

血液型の代数(?)の実装

https://github.com/yosukeueda33/bloodtype_algebra

AB型とO型の子供はA型かB型のみ

遺伝型

母親(AB型)



父親(O型)



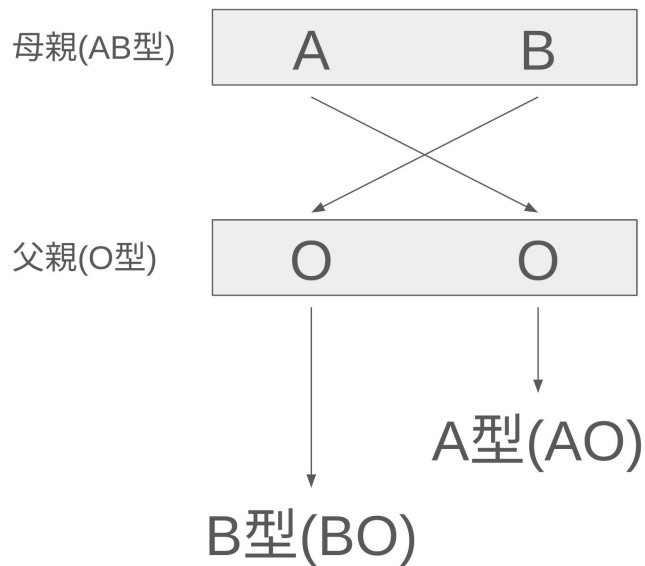
A型(AO)

B型(BO)

		A				B				AB		O	
		A A		A O		B B		B O		A B		O O	
A	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO
B	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO
AB	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
O	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO
	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO

遺伝型加算の定義

$$[A,B] \text{ } | + | \text{ } [O,O] = [[A,O], [B,O]]$$



数学のイコールを使って良いのか疑問だが
何故かそれで後の計算が整合する

遺伝型減算の定義

$$?? \text{ } | + | \text{ } [O, O] = [[A, O], [B, O]]$$

母親の遺伝型が分からない

$$?? = [[A, O], [B, O]] \text{ } | - | \text{ } [O, O]$$

$$?? = [[A, B]]$$

|+|を利用してありえるパターンを全て試せば演算可能。

利用例

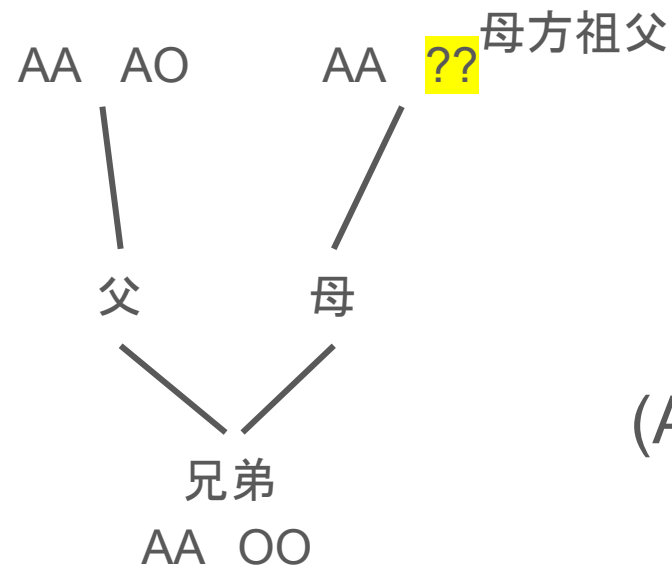
$$\begin{array}{lcl} [A,B] & | + | & [O,O] = [[A,O], [B,O]] \\ [[A,O], [B,O]] & | - | & [O,O] = [[A,B]] \end{array}$$

$$\begin{array}{lcl} [B,B] & | + | & [A,A] = [[A,B]] \\ [[A,B]] & | - | & [A,A] = [[A,B], [B,B], [B,O]] \end{array}$$

不均衡。型を分けて解消した。

$$\begin{array}{lcl} [A,O] & | + | & [B,O] = [[A,B], [A,O], [B,O], [O,O]] \\ [[A,B], [A,O], [B,O], [O,O]] & | - | & [A,O] = [[B,O]] \end{array}$$

