هوش مصنوعي

فصل سوم

حل مسئله با جستجو

فهرست

- عاملهای حل مسئله
- اندازه گیری کارایي حل مسئله
- جستجوی ناآگاهانه
- اجتناب از حالتهای تکراری
- جستجو با اطلاعات ناقص

عاملهای حل مسئله

- عاملهای حل مسئله یک نوع از عاملهای هدف گرا است.
- عاملهای حل مسئله با یافتن دنباله ای از فعالیت هایی که منجر به حالتهای مطلوب می شوند تصمیم می گیرند که چه کاری باید انجام دهند.

- **الگوریتم های آگاهانه و ناآگاهانه**

اگر تصمیم گیری عامل در حل مسئله مبتنی بر آگاهی و شواهد نزدیکی به جواب باشد ، این رهیافت حل مسئله را جستجوی آگاهانه می گویند و در غیر اینصورت جستجوی ناآگاهانه می گویند.

چهار گام اساسی برای حل مسائل

- **فرموله کردن هدف:** وضعیتهای مطلوب نهایي کدامند؟
- **فرموله کردن مسئله:** چه فعالیتها و وضعیتهایي برای رسیدن به هدف موجود است؟
- **جستجو:** انتخاب بهترین دنباله از فعالیتهایي که منجر به حالاتی با مقدار شناخته شده میشود. عاملی که با چند گزینه فوری و مقداری ناشناخته مواجه است ابتدا باید دنباله های ممکن از فعالیتهایی را که منجر به حالتهایی با مقدار شناخته شده می شوندرا بررسی کند و سپس بهترین دنباله را انتخاب کند.
- **اجرا:** وقتی دنباله فعالیت مطلوب توسط جستجو پیدا شد، فعالیتهای پیشنهادی آن میتواند اجرا شود.

شبه کد عاملهای ساده حل مسئله

Function SIMPLE-PROBLEM-SOLVING-AGENT(percept)returns an action

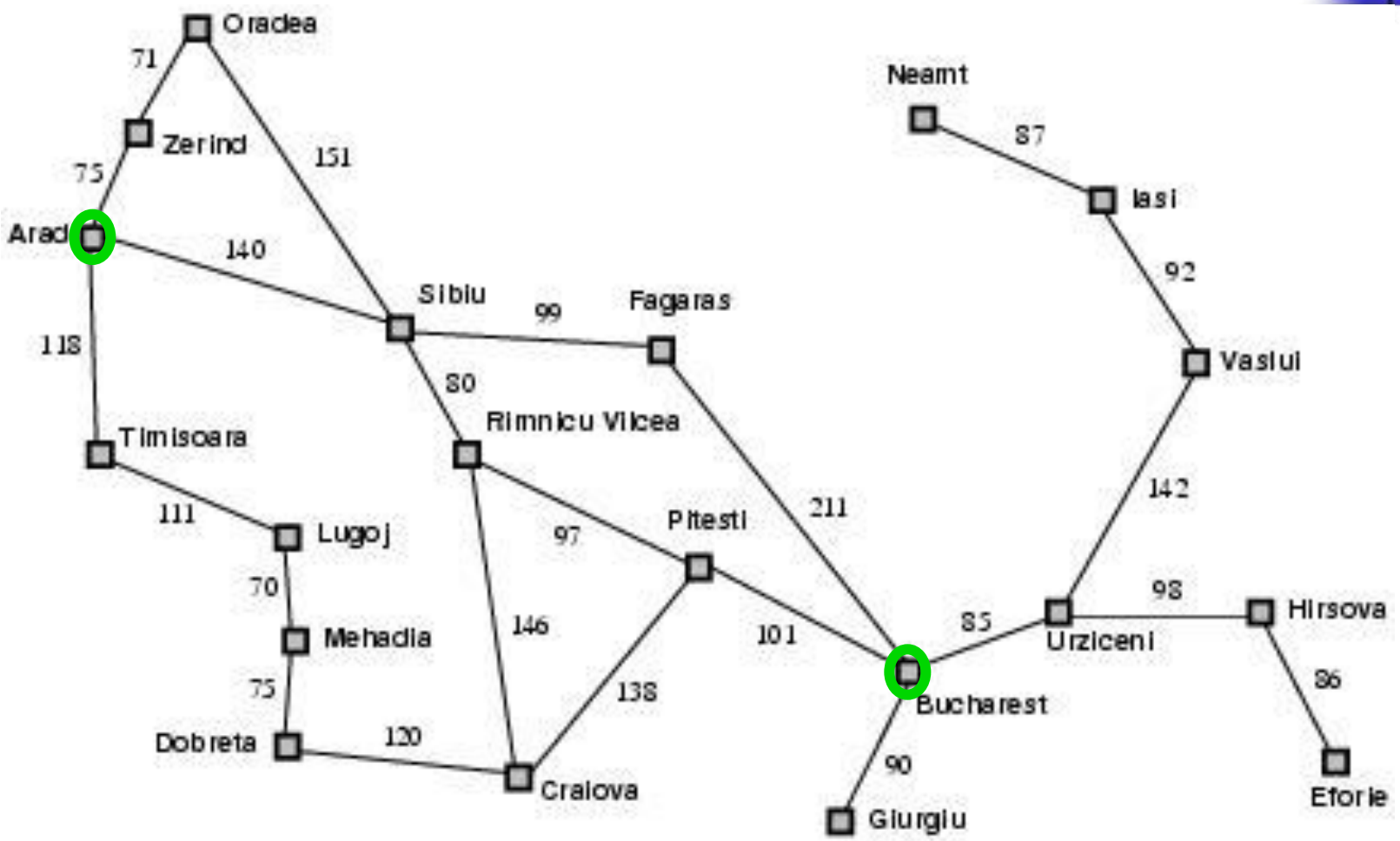
- Inputs:percept,a percept
- Static:seq,an action sequence,initially empty
- state,some descripetion of the current world state
- goal,a goal,initially null
- problem,a problem formulation

- State ← UPDATE-STATE(state,percept)
- If seq is empty then do
- goal ← FORMULATE-GOAL(state)
- problem ← FORMULATE-PROBLEM(state,goal)
- SEQ ← SEARCH(problem)
- action ← FIRST()seq)
- Seq ← REST(seq)
- return action

مفروضات شبه کد عاملهای ساده حل مسئله

- در طراحی عامل ارائه شده، فرموله کردن و حل مسئله بدون توجه به تغییرات در محیط صورت می گیرد پس فرض بر این است که محیط ایستا است.
- فرض می شود حالت اولیه مشخص است اگر محیط قابل مشاهده باشد، دانستن حالت اولیه ساده تر است.
- فرض می شود محیط گسسته است.
- فرض می شود محیط قطعی است.

