

۱- کسی که ما تعدادی رفتار را با ذکر مثال تشریح کنیم

نیاز به قدری های علمی در خصوص مفاهیم مادی دارد
 قدری قدری - هر قدری که از طریق سیستم ها اندازه گیری در ارتباط است به اندازه تحقیق و در حد فرد
 مقابل و اتفاقا اتفاق است با این روابط

۲- هدف از تعریف عاملان در سیستم و چه آورده ای در پی خواهد داشت

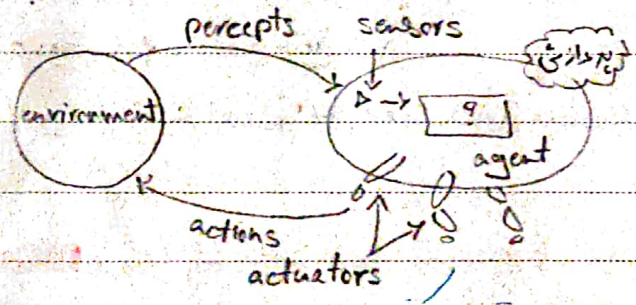
مختص به برنامه نویسی مبتنی بر سیستم است. تعریف عاملان به منظور رفتار عاملان در سیستم است. این رفتارها باعث انجام عمل در سیستم می شود و این باعث می شود که سیستم بتواند در محیط خود عمل کند. ① محدودیت مبتنی بر
 ② نیاز به قدری های پیشرفته علمی نیست

۳- اهداف اصلی و وظایف عامل را با رسم شکل و تامل در سیستم بررسی کنیم

شماره ۴ در اینجا مشخصه مادی از محیط / محله - به انجام اعمال در محیط
 عامل و وظیفه را در زمینه در اینجا مادی و در اینجا مادی از اعمال تعریف می نماید

$$f: p^* \rightarrow A$$
 عمل ما
 Action = Agent (percept)

عاملی می تواند از اعمال خود در محیط را درک کند اما تاثیر آن در محیط را می بیند. همیشه قابل چینی نیست



۴- PEAS برای رفتار تعریف می شود و توضیح می دهیم

- P = توانایی کنترل - هزینه یا پیچیدگی - محله در دست - یا نرخ در دست به عامل محیطی
- E = سطح لذت - محیط - محیط یا هزینه - محیط یا جاذبه های متفاوت - شرایط جوی مختلف
- A = حرکت - چرخش - ایستادن - صاف یا چینی - بلند کردن - انداختن
- S = در بین - نامیده یا ب - موقعیت یا ب

۵- هدف تعریف

- P = نیاز به توانایی - هزینه یا پیچیدگی - حرارت در دست - مقصد در دست
- E = بهینه سازی - بهینه سازی - بهینه سازی یا ب - آب و هوا
- A = حرکت - ایستادن - بهینه سازی یا ب
- S = در بین - نامیده یا ب - تحقیق خطوط

اینجا مثالی که زیر هر عامل مبتنی بر جدول است می توانیم به عنوان مثال یک برنامه بنویسیم
 و همچنین می توانیم خود را به هر کدام از دو شیء درگیر

Function TABLE-DRIVEN-AGENT(percept) returns an action

static: percepts, a sequence initially empty

table, a table of actions, indexed by percept sequence

append percept to the end of percepts

actions = LOOKUP(percepts, table)

return action

اینجا دقیقاً آنچه فعلت را می بیند که به یک یا راست برود یا چپ یا حرکت کند یا حالت به کار می آید اینجا

