

双向変換における 16個のレンズ則の含意関係

長田和樹 C1TB2172

東北大学

工学部電気情報物理工学科

電気コース

双方向変換

双方向変換：二つ以上の異なるデータ間の同期

国	首都	人口(万人)
日本	東京	12,435
韓国	ソウル	5,171
中国	北京	141,932

s

国	首都	人口(万人)
日本	東京	12,379
韓国	ソウル	5,171
中国	北京	141,932

s'

①get



国	人口(万人)
日本	12,435
韓国	5,171
中国	141,932

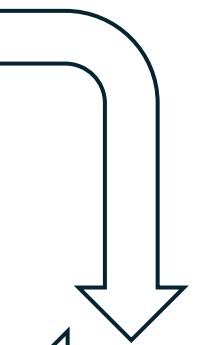
v

②ビューを
更新

国	人口(万人)
日本	12,379
韓国	5,171
中国	141,932

v'

③put



双方向変換

①ソース s から国、人口の情報を取り出し、再配置する(v)

国	首都	人口(万人)
日本	東京	12,435
韓国	ソウル	5,171
中国	北京	141,932

s

①get



国	人口(万人)
日本	12,435
韓国	5,171
中国	141,932

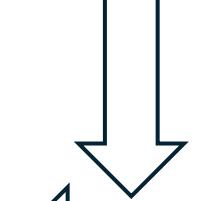
v

②ビューを
更新

国	首都	人口(万人)
日本	東京	12,379
韓国	ソウル	5,171
中国	北京	141,932

s'

③put



国	人口(万人)
日本	12,379
韓国	5,171
中国	141,932

v'

双方向変換

②ビュー v の一部を変更(v')

国	首都	人口(万人)
日本	東京	12,435
韓国	ソウル	5,171
中国	北京	141,932

s

国	首都	人口(万人)
日本	東京	12,379
韓国	ソウル	5,171
中国	北京	141,932

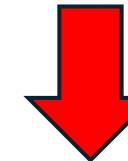
s'

①get



国	人口(万人)
日本	12,435
韓国	5,171
中国	141,932

v



国	人口(万人)
日本	12,379
韓国	5,171
中国	141,932

v'

②ビューを
更新

双方向変換

③ v' の変更の内容をソース s をもとに更新(s')

国	首都	人口(万人)
日本	東京	12,435
韓国	ソウル	5,171
中国	北京	141,932

s

①get

③put

国	人口(万人)
日本	12,435
韓国	5,171
中国	141,932

v

②ビューを
更新

国	首都	人口(万人)
日本	東京	12,379
韓国	ソウル	5,171
中国	北京	141,932

s'

国	人口(万人)
日本	12,379
韓国	5,171
中国	141,932

v'

レンズとは [Foster et al `07]

レンズ：以下のgetとputの組

- $\text{get} : S \rightarrow V$: 与えられたソース $s \in S$ から ビュー $v \in V$ を生成
- $\text{put} : S \times V \rightarrow S$: ビュー $v \in V$ を用いて ソース $s \in S$ を更新

S と V はデータベースだけではなく、自然数 \mathbb{N} や有理数 \mathbb{Q} でも可

レンズ(get,put)は全てが全域関数というわけではなく、部分関数の場合の方が多い

Core Lens Laws

Core Lens Laws:

レンズの一貫性を構成する以下の4つのレンズ則

$$(\text{StrongGetPut}): \forall s, s' \in S, \text{put}(s, \text{get}(s')) = s'$$

$$(\text{GetPut}): \forall s \in S, \text{put}(s, \text{get}(s)) = s$$

$$(\text{PutGet}): \forall s \in S, \forall v \in V, \text{get}(\text{put}(s, v)) = v$$

$$(\text{PutPut}): \forall s \in S, \forall v, v' \in V, \text{put}(\text{put}(s, v), v') = \text{put}(s, v')$$

レンズの一貫性:

- well-behaved: (GetPut)、(PutGet)を満たす
- very-well-behaved: (GetPut)、(PutGet)、(PutPut)を満たす
- bijective: (StrongGetPut)、(PutPut)を満たす

Core Lens Laws:(PutPut)

(PutGet): $\forall s \in S, \forall v, v' \in V, \text{put}(\text{put}(s, v), v') = \text{put}(s, v')$

ソースを複数回更新した際に得られるソースと、

最後と同じビューで一回だけ更新した際に得られるソースが等しい

例:

$$S = V = \mathbb{N}$$

$$s \in S, v \in V,$$

$$\text{put}(s, v)$$

$$= (s \sqcap 1)(v + 1)$$

$$x \sqcap y = \begin{cases} x & (x \leq y) \\ y & (x > y) \end{cases}$$

例



$$s, s' \in S$$

$$\text{put}(\text{put}(2,3), 4)$$

$$= \text{put}(4,4)$$

$$= 5$$

$$\text{put}(2,4) = 5$$

レンズの一貫性

- well-behaved : (GetPut)と(PutGet)を満たす
ソースとビューの間でデータの変換が矛盾なく行える
- very-well-behaved : (GetPut)と(PutGet)と(PutPut)を満たす
well-behavedに加えて、過去の更新に影響されない
- bijective : (StrongGetPut)と(PutGet)を満たす
ソースとビューの間に一対一の関係があり、ソースが一意的

レンズ則の含意関係

bijective : (StrongGetPut) と (PutGet) を満たす。

実際は

bijective : (StrongGetPut)、(PutGet)、(PutPut) を満たす。

これは、

$(\text{StrongGetPut}) \wedge (\text{PutGet}) \Rightarrow (\text{PutPut})$

の**含意関係**が成り立つため

レンズ則の含意関係

$$(\text{StrongGetPut}) \wedge (\text{PutGet}) \Rightarrow (\text{PutPut})$$

$$(\text{StrongGetPut}): \text{put}(s, \text{get}(s')) = s'$$

$$(\text{PutGet}): \text{get}(\text{put}(s, v)) = v$$

$$(\text{PutPut}): \text{put}(\text{put}(s, v), v') = \text{put}(s, v')$$

Proof.

$$\begin{aligned} & \text{put}(\text{put}(s, v), v') \\ = & \{\text{by}(\text{PutGet}): v' = \text{get}(\text{put}(s, v'))\} \\ & \quad \text{put}(\text{put}(s, v), \text{get}(\text{put}(s, v'))) \\ = & \{\text{by}(\text{StrongGetPut}): \text{put}(s, \text{get}(s')) = s'\} \\ & \quad \text{put}(s, v') \end{aligned}$$

■

レンズ則の含意関係（反例）

well-behaved : (GetPut)と(PutGet)を満たす

very-well-behaved : (GetPut)、(PutGet)、(PutPut)を満たす

この時、以下が成り立つことが期待される：

$$(GetPut) \wedge (PutGet) \not\Rightarrow (PutPut)$$

レンズ則の含意関係（反例）

$$(\text{GetPut}) \wedge (\text{PutGet}) \not\Rightarrow (\text{PutPut})$$

$$(\text{GetPut}) : \text{put}(s, \text{get}(s)) = s$$

$$(\text{PutGet}) : \text{get}(\text{put}(s, v)) = v$$

$$(\text{PutPut}) : \text{put}(\text{put}(s, v), v') = \text{put}(s, v')$$

Proof.