حل مسئله با جستجو(مسیریابی در نقشه رومانی)



حل مسئله با جستجو(مسیریابی در نقشه رومانی)

- صورت مسأله: رفتن از آراد به بخارست
- فرموله کردن هدف: رسیدن به بخارست
- فرموله کردن مسئله:
  - وضعیتها: شهرهای مختلف
  - فعالیتها: حرکت بین شهرها
- جستجو: دنباله ای از شهرها مثل:آراد، سیبیو، فاگارس، بخارست
- این جستجو با توجه به کم هزینه ترین مسیر انتخاب میشود

مسئله و راه حل های خوش تعریف

1. حالت اولیه: حالتی که عامل از آن شروع میکند.
   - در مثال رومانی: شهر آراد in(Arad)

2. تابع جانشین: توصیفي از فعالیتهای ممکن که برای عامل مهیا است.

متداولترین فرمول بندی از **تابع جانشین** استفاده می کند.

با توجه به حالت خاصی مثل x ،تابع successor(x) مجموعه ای از <جانشین و فعالیت> را بصورت زوج مرتب بر میگرداند که هر فعالیت ، یک فعالیت معتبر در حالت x و هر جانشین ، حالتی است که با انجام آن فعالیت از طریق x به آن می رسیم.

- در مثال رومانی:< GO(Sibiu) , in(Sibiu) >

- با توجه به حالت اولیه و تابع جانشین ، فضای حالت مسئله را می توان تعریف کرد
- فضای حالت : مجموعه ای از حالتهاست که از حالت اولیه می توان به آنها

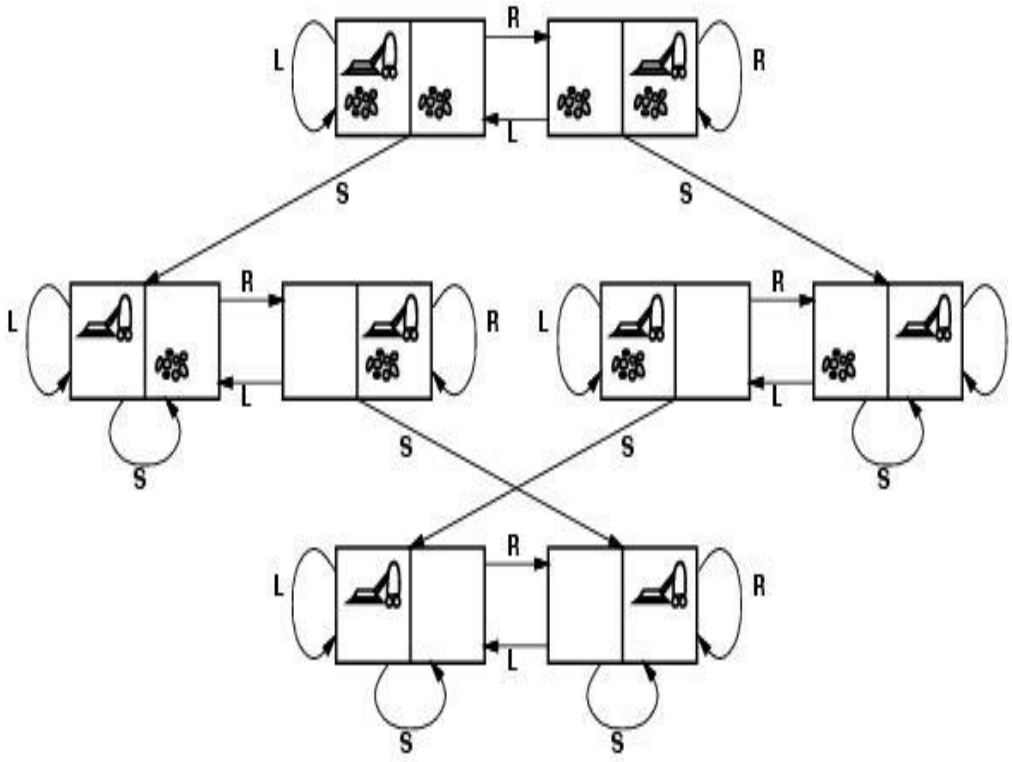
مسئله و راه حل های خوش تعریف(ادامه)

3. آزمون هدف: تعیین میکند که آیا حالت خاصی، حالت هدف است یا خیر
   - هدف صریح: در مثال رومانی، رسیدن به بخارست
   - هدف انتزاعی: در مثال شطرنج، رسیدن به حالت کیش و مات

4. تابع هزینه ی مسیر: برای هر مسیر یک هزینه عددی در نظر می گیرد.

- مسیر: دنباله ای از حالتها که دنباله ای از فعالیتها را به هم متصل میکند.
  - در مثال رومانی: Arad, Sibiu, Fagaras یک مسیر است
- **راه حل مسئله مسیری از حالت اولیه به حالت هدف است**
- **راه حل بهینه کمترین هزینه مسیر را دارد**

حل مسئله با جستجو(مثال جارو برقی)



حالتها: دو مکان که هر یک ممکن است کثیف یا تمیز باشند.لذا 8 = 2^3حالت در این جهان وجود دارد

حالت اولیه: هر حالتی میتواند به عنوان حالت اولیه طراحی شود

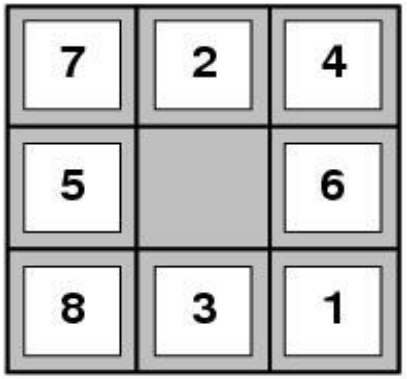
تابع جانشین: حالتهای معتبر از سه عملیات: راست،چپ، مکش(مکش,چپ,راست)

