

インターンシップ成果報告

知識表現言語「Mahverous」の企画および作成

2023/10/20

木内 康介



目次

1. 自己紹介
2. 知識表現言語企画内容
3. 言語概要
4. 実装上の工夫
5. 達成事項と課題点
6. 今後の展望
7. インターンシップ感想



1. 自己紹介

- 名前：木内康介
- 所属：東京工業大学 数理・計算科学コース (M1)
- 専門(研究室)：プログラミング言語理論
- 研究：SimSym の開発
- 小学生の時の将来の夢：ゲームを作る人



1-α. SimSym

物理系定義ペイン

名前を入力

物体追加

A

x_A [L]

y_A [L]

v_{Ax} [L/T]

v_{Ay} [L/T]

v_{0x} [L/T]

v_{0y} [L/T]

変数名

単位

変数追加

グローバル

t [T]

g [L/T²]

変数名

単位

変数追加

方程式

$$v_{Ax} = v_{0x}$$

$$v_{Ay} = v_{0y} - gt$$

$$x_A = v_{0x}t$$

$$y_A = v_{0y}t - \frac{gt^2}{2}$$

左辺

右辺

追加

観測ペイン

観測対象

A



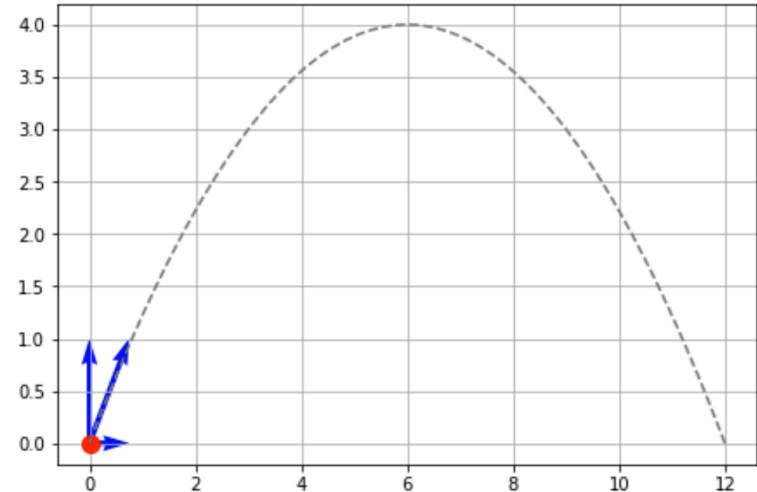
☒ 位置



☒ 速度

動作例

斜方投射



t

0.00

$v_{0x} =$ 3

$v_{0y} =$ 4

$g =$ 2





2.知識表現言語「Mahverous」企画内容



2-1. 知識表現言語「Mahverous」企画内容

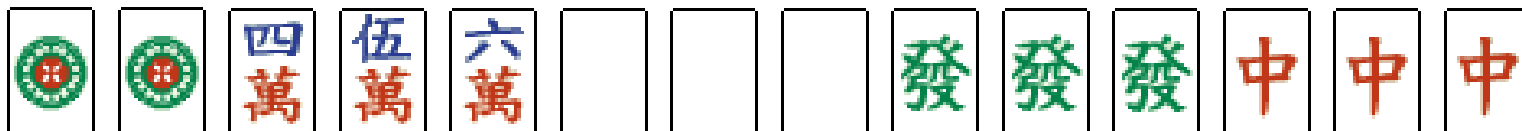
- 麻雀ライクなゲームを定義する言語(環境)
 - 表現できるもの
 - 麻雀(処理速度的に厳しい)
 - ドンジャラ・ポンジャンの類
 - 牌の種類や役の種類・条件を YAML で記述する
 - 記述内容を元に成立役を検査できる
- 実際にゲームとしてプレイできる



2-2. 表現できるゲーム例(1) 麻雀

- ・ 有名なボードゲーム
- ・ 完成形の枚数が14枚で、計算量が多い

大三元





2-3. 表現できるゲーム例(2) ドンジャラ

- ・ 麻雀の簡易版のようなもの
- ・ バリエーションが多い



©Fujiko-Pro, Shogakukan, TV-Asahi, Shin-ei, and ADK



©吾峠呼世晴／集英社・アニプレックス・ufotable



©バードスタジオ／集英社・東映アニメーション





2-3. 表現できるゲーム例(2) ドンジャラ(ドラえもん)

