4 行目は化学反応式の項に関するクラスの定義であり，6 行目から 8 行目は化学反応式に関するクラスの定義である．これらの定義と併せて Scala の機能である暗黙変換，演算子の多重定義，ピリオドの省略などを用いることで，以下の BNF 記法に従った化学反応式の記述が可能になっている．

```
Reaction ::= CTerm ==> CTerm | CTerm <=> CTerm
CTerm ::= Compound | Compound * Int |
    CTerm + CTerm | CSum(Compound * Int, ...)
```

$$
\begin{array}{cc}
 & \begin{array}{cccccccccc} R1 & R2 & R3 & R4 & R5 & R6 & R7 & R8 & R9 & R10 \end{array} \\
\mathbf{N} = & \begin{pmatrix}
1 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
\begin{array}{l} \to A \\ \to B \\ \to C \\ \to D \\ \to E \\ \to P \end{array}
\end{array}
$$

$Rev = \{R7\}$; $Irrev = \{R1, R2, R3, R4, R5, R6, R8, R9, R10\}$

図 5　代謝パスウェイ (図 1) の化学量論行列

4.3　Scarabio によるパスウェイの EM 解析

代謝パスウェイを解析するための代表的な制約モデルに Elementary Mode (EM) [29] [28] [27] がある．EM を用いた代謝パスウェイの解析では化学量論行列と呼ばれる表現を使う．図 5 に図 1 の代謝パスウェイを表した化学量論行列を示す．既存ツール [36] [5] [16] [18] では，Double Description (DD) 法 [9] を用いた EM アルゴリズムを実装しているが，DD 法では，一つだけ解を出力することが出来ず，常に全解列挙を行う．そのためパスウェイが大きくなると計算に時間がかかったり大量のメモリが必要になることがあった．

本論文では，EM を制約充足問題の形式に定式化・符号化し SAT ソルバー用いた EM の計算方法を提案する．提案方法は SAT 型制約プログラミングシステム Scarab を用いて Scarabio 上に実装する．

SAT ソルバーを用いた EM の計算方法を説明する．いまパスウェイの化学反応が $c \times q$ の化学量論行列 $\boldsymbol{N}$ で与えられたとする．ここで，c は化合物の数，q は反応の数である．$\boldsymbol{N}$ の要素は n_{ij} $(1 \leq i \leq m, 1 \leq j \leq q)$ で表す．また x_j $(1 \leq j \leq q)$ を反応の度合いを表す整数変数とする．パスウェイの定常状態における制約式は以下になる．

$$\bigwedge_{i=1}^{m} \sum_{j=1}^{q} n_{ij} x_j = 0 \tag{1}$$

$$x_j \geq 0 \quad (\mathrm{Rj} \in Irrev) \tag{2}$$

上式は定常状態においては細胞内の化合物の生成と消費が均衡していることを表している．EM 解析では，このような制約を満たす極小な反応集合を求める必要がある．そのため反応の度合いを表す変数 x_i に対して，反応の活性・不活性を表す命題変数 b_i を導入する．二つの変数を紐付けるチャネリング制約は以下になる．

