トップエスイー修了制作

多種サービス間でBASEを保証するための更新トランザクションの設計ルールと検証手法

NTTコムウェア株式会社　高鶴哲也　takatsuru.tetsuya@nttcom.co.jp

開発における問題点

多くのサービスが連携するサービス群全体でBASEを保証するには，各サービスの全ての更新トランザクションがリトライ可能な設計になっている必要がある．しかし，それをレビューやテストで網羅するのは困難である．

手法・ツールの適用による解決

業務系に多いRDBを使うシステムを対象とした検証モデルを定理証明支援系Coqの上に構築し，SQL文と設計から検証コードに機械的に写し取ったものを半自動的に証明できるようにした．このモデル化により，現実的な工数で網羅性が担保できることをケーススタディで確認した．

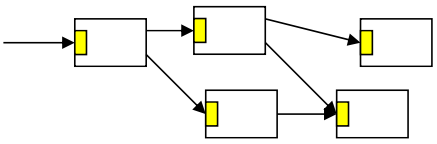
サービス間通信の設計

【ACID】
整合性 強
障害に弱い

→

【BASE】
整合性 弱（結果整合性）
障害に強い
（回復後にリトライ）

リトライ時に不整合を発生させない更新ロジックの設計が必要．



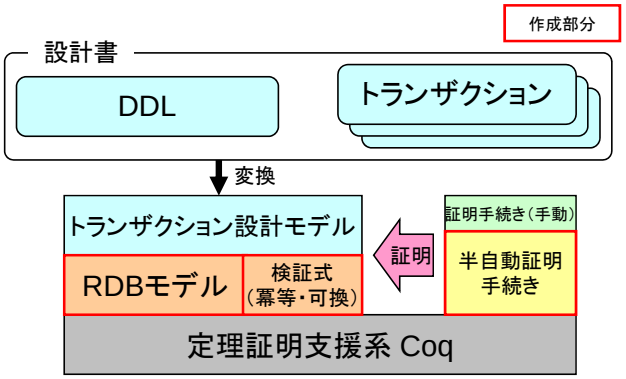
冪等: f(f(x))=f(x)
可換: f(g(x))=g(f(x))
⇒リトライ可能
f, g : 更新トランザクション
x : 任意のシステム状態

設計の困難性

- レビューやテストで**全条件を網羅**するのは困難．
- 検証課題の**組み合わせ爆発**が起こる．
（更新操作数nに対して計$2n+{}_nC_2$個の検証課題）

設計検証ツール

RDBを使うシステムをターゲットとして，更新トランザクションの設計をCoqで検証するため，RDBモデルと検証式を定義．
Coqのタクティク（証明手続き記述言語）により，証明の自動化を試みた．



ツールの効果

ツール作成のポイント

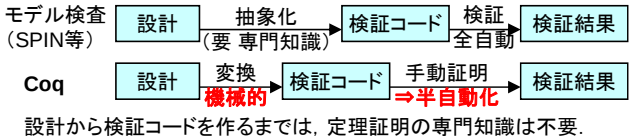
- SQLから機械的に変換できるよう，RDBモデルを作成
⇒**設計者の負担軽減**
- 汎用的な自動化が難しい定理証明を，ドメインを限定することで半自動化 ⇒**検証者の負担軽減**

ケーススタディによる効果測定

- 実際によく使われるロジックを一部抜き出しで検証
通話履歴サービス，トラブルチケットシステム，銀行振り込みサービス
- 自動証明 ⇒ 検証課題1個あたり**数十秒**（×$2n+{}_nC_2$）
- 手動証明（一部）⇒ 3～7ステップ程度で証明完了．
- 不成立時 ⇒ 反例分析は，前提を目視により確認し，原因を設計者にフィードバック．

考察

ツール比較



設計から検証コードを作るまでは，定理証明の専門知識は不要．
検証は定理証明の専門家が実施（業務知識は不要）．
設計者⇔検証者の分業が可能．　⇒**導入・展開が比較的容易**

今後の課題

- 複雑なトランザクションロジック変換時のミス低減
  - ➢DSLを定義し，プログラムと検証コードの両方を**自動的に生成できるツール**を作ることで解決可能．
- 検証時の不成立を減らすことによる検証工数の低減
  - ➢**証明済みデザインパターン**を用意し，検証前に可能な限り正しく設計することで解決可能．