- ・ ドラえもんの登場人物で役を作る
- ・ 題材の作品によって役が大きく変わる

① 空の旅セット タケコブターをつけているキャラクターのみで3セット。	  	  	  	600 点
② 野比家セット のび太、のび太のママ、のび太のパパの3セット。	  	  	  	420 点
③ 剛田家セット ジャイアン、ジャイアンのかあちゃん、ジャイコ	  	  	  	420 点



2-3. 表現できるゲーム例(2) ドンジャラ(ワンピース)

- ・ ドラえもんと概ね同じ
- ・ パーツ単位の加点がある

2 ^{むぎ} ^{いち み} **麦わらの一味セット** ^{むぎ} ^{いち み} ^{あか} **麦わらの一味(赤)のパイ6コ** + ^{あか い} ^{がい} ^{はいけいしよく} ^{どうしよく} **赤以外の背景色の同色パイ3コ**または^{とくしゅやく} ^{こうせい} **特殊役1セット**で構成。



+



21 ^{まん} ^{サイ} **万点**

9 ^{せんそう} ^あ ^き **「この戦争を終わらせに来た!!!」**
^{かず} **下図4キャラクターから3キャラクターのパイを1コずつ**



+6 ^{まん} ^{サイ} **万点**



3.言語概要



3-1. 牌の記述 (YAML)

- 必須
 - 牌の名前
 - 枚数
- その他任意のパラメータ
- 共通する値は COMMON で一括定義できる

```
COMMON:  
枚数: 3  
色: なし  
タケコプター: False
```

```
ドラえもん:  
枚数: 6  
色: 赤
```

```
ドラえもんT:  
色: 赤  
タケコプター: True
```

3-2. パーツの記述 (YAML)

- 必須
 - パーツの名前
 - 制約
- オプション
 - 前処理 (後述)
 - 後処理 (後述)

同色 a b c:

制約:

- a.色 == b.色 == c.色

タケコプター同色 a b c:

制約:

- and

-

- a.色 == b.色 == c.色

- a.タケコプター

- b.タケコプター

- c.タケコプター



3-3. 役の記述 (YAML)

- 必須
 - 役の名前
 - 構造
 - 制約
- オプション
 - 前処理 (後述)
 - 後処理 (後述)

なかよしセット:

構造: 3 3 3

制約:

- and
-
- 同色(a, b, c)
- a.色 == '赤'
- 同色(d, e, f)
- d.色 == '黄'
- 同色(g, h, i)

3-4. ルールの記述 (YAML)

- 必須
 - 完成形の枚数
 - オールマイティの枚数
 - ゲームの設定値
 - ▶ 局数
 - ▶ プレイヤー数
 - ▶ 初期点数
 - ▶ 表示するパラメータ
 - ▶ 整列させるパラメータ
- オプション
 - 前処理 (後述)
 - 後処理 (後述)

完成形の枚数: 9
オールマイティの枚数: 3

ゲーム:
局数: 4
プレイヤー数: 2
初期点数: 1000
表示するパラメータ:

- 名前
- 色

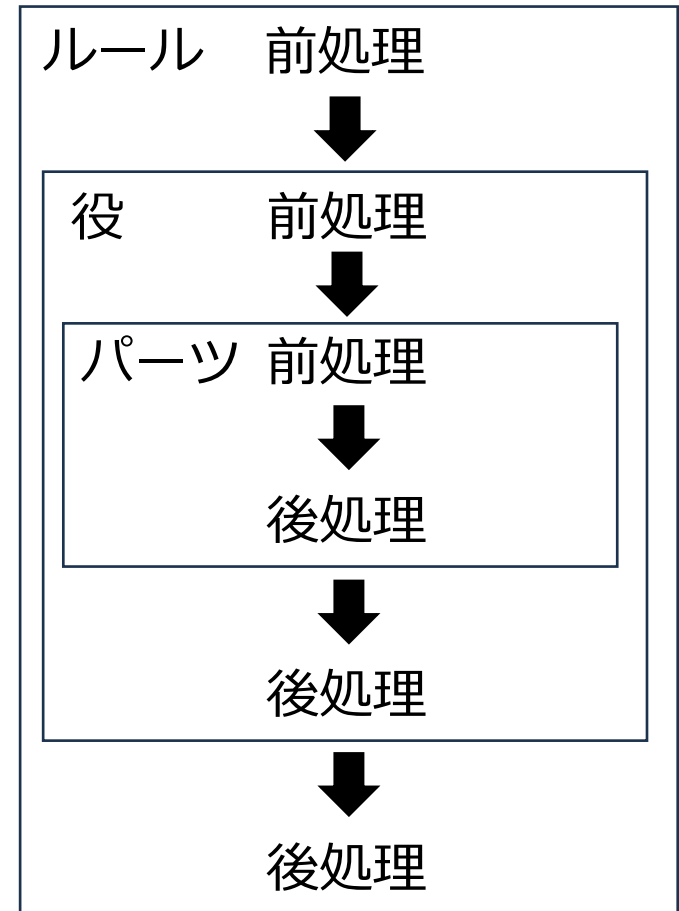
整列させるパラメータ:

- 色



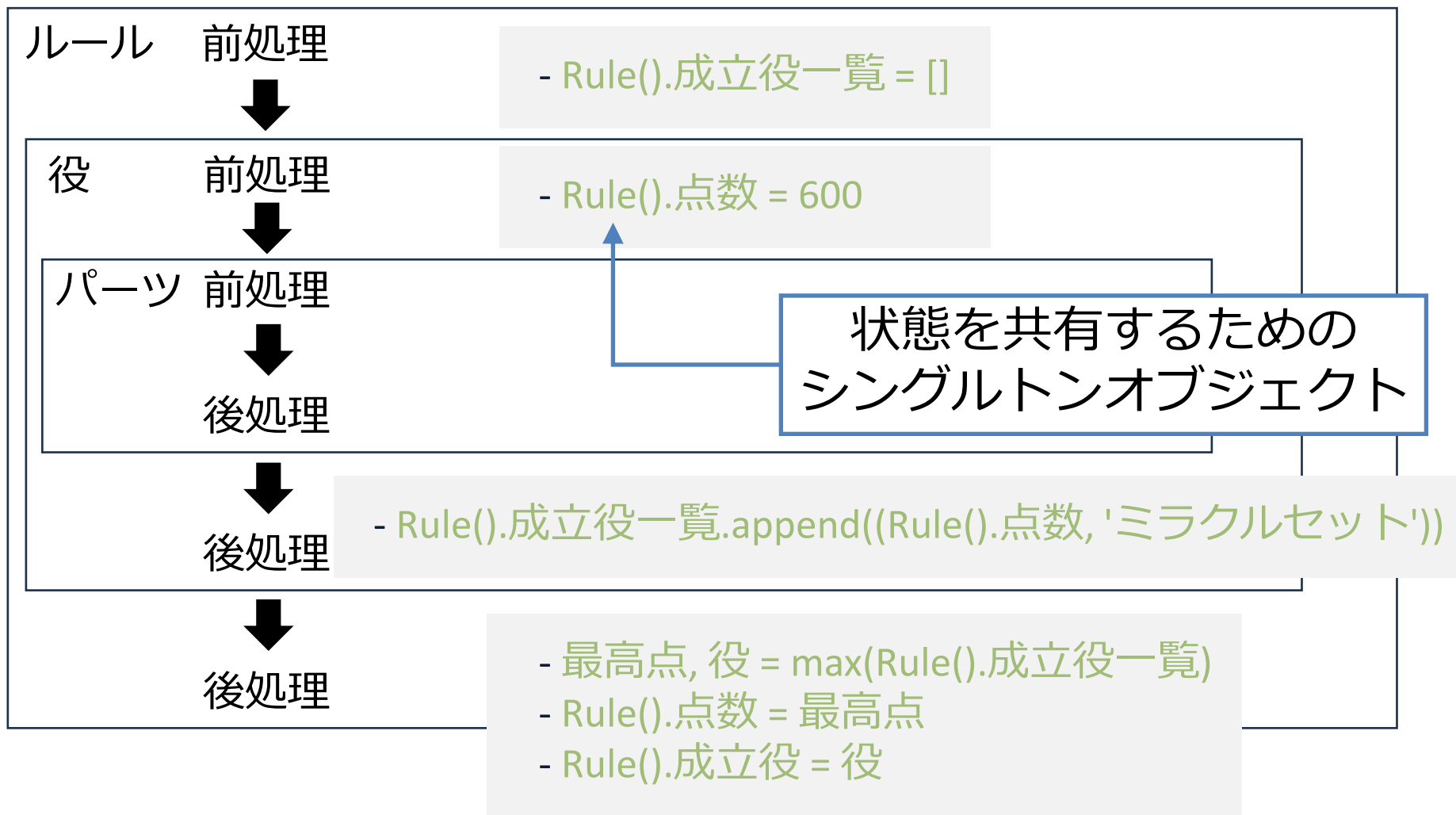
3-5. 前処理と後処理

- Python のコード片を記述
- 柔軟性を確保
- 前処理
 - 各種判定前に実行
- 後処理
 - 判定に通過したら実行





3-6. 処理の実例1 – ドラえもん





3-7. 処理の実例2 – ワンピース





4.実装上の工夫



4-1. 実装言語の選定

- 実装には Python 3.12.0 を利用(10/2 リリース)
 - 手に馴染んでいる
 - 制約がゆるいので黒魔術が使える

- かなり柔軟な定義が可能に
 - YAML に書くだけでパラメータを増やせる

ドラえもん:

枚数: 6

色: 赤

ドラえもんT:

色: 赤

タケコプター: True

- 内部実装では型ヒントと mypy をフルに活用



4-2. 役の判定について

- 存在する全役の判定を順番に行う
- 前処理 -> 条件判定 -> (通ったら)後処理
- 役の構造を YAML で指定することで高速化



4-3. 役の判定手順(1)

- 手牌から、構造の先頭の枚数抽出する

構造: 3 3 3

ド	ド	の	ド	の	の	し	し	し
ラ	ラ	び	ラ	び	び	ず	ず	ず



a	b	c
ド	ド	の
ラ	ラ	び

なかよしセット:

構造: 3 3 3

前処理:

- ...

制約:

- and

-

- 同色(a, b, c)

- a.色 == '赤'

- 同色(d, e, f)

- d.色 == '黄'

- 同色(g, h, i)

後処理:

-



4-3. 役の判定手順(2)

- 制約を順番にチェック

構造: 3 3 3

ド	ド	の	ド	の	の	し	し	し
ラ	ラ	び	ラ	び	び	ず	ず	ず



a	b	c
ド	ド	の
ラ	ラ	び

- 同色(a, b, c) [FALSE]

なかよしセット:

構造: 3 3 3

前処理:

- ...

制約:

- and