$$b_i \to x_i \neq 0 \tag{3}$$

$$\neg b_i \to x_i = 0 \tag{4}$$

最終的に上式 (1), (2), (3), (4) を SAT 符号化し，命

```
import jp.kobe_u.scarab.csp._
import jp.kobe_u.scarab.solver._

case class EmSolver(p: Pathway) {
  val csp = CSP()
  val satSolver = new Sat4j
  val encoder = new LogEncoder(csp,satSolver)
  val solver = new Solver(csp,satSolver,encoder)
  val d = 5

  var bmap = Map.empty[String, Bool]

  def define {
    for (r <- p.Reactions)
      csp.int('x(r.name), if(r.isRev)-d else 0,d)

    for (r <- p.Reactions) {
      val b = Bool(r.name)
      bmap += r.name -> b
      csp.bool(b)
      csp.add(b ==> ('x(r.name) !== 0))
      csp.add(Not(b) ==> ('x(r.name) === 0))
    }

    for (c <- p.Compounds) {
      val eq = p.rsInvolving(c).map(
                r => 'x(r.name) * r.coef(c))
      csp.add(Sum(eq.toSeq) === 0)
    }
    csp.add(Or(bmap.keys.map(bmap(_))))
  }

  def findEM = {
    define
    val bset = bmap.values.toSeq
    while (solver.findMinimal(bset.toSeq)) {
      csp.add(Or(bset.filter(i =>
      solver.solution.boolMap(i)).map(Not(_))))
    }
  }

}
```

図 6　SAT ソルバーを用いた EM の解法を実装した EmSolver クラス

題変数集合 $\{b_i \mid (1 \leq i \leq q)\}$ に関する極小モデルを計算することで SAT ソルバーを用いて EM を計算することができる．SAT 符号化には順序符号化 [8] [3] [1] [11] [31] [34] や対数符号化 [14] [10] が利用可能であり，極小モデルの計算には文献 [19] の方法を利用できる．SAT 型制約プログラミングシステム Scarab は，これらの機能をデフォルトで提供している．図 6 に Scarabio における，EM ソルバーの実装を示す．1 行目から 8 行目までで，Scarab の関連クラスのインポートと CSP ソルバーの定義を行っている．14，15 行目では反応の度合いを表す整数変数 x_i を定義している．17 行目から 23 行目は式 (3), (4) を表している．25 行目から 29 行目は式 (1) を表している．33 行目から 39 行目で b_i の集合に関する極小モデルを Scarab を用いて求め，EM を計算している．

このように EM を解析する Scarabio のプログラム自体も 40 行ほどで簡潔に記述できる．これにより，利用者が直接 EmSolver を編集することも可能であり，文献 [2] [13] [37] [32] [21] で研究されているような遺伝子欠損の影響を考慮することも可能である．

4.4 EM 求解性能の評価

Scarabio の EM 解析ツールとしての基本性能を評価するために，制約モデル解析のベンチマークとしてよく使われる大腸菌の代謝パスウェイ *E. coli core* [25] を用いて簡単な計算機実験を行った．この代謝パスウェイは 72 の化合物と 94 の反応で構成される．図 7 にこの代謝パスウェイを示す．

文献 [15] では，6 コアの Intel Xeon CPU を 2 個とメモリ 190GB (153GB を実際に使用) を搭載した計算機を使って，上記のパスウェイの EM を現在最も高速といわれる efmtool を用いて計算している．結果として，全ての EM (226.6×10^6 個) を求めるのに 34 時間かかることが報告されている．efmtool ツールは他のツールと同様に Double Description 法を用いており全ての EM を列挙するまでは 1 つも解を返すことができないことに注意されたい．現実的には 226.6×10^6 個もの全ての EM を解析するのは不可能であり，高速に興味がある EM が見つかるほうが望ましい．

Scarabio をノート PC （CPU は Intel Core i7 3.1GHz，メモリ 8GB）上で実行したところ，上記パスウェイの 1 つの EM は 1 秒未満で計算することができた．また 100 個の EM を求めるのに要した CPU 時間は 74 秒であった．Scarabio では，興味のある反応が含まれる EM を計算したり，EM に含まれる反応の数を制限することも可能である．このことから SAT ソルバーを用いた EM の計算方法，またそれを実装したツール Scarabio は代謝パスウェイ解析における有効な手法になると考えられる．