代数方程式的な扱い



$$(AA \overset{\text{父}}{|+|} AO) \overset{\text{母}}{|+|} (AA \overset{\text{母}}{|+|} ??) = [AA, OO]$$

$$\overset{\text{母方祖父}}{??} = ([AA, OO] \overset{\text{父}}{|-|} (AA \overset{\text{父}}{|+|} AO)) \overset{\text{母方祖母}}{|-|} AA$$

$$?? = [AA, AO, BO]$$

[AA,BB,OO]である可能性はない

この挙動は以下の点で大変興味深い。

- 遺伝型が実数と同様な代数として扱えている。
- 定義した演算子は実数のように単純な演算ではなく、その裏に直積や総当りの処理を行っている。しかし利用においては実数の演算子と同様に平意に認識できる。

また実装においても以下の面白さがあった。

- Haskellの特にListモナドが効果的に利用できた。
- all, anyの利用が効果的であり、その量化の概念が遺伝型の型にも必要となった。
-

```
(|+|) :: QGenos -> QGenos -> QGenos  
(QGenos _ gls1) |+| (QGenos _ gls2) = QGenos EXIST . nub $ do  
  gs1 <- gls1 :: [ProbableGenos]  
  gs2 <- gls2 :: [ProbableGenos]  
  Genos g1 <- gs1 :: ProbableGenos  
  Genos g2 <- gs2 :: ProbableGenos  
  ge1 <- g1 :: [GenoTypeElem]  
  ge2 <- g2 :: [GenoTypeElem]  
  return [Genos $ sort [ge1, ge2]]
```



```

(|-|) :: QGenos -> QGenos -> QGenos

childs@(QGenos _ proba_childs) |-| parent = QGenos EXIST . nub $ mhelper childs parent
  where
    mhelper (QGenos ALL real_childs) (QGenos ALL (p1:[])) = mmhelper (all) p1
    mhelper (QGenos EXIST real_childs) (QGenos ALL (p1:[])) = mmhelper (any) p1
    mhelper (QGenos ALL real_childs) (QGenos EXIST possible_parent) = concatMap (mmhelper (all)) possible_parent
    genosAll = nub [Genos $ sort [g1, g2] | g1 <- [A,B,0] , g2 <- [A,B,0]]
    mmhelper :: ((ProbableGenos -> Bool) -> [ProbableGenos] -> Bool) -> ProbableGenos -> [ProbableGenos]
    mmhelper quant_f p1 = do
      p2 <- genosAll
      let (QGenos EXIST gs_possible) = (QGenos ALL [p1]) |+| (QGenos ALL [[p2]])
      guard $ quant_f (any (`elem` [g|g_proba <- gs_possible, g<-g_proba])) proba_childs
      return [p2]

```

引き算の定義

父遺伝型 **+** 母遺伝型 = 子遺伝型の可能性リスト
兄弟の遺伝型リスト **-** 父遺伝型 = 母遺伝型の可能性リスト

定義:
兄弟全員の遺伝型が合致する母親を全検索する。
兄弟 **-** 父 = [母 | 母 ∈ 全母リスト, 父 **+** 母 = 子リスト, ∀ 兄弟 ∈ 子リスト]

例:
[A,B] **+** [O,O] = [[A,O], [B,O]]
[[A,O], [B,O]] **-** [O,O] = [[A,B]]

[B,B] **+** [A,A] = [[A,B]]
[[A,B]] **-** [A,A] = [[A,B], [B,B], [B,O]]

不均衡。型で解消できるが少々複雑

[A,O] **+** [B,O] = [[A,B], [A,O], [B,O], [OO]]
[[A,B], [A,O], [B,O], [O,O]] **-** [A,O] = [[B,O]]

足し算の定義

AB型とO型間の子供はA型かB型のみ

血液型:=f({遺伝型,遺伝型})
子供の遺伝型 ∈ 親遺伝型 × 親遺伝型
子供の血液型=f(子供の遺伝型)

遺伝型の足し算が定義できる。
直積(両方から1つずつとる全パターン)して重複削除。

父遺伝型 | + | 母遺伝型 = 子遺伝型の可能性リスト

例:

- [A,B] | + | [O,O] = [[A,O], [B,O]]
- [A,A] | + | [B,B] = [[A,B]]
- [A,O] | + | [B,O] = [[A,B], [A,O], [B,O], [OO]]

		A				B				AB		O	
		A A		A O		B B		B O		A B		O O	
A	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO
B	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO
AB	A	AA	AA	AA	AO	AB	AB	AB	AO	AA	AB	AO	AO
	B	AB	AB	AB	BO	BB	BB	BB	BO	AB	BB	BO	BO
O	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO
	O	AO	AO	AO	OO	BO	BO	BO	OO	AO	BO	OO	OO