آزمون هدف: بررسی می کند که آیا تمام مربعها تمیز هستند یا خیر

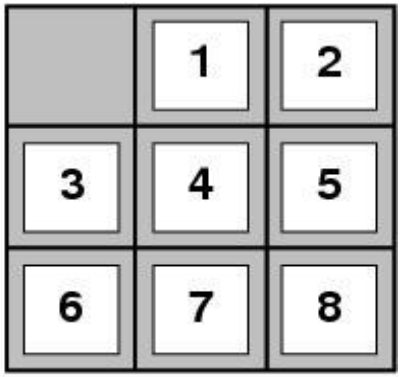
هزینه مسیر: تعداد مراحل در مسیر

(در واقع هزینه ی هر مرحله 1 است)

حل مسئله با جستجو(مثال معمای 8)



Start State



Goal State

حالتها: مکان هر هشت خانه شماره دار و خانه خالی در یکي از 9 خانه

حالت اولیه: هر حالتي را میتوان به عنوان حالت اولیه در نظر گرفت

تابع جانشین: حالتهای معتبر از چهار عمل را تولید می کند

(انتقال خانه خالی به چپ، راست، بالا یا پایین)

آزمون هدف: بررسی میکند که آیا اعداد به ترتیب چیده شده اند (طبق شکل روبرو) یا خیر

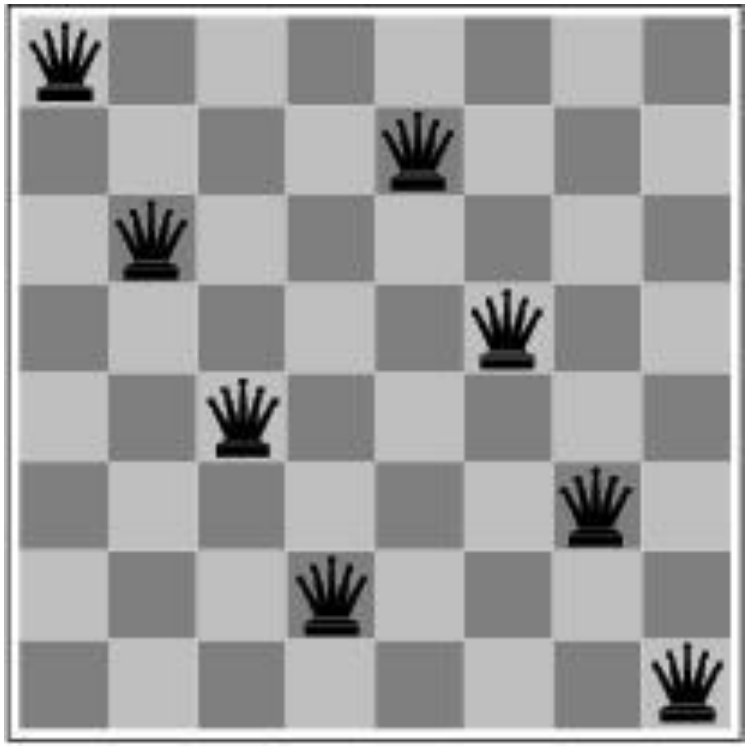
هزینه مسیر: برابر با تعداد مراحل در مسیر

(هزینه هر مرحله 1 است)

حل مسئله با جستجو(مثال معمای 8-ادامه)

- معمای 8 به خانوادهی معماهای جابجایی بلوک تعلق دارد که برای آزمون الگوریتم های جستجوی جدید در AI بکار می رود.
- این رده از مسائل را **NP** کامل می گویند.
- معمای 8 دارای!9 حالت قابل رسیدن است.
- معمای 15(صفحه ی4*4) تقریبا دارای 3/1 تریلیون حالت است اما در چند میلی ثانیه حل می شوند.
- معمای 24 (صفحه ی 5*5) تقریبا 10^25حالت دارد و حل نمونه های آن با بهترین الگوریتم ها و ماشین های فعلی دشوار است.

حل مسئله با جستجو(مثال 8 وزیر)



فرمول بندی افزایشي

حالتها: هر ترتيبي از 0 تا 8 وزير در صفحه، يک حالت است

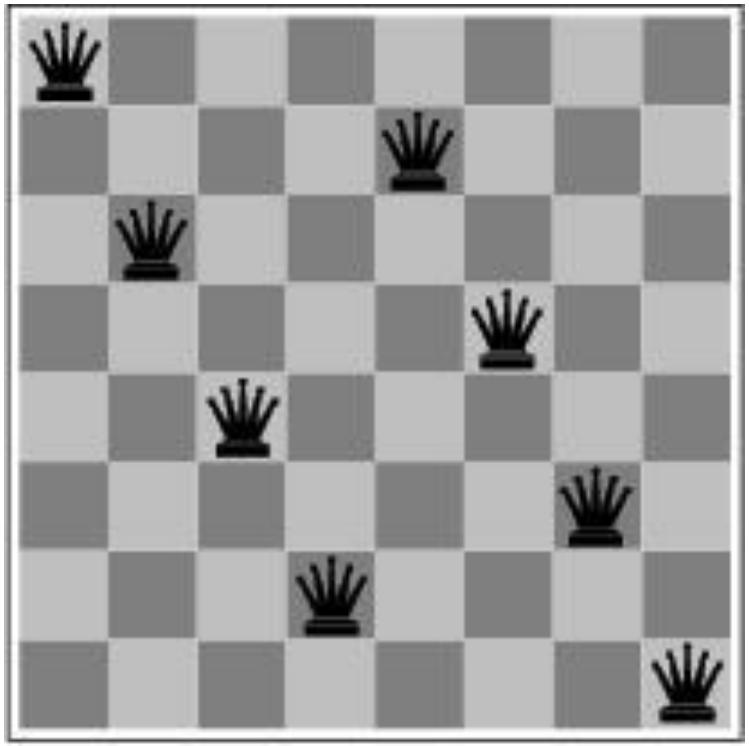
حالت اولیه: هیچ وزیری در صفحه نیست

تابع جانشین: وزیری را به خانه خالی اضافه میکند

آزمون هدف: 8وزیر در صفحه وجود دارند و هیچ کدام به یکدیگر گارد نمیگیرند

در این فرمول بندی باید 3*10^14 دنباله ممکن بررسی میشود

حل مسئله با جستجو(مثال 8 وزیر-ادامه)



فرمول بندی حالت کامل

حالتها: چیدمان n وزیر (0≤ n≤ 8) ، بطوریکه در هر ستون از n ستون سمت چپ، یک وزیر قرار گیرد و هیچ دو وزیری بهم گارد نگیرند

حالت اولیه: با 8 وزیر در صفحه شروع میشود

تابع جانشین: ?

