



سوال اول (۲۰ نمره)

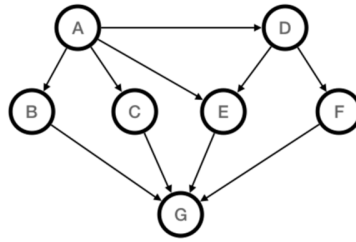
شما سرمربی تیم ساوتهمپتون هستید و قصد دارید جلسه تمرینی برای کسب بهترین نتیجه برای بازی بعدی خود در لیگ انجام دهید. می توانید تمرین دفاع یا تمرین حمله کنید. اگر تمرین دفاع کنید تیم حریف با احتمال 0.2 حداقل دو گل می زند. اما اگر تمرین دفاع نکنید با احتمال 0.7 دو گل به ثمر می رساند. از سوی دیگر، اگر دفاع تمرین کنید (و اتوبوسی دفاع کنید!)، بهترین مهاجم شما که **AI TAREMI** نام دارد، با احتمال 0.6 بازی ضعیفی از خود به نمایش خواهد گذاشت. اما اگر دفاع تمرین نکنید، این احتمال 0.1 خواهد بود. اگر تمرین حمله کنید و **AI TAREMI** ضعیف بازی نکند، با احتمال 0.84 دو گل به ثمر می رساند. اما اگر تمرین حمله کنید و او ضعیف بازی کند، این احتمال 0.5 خواهد بود. همچنین اگر حمله تمرین نکنید و **AI TAREMI** ضعیف بازی نکند، با احتمال 0.4 دو گل می زنید. در صورت عدم تمرین حمله و ضعیف بازی کردن مهاجمتان، این احتمال به 0.15 کاهش می یابد. اگر هم تیم شما و هم تیم حریف حداقل دو گل به ثمر برسانند، شما با احتمال 0.55 برنده بازی می شوید، چراکه بازی در خانه شما برگزار می شود. اگر هر دو تیم کمتر از دو گل بزنند، با احتمال 0.35 بازی را می برید. مسلماً در حالتی که یک تیم حداقل دو گل بزند و تیم حریف کمتر از دو گل بزند، تیم اول با احتمال ۱ برنده می شود.

۱. یک مدل احتمالاتی بسازید و با استفاده از یک شبکه بیز، همه متغیرهای تصادفی مدل و وابستگی بین آنها را توضیح دهید. جداول احتمالات شرطی را نیز مشخص کنید. (۱۵ نمره)

۲. با استفاده از مدل خود توضیح دهید که تمرین دفاع به سود شماست یا تمرین حمله. (۵ نمره)

سوال دوم (۱۵ نمره)

۱. شبکه زیر چه فاکتورگیری از توزیع توام $P(A, B, C, D, E, F, G, H)$ نشان می‌دهد؟ (۱ نمره)



۲. دو شبکه بیز متفاوت رسم کنید که داشته باشیم:

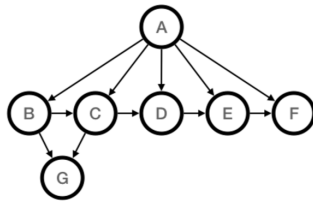
$$P(A, B, C, D, E) = P(E|C, D)P(D|C)P(C|A, B)P(B|A)P(A)$$

(۱ نمره)

۳. فرض کنید هر متغیر تصادفی در توزیع توام قسمت قبل، دامنه‌ای از ۱۰ عضو داشته باشد. جدول نمایش توزیع توام چه تعداد سطر دارد؟ (۱ نمره)

۴. فرض کنید هر متغیر تصادفی در توزیع توام قسمت ۲، دامنه‌ای از ۱۰ عضو داشته باشد. مجموعاً چه تعداد سطر برای نمایش جدول احتمال شرطی شبکه‌ای که شما رسم کردید لازم است؟ (۲ نمره)

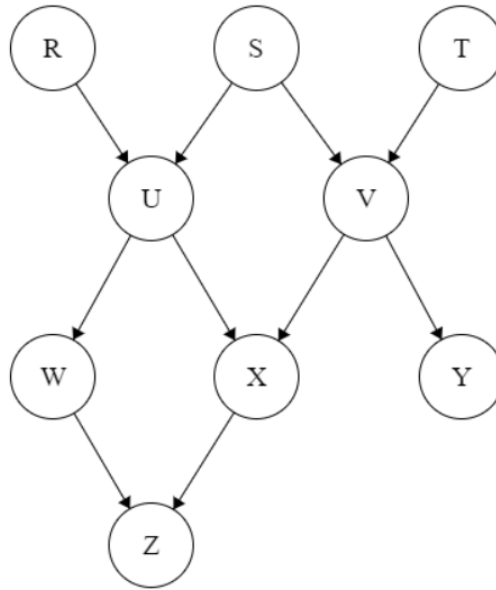
۵. شبکه زیر را در نظر بگیرید.