$S = \{s_1, s_2, s_3\}, V = \{v_1, v_2\}$ とし、以下のレンズを考える。

$$\text{get}(s_1) = v_1, \quad \text{get}(s_2) = v_1, \quad \text{get}(s_3) = v_2$$

$$\text{put}(s_1, v_1) = s_1, \quad \text{put}(s_2, v_1) = s_2, \quad \text{put}(s_3, v_1) = s_2$$

$$\text{Put}(s_1, v_2) = s_2, \quad \text{put}(s_2, v_2) = s_3, \quad \text{put}(s_3, v_2) = s_3$$

Weaker Lens Laws

レンズ則はCore Lens Laws以外にも必要

- Core Lens Lawsでは強すぎることがある
- レンズやCore Lens Lawsの特徴をより明らかにしたい
 - $P \Rightarrow Q$ の時、 Q は P の「特徴」と思える
- 特定の応用で必要とされる特性を表現したい

よってCore Lens Lawsよりも弱いレンズ則が必要

→ Weaker Lens Laws

これまでに 1~2 個のWeaker Lens Lawsが提案してきた

問題点と目的 [Nakano '25][Hashiba '24]

現在、Core Lens Laws4個Weaker Lens Laws12個の
計16個のレンズ則が存在し、その含意関係が示されてきた

しかし、getとputのいずれかが部分関数の場合、16個全てのレンズ則の含意関係が示されたわけではない

- (get,put)が(全域,部分)の場合、12個
- (get,put)が(部分,部分)の場合、13個

(get,put)が(全域,全域)(部分,全域)(全域,部分)(部分,部分)
それぞれの場合で16個のレンズ則の含意関係を考察する

目次

- Weaker Lens Lawsについて
- 部分関数のレンズ則について
- Familyについて
 - GetPut Family
 - PutGet Family
 - PutPut Family
- レンズ則全体の含意関係について
- 今後の課題

目次

- Weaker Lens Lawsについて
- 部分関数のレンズ則について
- Familyについて
 - GetPut Family
 - PutGet Family
 - PutPut Family
- レンズ則全体の含意関係について
- 今後の課題

Weaker Lens Laws

Weaker Lens Lawsは以下の12個

- 1.(WeakPutGet)
- 2.(PutGetPut)
- 3.(GetPutGet)
- 4.(Undoability)
- 5.(GetInjectivity)
- 6.(GetSurjectivity)
- 7.(PutTwice)
- 8.(SourceStability)
- 9.(WeakSourceStability)
- 10.(ViewDetermination)
- 11.(PutInjectivity)
- 12.(PutSurjectivity)

(GetPutGet):

$$\forall s, s' \in S, \\ \text{get}(\text{put}(s, \text{get}(s))) \\ = \text{get}(s)$$

(GetSurjectivity):

$$\forall v \in V, \exists s \in S, \\ \text{get}(s) = v$$

目次

- Weaker Lens Lawsについて
- 部分関数のレンズ則について
- Familyについて
 - GetPut Family
 - PutGet Family
 - PutPut Family
- レンズ則全体の含意関係について
- 今後の課題

部分関数のレンズ則の解釈

get、putが部分関数の場合、今までに定義したレンズ則では計算できなくなる場合がある

例:

$$S = \{s_1, s_2, \dots, s_n\},$$

$$V = \{v_1\}$$

$$\text{get}(s_n) = \begin{cases} v_1 & (2 \leq m \leq n) \\ \perp & (m = 1) \end{cases}$$
$$\text{put}(s_m, v_1) = s_m$$

例



(StrongGetPut) :

$$\forall s \in S, \text{put}(s, \text{get}(s')) = s$$

$$\text{put}(s_2, \text{get}(s_1)) = \text{put}(s_2, \perp)$$
$$=?$$

レンズ則を部分関数へ拡張 [Hashiba '24]

新たな表記法として「 $\text{get}(s) \downarrow v$ 」を導入

これは「 $\text{get}(s)$ が部分関数で定義されていて、
かつ計算結果が v になる」ことを意味する

(StrongGetPut):

$$\text{put}(s, \text{get}(s')) = s'$$

$$\forall s, s' \in S, \forall v \in V, \text{get}(s') \downarrow v \Rightarrow \text{put}(s, v) \downarrow s'$$

部分関数のレンズ則は、 get と put が全域関数の場合、対応する全域関数のレンズ則と同値になる。

レンズ則を部分関数へ拡張 [Hashiba '24]

新たな表記法として「 $\text{get}(s) \downarrow v$ 」を導入

これは「 get が部分関数で定義されていて

かつ計算結果が v になる」ことを意味する

例:

$$S = \{s_1, s_2, \dots, s_n\},$$

$$V = \{v_1\}$$

$$\text{get}(s_m) = \begin{cases} v_1 & (2 \leq m \leq n) \\ \perp & (m = 1) \end{cases}$$

$$\text{put}(s_m, v_1) = s_m$$

例



(StrongGetPut) :

$$\forall s, s' \in S, \forall v \in V,$$

$$\text{get}(s') \downarrow v \Rightarrow \text{put}(s, v) \downarrow s'$$

$$\text{get}(s_1) \downarrow \perp \not\Rightarrow \text{put}(s, v)$$

レンズ則を部分関数へ拡張 [Hashiba '24]

(PutPut)は定義される前提を2通り用いる

$$\text{put}(\text{put}(s, v), v') = \text{put}(s, v')$$



$$\text{put}(s', v') = \text{put}(s, v')$$



$$s'' = \text{put}(s, v')$$

$$\text{put}(s', v') = s''$$



$$\text{put}(s, v) = s'$$

$$\text{put}(s', v') = s''$$

$$\text{put}(s', v') = s''$$

(PutPut-1)

$$\forall s, s', s'' \in S, \forall v, v' \in V,$$

$$\text{put}(s, v) \downarrow s'$$

$$\Rightarrow (\text{put}(s', v') \downarrow s'' \Rightarrow \text{put}(s, v') \downarrow s'')$$

(PutPut-2)

$$\forall s, s', s'' \in S, \forall v, v' \in V,$$

$$\text{put}(s, v) \downarrow s'$$

$$\Rightarrow (\text{put}(s, v') \downarrow s'' \Rightarrow \text{put}(s', v') \downarrow s'')$$

目次

- Weaker Lens Lawsについて
- 部分関数のレンズ則について
- Familyについて
 - GetPut Family
 - PutGet Family
 - PutPut Family
- レンズ則全体の含意関係について
- 今後の課題

レンズ則のFamily [Nakano '21]

レンズ則の含意関係をむやみやたらに示すのは効率が悪い

→16個のレンズ則の含意関係を分かりやすくまとめるために、
レンズ則を3つのグループに分ける

そのグループを**Family**と呼ぶ

FamilyはCore Lens Lawsを一番強いレンズ則とし、
そのCore Lens Lawsと単純な含意関係を持つWeaker Lens Law
をそのCore Lens LawsのFamilyに分類する

Family

16個のレンズ則をその含意関係を元に3つのFamilyに分類

- GetPut Family:(StrongGetPut)と単純な含意を持つレンズ則
(GetPut)もこのFamilyのレンズ則
- PutGet Family:(PutGet)と単純な含意を持つレンズ則
- PutPut Family:(PutPut)と単純な含意を持つレンズ則

任意のFamilyに分類されるレンズ則から他のFamilyのレンズ則への含意関係は存在しない

Familyの内訳

16個のレンズ則は以下の3つのFamilyに分類

- GetPut Family
(StrongGetPut), (GetPut), (**WeakPutGet**), (PutGetPut),
(**GetPutGet**), (Undouability), (GetInjectivity), (SourceStability),
(**WeakSourceStability**), (PutInjectivity)
- PutGet Family
(PutGet), (**WeakPutGet**), (**GetPutGet**), (GetSurjectivity),
(ViewDetermination), (PutInjectivity)
- PutPut Family
(PutPut), (PutTwice), (**WeakSourceStability**)

