

# SMT Solver を利用した Web アプリケーション用テストデータの生成

藤原 翔一朗<sup>†</sup> 宗像 一樹<sup>†</sup> 片山 朝子<sup>†</sup> 前田 芳晴<sup>†</sup> 大木 憲二<sup>†</sup> 上原 忠弘<sup>†</sup> 山本 里枝子<sup>†</sup>

富士通研究所 ソフトウェアイノベーション研究部<sup>†</sup>

## 1 はじめに

近年, Web 技術を用いた業務システムの開発が盛んになっており, 十分な開発リソースが確保できない開発案件も増えている. 結果として, 開発した Web アプリケーション(Web アプリ)のテストが不十分なまま出荷され, 実運用時に障害が頻発するといった事態も増えている. そのため, Web アプリの効率的, 効果的なテストに対する期待が高まっている.

そこで, 本研究では, Web アプリの結合テストに関して, テストのボトルネックの一つであるテストデータ生成の自動化ツールを開発した. 本ツールでは, テストデータ生成時に考慮すべき各種制約を述語論理式に変換する. そして, SMT Solver を用いて変換した述語論理式の充足値を得ることでデータを生成する.

## 2 テストデータ生成とその課題

### 2.1 Web アプリの仕様

Web アプリの結合テストで検査される仕様は, 画面遷移等のイベント発生前後における画面と DB の状態変化という形で表現される.

図 1 は, 商品の在庫管理を行う Web アプリの一部であり, 以下のような仕様を持つ:

「確認ボタン」押下時(イベント), 「画面中の倉庫と商品に対応するレコードが在庫状況テーブル(T)に存在し, かつ当該レコードの在庫数が画面中の発注数以上である」ならば(事前条件), 「発注確認画面へ遷移し, 結果コードに"CN001"が表示される」(事後条件).

### 2.2 テストデータ生成とその課題

例のような仕様のテストを実施するためには, 事前条件を満たす画面入力(図 1 の「発注入力画面」)と DB データ(図 2)がテストデータとして必要である. これらテストデータは, 「在庫状況 T の商品名は商品マスタ T の商品名の外部キー」という外部キー制約(FK)や主キー制約といったデータ制約も満たさなければならない.

このように, テストで注目する仕様以外にも様々な制約を考慮する必要があるため, テストデータの作成は未だ人手に負うところが大きい.

Test Data Generation for Web Applications using an SMT Solver

<sup>†</sup>Shoichiro FUJIWARA, Kazuki MUNAKATA, Asako KATAYAMA, Yoshiharu MAEDA, Kenji OKI, Tadahiro UEHARA, Rieko YAMAMOTO

(<sup>†</sup>)Software Innovation Laboratory, Fujitsu Laboratories

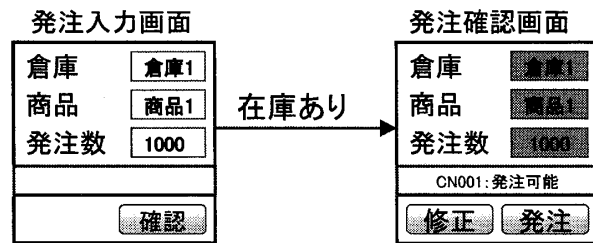


図 1 : Web アプリの一例

倉庫マスタ T		商品マスタ T		在庫状況 T		
倉庫名	地域	商品名	単価	倉庫名	商品名	在庫数
倉庫1	地域1	商品1	100	倉庫1	商品1	2000
倉庫2	地域2	商品2	100	倉庫1	商品2	1500
				倉庫2	商品2	900

図 2 : テスト用 DB データの一例

それゆえ作成に工数を要し, 誤ったデータを作ってしまうことも頻発するという課題がある.

## 3 SMT Solver とデータ生成

SMT (Satisfiability Modulo Theories) とは, 述語論理式の充足可能性問題である. 研究の進歩により近年の SMT Solver は従来と比較してより大きな規模の問題を解くことができるようになっており, 様々な技術分野で利用されている.

SMT Solver は入力された論理式の充足値を出力する. ゆえに, 考慮すべき制約全てを SMT Solver で解けるクラスの論理式に変換できれば, 変換した論理式を SMT Solver で解くことで, 従来人手で作成せざるを得なかったテストデータを自動生成できる.

## 4 SMT Solver によるデータ生成

テストデータ生成の課題を解決するため, 本章に述べるテストデータ生成ツール TDG (Test Data Generator) を開発した.

### 4.1 全体像

図 3 に TDG の全体像を示す. TDG の入力, テストする仕様の事前条件およびデータ制約となる. TDG は, (1) SMT Solver が扱えるようテーブルサイズ (テーブルが持つレコード数) の上限を与え, その上限の下で各制約を等価な論理式に変換する. さらに, (2) 変換した論理式を SMT Solver で解き, 各変数の制約充足値を得る. そして, (3) 充足値を文字列等に変換し, テストデータを得る. なお, TDG では SMT Solver として Yices[1] を用いた.