- در صورتی که الگوریتم **variable elimination** روی این شبکه اعمال شود تا به پرسمان $P(B|G, E)$ پاسخ داده شود، تمام عامل‌های تولیدشده و عملیات تولید آن‌ها را بنویسید. از ترتیب G, E, A, B, C, D, F استفاده کنید. (۴ نمره)
- در صورتی که الگوریتم **variable elimination** روی این شبکه اعمال شود تا به پرسمان $P(B|G, E)$ پاسخ داده شود، تمام عامل‌های تولیدشده و عملیات تولید آن‌ها را بنویسید. از ترتیب G, E, F, D, C, B, A استفاده کنید. (۴ نمره)
- کدام یک از ترتیب‌های داده‌شده برای این پرسمان بهتر است؟ دلیل خود را بنویسید. (۲ نمره)

سوال سوم (۱۰ نمره)

با توجه به شکل زیر و با استفاده از D-Separation، درستی یا نادرستی هر یک از گزاره‌های زیر را مشخص کنید. در صورت نادرست بودن، یک مسیر معتبر بین دو متغیر تصادفی بنویسید.



۱. $R \perp\!\!\!\perp Z$

۲. $R \perp\!\!\!\perp Z \mid U$

۳. $R \perp\!\!\!\perp T$

۴. $R \perp\!\!\!\perp Y \mid X$

۵. $U \perp\!\!\!\perp V \mid S, X$

۶. $W \perp\!\!\!\perp Y \mid S$

۷. $W \perp\!\!\!\perp Y \mid Z$

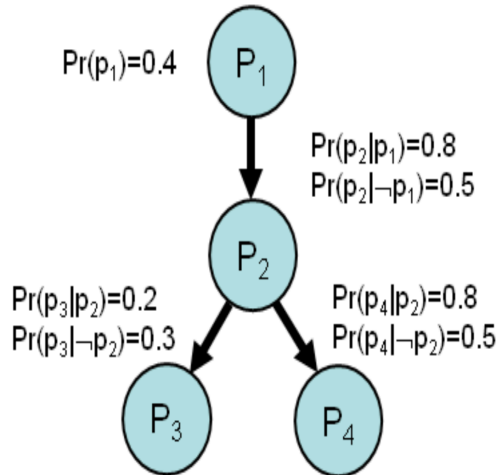
۸. $W \perp\!\!\!\perp Y \mid U, V$

۹. $W \perp\!\!\!\perp T \mid U, X, Z$

۱۰. $T \perp\!\!\!\perp Z \mid X$

سوال چهارم (۱۵ نمره)

فرض کنید می‌خواهیم برای محاسبه احتمال شرطی $Pr(P_1|P_2, \neg P_3)$ در شبکه بیز داده شده، از روش تقریبی استفاده کنیم. فرض کنید جدول زیر، خروجی یک *uniform random generator* در بازه $(0, 1)$ باشد.



r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
0.2551	0.5060	0.6991	0.8909	0.9593	0.5472	0.1386	0.1493	0.2575	0.8407
r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}	r_{20}
0.0827	0.9060	0.7612	0.1423	0.5888	0.6330	0.5030	0.8003	0.0155	0.6917

با فرض استفاده از نمونه‌های بالا، محاسبات را با سه روش *likelihood weighting*، *rejection sampling* و *Gibbs sampling* انجام داده و در مورد مزایا و معایب هریک از این روش‌ها بحث کنید.

سوال پنجم – مکان‌یابی ربات (۵۰ نمره)

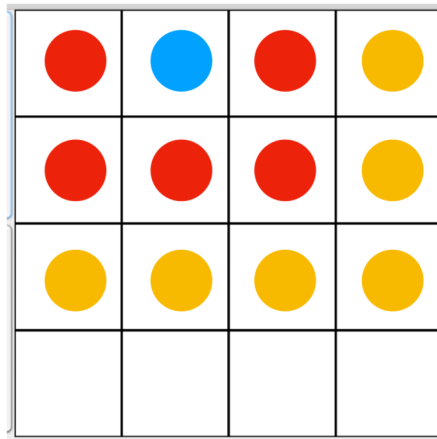
این مسئله مشابه تمرین 15.9 کتاب درسی (موجود در سایت درس) است و مدل سنسور آن را با جزئیات بیشتری توصیف می‌کند. روش حل این سوال بر پایه عملیات matrix based forward filtering در بخش 15.3.1 کتاب است.