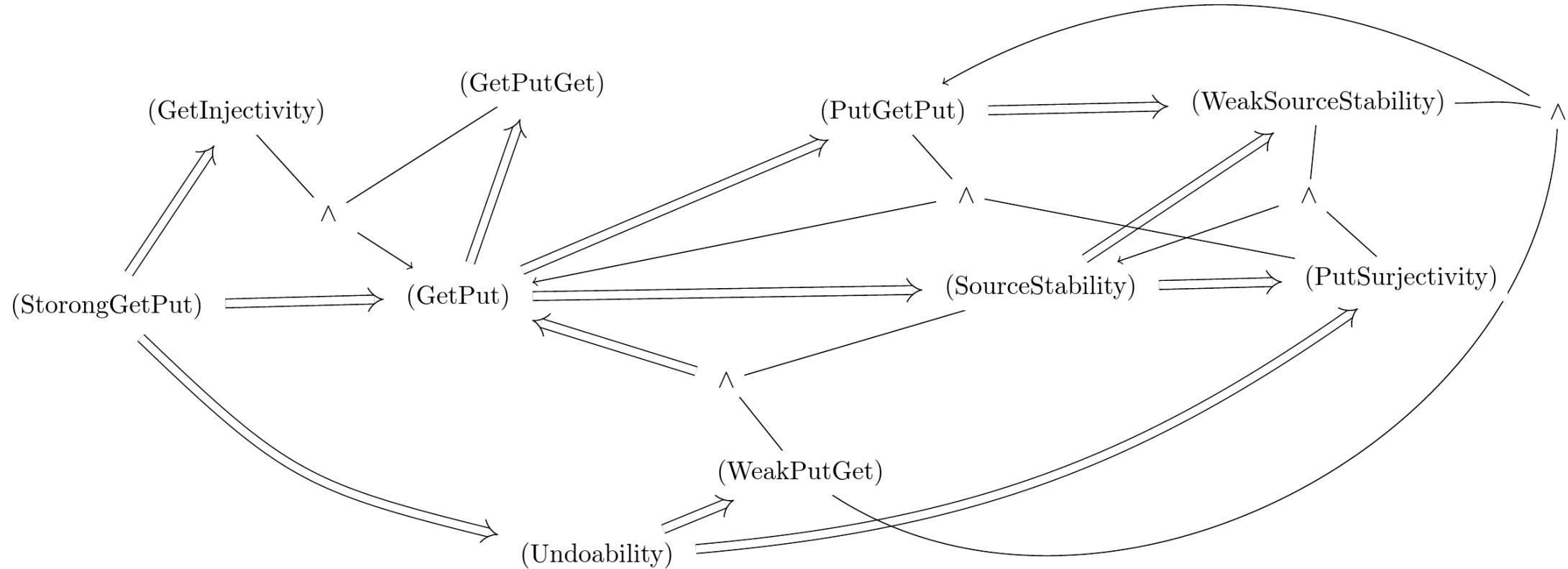
Familyの内訳

- GetPut Family
(StrongGetPut), (GetPut), (**WeakPutGet**), (PutGetPut),
(**GetPutGet**), (Undouability), (GetInjectivity), (SourceStability),
(**WeakSourceStability**), (PutInjectivity)
- PutGet Family
(PutGet), (WeakPutGet), (GetPutGet), (GetSurjectivity),
(ViewDetermination), (PutInjectivity)
- PutPut Family
(PutPut), (PutTwice), (WeakSourceStability)

GetPut Family

get:全域関数、 put:全域関数

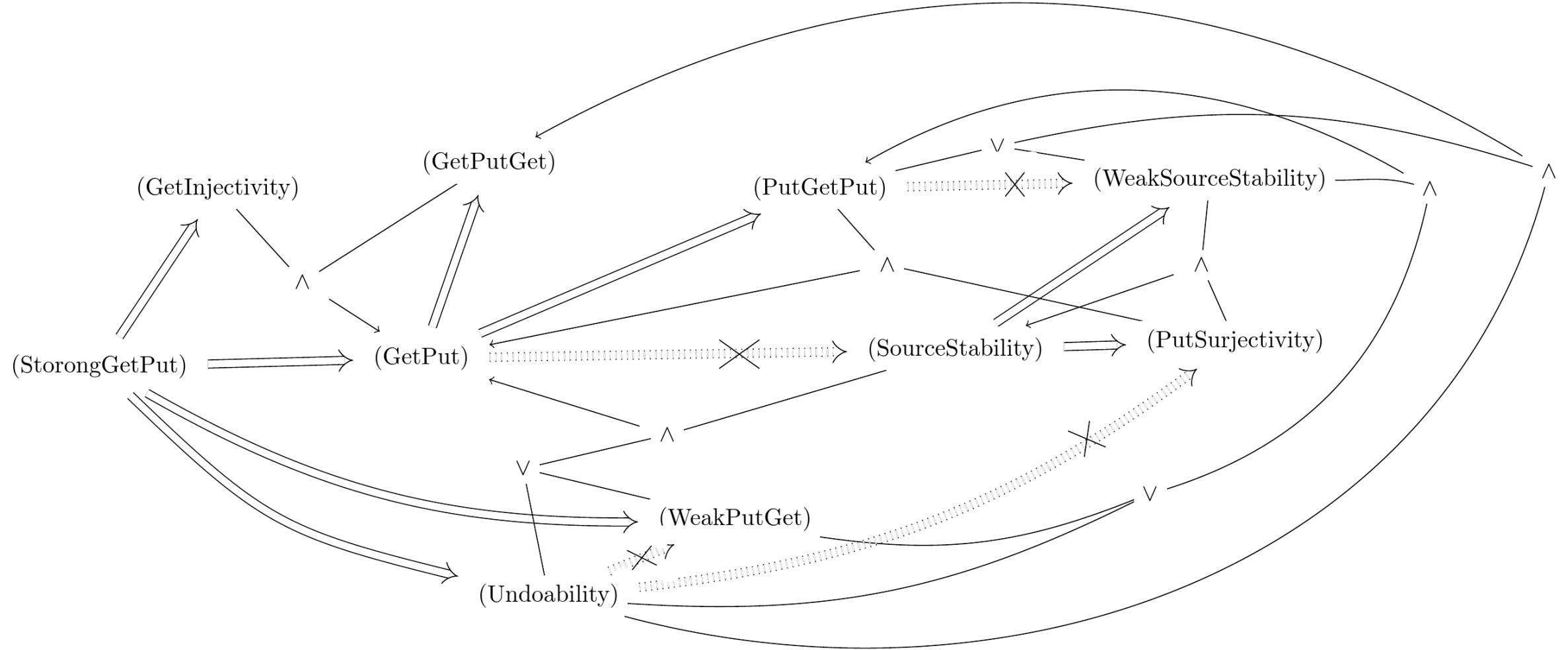
$(\text{Getinjectivity}) \wedge (\text{GetPutGet}) \Rightarrow (\text{GetPut})$



GetPut Family

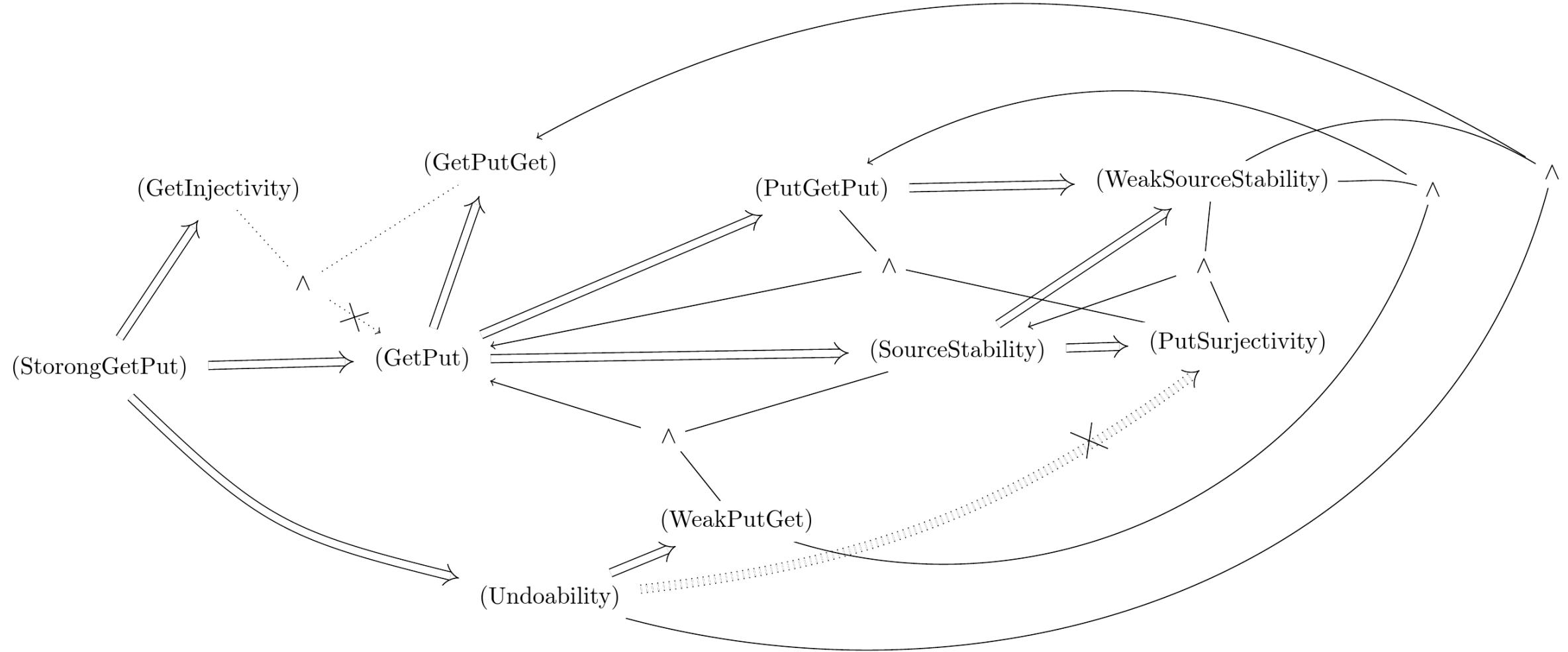
get:部分関数、 put:全域関数

$$((\text{PutGetPut}) \vee (\text{WeakSourceStability})) \wedge (\text{Undoability}) \Rightarrow (\text{GetPutGet})$$