-

- 同色(a, b, c)

- a.色 == '赤'

- 同色(d, e, f)

- d.色 == '黄'

- 同色(g, h, i)

後処理:

-



4-3. 役の判定手順(3)

- 一つでも False があったら
再抽出

構造: 3 3 3

ド	ド	の	ド	の	の	し	し	し
ラ	ラ	び	ラ	び	び	ず	ず	ず



a	b	c
ド	ド	ド
ラ	ラ	ラ

なかよしセット:

構造: 3 3 3

前処理:

- ...

制約:

- and

-

- 同色(a, b, c)

- a.色 == '赤'

- 同色(d, e, f)

- d.色 == '黄'

- 同色(g, h, i)

後処理:

-



4-3. 役の判定手順(4)

- 再度制約をチェック
- NameError (未定義によるエラー)は無視

構造: 3 3 3

ド	ド	の	ド	の	の	し	し	し
ラ	ラ	び	ラ	び	び	ず	ず	ず



a	b	c
ド	ド	ド
ラ	ラ	ラ

```
- 同色(a, b, c) [TRUE]
- a.色 == '赤' [TRUE]
- 同色(d, e, f) [NameError]
- d.色 == '黄' [NameError]
- 同色(g, h, i) [NameError]
```

なかよしセット:

構造: 3 3 3

前処理:

- ...

制約:

- and

-

- 同色(a, b, c)

- a.色 == '赤'

- 同色(d, e, f)

- d.色 == '黄'

- 同色(g, h, i)

後処理:

-



4-3. 役の判定手順(5)

- NameError 込みで通ったら次の枚数を抽出
- 制約をチェック

構造: 3 3 3

ド	ド	の	ド	の	の	し	し	し
ラ	ラ	び	ラ	び	び	ず	ず	ず



a	b	c	d	e	f
ド	ド	ド	の	の	の
ラ	ラ	ラ	び	び	び

なかよしセット:

構造: 3 3 3

前処理:

- ...