به طور خلاصه

شما روی مسئله مکان‌یابی ربات با استفاده از forward filtering و HMM کار می‌کنید.

به طور دقیق

شما باید یک HMM پیاده‌سازی کنید تا در یک محیط بدون نشانه مشخص، با استفاده از forward filtering مسئله مکان‌یابی را حل کنید. مسئله یک جاروبرقی در یک اتاق خالی به صورت یک جدول $n * m$ را در نظر بگیرید. پیش از این با مسئله مشابه در اسلایدهای درس مواجه شده‌اید. مکان ربات مشخص نیست و تنها نشانه‌ای که داریم، یک سنسور noisy است که مکان ربات را به صورت تقریبی و مبهم مشخص می‌کند. سنسور به جای مکان اصلی $L = (x, y)$ ، مکان تقریبی ربات $S' = (x', y')$ را مشخص می‌کند. به علاوه سنسور خانه‌های مجاور Ls و خانه‌های مجاورِ مجاور ربات $Ls2$ را نیز نشان می‌دهد. قوانین مشخص کردن این موارد در ادامه ذکر می‌شوند. در این جا nLs تعداد خانه‌های مجاور L را نشان می‌دهد که این عدد می‌تواند بسته به مکان ربات برابر با 3 یا 5 یا 8 باشد (ربات در گوشه اتاق، کنار یک دیوار یا به دور از دیوار باشد). همچنین $nLs2$ نیز تعداد خانه‌های مجاورِ مجاورِ مجاورِ مجاور L را نشان می‌دهد که می‌تواند یکی از اعداد $\{5, 6, 7, 9, 11, 16\}$ باشد. در شکل زیر، نقطه آبی L و نقاط قرمز Ls را نشان می‌دهد. به علاوه نقاط زرد نشان‌دهنده $Ls2$ هستند.



سنسور موارد زیر را گزارش می‌کند:

- مکان دقیق ربات با احتمال 0.1
- هر کدام از $nLs \in \{3, 5, 8\}$ خانه‌های مجاور با احتمال 0.05
- هر کدام از $nLs \in \{5, 6, 7, 9, 11, 16\}$ خانه‌های مجاورِ مجاور با احتمال 0.025
- در نهایت با احتمال $1.0 - 0.1 - nLs * 0.05 - nLs2 * 0.025$ هیچ چیزی گزارش نمی‌کند.

این بدان معناست که سنسور هنگامی که ربات در گوشه اتاق قرار دارد یا کمتر از دو خانه با دیوار فاصله دارد، با احتمال بالایی هیچ چیزی گزارش نمی‌کند. ربات طبق استراتژی زیر حرکت می‌کند:

ربات در ابتدا یک heading شروع (h_0) را به صورت تصادفی انتخاب می‌کند و در مراحل بعدی، یک heading جدید (h_{t+1}) را طبق heading کنونی (h_t) و روابط زیر انتخاب می‌کند. heading جهت سر ربات (شمال - شرق - جنوب - غرب) را با عددی بین ۰ تا ۳ نشان می‌دهد:

$$(1) \quad \begin{cases} P(h_{t+1} = h_t | \text{دیوار با مواجهه}) = 0.7 \\ P(h_{t+1} \neq h_t | \text{دیوار با مواجهه}) = 0.3 \\ P(h_{t+1} = h_t | \text{مواجهه با دیوار}) = 0.0 \\ P(h_{t+1} \neq h_t | \text{مواجهه با دیوار}) = 1.0 \end{cases}$$

پس از آن ربات در جهت h_{t+1} یک گام در جدول حرکت می‌کند. توجه کنید که این امر به این معناست که ربات همواره یک گام حرکت می‌کند و تنها می‌تواند مستقیم حرکت کند.

در مواقعی که یک heading جدید بخواهیم پیدا کنیم، این heading به صورت تصادفی از بین گزینه‌های ممکن انتخاب خواهد شد. (در صورت مواجهه به دیوار در محلی وسط دیوار، ۳ انتخاب و مواجهه با دیوار در گوشه، ۲ انتخاب به ما خواهد داد. یک مثال از این مورد را در راهنمایی ۱ مشاهده کنید.) این مسئله را به صورت یک HMM پیاده‌سازی کنید (طبق نوشتار ماتریس-بردار در بخش 15.3.1 کتاب درس) و forward filtering ساده را برای یافتن مسیر ربات اعمال کنید. به وضوح، پیاده‌سازی شما شامل دو بخش خواهد بود زیرا باید ربات و حرکت آن (که می‌توانید از روی آن خواندن‌های سنسور را نیز شبیه‌سازی کنید) را شبیه‌سازی کنید تا حقایق اولیه‌ای برای ارزیابی مسیریابی خود پیدا کنید. الگوریتم شما باید در یک حلقه سه مرحله زیر را اجرا کند:

۱. حرکت (شبیه‌سازی‌شده) ربات به محل جدید طبق مدل حرکت
۲. به‌دست‌آوردن (شبیه‌سازی‌شده) خواندن سنسور طبق وضعیت حقیقی داده‌شده و مدل سنسور
۳. به‌روزرسانی تخمین وضعیت (بردار f به کمک الگوریتم forward طبق خواندن سنسور از مرحله ۲، با استفاده از مدل سنسور و گذار (transition) شناخته شده.

راهنمایی ۱: حالات را به گونه‌ای کد کنید که هر حالت ممکن، نشانگر محل ربات در جدول به همراه یک heading باشد. این به این معناست که شما به تعداد ($4 \times \text{ستون‌ها} \times \text{سطرها}$) حالت ممکن و به تعداد ($1 + \text{ستون‌ها} \times \text{سطرها}$) خوانده سنسور ممکن خواهید داشت. بنابراین ماتریس transition های شما، دارای ابعاد ($4 \times \text{ستون‌ها} \times \text{سطرها}$) \times ($4 \times \text{ستون‌ها} \times \text{سطرها}$) خواهد بود و برای ضرب ماتریس‌ها نیازمند تعداد ($1 + \text{ستون‌ها} \times \text{سطرها}$) ماتریس observation خواهید بود که هریک ماتریسی قطری به ابعاد ($4 \times \text{ستون‌ها} \times \text{سطرها}$) \times ($4 \times \text{ستون‌ها} \times \text{سطرها}$) می‌باشد.

اما به نکته زیر توجه کنید که شاید ذخیره‌سازی ماتریس‌های observation به صورت بردار منطقی باشد که هر بردار نشان دهنده قطر مربوطه در ماتریس قطری اصلی است. هم‌چنین الگوریتم filtering تنها خواندن‌های سنسور را می‌داند که نشانگر محل‌های مختلف جدول است. بنابراین یک خواندن سنسور، برای ۴ حالت (heading) متناظر با یک محل در جدول، دارای احتمال یکسان خواهد بود. بنابراین مقادیر درون هر ۴ درایه متوالی قطر ماتریس observation یکسان هستند که می‌توان از این امر برای کاهش

فضای مورد استفاده، بهره برد.

0.00	0.00	0.00	0.00	0.025	0.050	0.050	0.050	0.625	0.500	0.500	0.625
0.00	0.00	0.00	0.70	0.025	0.025	0.050	0.050	0.625	0.625	0.500	0.500
0.00	0.00	0.00	0.00	0.025	0.050	0.050	0.050	0.625	0.500	0.500	0.625
0.00	0.00	0.00	0.00	0.025	0.050	0.100	0.050	0.500	0.325	0.325	0.500
0.00	0.00	0.00	0.00	0.025	0.025	0.050	0.100	0.500	0.500	0.325	0.325
0.30	0.00	0.00	0.00	0.025	0.050	0.100	0.050	0.500	0.325	0.325	0.500
0.00	0.00	0.00	0.00	0.025	0.050	0.050	0.050	0.500	0.325	0.325	0.500
0.00	0.00	0.00	0.00	0.025	0.025	0.050	0.050	0.500	0.500	0.325	0.325
0.00	0.00	0.00	0.00	0.025	0.050	0.050	0.050	0.500	0.325	0.325	0.500
0.00	0.00	0.00	0.00	0.025	0.025	0.025	0.025	0.625	0.500	0.500	0.625
0.00	0.00	0.00	0.00	0.025	0.025	0.025	0.025	0.625	0.625	0.500	0.500
0.00	0.00	0.00	0.00	0.025	0.025	0.025	0.025	0.625	0.500	0.500	0.625

در سه شکل بالا یک سطر از ماتریس transition و قطر یک ماتریس observation نشان داده شده است. (۴ سمت هر سلول (x, y) ، نمایانگر ۴ حالتی هستند که با احتمالات متفاوتی می‌توان به آن‌ها رسید، اما دارای احتمال یکسانی هستند که باعث یک خواندن سنسور شوند.) شکل مربوط به transition (سمت چپ) احتمال رفتن از خانه $(0, 0, EAST)$ (آبی روشن) به هر حالت دیگر را نشان می‌دهد. برای مثال تنها دو حالت $(0, 1, EAST)$ و $(1, 0, SOUTH)$ ممکن هستند و احتمال بیشتر از ۰ دارند. در مثال‌های ماتریس observation (وسط و سمت راست) به ازای هر حالت (x, y, h) احتمال تولید $r = (1, 2)$ (وسط) و احتمال تولید "nothing" (سمت راست) نشان داده شده است.

