

2013年7月8日現在

Computer Science Library 13

人工知能の基礎（初版）

※赤色・青色の付いた頁・行番号以外の箇所は初版第4刷までに修正済みです。

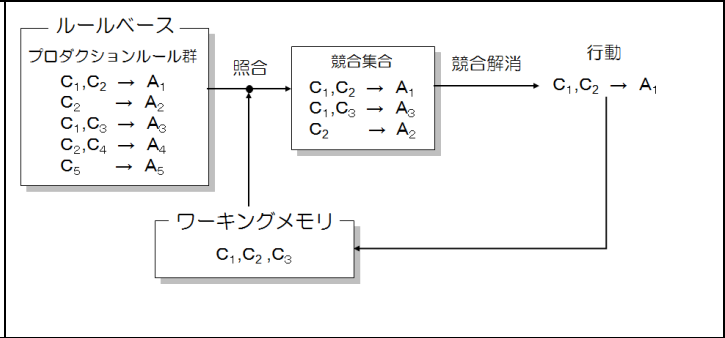
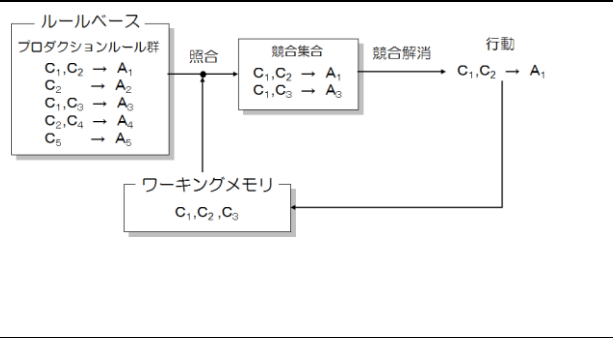
正誤表

箇所	誤	正
vi ↓ 13行	(日)	削除
p.1 ↓ 1 行	欲望のだろうか,	欲望なのだろうか,
p.1 ↑ 6 行	世界発	世界初
p.10 ↓ 10 行	carnivals	cannibals
p.13 ↑ 3行	総和は2以下であると	総和は1以上2以下であると
p.13 ↑ 1行	$ \text{宣教師}(t) - \text{宣教師}(t+1) $	$1 \leq \text{宣教師}(t) - \text{宣教師}(t+1) $
p.13 脚注	3.2.4項にて	3.2.5項にて
p.14 ↓ 9行	$3 - \text{先住民}(t) < 3 - \text{先住民}(t)$	$3 - \text{宣教師}(t) < 3 - \text{先住民}(t)$

<p>p.16 図中の表記</p>		
<p>p.20 図 3.2 図中の表記を 図 3.3、 図 3.4 に統一</p>		
<p>p.26 図 3.8 ノードのリンク位置修正</p>		
<p>p.31 ↓4行</p>	<p>・探索において評価関数で表わされる指針を持っているので、縦型探索や横型探索よりも早く解が求まる。</p>	<p>・探索したノードに隣接するノードの値をすべて調べるため(評価値を決めないと探索の方向を決定できないため)、対象とする問題によっては探索コストがそれなりにかかってしまうことがあるが、最適経路を発見できるということが大きな利点となる。</p>
<p>p.32 問題 7</p>	<p>このとき、評価関数を</p>	<p>このとき、p点における評価関数を</p>