制約:

- and

-

- 同色(a, b, c)

- a.色 == '赤'

- 同色(d, e, f)

- d.色 == '黄'

- 同色(g, h, i)

後処理:

-



4-3. 役の判定手順(6)

- この手順を繰り返し、全制約が True になったら役が成立

構造: 3 3 3

ド	ド	の	ド	の	の	し	し	し
ラ	ラ	び	ラ	び	び	ず	ず	ず



a	b	c	d	e	f	g	h	i
ド	ド	ド	の	の	の	し	し	し
ラ	ラ	ラ	び	び	び	ず	ず	ず

なかよしセット:

構造: 3 3 3

前処理:

- ...

制約:

- and

-

- 同色(a, b, c)

- a.色 == '赤'

- 同色(d, e, f)

- d.色 == '黄'

- 同色(g, h, i)

後処理:

-



4-4. パーツの判定について

- 基本的には bool を返す関数のように振る舞う
 - 複数の制約が入れ子になっている場合、CNF(連言標準形) に変換される
- 早期に枝切りができる

同色 a b c:

制約:

- a.色 == b.色 == c.色

特殊役 a b c:

制約:

- or

-

- 三船長(a, b, c)

- 未来の海賊王と両翼(a, b, c)

三船長 a b c:

制約:

- and

-

- len({a, b, c}) == 3

- '{a, b, c} == {..., ..., ...}'

$$(A \wedge B) \vee (C \wedge D) \Leftrightarrow (A \vee C) \wedge (A \vee D) \wedge (B \vee C) \wedge (B \vee D)$$



4-5. 前処理・後処理の重複問題

- 「判定前後に処理をする」という実装
 - 判定が複数回行われると、前処理・後処理も複数回行われてしまう
 - 役やパーツの記述が非常に柔軟なため、パーツ単位で成立をトレースするのが難しい

例パーツ a b c:

制約:

- and
- a.色 == b.色 == c.色
- or
-
- 加点のある条件(a, b, c)
- 加点のない条件(a, b, c)



4-6. 解決方法

- 役の成立が判定できたら、牌の順番が**整っている**状態で再判定
- 後処理は1度しか実行されない
 - 前処理は複数回実行されても大丈夫な内容を記述

- 整っていない状態

ド	ド	ド	の	の	し	の	し	し
ラ	ラ	ラ	び	び	ず	び	ず	ず

- ドラの同色判定に複数回通過する

- 整っている状態

ド	ド	ド	の	の	の	し	し	し
ラ	ラ	ラ	び	び	び	ず	ず	ず

- 各判定は高々一度しか通過しない



4-7. キャッシュの作成と利用

- 牌の集合を key 、その牌での判定結果を value とするキャッシュを作成
- ゲームプレイ時、牌山の先頭の数枚に対して予め判定させておくことで、ツモ時の計算を削減



5. 達成事項と課題点



5-1. 達成事項

- 各種麻雀ライクなゲームを統一的に記述できる
- 麻雀も(能力的には)記述できる
- 点数計算ができる
- 記述をもとに実際にプレイができる



5-2. 課題点

- 速度が遅い
 - 一般性を重視しているため、特定のゲームに特化した高速化が適用できない
- 実装できていないもの
 - 鳴き
 - ▶ 複数プレイヤーが別々にプレイできる環境が必要
 - 立直
 - ▶ テンパイの判定が難しい
 - ドラ
 - ▶ 一般性を持たせたまま記述できるか不明



6. 今後の展望



6. 今後の展望

- ・ オンラインで定義・プレイできる環境の提供
 - ブラウザ上でのプレイ
 - Discord 等コミュニケーションツールの bot として
- ・ 他に定義できるゲームがないか考える



7. インターンシップ感想

7. インターンシップ感想

- 特に速度面等、詰め切れていない部分が多く残ってしまった
 - アルゴリズム力の弱さを実感
- 実装力はかなり成長した
- 「ゲーム」というトピックに対しての自身のモチベーションを再確認した