10 ی. دود

۹- در اینجا چاره دیگری را به خود می بینیم که در مرحله اول به خود می بینیم که

حالات ۸-۴ حالت مختلف

حالات شروع ۴-۶ حرکت از حالات

اعمال ۲-۴ به چپ، راست، حرکت، هیچ کار

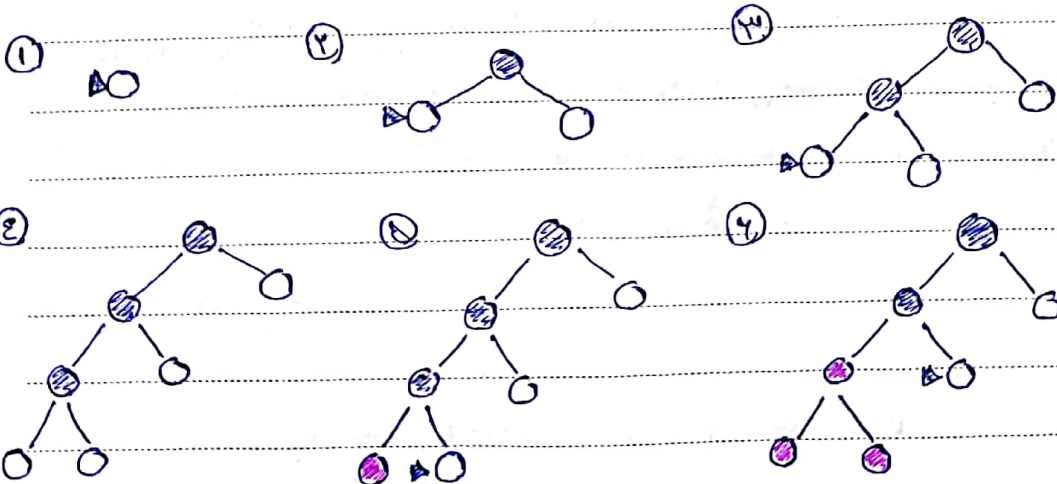
آزمایش خود را ۴-۶ حالات ۸، ۷، ۶

15

فهرست میوه به تعداد اعمال انجام شده تا رسیدن به هدف

۱۰- در اینجا می بینیم که در مرحله اول به خود می بینیم که در مرحله اول به خود می بینیم که

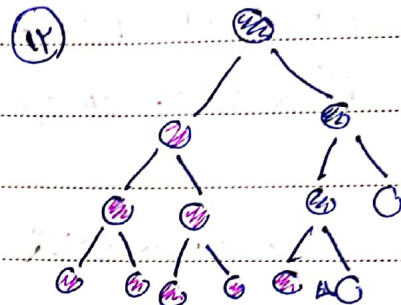
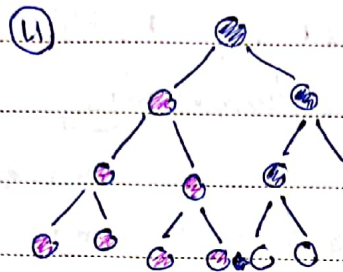
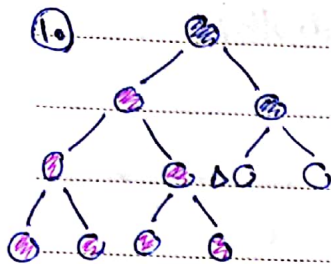
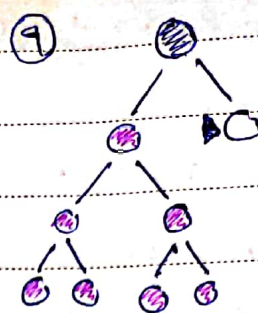
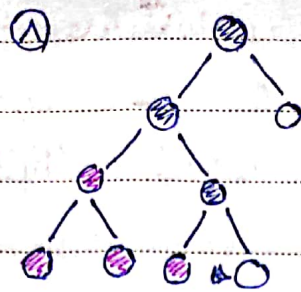
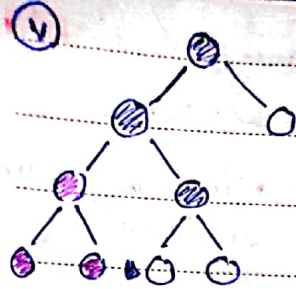
بیان کنید



20

25

(صفحه ۶ تا ۷ جزوه) فصل ۳



کامل بودن P ، غیر محدودیت فضای حالت محدود باشد و حالت تکرار وجود نداشته باشد

جبهه بودن P ، غیر محدودیت فضای حالت محدود باشد

پسندگی زمانی P ، اگر m فضای بزرگتر از n باشد و مدلها بدتر است در بسیاری مسائل مربوط به جستجوی BF است

پسندگی حافظه P ، در زمان عملکرد حافظه آزاد می شود

15

۱. فرض بر این است که جستجو در فضای مسئله که زیر درختی گسترده است و گسترده تر از آن است و گسترده تر از آن است و گسترده تر از آن است