↓ 1 行		
p.32 問題 7 ↓ 2 行	$f'(p)=g(p)+h'(p)$ として、探索を行うとする.	$f'(p)=g(p)+h'(p)$ として、ノード G を必ず通過する制約の下に探索を行うとする.
p.32 問題 7 ↓ 4 行	$f'(p)$ は Goal に到達する	$f'(p)$ は Goal (Y) に到達する
p.32 問題 7 ↓ 5 行	$g(p)$ は A 点から	$g(p)$ は Start (A) から
p.32 問題 7 ↓ 6 行	$h'(p)$ は、迷路に壁がないと仮定したときの p 点から Y までの移動距離	$h'(p)$ は、p 点が G へ到達以前では迷路に壁がないと仮定したときの p 点から G および G から Y までの移動距離. p 点が G を通過後では、迷路に壁がないと仮定したときの p 点から Y までの移動距離
p.32 問題 7 ↓ 10 行	探索の順番をノードの上に付記	探索の順番とその評価値をノードに付記
p.32 問題 7 ↓ 11 行	ただし、	ここで、
p.32 問題 7 ↓ 13 行	探索の条件として、評価値が良い方向へ進むことを基本とし、評価値が同じ場合は、より深い節点を選んで探索を進める. また、同じ深さの時はアルファベット順に探索するものとする.	探索の条件として、評価値が良いノードを選択することを基本とし、探索方向に深さを優先（縦型探索）する. また、同じ深さの時はアルファベット順に従うものとする.
p.32 問題 8	問題 8	削除
p.80 ↓ 15 行	語彙の類犠牲など	語彙の類義性など

p.94
図 9.2
競合集
合に
C2 →
A2
を追加

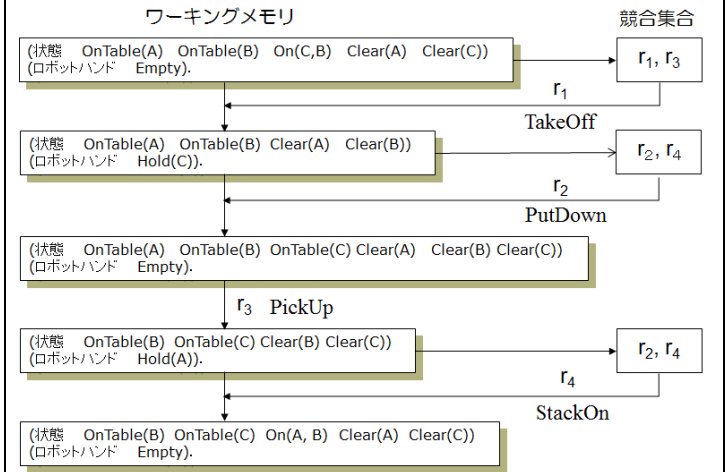
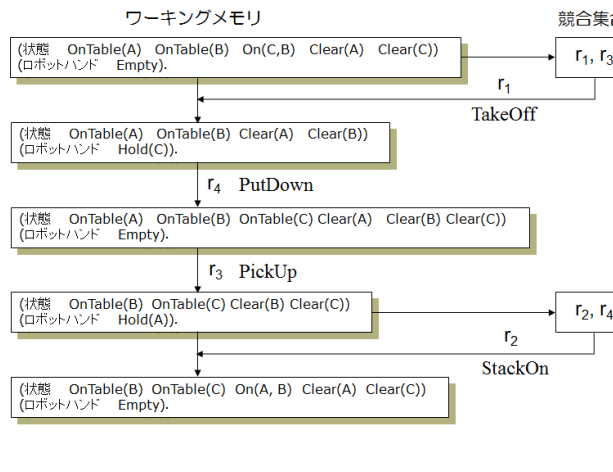


p.96
表 9.1
(詳細
を下に
示す)

ルール名称&ID	操作内容
start	初期状態の設定。(ワーキングメモリ内へファクトの追加)
r0	start ⇒ (make (状態 OnTable(A) OnTable(B) On(C,B) Clear(A) Clear(C)) (make (ロボットハンド Empty)))
TakeOff(X,Y)	Xを掴んでYの上から取り除く。 r1 (状態 Clear(X) On(X, Y), (ロボットハンド Empty) ⇒ (delete 1 Clear(X) On(X, Y), (add 1 Clear(Y)), (modify 2 Hold(X)).
StackOn(X,Y)	XをYの上に積み上げる。 r2 (状態 Clear(Y), (ロボットハンド Hold(X)) ⇒ (delete 1 Clear(Y), (add 1 Clear(X) On(X,Y)), (modify 2 Empty).
PickUp(X)	テーブルの上のXを掴み上げる。 r3 (状態 Clear(X) OnTable(X), (ロボットハンド Empty) ⇒ (delete 1 Clear(X) OnTable(X), (modify 2 Hold(X)).
PutDown(x)	テーブルにXを置く。 r4 (ロボットハンド Hold(X)) ⇒ (add 1 Clear(X) OnTable(X), (modify 2 Empty).