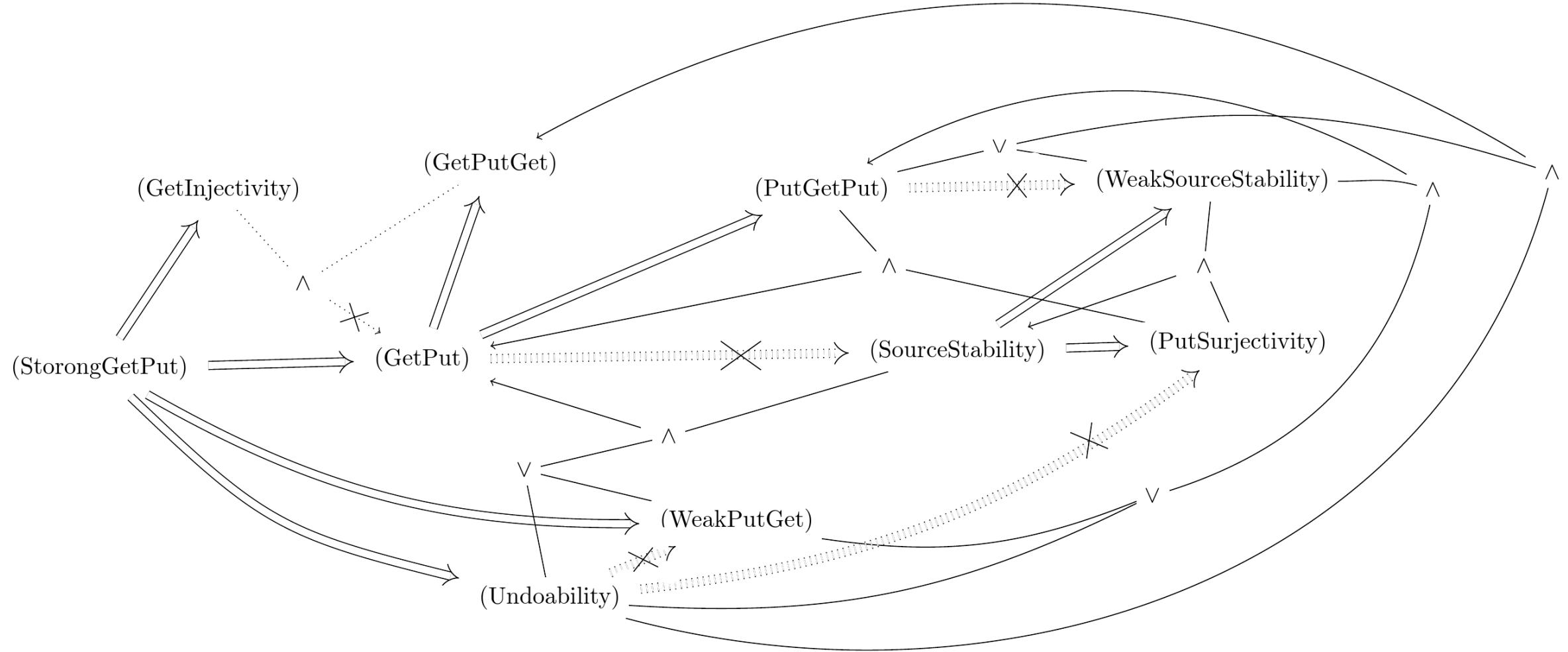
GetPut Family

get:全域関数、 put:部分関数



GetPut Family

get:部分関数、 put:部分関数

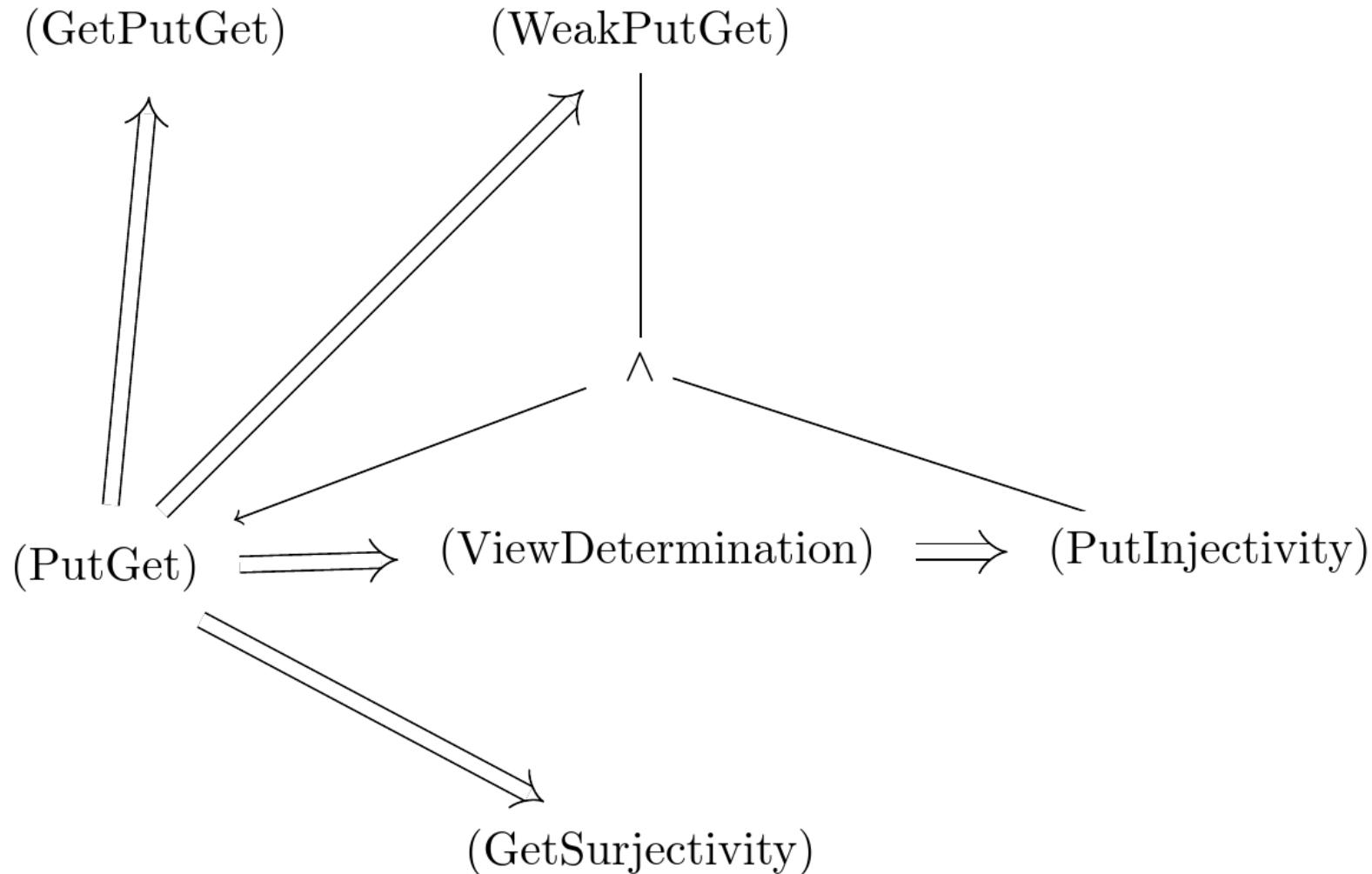


Familyの内訳

- GetPut Family
(StrongGetPut), (GetPut), (WeakPutGet), (PutGetPut),
(GetPutGet), (Undouability), (GetInjectivity), (SourceStability),
(WeakSourceStability), (PutInjectivity)
- PutGet Family
(PutGet), (**WeakPutGet**), (**GetPutGet**), (GetSurjectivity),
(ViewDetermination), (PutInjectivity)
- PutPut Family
(PutPut), (PutTwice), (WeakSourceStability)