آزمون هدف: 8وزیر در صفحه وجود دارند و هیچ کدام به یکدیگر گارد نمیگیرند

این فرمول بندی فضای حالت را از 3*10^14 به 2057 کاهش میدهد

اندازه گیری کارآیی حل مسئله

- کامل بودن: آیا الگوریتم تضمین میکند که در صورت وجود راه حل، آن را بیابد؟(در صورتی که الگوریتم در یک LOOP بینهایت قرار گیرد،کامل نیست)
- بهینگي: آیا این راهبرد، راه حل بهینه ای را ارائه میکند.
- پیچیدگي زمانی: چقدر طول میکشد تا راه حل را پیدا کند؟
  - تعداد گره های تولید شده در اثنای جستجو
- پیچیدگی فضا: برای جستجو چقدر حافظه نیاز دارد؟
  - حداکثر تعداد گره های ذخیره شده در حافظه

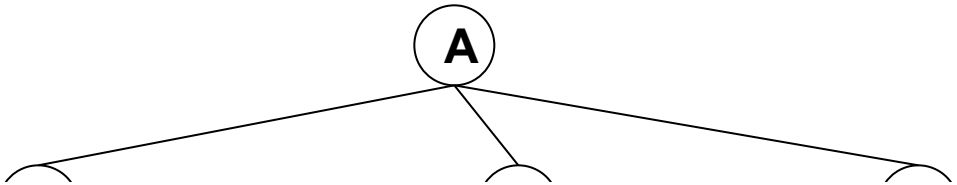
جستجوی ناآگاهانه(کورکورانه)

- ناآگاهی این است که الگوریتم هیچ اطلاعاتی غیر از تعریف مسئله در اختیار ندارد
- این الگوریتمها فقط میتواند جانشینهایی را تولید کنند و هدف را از غیر هدف تشخیص دهند
- راهبردهایی که تشخیص می دهند یک حالت غیر هدف نسبت به گره غیر هدف دیگر، امید بخش تر است، جست و جوی آگاهانه یا جست و جوی اکتشافی نامیده میشود.

راهبردها

- جست و جوی عرضی(BFS)
- جست و جوی عمقی(DFS)
- جست و جوی عمیق کننده تکراری(IDFS)
- جست و جوی هزینه یکنواخت
- جست و جوی عمقی محدود
- جست و جوی دو طرفه

جستجوی عرضی



جستجوی عرضی(ادامه)

کامل بودن: بله

بهینگی: بله (مشروط)

جستجوی عرضی ابتدا کم عمق ترین وضعیت هدف را پیدا می کند و در صورتی بهینه است که هزینه مسیر ، تابعی غیر نزولی از عمق گره باشد (مثل وقتي كه فعاليتها هزينه يكسانی دارند)

پیچیدگي زماني: $O(b^{d+1})$

پیچیدگی فضا: $O(b^{d+1})$

زمان و فضای لازم برای جستجوی عرضی

عمق	گره ها	زمان	حافظه
2	1100	11 seconds	1 mb
4	111،100	11 seconds	106 mb
6	10^7	19 minutes	10 gb
8	10^9	31 hours	1 tb
10	10^11	129 days	101 tb
12	10^13	35 years	10 pb
14	10^15	3،523 years	1 eb

اعدادی که نشان داده شده اند با فاکتور B=10 تعداد 000/10 گره در هر ثانیه و هر گره 1000 بایت فضا را اشغال می کند .این اعداد نشان می دهند که هر هر چند مسئله زمان مسئله ی مهمی است اما مسئله ی فضا مسئله ی جدی تری است.

جستجوی با هزینه یکنواخت

- جستجو با هزینه ی یکنواخت ، استراتژی جستجوی عرضی را توسط بسط کم هزینه ترین گره و نه کم عمق ترین گره بهبود می بخشد درواقع جستجوی عرضی، همان جستجوی یکسان با **g(n)=DEPTH(n)** است.(**g(n)** تابع هزینه ی مسیر است)

- کامل بودن وقتی تضمین می شود که هزینه ی هر مرحله بزرگتر یا مساوی یک مقدار ثابت و مثبت ε باشد. این شرط برای بهینگی نیز کافی است.

جستجوی با هزینه یکنواخت(ادامه1)

