

双方向変換における一貫性保証の為のレンズ則に関する研究

背景

双方向変換は、データベースやプログラム変換、システム開発(MDD)において重要な役割を果たしている。双方向変換は、レンズと呼ばれる2つの関数(get,put)の組で表すことができ、この二つの関数の間で一貫性が成立することで実現できる。これらの関数が一貫性を持つ時にはある規則が成立している。本研究では、この規則に関する研究を行なっている。

目的

これまでに複数のレンズ則が見つかっており、これらのレンズ則における関係が明らかになってきた。本研究ではその中のレンズ則がデータベースにおけるデータの追加や削除などによる更新においてどのような働きをするかを明確にする

ビュー更新問題 (View-update problem)

双方向変換はデータベースにおけるビュー更新問題に
ビュー更新問題とは、Viewテーブル上のデータに変更を加えた時に
Sourceテーブル上で適切なデータの更新を行うことである。
図2において次の関係がある:
Sourceテーブル={Staffテーブル,Projectsテーブル} \hookrightarrow Viewテーブル
例1. Viewテーブルに一人新しく追加した時にStaffテーブルが更新される
が、Salaryはどうなるか？

```
SELECT Name, Room, Role
FROM Staff, Projects
WHERE Name = Person
AND Code = "Plum"
```

Staff			Projects			View		
Name	Room	Salary	Code	Person	Role	Name	Room	Role
Sam	314	£30k	Plum	Sam	Lead	Sam	314	Lead
Pat	159	£25k	Plum	Pat	Test	Pat	159	Test
Max	265	£25k	Pear	Pat	Lead			

(a) (b) (c)

図1. SQLでデータ検索を行うクエリ
(=Viewテーブルを作成するコード)

図2. クエリによって得られる3つのテーブル

ビュー更新問題をレンズで考える！

- get (view) はsourceテーブルからviewテーブルを取得する関数
 - put (update) はviewテーブルに加えられた変更を元にsourceテーブルを更新する関数
- 例1についてgetとputを用いて考えてみる。
まず、図2(c)Viewテーブルを取得する為に、sは図1のクエリとして、 $get(s) = v$
次に作成したViewテーブルに一人追加しsを更新する。
 $put(s, v) = s'$

レンズ (lens)

レンズは、source Sからview Vにおける二つの関数 (getとputの組) で表すことができる

- lens(get, put): $S \Leftrightarrow V$
- get : $S \rightarrow V$ (sourceからviewを取得する)
- put : $S \times V \rightarrow S$ (viewに加えられた変更に対してsourceを更新する)

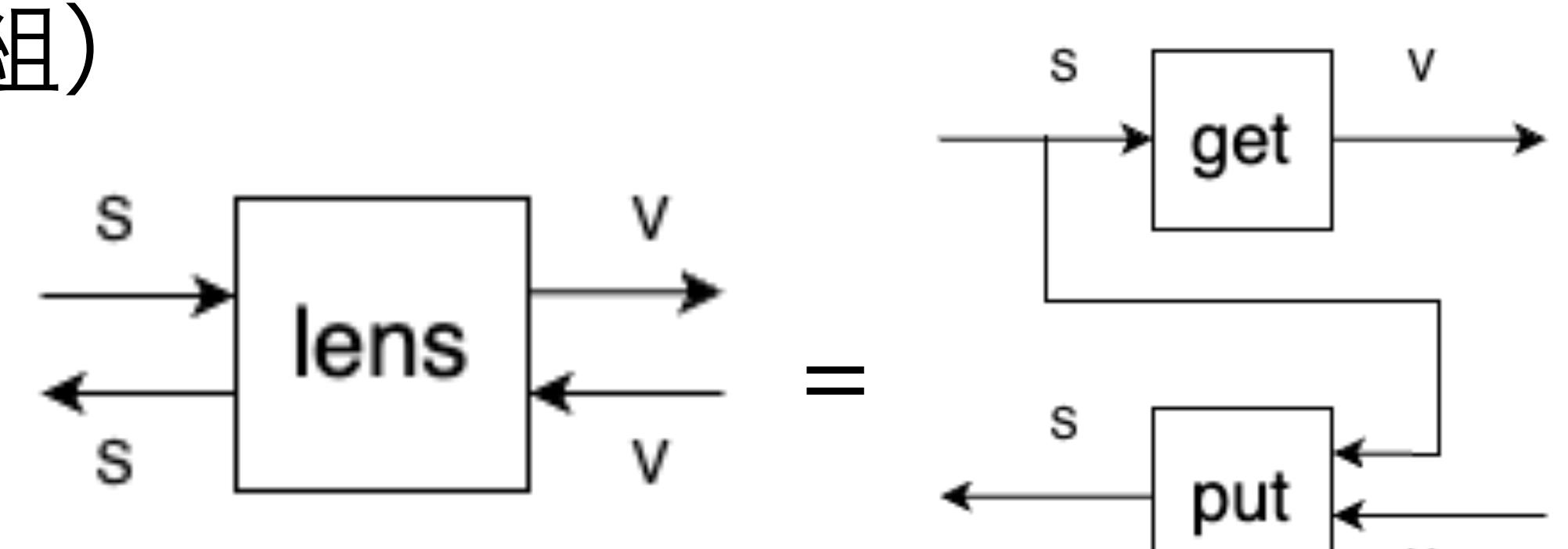


図3. レンズの構成図

一貫性のあるレンズ則(GetPut,PutGet)