5 おわりに

本論文では，生物の代謝パスウェイ解析用の DSL Scarabio について述べた．Scarabio は Scala の内部 DSL として実装されており，近年急速に発展している SAT ソルバーを用いた SAT 型制約プログラミングシステム Scarab [33] を制約モデルの記述および計算エンジンとして採用することで，簡潔なパスウェイ・制約モデルの記述と高速な EM の計算が可能である．今後の課題は，極小非充足コアなどの高度な SAT 技術をパスウェイ解析に応用することである．Scarabio は以下の URL からダウンロード可能である．`http://kix.istc.kobe-u.ac.jp/~soh/scarab/scarabio/`

参 考 文 献

[1] Ansótegui, C. and Manyà, F.: Mapping Problems with Finite-Domain Variables into Problems with Boolean Variables, *SAT 2004 - The Seventh International Conference on Theory and Applications of Satisfiability Testing, 10-13 May 2004, Vancouver, BC, Canada, Online Proceedings*, 2004.

[2] Baba, T., Ara, T., Hasegawa, M., Takai, Y., Okumura, Y., Baba, M., Datsenko, K. A., Tomita, M., Wanner, B. L., and Mori, H.: Construction of *Escherichia coli* K-12 in-frame, single-gene knockout mutants: the Keio collection, *Molecular Systems Biology*, Vol. 2, No. 2006.0008(2006).

[3] Bailleux, O. and Boufkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints, *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP 2003), LNCS 2833*, 2003, pp. 108–122.

[4] 番原睦則, 田村直之: SAT によるシステム検証, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 122–129.

[5] Becker, S. A., Feist, A. M., Mo, M. L., Hannum, G., Palsson, B. Ø., and Herrgard, M. J.:

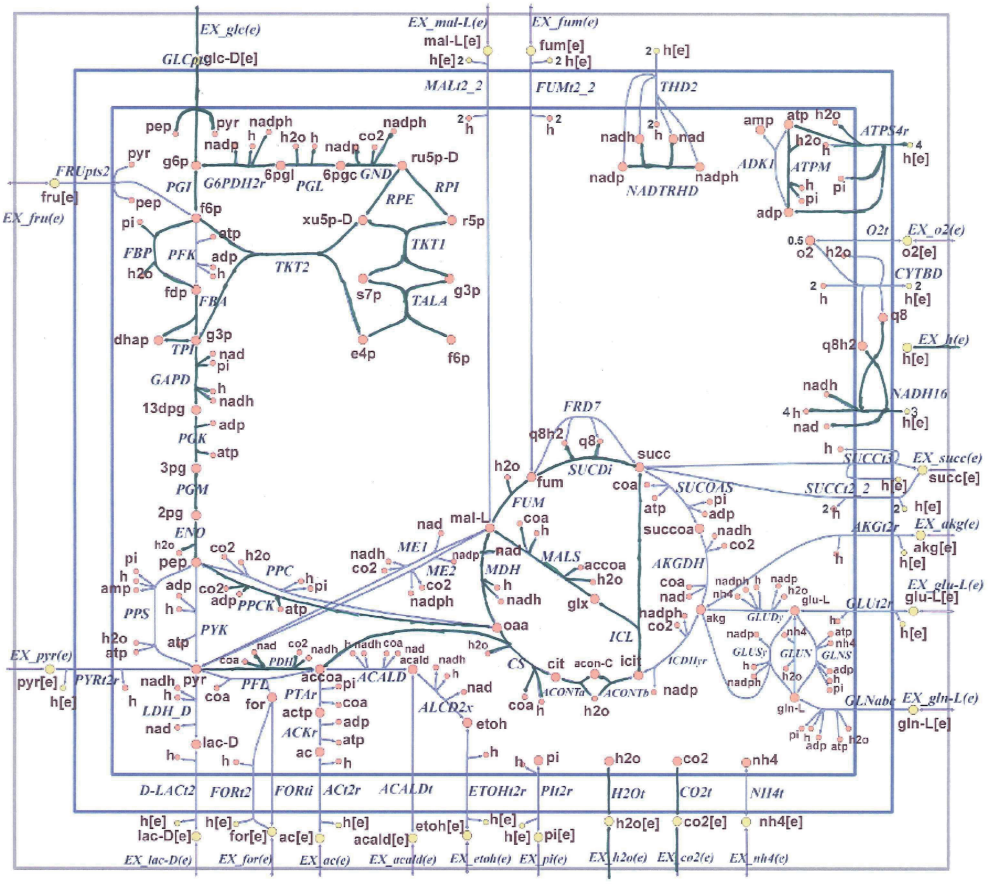

図 7　計算機実験のベンチマークとして用いた *E. coli core* の代謝パスウェイ

`http://2014.igem.org/Team:TU_Delft-Leiden/Modeling/EET/FBA#EETCoreModel` **より引用**

Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox, *Nature Protocols*, Vol. 2, No. 3(2007), pp. 727–738.

[6] Benner, P., Findeisen, R., Flockerzi, D., Reichl, U., and Sundmache, K.: *Large-Scale Networks in Engineering and Life Sciences*, Springer, 2014.

[7] Biere, A., Heule, M., van Maaren, H., and Walsh, T.(eds.): *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications (FAIA), Vol. 185, IOS Press, 2009.

[8] Crawford, J. M. and Baker, A. B.: Experimental Results on the Application of Satisfiability Algorithms to Scheduling Problems, *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 1994)*, 1994, pp. 1092–1097.

[9] Fukuda, K. and Prodon, A.: Double description method revisited, *Combinatorics and Computer Science*, Vol. 1120(1996), pp. 91–111.

[10] Gelder, A. V.: Another Look at Graph Coloring via Propositional Satisfiability, *Discrete Applied Mathematics*, Vol. 156, No. 2(2008), pp. 230–243.

[11] Gent, I. P. and Nightingale, P.: A New Encoding of Alldifferent into SAT, *Proceedings of the 3rd International Workshop on Modelling and Reformulating Constraint Satisfaction Problems*, 2004.

[12] 井上克巳, 田村直之: SAT ソルバーの基礎, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 57–67.

[13] Ishii, N., Nakahigashi, K., Baba, T., Robert, M., Soga, T., Kanai, A., Hirasawa, T., Naba, M., Hirai, K., Hoque, A., Ho, P. Y., Kakazu, Y., Sugawara, K., Igarashi, S., Harada, S., Masuda, T., Sugiyama, N., Togashi, T., Hasegawa, M., Takai, Y., Yugi, K., Arakawa, K., Iwata, N., Toya, Y., Nakayama, Y., Nishioka, T., Shimizu, K., Mori, H., and Tomita, M.: Multiple High-Throughput Analyses Monitor the Response of *E. coli* to Perturbations, *Science*, Vol. 316, No. 5824(2007), pp. 593–597.

[14] Iwama, K. and Miyazaki, S.: SAT-Variable Complexity of Hard Combinatorial Problems, *Proceedings of the IFIP 13th World Computer*

Congress, 1994, pp. 253–258.

[15] Jungreuthmayer, C., Ruckerbauer, D. E., and Zanghellini, J.: Utilizing gene regulatory information to speed up the calculation of elementary flux modesx, *ArXiv e-prints*, (2012).

[16] Kamp, A. v. and Schuster, S.: Metatool 5.0: fast and flexible elementary modes analysis, *Bioinformatics*, Vol. 22, No. 15(2006), pp. 1930–1931.

[17] Karr, J. R., Sanghvi, J. C., and Derek N. Macklin, e. a.: A whole-cell computational model predicts phenotype from genotype, *Cell*, Vol. 150, No. 2(2012), pp. 389–401.

[18] Klamt, S., Saez-Rodriguez, J., and Gilles, E. D.: Structural and functional analysis of cellular networks with CellNetAnalyzer, *BMC Systems Biology*, Vol. 1, No. 2(2007).