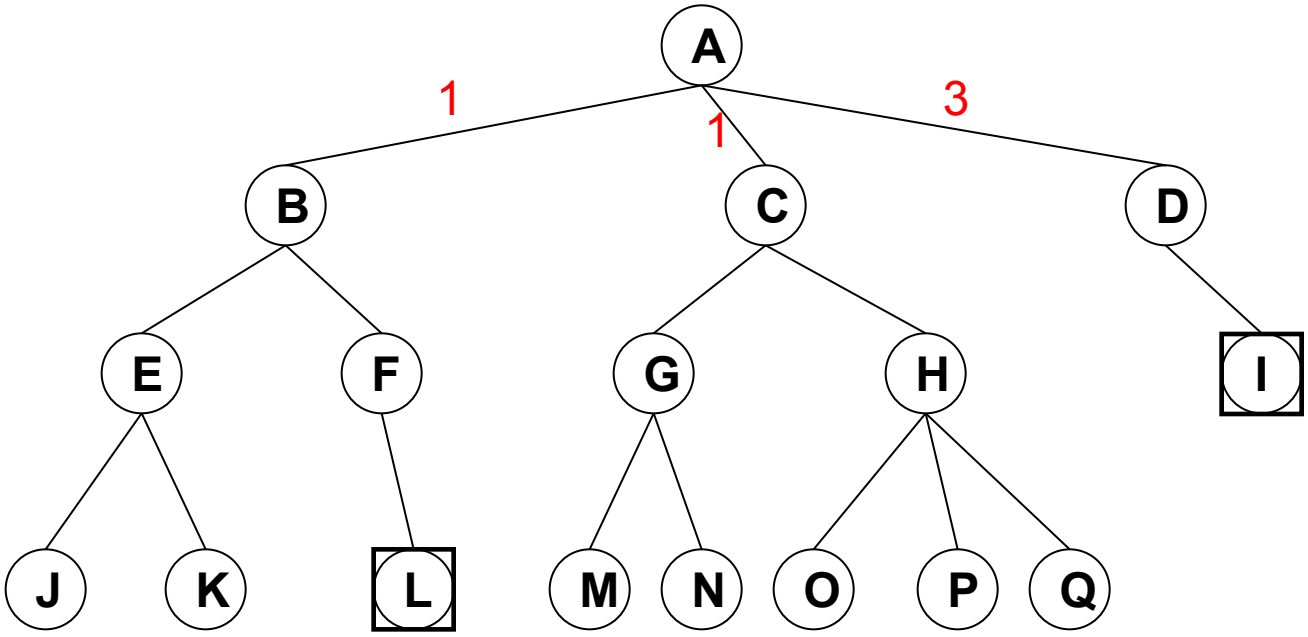
- این استراتژی بجای عمق از هزینه ی های مسیر پیروی می کند لذا پیچیدگی آن را نمی توان براحتی بر حسب b و d بدست آورد.

- فرض کنید هزینه ی راه حل بهینه c * و هزینه هر فعالیت حداقل ε است آنگاه پیچیدگی زمان و فضای الگوریتم در بدترین حالت برابر با $O(b^{\lceil C^*/\varepsilon \rceil})$ است.

- وقتی هزینه تمام مراحل یکسان باشد $O(b^{\lceil C^*/\varepsilon \rceil})$ برابر با b ^d خواهد بود.

جستجوی با هزینه یکنواخت(ادامه2)

این جستجو گره L را با کمترین هزینه مسیر بسط میدهد



جستجوی عمقی

- جستجوی عمقی همیشه عمیق ترین گره را در حاشیه ی فعلی درخت جستجو بسط می دهد.
- این بسط دادن تا جایی ادامه می یابد که گره ها فاقد جانشین هستند و سپس جستجو به عمیق ترین گره ی بعدی که هنوز بسط داده نشده باز می گردد.
- برای جستجوی عمقی معمولا از توابع بازگشتی استفاده می شود.

جستجوی عمقی(ادامه1)

جستجوی عمقی(ادامه2)

کامل بودن: خیر

اگر زیر درخت چپ عمق نامحدود داشت و فاقد هر گونه راه حل باشد، جستجو هرگز خاتمه نمي يابد.

بهینگی: خیر

پیچیدگي زماني: $O(b^m)$

پیچیدگی فضا: $O(bm)$

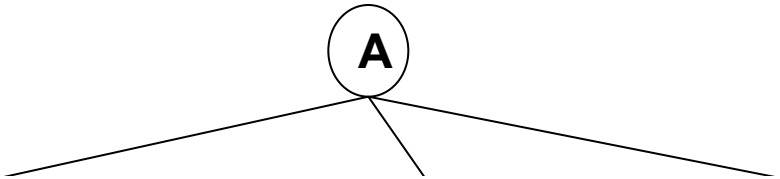
m نشان دهنده ی حداکثر عمق است.

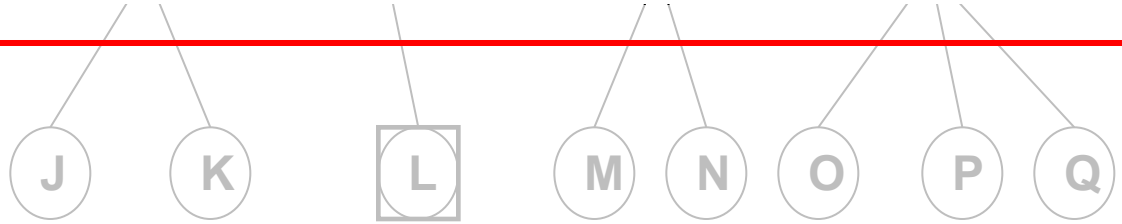
جستجوی عمقی محدود

این استراتژی شبیه استراتژی جستجوی عمقی است با این تفاوت که فرض می شود گره های سطح فاقد جانشین هستند یعنی درخت تا عمق بسط داده می شود و از آنجا پایین تر نمی رود.

- مسئله درختهای نامحدود میتواند به وسیله جست و جوی عمقي با عمق محدود L بهبود یابد

جستجوی عمقی محدود(ادامه1)





جستجوی عمقی محدود(ادامه2)

کامل بودن: خیر

اگر L>d باشد و سطحی ترین هدف در خارج از عمق محدود شده قرار داشته باشد، این راهبرد کامل نخواهد بود.

بهینگی: خیر

این راهبرد بهینه نخواهد بود.