ルール名称&ID	操作内容
start	初期状態の設定。(ワーキングメモリ内へファクトの追加)
r0	start ⇒ (make (状態 OnTable(A) OnTable(B) On(C,B) Clear(A) Clear(C)) (make (ロボットハンド Empty)))
TakeOff(X,Y)	Xを掴んでYの上から取り除く。 r1 (状態 Clear(X) On(X, Y), (ロボットハンド Empty) ⇒ (delete 1 Clear(X) On(X, Y), (add 1 Clear(Y)), (modify 2 Hold(X)).
PutDown(x)	テーブルにXを置く。 r2 (ロボットハンド Hold(X)) ⇒ (add 1 Clear(X) OnTable(X), (modify 2 Empty).
PickUp(X)	テーブルの上のXを掴み上げる。 r3 (状態 Clear(X) OnTable(X), (ロボットハンド Empty) ⇒ (delete 1 Clear(X) OnTable(X), (modify 2 Hold(X)).
StackOn(X,Y)	XをYの上に積み上げる。 r4 (状態 Clear(Y), (ロボットハンド Hold(X)) ⇒ (delete 1 Clear(Y), (add 1 Clear(X) On(X,Y)), (modify 2 Empty).

p.97
図 9.5
(詳細
を下に
示す)



p.117
↓ 12
行

$$P(\text{食事}=1 | \text{イタリアン}=0) = 1.74 \times 0.8 \times 0.44 = 0.61$$

$$P(\text{食事}=0 | \text{イタリアン}=1) = 1.74 \times 0.8 \times 0.44 = 0.61$$

p.-135
(11.7)
(式の
右にあ
るQの
肩に着

$$Q^*(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1} \in A} Q^*(s_{t+1}, a_{t+1})$$

$$Q^*(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1} \in A} Q^*(s_{t+1}, a_{t+1})$$

<p>いた た「*」 を削除)</p>		
<p>p.144 ↑ 7 行</p>	<p>ここで、$y_{j,s}$は教師信号</p>	<p>ここで、y_jは教師信号</p>
<p>p.144 ↑ 7 行</p>	<p>(「希望出力」ともいう)であり、</p>	<p>(「希望出力」ともいう)である。</p>
<p>p.144 ↑ 7 行</p>	<p>Sはある入力信号ベクトルである。</p>	<p>削除</p>
<p>p.144 ↑ 8 行</p>	$E = \frac{1}{2} \sum_{j,s} (X_{j,s} - y_{j,s})^2$	$E = \frac{1}{2} \sum_j (X_j - y_j)^2$
<p>p.145 ↓ 11行</p>	$\frac{\partial E}{\partial w_{ij}} = \sum_j \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial u_j} \frac{\partial u_j}{\partial w_{ij}}$	$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial u_j} \frac{\partial u_j}{\partial w_{ij}}$
<p>p.145 ↓ 16行</p>	$\Delta w_{ij} = -\eta \sum_j \frac{\partial E}{\partial x_j} f'(u_j) x_i$	$\Delta w_{ij} = -\eta \frac{\partial E}{\partial x_j} f'(u_j) x_i$
<p>p.146 ↓ 8 行</p>	$\Delta w_{ij} = -\eta \sum_j (x_j - y_j) f'(u_j) x_i$	$\Delta w_{ij} = -\eta (x_j - y_j) f'(u_j) x_i$
<p>p.148 (12.20) ↓ 1 行</p>	<p>ユニット i の状態は相互結合されているユニット j</p>	<p>ユニット j の状態は相互結合されているユニット i</p>
<p>p.148 (12.20) ↓ 2行</p>	<p>ユニット i が</p>	<p>ユニット j が</p>