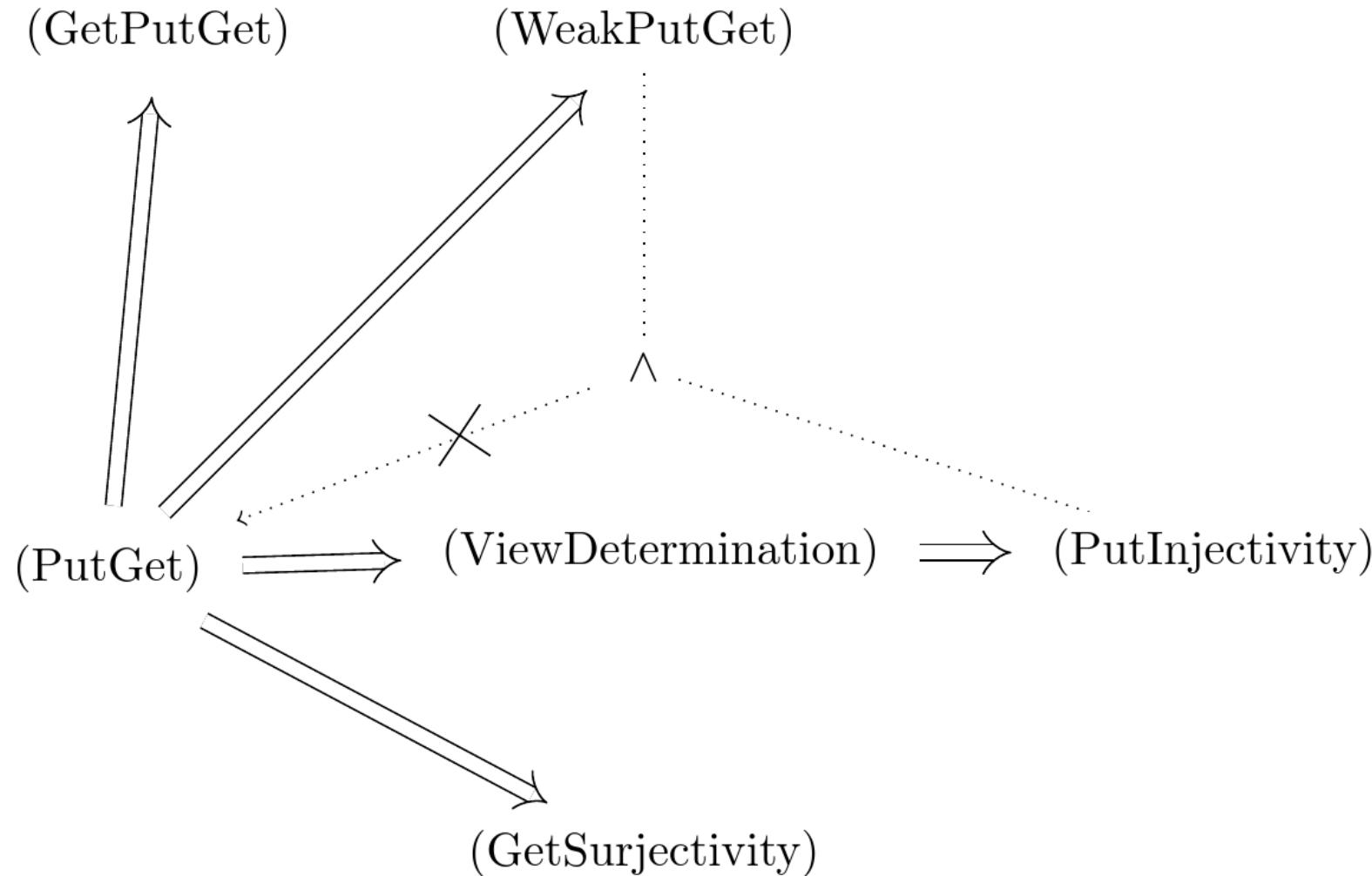
PutGet Family

get:全域関数、 put:全域関数



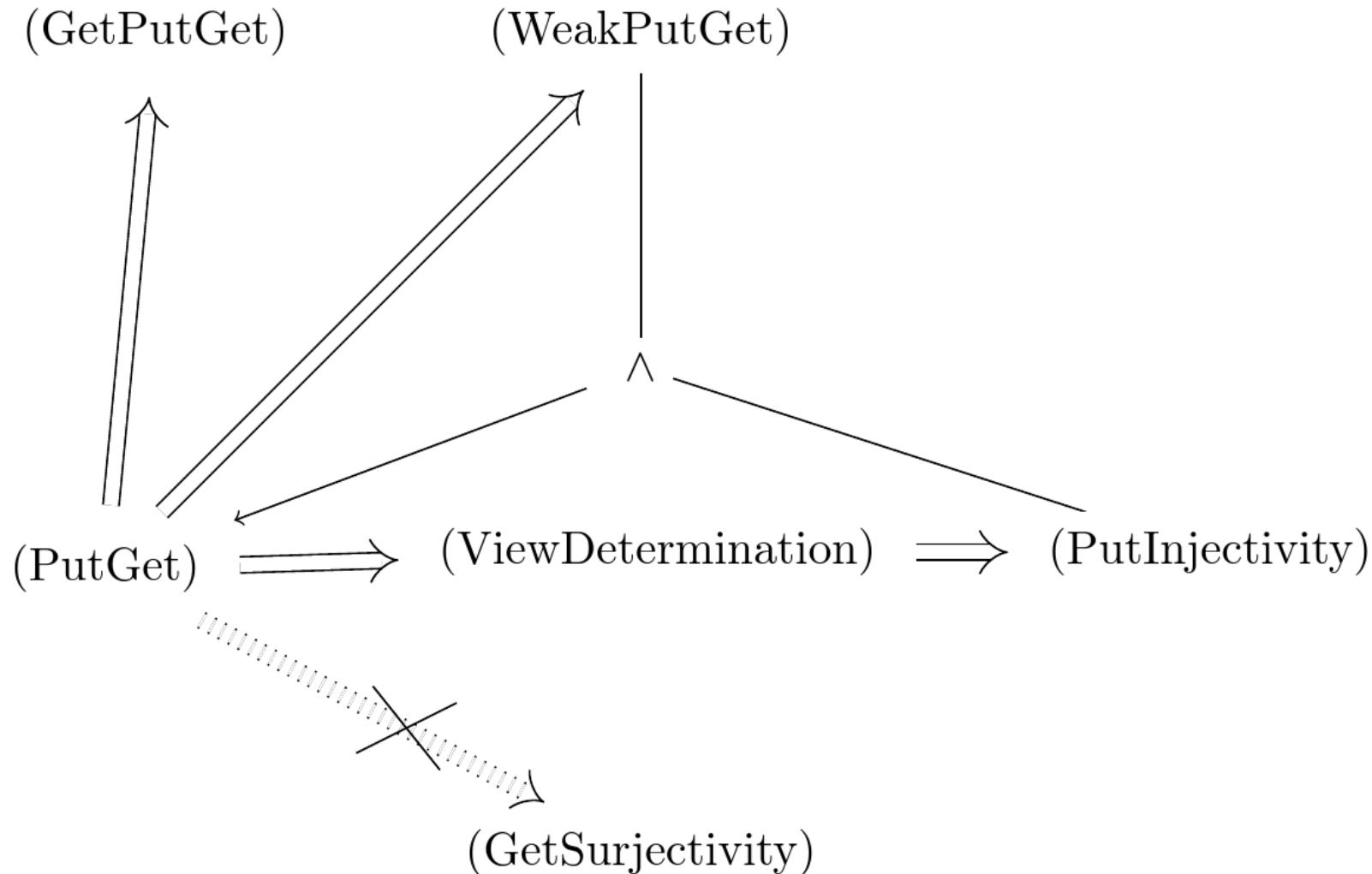
PutGet Family

get:部分関数、 put:全域関数



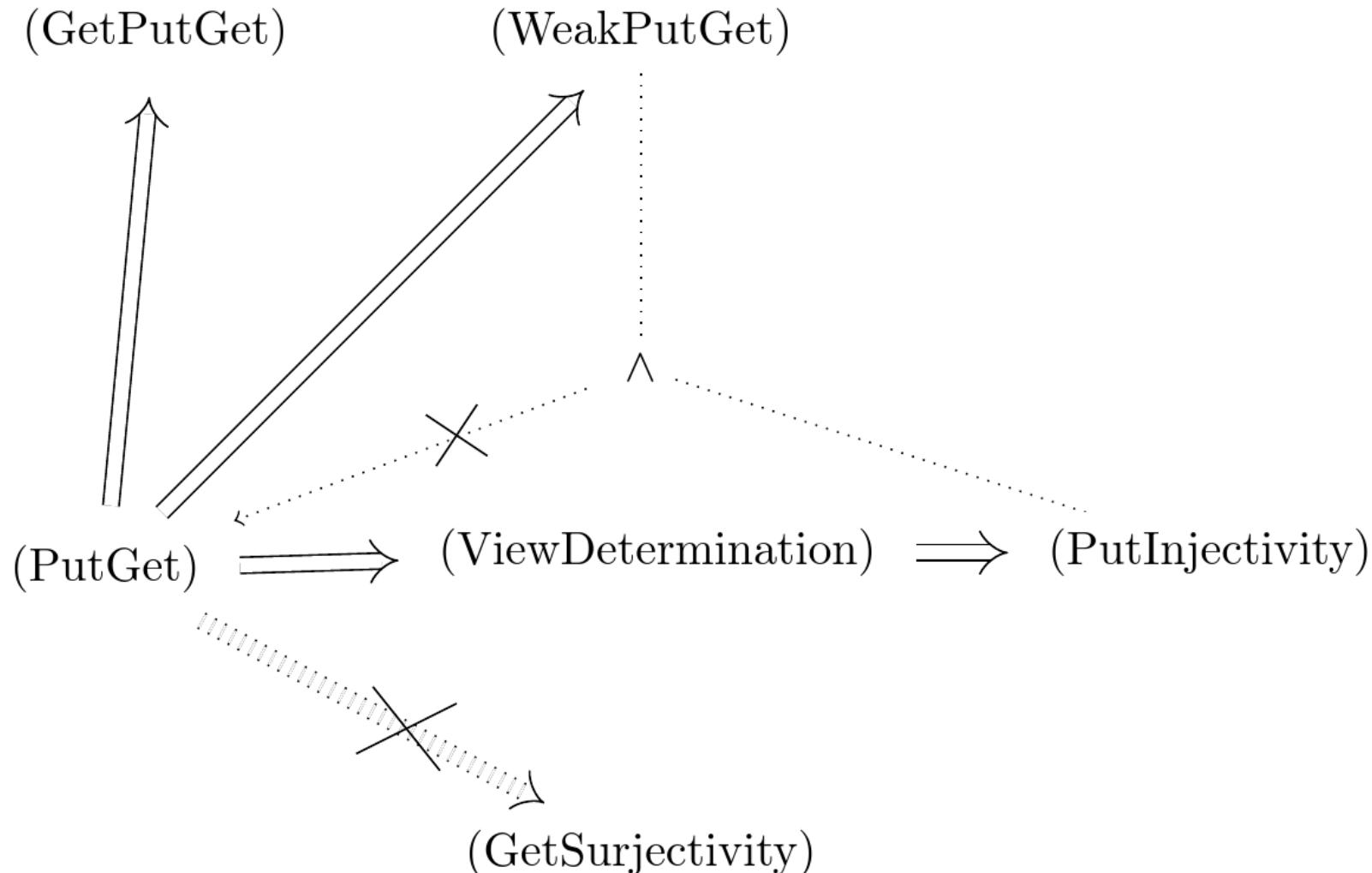
PutGet Family

get:全域関数、 put:部分関数



PutGet Family

get:部分関数、 put:部分関数



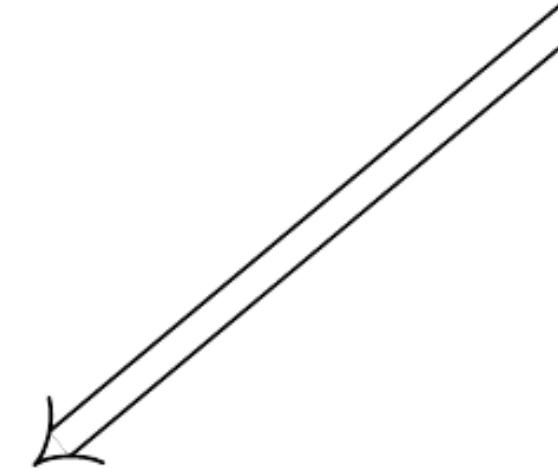
Familyの内訳

- GetPut Family
(StrongGetPut), (GetPut), (WeakPutGet), (PutGetPut),
(GetPutGet), (Undouability), (GetInjectivity), (SourceStability),