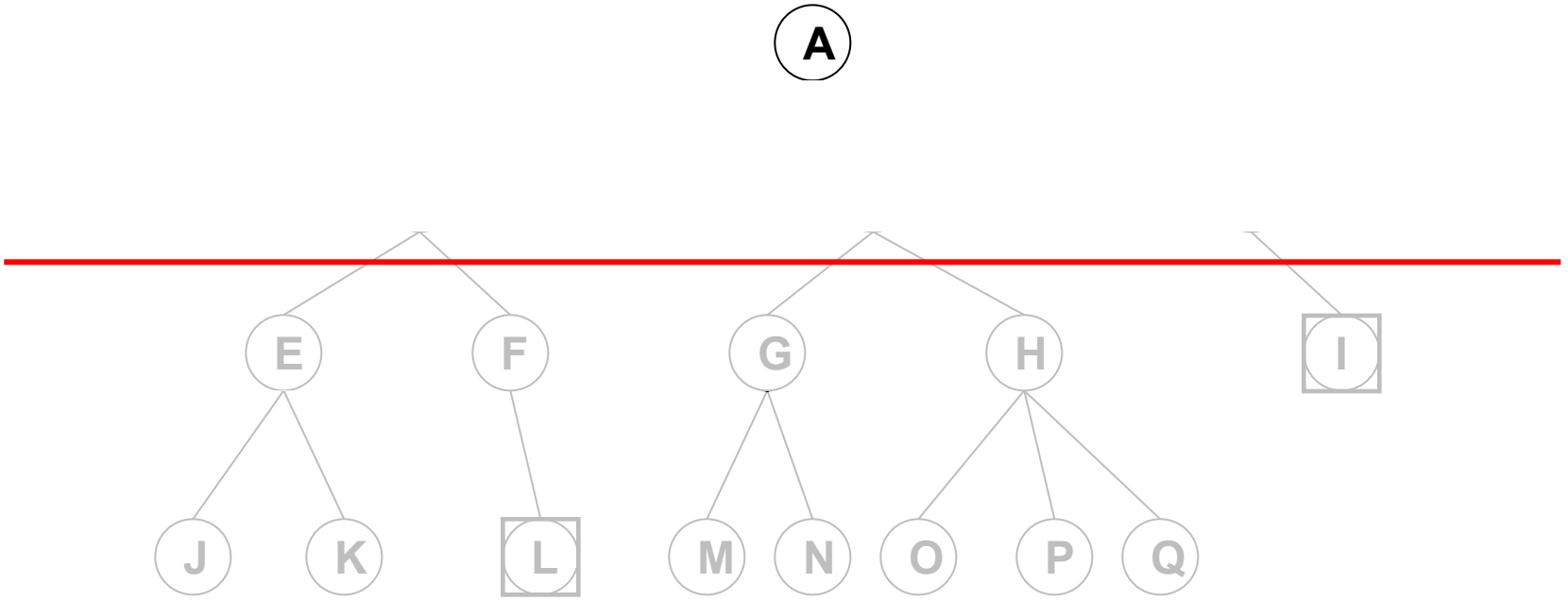
پیچیدگی زمانی: $O(b^L)$

پیچیدگی فضا: $O(bL)$

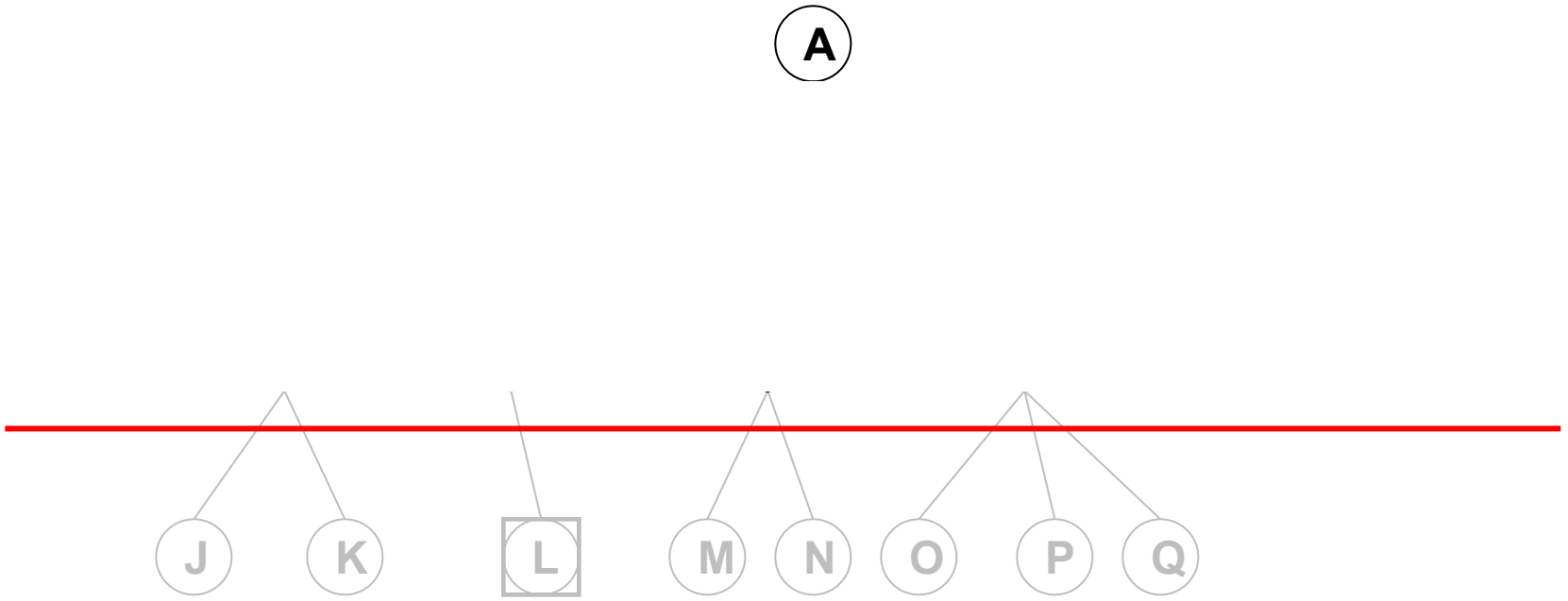
جستجوی عمیق کننده تکراری

- این استراتژی شبیه به جستجوی عمقی محدود عمل می کند با این تفاوت که عمق محدود شده L بتدریج اضافه شده و جستجو از سر گرفته می شود.
- روند جستجو تا رسیدن به اولین پاسخ ادامه پیدا می کند.
- جستجوی عمیق کننده ی تکراری مزایای جستجوی عمقی و جستجوی سطحی را در بردارد.
- بطور کلی وقتی فضای جستجو بزرگ باشد و عمق راه حل مشخص نباشد جستجوی عمیق کننده ی تکراری روش ناآگاهانه مناسبی است.