[19] Koshimura, M., Nabeshima, H., Fujita, H., and Hasegawa, R.: Minimal model generation with respect to an atom set, *Proceedings of the the 7th International Workshop on First-Order Theorem Proving (FTP '09)*, 2009, pp. 49–59.

[20] Mernik, M., Heering, J., and Sloane, A. M.: When and How to Develop Domain-Specific Languages, *ACM Computing Surveys*, Vol. 37, No. 4(2005), pp. 316–344.

[21] Mori, H., Baba, T., Yokoyama, K., Takeuchi, R., Nomura, W., Makishi, K., Otsuka, Y., Dose, H., and Wanner, B.: Identification of Essential Genes and Synthetic Lethal Gene Combinations in Escherichia coli K-12, *Gene Essentiality*, Lu, L. J.(ed.), Methods in Molecular Biology, Vol. 1279, Springer New York, 2015, pp. 45–65.

[22] 鍋島英知: SAT によるプランニングとスケジューリング, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 114–121.

[23] 鍋島英知, 宋剛秀: 高速 SAT ソルバーの原理, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 68–76.

[24] Odersky, M., Spoon, L., and Venners, B.: *Programming in Scala*, Artima, Inc., second edition, 2010.

[25] Orth, J., Fleming, R., and Palsson, B.: Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide, *EcoSal Plus*, (2010).

[26] Schmid, A., Dordick, J. S., Hauer, B., Kiener, A., Wubbolts, M., and Witholt, B.: Industrial biocatalysis today and tomorrow, *Nature*, Vol. 409(2001), pp. 258–268.

[27] Schuster, S.: A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks, *Nature Biotechnology*, Vol. 18, No. 3(2000), pp. 326–332.

[28] Schuster, S., Dandekar, T., and Fell, D. A.: Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering, *Trends in Biotechnology*, Vol. 17, No. 2(1999), pp. 53–60.

[29] Schuster, S. and Hillgetag, C.: On Elementary Flux Modes in Biochemical Reaction Systems at Steady State, *Journal of Biological Systems*, Vol. 02, No. 02(1994), pp. 165–182.

[30] Schwarz, R., Musch, P., von Kamp, A., Engels, B., Schirmer, H., Schuster, S., and Dandekar, T.: YANA a software tool for analyzing flux modes, gene-expression and enzyme activities, *BMC Bioinformatics*, Vol. 6, No. 135(2005).

[31] Sinz, C.: Towards an Optimal CNF Encoding of Boolean Cardinality Constraints, *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP 2005), LNCS 3709*, 2005, pp. 827–831.

[32] Soh, T., Inoue, K., Baba, T., Takada, T., and Shiroishi, T.: Evaluation of the Prediction of Gene Knockout Effects by Minimal Pathway Enumeration, *International Journal On Advances in Life Sciences*, Vol. 4, No. 3-4(2012), pp. 154–165.

[33] Soh, T., Tamura, N., and Banbara, M.: Scarab: A Rapid Prototyping Tool for SAT-based Constraint Programming Systems, *Proceedings of the 16th International Conference on Theory and Applications of Satisfiability Testing (SAT 2013), LNCS 7962*, July 2013, pp. 429–436.

[34] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, *Constraints*, Vol. 14, No. 2(2009), pp. 254–272.

[35] 田村直之, 丹生智也, 番原睦則: 制約最適化問題と SAT 符号化, 人工知能学会誌, Vol. 25, No. 1(2010), pp. 77–85.

[36] Terzer, M. and Stelling, J.: Accelerating the Computation of Elementary Modes Using Pattern Trees, *Algorithms in Bioinformatics*, Lecture Notes in Computer Science, Vol. 4175, Springer Berlin Heidelberg, 2006, pp. 333–343.

[37] 宋剛秀, 井上克巳: モデル生成を用いた代謝ネットワークにおける極小活性パスウェイの列挙, 人工知能学会論文誌, Vol. 27, No. 3(2012), pp. 204–212.