(WeakSourceStability), (PutInjectivity)
- PutGet Family
(PutGet), (WeakPutGet), (GetPutGet), (GetSurjectivity),
(ViewDetermination), (PutInjectivity)
- PutPut Family
(PutPut), (PutTwice), (WeakSourceStability)

PutPut Family

put:全域関数

(PutPut) \Rightarrow (PutTwice)

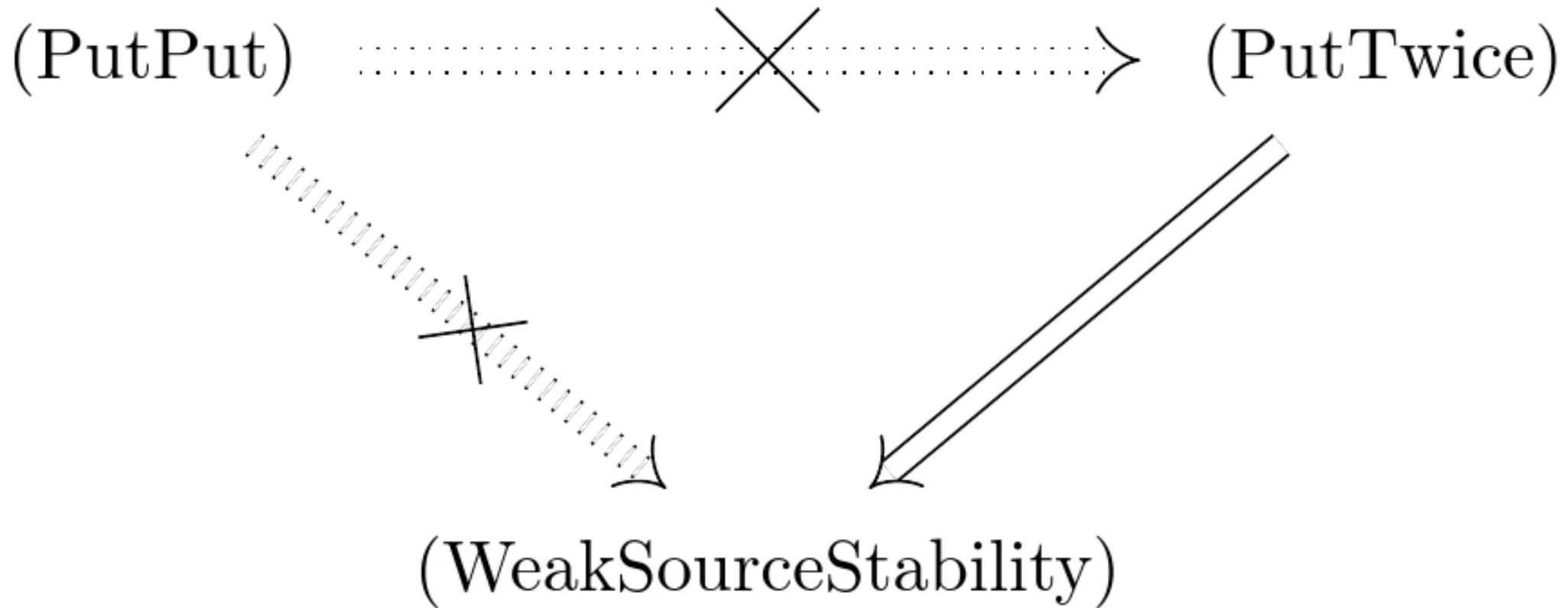


(WeakSourceStability)

PutPut Family

put:部分関数

putが部分関数の場合、(PutPut)は(PutPut-1)