جستجوی عمیق کننده تکراری(ادامه1)



جستجوی عمیق کننده تکراری(ادامه2)



جستجوی عمیق کننده تکراری(ادامه3)

جستجوی عمیق کننده تکراری(ادامه4)

کامل بودن: بله

در صورتی که فاکتور انشعاب محدود باشد

بهینگی: بله

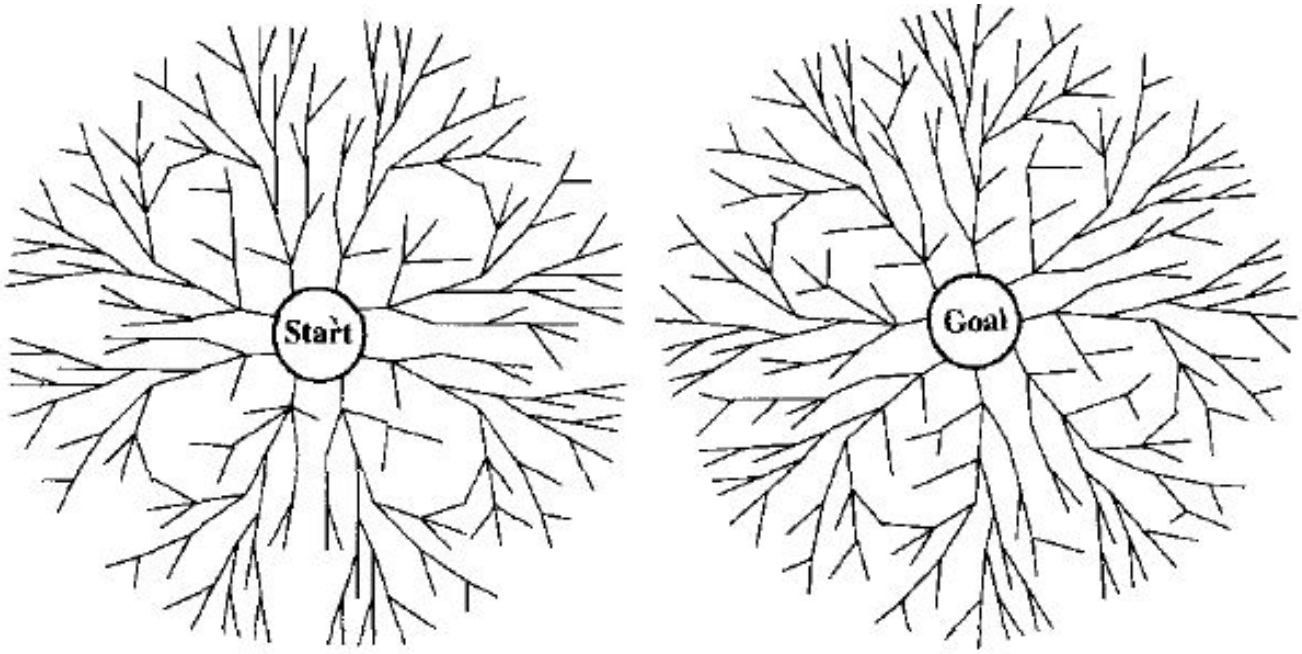
وقتی که هزینه مسیر، تابعی غیر نزولی از عمق گره باشد

پیچیدگي زماني: $O(b^d)$

پیچیدگی فضا: $O(bd)$

جستجوی دو طرفه

انجام دو جست و جوی همزمان، یکي از حالت اولیه به هدف و دیگری از هدف به حالت اولیه تا زمانی که دو جست و جو به هم برسند



جستجوی دو طرفه(ادامه1)

❖**جستجو از سمت هدف به چه معناست؟**

- تعریف)- ما قبل های گره n ، گره هایی هستند که n ، ما بعد آن ها باشد.

- جستجو به سمت عقب به این معناست که تولید ما قبل ها از گره هدف آغاز شود.

- اگر تمام عملگرها قابل وارونه شدن باشند محاسبه ی ماقبل ها به سادگی امکان پذیر است اما این کار در برخی مسائل بسیار مشکل است.

- مسئله ی دیگر در جستجوی دوطرفه این است که هدف باید مشخص باشد.

جستجوی دو طرفه(ادامه2)

کامل بودن: بله

اگر هر دو جستجو، عرضی باشند

بهینگی: بله

اگر هر دو جستجو، عرضی باشند و هزینه تمام مراحل یکسان باشد

پیچیدگي زماني: $O(b^{d/2})$

پیچیدگی فضا: $O(b^{d/2})$

ارزیابی راهبردهای جستجو

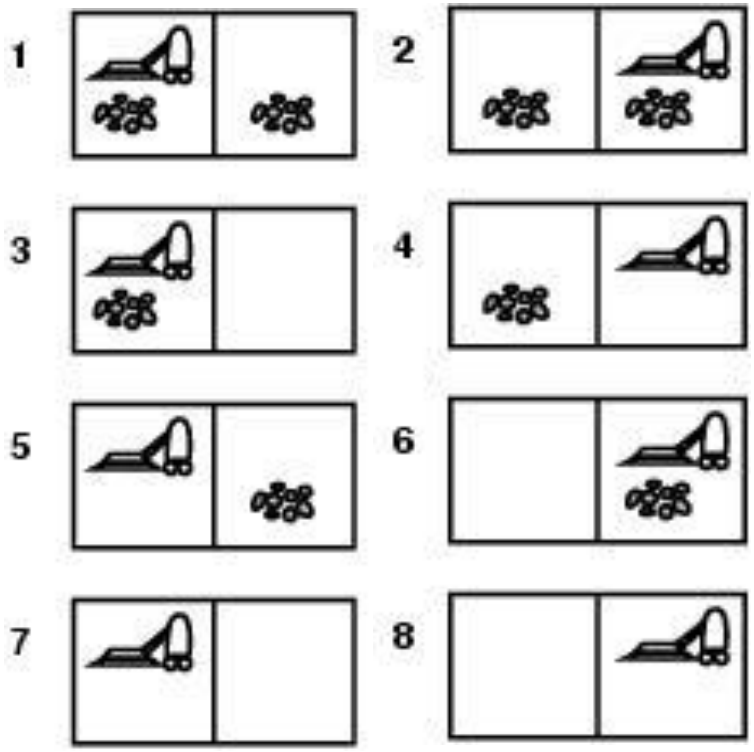
BFS	DFS	DLS	IDFS	BID
ملاک	عرضی	هزینه یکنواخت	عمقی	عمق محدود	عمیق‌کننده تکراری	دوطرفه (درصورت‌امکان)
کامل؟	Yes[a]	Yes [a, b]	No	No	Yes [a]	Yes [a, d]
زمان	$O(b^{d+1})$	$O(b^{\lceil C^*/\varepsilon \rceil})$	$O(b^{\ell})$	$O(b^{m})$	$O(b^{d})$	$O(b^{d/2})$
فضا	$O(b^{d+1})$	$O(b^{\lceil C^*/\varepsilon \rceil})$	$O(b\ell)$	$O(bm)$	$O(bd)$	$O(b^{d/2})$
بهینگی؟	Yes[c]	Yes	No	No	Yes [c]	Yes [c, d]