راه حل یا اشغال می شود

فرایز زیر درختی انجام بده

loop do

if $\text{EMPTY?}(\text{fringe})$ then return failure

node $\leftarrow \text{REMOVE-FIRST}(\text{fringe}) \rightarrow$

if $\text{GOAL-TEST}[\text{node}]$ then

return $\text{SOLUTION}(\text{node})$

else

این تابع مجدداً از نو، را برمی گرداند. گسترده تر از آن است

$\text{fringe} \leftarrow \text{INSERT-ALL}(\text{EXPAND}(\text{node}, \text{problem}), \text{fringe})$

Remove-First

Expand

G

Insert-All

مسئله: در زیر مدیوفا به دنبال جستجوی ناگهانی می باشد و از میانجی که نام جستجوهای دیده شده برده است عبارت زیر را شکل داده و شرح دهید.

function ITERATIVE_DEEPENING_SEARCH(problem) return a solution or failure

Inputs: problem

for depth ← 0 to ∞ do

result ← DEPTH-LIMITED_SEARCH(problem, depth)

if result ≠ cutoff then return result

جستجوی عمیق تکرار شونده

(مسئله ۷۸، ۷۹، ۸۱ جز ۳ فصل ۳)

نمود ۱۰

۱- انواع جستجوهای ناگهانی مدیوفا را به تفصیل با ۴ مثال مدیوفا به اعتبار شرح دهید.

۱) جستجوی سطحی - به کامل است به شرطی که جواب بهینه در عمق d قابل دسترسی باشد و حداکثر انتخاب b محدود باشد / بهینه است به شرطی که مسیر صاف باشد / پیچیدگی زمانی و حافظه $O(b^d)$

۲) جستجوی عمیق محدود - به کامل است به شرطی که جواب بهینه در عمق d قابل دسترسی باشد و عمق بهینه نامتناهی داشته باشد / بهینه است به شرطی که مسیر صاف باشد / پیچیدگی زمانی و حافظه $O(b^d)$

۳) جستجوی عمیق - به کامل نیست بلکه فضای حالت محدود باشد و حلقه تکرار وجود نداشته باشد / بهینه نیست زیرا کامل نیست / پیچیدگی زمانی $O(b^m)$ / پیچیدگی حافظه $O(b^m)$

۴) جستجوی عمیق محدود - اگر L غیر کامل است، اگر L کامل اما غیر بهینه است، اگر L کامل و بهینه است / پیچیدگی زمانی $O(b^L)$ / پیچیدگی حافظه $O(b^L)$

۵) جستجوی عمیق تکراری - به کامل است به شرطی که حلقه تکرار وجود نداشته باشد و بهینه است اگر بهینه نامتناهی باشد / بهینه است به شرطی که مسیر صاف باشد / پیچیدگی زمانی $O(b^d)$ / پیچیدگی حافظه $O(b^d)$

۶) جستجوی دو طرفه - به کامل است به شرطی که از جستجوی سطحی استفاده شود / بهینه است به شرطی که از جستجوی سطحی استفاده شود / پیچیدگی زمانی $O(b^{\frac{d}{2}})$ / پیچیدگی حافظه $O(b^{\frac{d}{2}})$

۱۱- جستجوی A^* با با توجه به مدل h SLN با جستجوی در میان با رسم مدیت به طور کامل قد شرح داده و مقایسه با سایر الگوریتم ها ذکر کنید. جستجوی در میان - به کامل نیست (حلقه تکرار) / بهینه نیست (کامل نیست)

مرتبه زمانی $O(b^m)$ / مرتبه حافظه $O(b^m)$

جستجوی A^* - به کامل است / بهینه است / مرتبه زمانی به مرتبه تابع مسئله جستجو دارد / مرتبه حافظه به مرتبه تابع مسئله جستجو دارد

۱۲. الگوریتم زیر را برای حل مسئله ۱۱ با نام درخت جستجو کنید.

Function RECURSIVE-BEST-FIRST-SEARCH(problem) return a solution or failure

return RFBFS(problem, -NODE(INITIAL-STATE[problem]), ∞)

Function RFBFS(problem, node, f-limit) return a solution or failure and a new f-cost limit

5 if GOAL-TEST[problem](STATE[node]) then return node

successors \leftarrow EXPAND(node, problem)

if successors is empty then return failure, ∞

for each s in successors do

$f[s] \leftarrow \max(g[s] + h[s], f[node])$

10 repeat

best \leftarrow the lowest f-value node in successors

if $f[best] > f_limit$ then return failure, $f[best]$

alternative \leftarrow the second lowest f-value among successors

result, $f[best] \leftarrow$ RFBFS(problem, best, min(f-limit, alternative))

15 if result \neq failure then return result

الگوریتم RFBFS

۱. بهترین گره به عنوان گره فعلی انتخاب شود. ۲. آن مقدار هزینه برای آن گره از جانشین آن

محاسبه شود. ۳. در بین عقبات مقدار f برای هر یک از جانشینان

محاسبه شود.