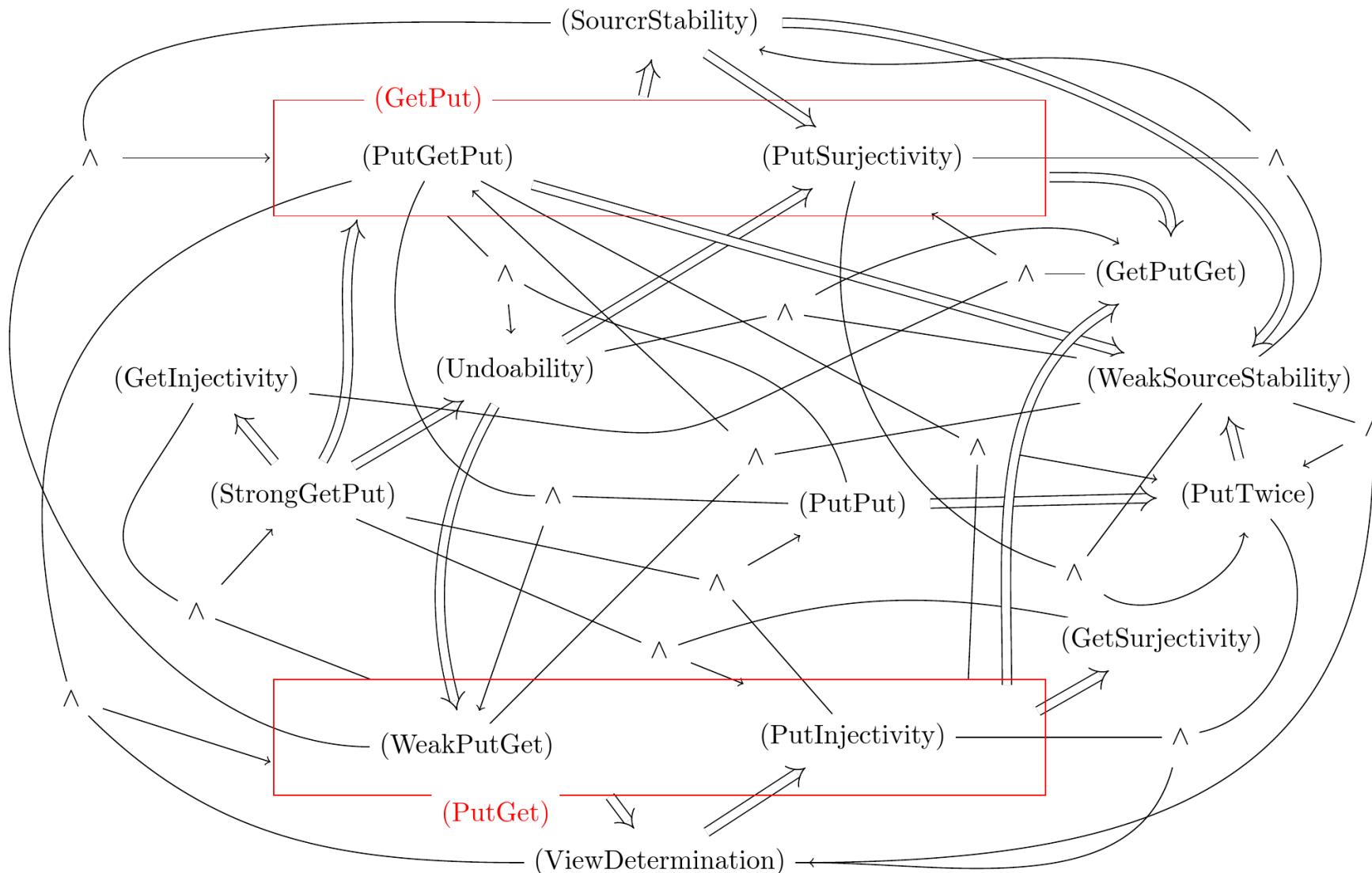


目次

- Weaker Lens Lawsについて
- 部分関数のレンズ則について
- Familyについて
 - GetPut Family
 - PutGet Family
 - PutPut Family
- レンズ則全体の含意関係について
- 今後の課題