شکل ۱۷-۳ ارزیابی راهبردهای جست‌وجو. b فاکتور انشعاب، d عمق سطحی‌ترین راه‌حل، m حداکثر عمق درخت جست‌وجو، ℓ عمق محدود است. مفهوم اندیس‌های بالا: [a] کامل است اگر b کامل باشد، [b] کامل است اگر برای ε مثبت، $\varepsilon <$ هزینه مرحله. [c] بهینه است اگر هزینه تمام مراحل یکسان باشند. [d] اگر هر دو طرف از جست‌وجوی عرضی استفاده کنند.

اجتناب از حالتهای تکراری

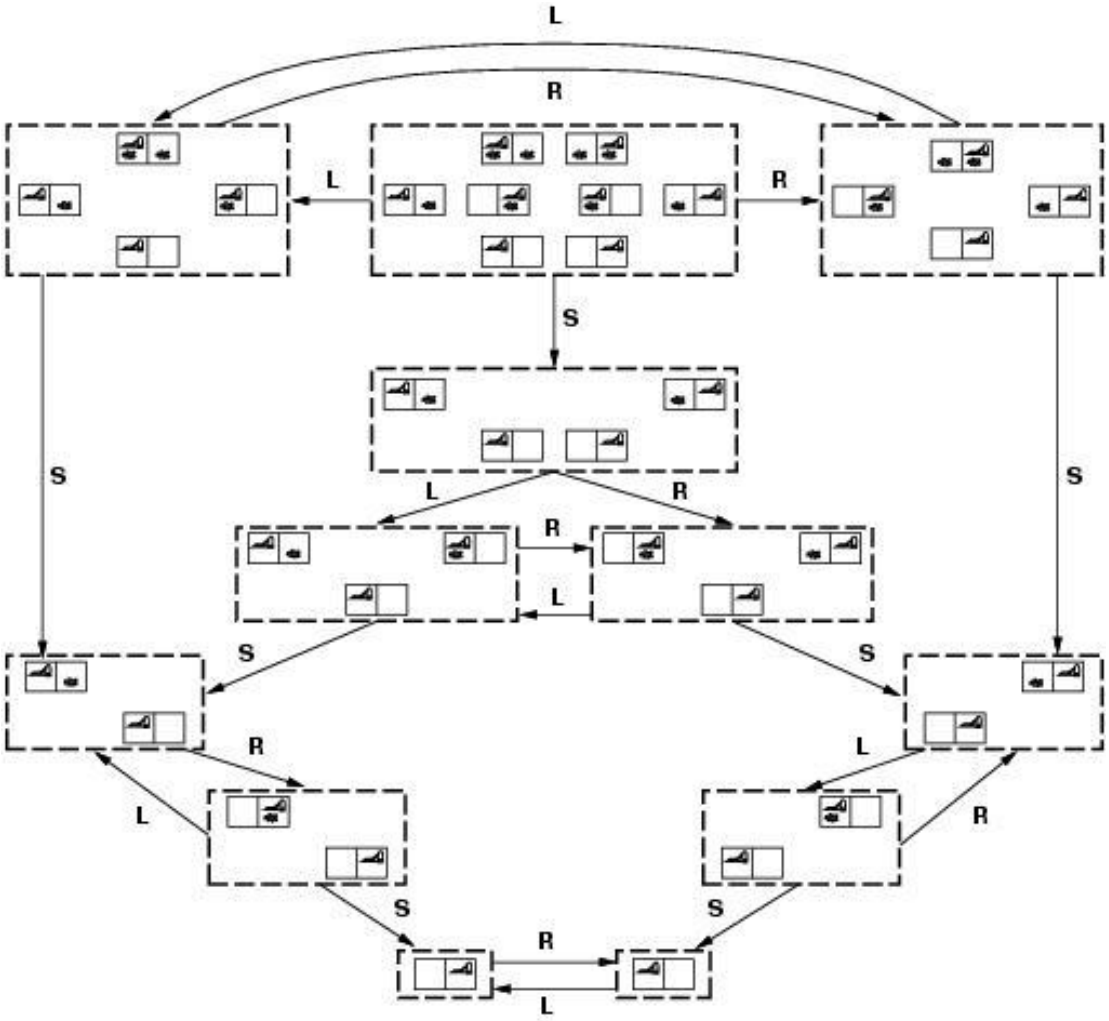
- وجود حالتهای تکراری در یک مسئله قابل حل ، می تواند آنرا به مسئله غیر قابل حل تبدیل کند.

دنیای جارو برقی فاقد حسگر



- عامل جارو تمام اثرات فعالیتهایش را میداند اما فاقد حسگر است.
- حالت اولیه آن یکي از اعضای مجموعه {1،2،3،4،5،6،7،8} میباشد
- فعالیت (راست) ⟵ {2،4،6،8}
- فعالیت (راست,مکش) ⟵ {4،8}
- فعالیت (راست,مکش,چپ,مکش) تضمین میکند که صرف نظر از حالت اولیه، به حالت هدف، یعنی 7 برسد

دنیای جارو برقی فاقد حسگر(ادامه)



عامل باید راجع به مجموعه هاي حالتی که میتواند به آنها برسد استدلال کند. این مجموعه از حالتها را حالت باور گوییم.