<p>p.195 第3章 問題7 (b)</p>		
<p>p.195 第3章 問題8</p>	<p>問題8 迷路の一つの経路が深く入り組んでいる場合は、縦型探索は探索に時間がかかり、迷路の規模が大きくなると横型探索は時間がかかってしまう。それに比べて、A*アルゴリズムは Goal までの予測コストが小さい方へ進むことができるため、探索木の余計なノードを探索することがなくなり、探索時間が短くなる。</p>	<p>削除</p>
<p>p.196 第5章 問題 1(a)</p>	$(\neg p \vee q) \wedge (p \vee \neg q) \vee (p \vee \neg r)$	$(\neg p \vee q) \wedge (p \vee \neg q) \wedge (p \vee \neg r)$
<p>p.196 第6章 問題 2(a)</p>	$\forall x_1 \forall y_1 ([\neg P(x_1) \vee Q(x_1, f(x_1))] \wedge [\neg P(a) \vee R(g(y_1))])$ <p>節集合表現： $\{\neg P(x_1) \vee Q(x_1, f(x_1)), \neg P(a) \vee R(g(y_1))\}$</p>	$\forall x_1 \forall y_1 ([\neg P(x_1) \vee Q(x_1, f(x_1))] \wedge [\neg P(a) \vee \neg R(g(y_1))])$ <p>節集合表現： $\{\neg P(x_1) \vee Q(x_1, f(x_1)), \neg P(a) \vee \neg R(g(y_1))\}$</p>
<p>p.196 第6章 問題 2(b)</p>	$\forall x_1 \forall y_1 ([\neg P(x_1, f(x_1)) \vee Q(x_1)] \wedge [\neg P(a, y_1) \vee R(g(y_1))])$ <p>節集合表現： $\{\neg P(x_1, f(x_1)) \vee Q(x_1), \neg P(a, y_1) \vee R(g(y_1))\}$</p>	$\forall x_1 \forall y_1 ([\neg P(x_1, f(x_1)) \vee Q(x_1)] \wedge [\neg P(a, y_1) \vee \neg R(g(y_1))])$ <p>節集合表現： $\{\neg P(x_1, f(x_1)) \vee Q(x_1), \neg P(a, y_1) \vee \neg R(g(y_1))\}$</p>
<p>p.196 第6章 問題 2(c)</p>	<p>節集合表現： $\{\neg P(x_1, f(x_1)) \vee Q(x_1), \neg P(a, y_1) \wedge R(g(y_1))\}$</p>	<p>節集合表現： $\{\neg P(x_1, f(x_1)) \vee Q(x_1), \neg P(a, y_1), \neg R(g(y_1))\}$</p>

表 9.1 (誤)

ルール名称&ID	操作内容
start	初期状態の設定. (ワーキングメモリ内へファクトの追加)
r_0	start ⇒ (make (状態 OnTable(A) OnTable(B) On(C,B) Clear(A) Clear(C)) (make (ロボットハンド Empty))
TakeOff(X,Y)	X を掴んで Y の上から取り除く.
r_1	(状態 Clear(X) On(X, Y)), (ロボットハンド Empty) ⇒ (delete 1 Clear(X) On(X, Y)), (add 1 Clear(Y)), (modify 2 Hold(X)).
StackOn(X,Y)	X を Y の上に積み上げる.
r_2	(状態 Clear(Y)), (ロボットハンド Hold(X)) ⇒ (delete 1 Clear(Y)), (add 1 Clear(X) On(X,Y)), (modify 2 Empty).
PickUp(X)	テーブルの上の X を掴み上げる.
r_3	(状態 Clear(X) OnTable(X)), (ロボットハンド Empty) ⇒ (delete 1 Clear(X) OnTable(X)), (modify 2 Hold(X)).
PutDown(x)	テーブルに X を置く.
r_4	(ロボットハンド Hold(X)) ⇒ (add 1 Clear(X) OnTable(X)), (modify 2 Empty).