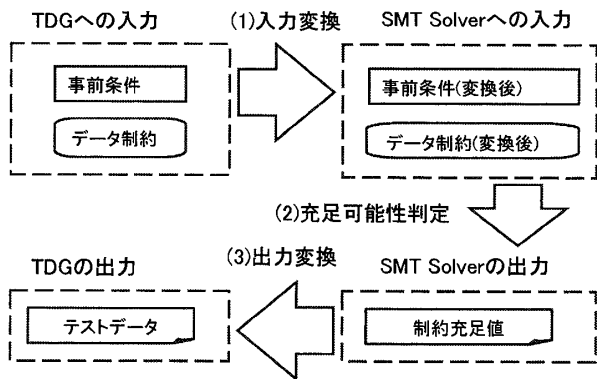


図 3 : TDG 全体像

#### 4.2 文字列の取扱い

業務システムでは多くの文字列データが処理されるため、必然的に TDG は文字列関係の制約を扱う必要がある。ところで、Web アプリの事前条件に出現する制約の殆どは、入力データに対応するレコードの存在確認やマスタテーブルとの整合性確認である。すなわち、文字列長や部分文字列の比較といった制約は殆ど出現しない。

そのため、文字列関係の制約に関しては、単純な等価関係や大小関係程度の対応で十分実用可能と判断した。これらの関係に限れば、各文字列を代表する数値型変数を文字列変数の代わりに用いて制約を構成し、充足値の数値的な大小関係を維持しながら文字列へ置換するというアプローチが可能になり、より大規模のデータが生成できる。

#### 4.3 DB テーブルの限量子除去

図 4 は、2 節で述べた事前条件および外部キー制約 FK の論理式表現である。このように、TDG の入力である制約の多くは DB テーブルに関する限量子を含む。限量されたテーブルサイズの上限が定まらない場合、SMT Solver が完全性を失い、充足可能性を判定できないケースが発生する。しかし、通常テーブルサイズの上限は DBMS 等の性能限界に起因するもの以外は規定されない。そのため、TDG ではテーブルサイズの上限を決めて限量子除去を行う。図 5 は、在庫状況 T、商品マスタ T のテーブルサイズ上限をそれぞれ 3、2 として図 4 の論理式の限量子除去を行った結果である。

### 5 試行結果

実開発資産に対して TDG を試行した。ある Web アプリの一機能を構成する 23 の仕様について、各仕様に対応するテストデータを数分で生成できた。なお、各仕様の事前条件は、項目間の単純な等価関係および特定の条件を満たすレコードの数に関する条件で記述できた。

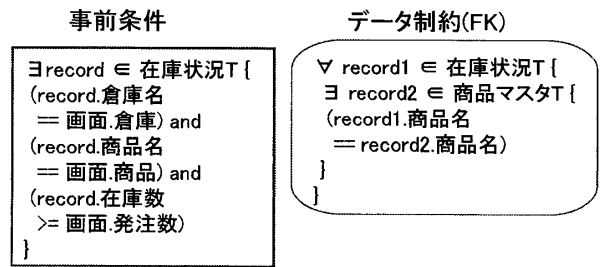


図 4 : TDG への入力

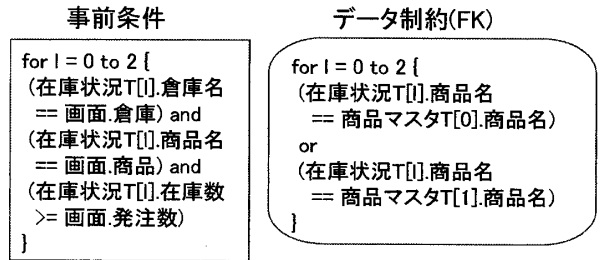


図 5 : SMT Solver への入力

### 6 おわりに

本研究では、SMT Solver を利用して、Web アプリケーションの仕様およびデータ制約から結合テスト用のデータを生成するツールを開発した。さらに、開発したツールを実プロジェクト資産に対して試行し、有効性を確認した。

SMT (SAT) Solver を利用したデータ生成の技術としては、Pex[2]や TestEra[3]がある。Pex は、ソースコードの分岐条件等を重ね合わせて構成した制約からソースコードの各パスを実行するための入力データを生成する。TestEra は、Java ソースと論理式で記述したメソッドの事前条件からテストデータを生成し、事後条件を用いてテスト結果を判定する。これらのツールが主にソースコードレベルでのテストデータを生成する一方で、本ツールは業務的な観点による仕様のテストデータを生成することが特徴である。

今後は、対応可能な制約を広げる等、より複雑な仕様のテストデータや期待値も生成できるように技術開発を行い、開発現場への導入を行う予定である。

#### 参考文献

- [1] <http://yices.csl.sri.com/>
- [2] <http://research.microsoft.com/en-us/projects/pex/>
- [3] Sarfraz Khurshid and Darko Marinov. TestEra: Specification-based Testing of Java Programs Using SAT, Automated Software Engineering Journal, Vol.11, No.4, pp.403-434 (2004).