(مقدار هزینه) (مقدار فاصله)

۱۳- چند نوع تابع تعریف کنید. برای مثال از این جدول برای مثال استفاده کنید.

| | | |
|---|---|---|
| ۷ | ۲ | ۴ |
| ۵ | | ۶ |
| ۸ | ۳ | ۱ |

| | | |
|---|---|---|
| | ۱ | ۲ |
| ۳ | ۴ | ۵ |
| ۶ | ۷ | ۸ |

۱- روند قابل قبول - ۲- برای یازده تابعی
 ۱) $h_1(x)$ عددی که جای خالی دارد است
 ۲) $h_2(x)$ مجموع حاصله تا به آخر

5

| | | |
|---|---|---|
| * | ۲ | ۴ |
| * | | * |
| * | ۳ | ۱ |

| | | |
|---|---|---|
| | ۱ | ۲ |
| ۳ | ۴ | * |
| * | * | * |

۱۴- سه راه حل جهت ابداع تابع تعریف کنید و شرح دهید
 ۱- از طریق چند کد حلقه از مسئله - ۲
 ۲- از طریق چند ساد شده از مسئله - ۲
 ۳- h_1 هر کد می تواند در جای مسئله شود

10

h_2 هر کد می تواند به هر خانه هم به منتقل شود
 ABSolver هزینه راه حل برای ملاب رویل را تخمین می زند

۳- از طریق یادگیری از تجربه - ۲

حل مسئله زیادی از ساله (تجربه)

15

۱۵- انواع جستجوی محلی را نام ببرید و اینکه هر یک را بیان کنید

تجربه زوری - ۲- به طور متناوب در جهت جستجو حرکت می کند. زمانی که به قله میرسد متوقف می شود. به آن قله
 کوه های بزرگ قدم می کند. به همین دلیل به الگوریتم جستجوی هیل کلایم هم مشهور است. در هر مرحله جستجو از یک کوه
 به قله دیگری می رود تا به قله ای که به صورت تصادفی انتخاب می شود

SA - ۲- انتخاب از گزیده کردن به بهترین حالت محلی با اجازه دادن به انجام حرکت های بدی (نامناسب)
 که در همین گذشت زمان احتمال رسیدن به آن کاهش می یابد. جستجو الگوریتم به عدم متناهی می رود
 به تدریج محلی - ۲- از حالت شروع به جای ۱ حالت شروع به بدی می شود. است الگوریتم از عدم
 تدریج گامی به خود دارد. تفاوت با تدریج بدی با شروع مجدد تصادفی این است که اطلاعات به اشتراک
 گذار نمی شوند

20

تدریج

25

۱۶- الگوریتم زیر را شرح داده و انواع آن را نام ببرید و برای آن بنویسید.

Function HILL-CLIMBING (problem) return a state that is a local maximum

input: problem, a problem

local variables: current, a node.

neighbor, a node

current ← MAKE-NODE (INITIAL-STATE [problem])

loop do

neighbor ← a highest valued successor of current

if VALUE [neighbor] > VALUE [current] then return STATE [current]

current ← neighbor

به طور متناوب در جهت بهبود حرکت می کنند زمانی که به قله برسند متوقف می شوند و به آن پیکان اشاره می کنند.

مثلاً فرض کنید در مسیر رسیدن به الگوریتم جستجوی محلی در یک فضای وسیع می شویم و در صورتی که بین آن پیکان جستجو می وجود داشته باشد، به جهت آن پیکان حرکت می کنیم و انتخاب می کنند.

تیم فردی غیر واقعی - که در بین حرکت های روبه بالا و پایین به صورت تصادفی انتخاب شود البته احتمال انتخاب با بیشترین متناوب است.

تیم فردی با انتخاب اولی که پیدا می شود - که در میان حالت های مختلف یک کوه به جهت بالا بردن می رود.

تیم فردی تصادفی - که از حالت شروع مجدد تصادفی تا رسیدن به پیکان مجدداً شروع می کند تا به حادینه بخورد.