$\forall s \in S, v \in V$ に対して、以下を満たす時lensはwell-behavedである

$$(GetPut) \quad put(s, get(s)) = s$$

$$(PutGet) \quad get(put(s, v)) = v$$

その他の有名なレンズ則

$$(PutPut) \quad put(put(s, v), v') = put(s, v')$$

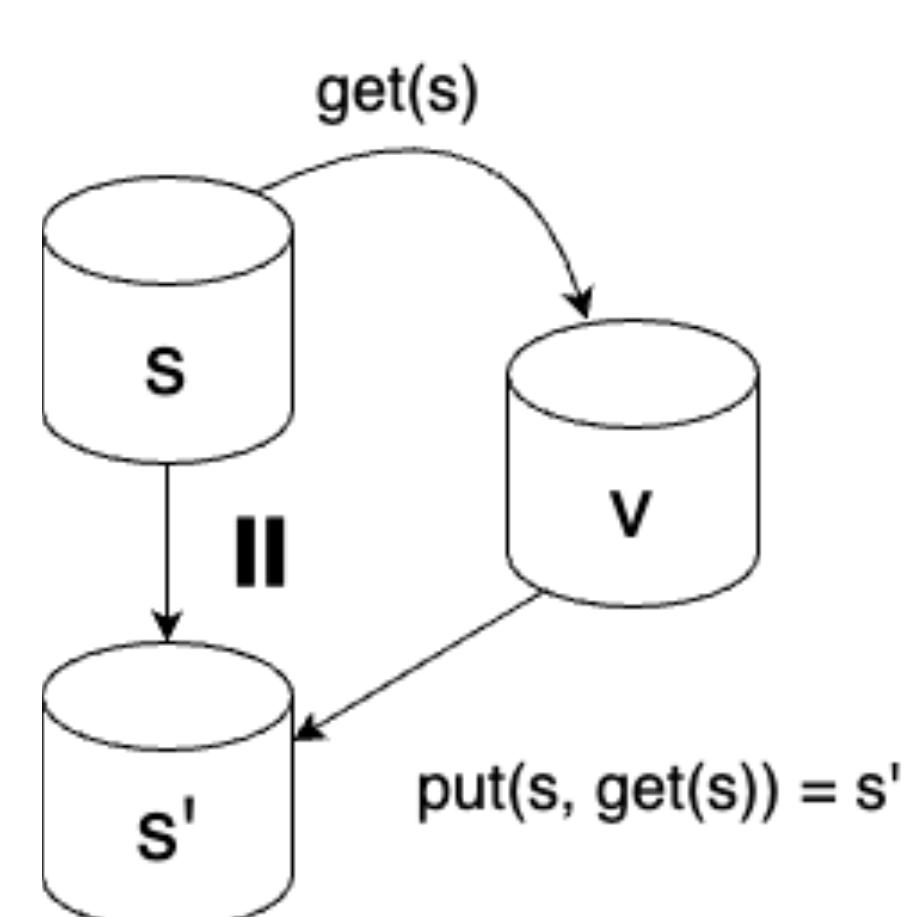


図4. GetPutのイメージ

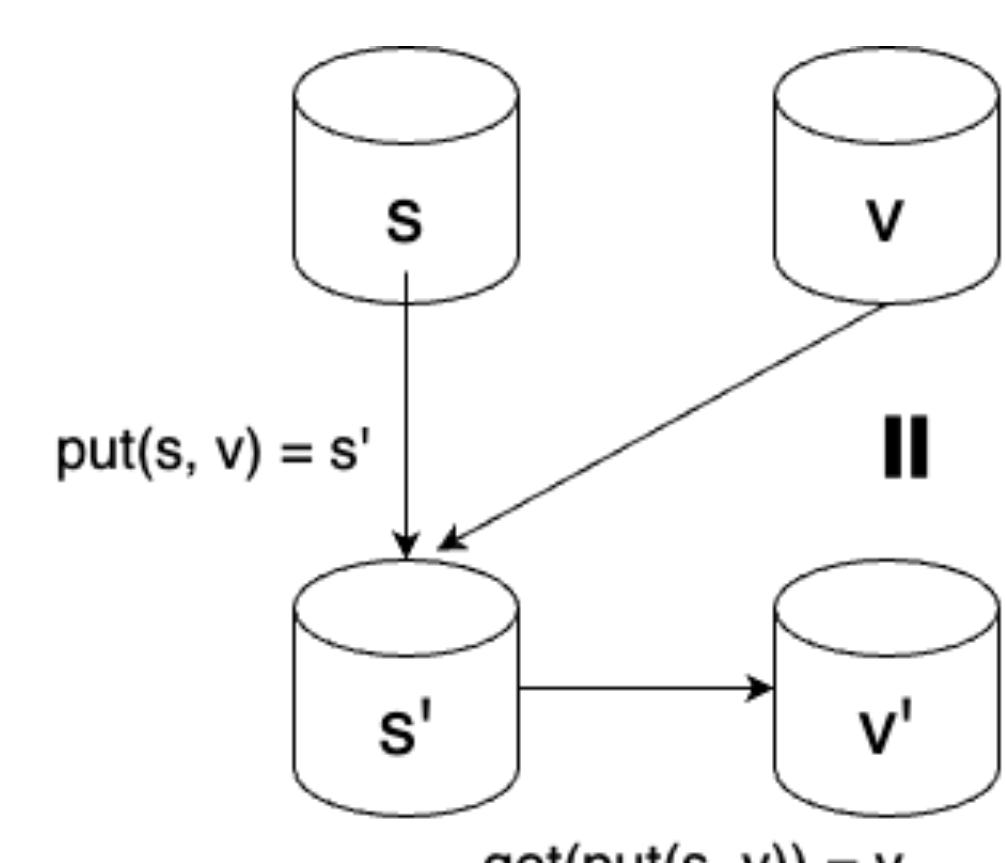


図5. PutGetのイメージ