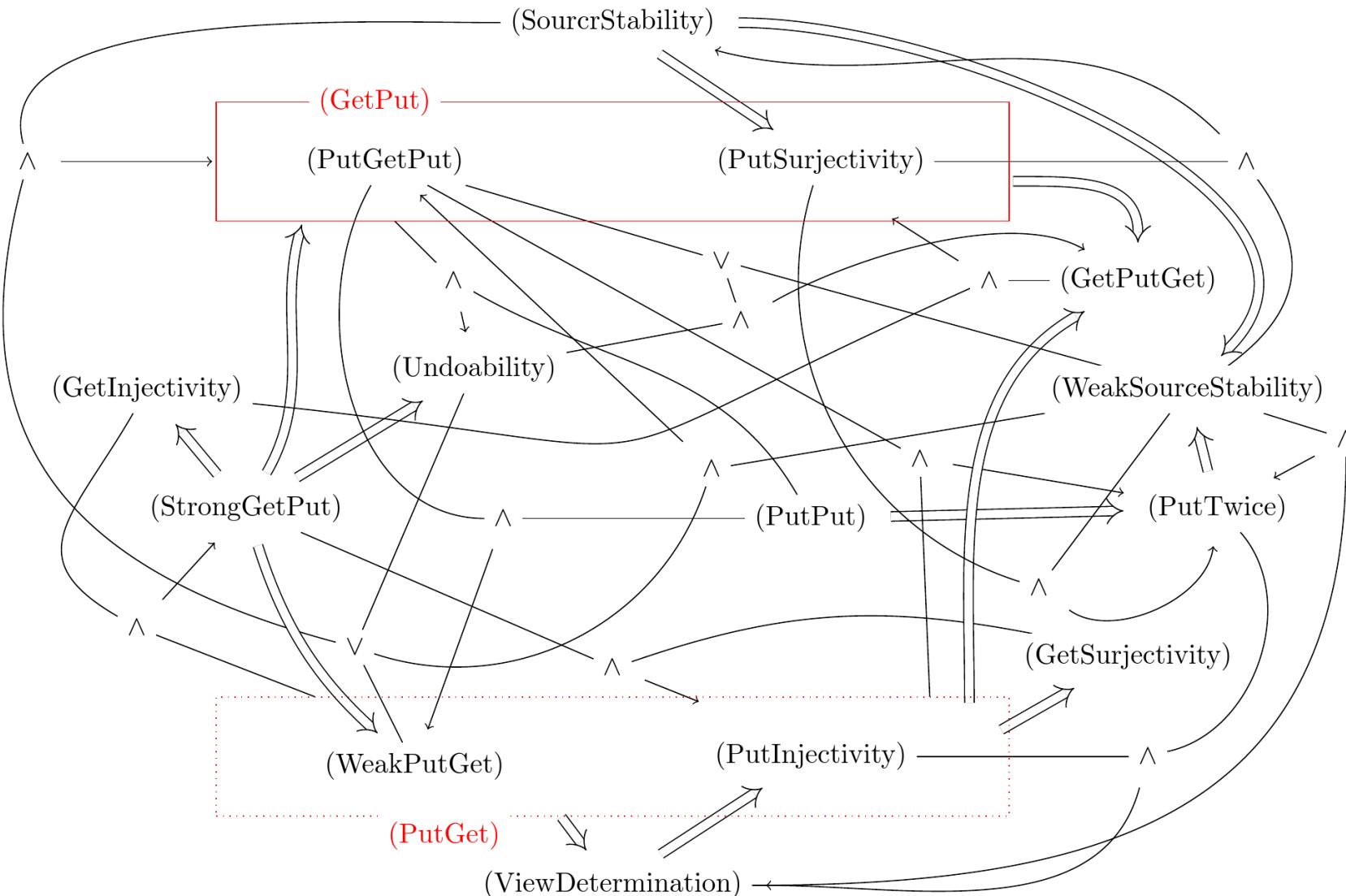
16個のレンズ則全体の含意関係

get:全域関数、 put:全域関数



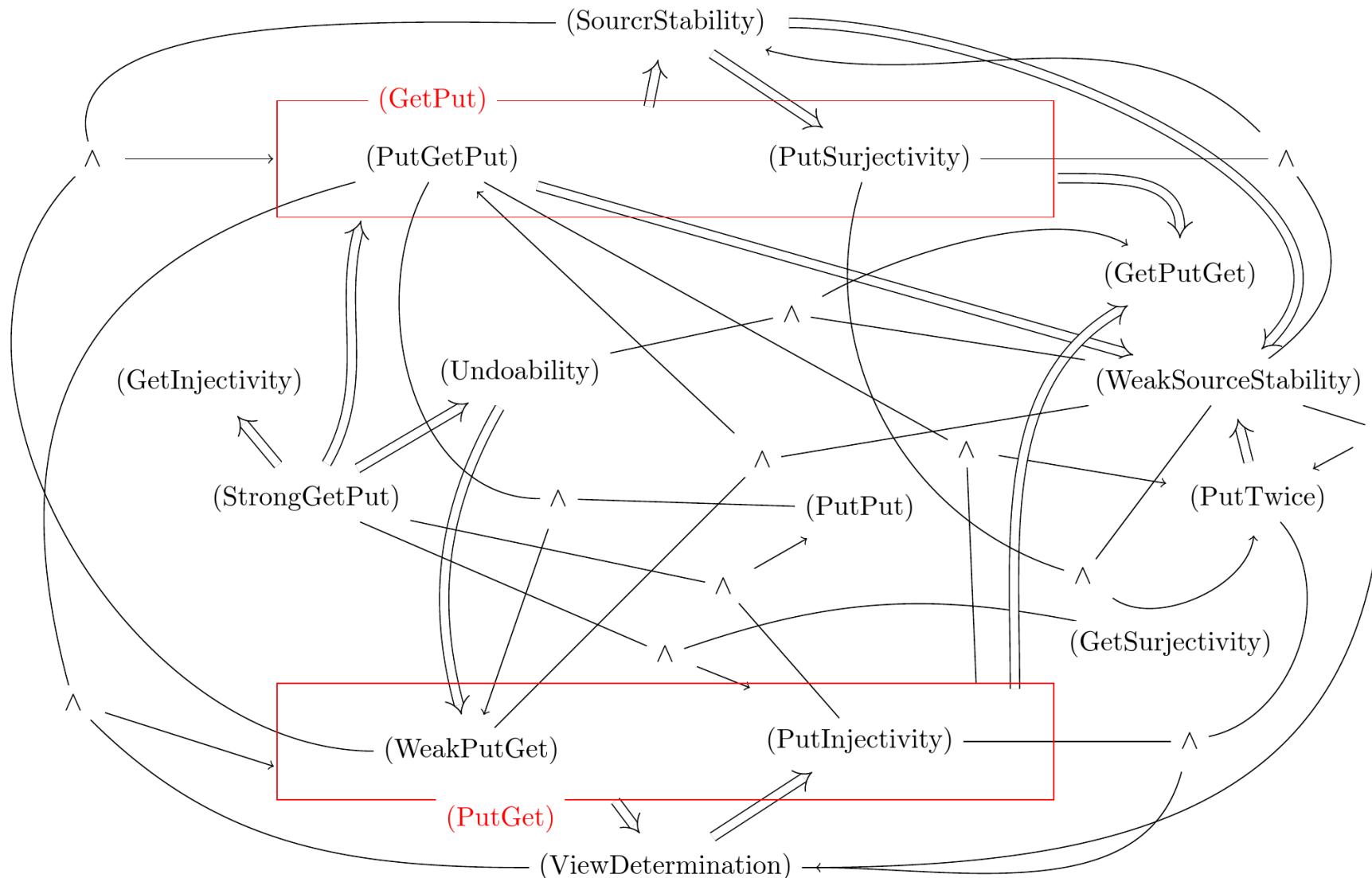
16個のレンズ則全体の含意関係

get:部分関数、put:全域関数



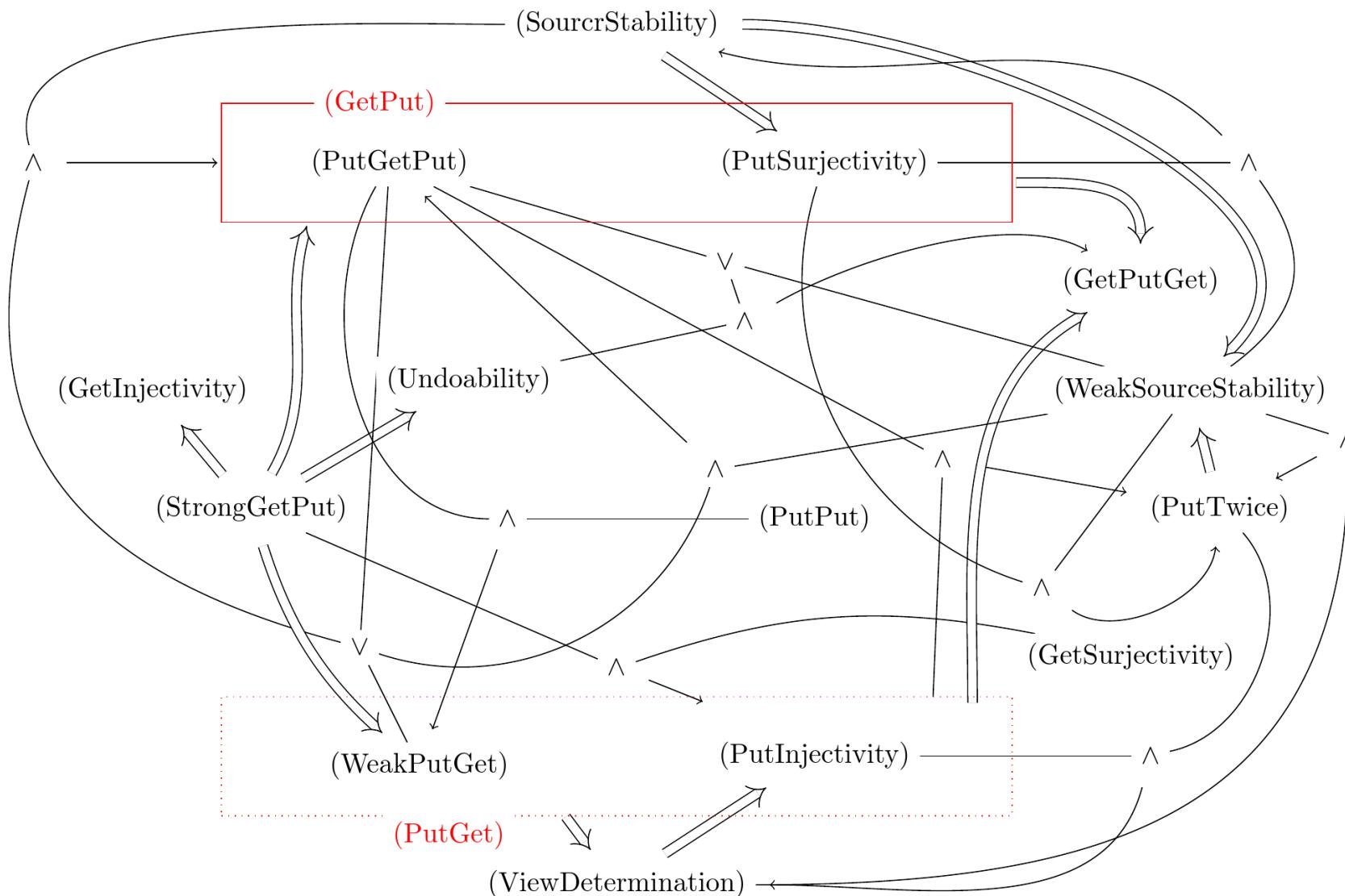
16個のレンズ則全体の含意関係

get:全域関数、 put:部分関数 ここでの(PutPut)は(PutPut-1)



16個のレンズ則全体の含意関係

get:部分関数、 put:部分関数 ここでの(PutPut)は(PutPut-1)



目次

- Weaker Lens Lawsについて
- 部分関数のレンズ則について
- Familyについて
 - GetPut Family
 - PutGet Family
 - PutPut Family
- レンズ則全体の含意関係について
- 今後の課題

今後の課題

- ・全体の含意関係の図は完全ではない
 - ・3個以上のレンズ則から1つのレンズ則への含意関係が存在するか
 - ・図から導かれない含意関係には反例があるか
- ・putが部分関数の場合、(GetSurjectivity)に対して新たなレンズ則を定義する必要はあるのか
- ・レンズの一貫性について新たな考察が必要となるか
- ・(PutPut-1)と(PutPut-2)に関する含意関係についての考察

参考文献

- [1] Fran^{çois} Bancilhon and Nicolas Spryatos. Update semantics of relationalviews. ACM Trans. Database Syst., Vol. 6, No. 4, pp. 557–575, 1981.
- [2] Sebastian Fischer, Zhenjiang Hu, and Hugo Pacheco. A clear picture of lens laws. In Ralf Hinze and Janis Voigtländer, editors, Mathematics of Program Construction, pp. 215–223, Cham, 2015. Springer International Publishing.
- [3] Keisuke Nakano. Lens Laws Zoo, pp. 37–59. Springer Nature Singapore, Singapore, 2025.
- [4] Hashiba Keishi. Relations among lens laws for bidirectional transformations consisting of partial functions. Vol. 39, , 2024.