表 9.1 (正)

ルール名称&ID	操作内容
start	初期状態の設定. (ワーキングメモリ内へファクトの追加)
r_0	start \Rightarrow (make (状態 OnTable(A) OnTable(B) On(C,B) Clear(A) Clear(C)) (make (ロボットハンド Empty))
TakeOff(X,Y)	X を掴んで Y の上から取り除く.
r_1	(状態 Clear(X) On(X, Y)), (ロボットハンド Empty) \Rightarrow (delete 1 Clear(X) On(X, Y)), (add 1 Clear(Y)), (modify 2 Hold(X)).
PutDown(x)	テーブルに X を置く.
r_2	(ロボットハンド Hold(X)) \Rightarrow (add 1 Clear(X) OnTable(X)), (modify 2 Empty).
PickUp(X)	テーブルの上の X を掴み上げる.
r_3	(状態 Clear(X) OnTable(X)), (ロボットハンド Empty) \Rightarrow (delete 1 Clear(X) OnTable(X)), (modify 2 Hold(X)).
StackOn(X,Y)	X を Y の上に積み上げる.
r_4	(状態 Clear(Y)), (ロボットハンド Hold(X)) \Rightarrow (delete 1 Clear(Y)), (add 1 Clear(X) On(X,Y)), (modify 2 Empty).

図 9.5 (誤)

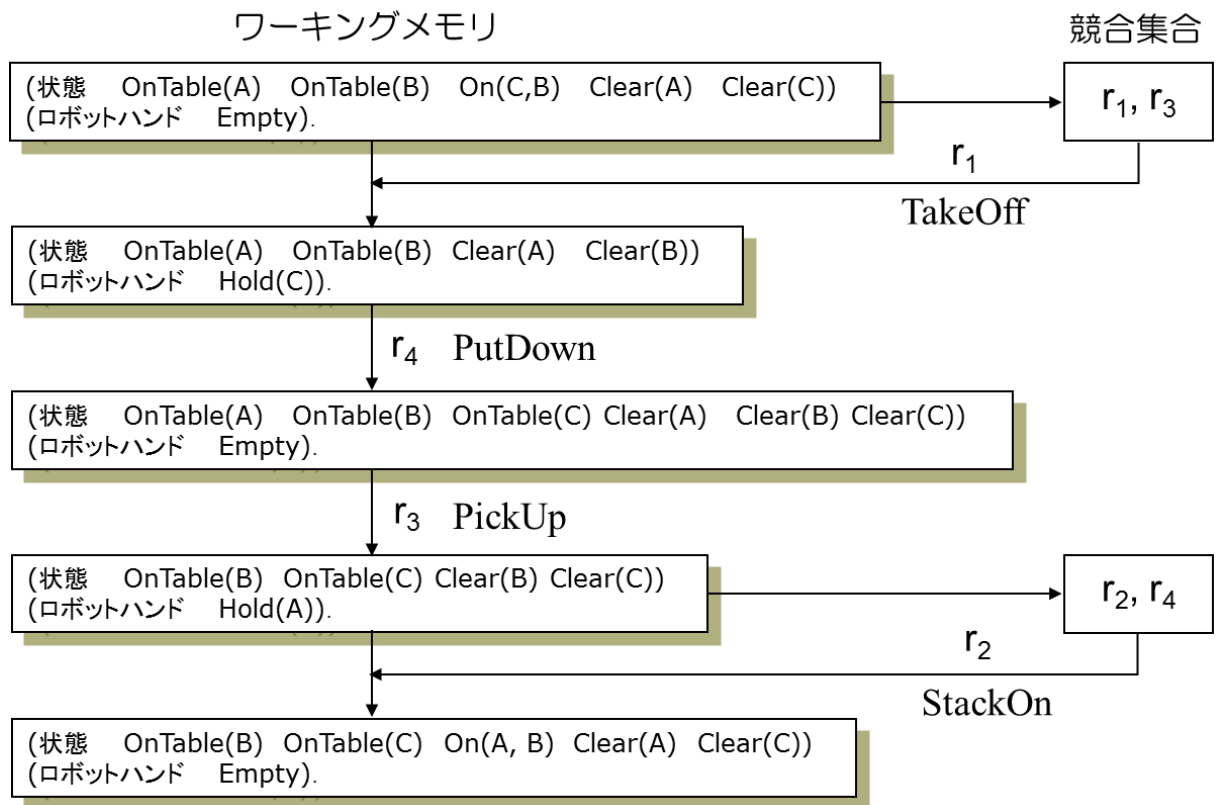


図 9.5 (正)