راهنمایی ۲: گزارش شدن "nothing" توسط سنسور به طور معمول به معنی انجام یک *forwardstep* بدون آپدیت است، اما باید همانگونه که در بالا گفته شده است باید از اطلاعات داده شده در مدل سنسور استفاده کنید. برای مثال خوانده شدن "nothing" هنگامی که ربات نزدیک به دیوار است، کمی محتمل‌تر است. بنابراین حتی خوانده شدن "nothing" توسط سنسور هم باید موجب یک پیشبینی مناسب و یک مرحله آپدیت شود.

راهنمایی ۳: یک شبکه ترحیحا ۸ در ۸ (حداقل ۵ در ۵) فرض کنید و ارزیابی خود را بر مبنای آن انجام دهید. اگر از Java و یک شبکه ۸ در ۸ استفاده می‌کنید، باید به سرعت (در حدود ۱۰۰ مرحله) حدود ۳۰ تا ۳۵ درصد از تخمین درست را مشاهده کنید. سپس فاصله منتهن میانگین (برای مثال تعداد مراحل لازم ربات برای رسیدن از محل تخمینی به محل واقعی) باید بین ۱.۶ تا ۲.۰ باشد. مشاهده شدن یک صفحه همانند صفحه checker معمول است. (شکل پایین را نگاه کنید که رنگ‌های تیره تر نشان‌دهنده احتمال بیشتر و رنگ‌های روشن نشان‌دهنده احتمال کمتر است. طوسی بیشترین مقدار، مشکی مکان حقیقی و آبی روشن نشان‌دهنده مکان خوانده شده توسط سنسور است)

0.0000	0.0000	0.0000	0.0000
0.0000	0.0124	0.1127	0.0136
0.0000	0.1187	0.0289	0.3025
0.0000	0.0192	0.3222	0.0698

یک فایل zip شامل کد جاوا برای نمایش مدل انتقال و مدل سنسور در اختیار شما قرار گرفته است.

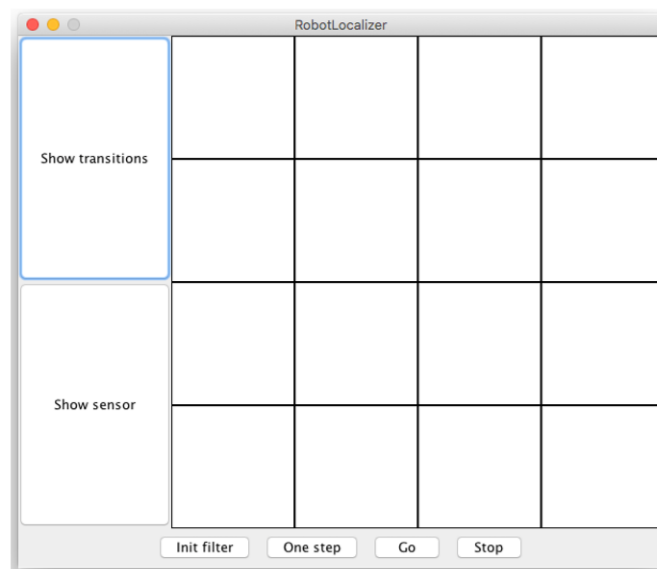
توضیحات اجزای آن در ادامه می‌آید.

راهنمای استفاده از RobotLocalizationViewer

RobotLocalizationViewer از حالت‌هایی به شکل (x, y, h) که x و y نشان‌دهنده سطر و ستون است و h جهت ربات را نشان می‌دهد. $x = 0$ سطر بالا و $y = 0$ ستون چپ است. مقدار h نیز عددی بین 0 تا 3 است که به ترتیب شمال، شرق، جنوب و غرب را نشان می‌دهد. هر روشی برای نمایش حالت‌ها استفاده می‌کنید، باید آن‌ها را به صورت گفته شده به ابزار مصورسازی تحویل دهید. به عنوان خواننده‌های سنسور، کلاس گفته شده $n * m$ مقدار احتمال (هر یک به ازای هر خانه از جدول) یا مقدار $(-1, -1)$ (به عنوان هیچ) را دریافت می‌کند. برای اطلاعات بیشتر، کامنت‌های EstimatorInterface را ببینید.

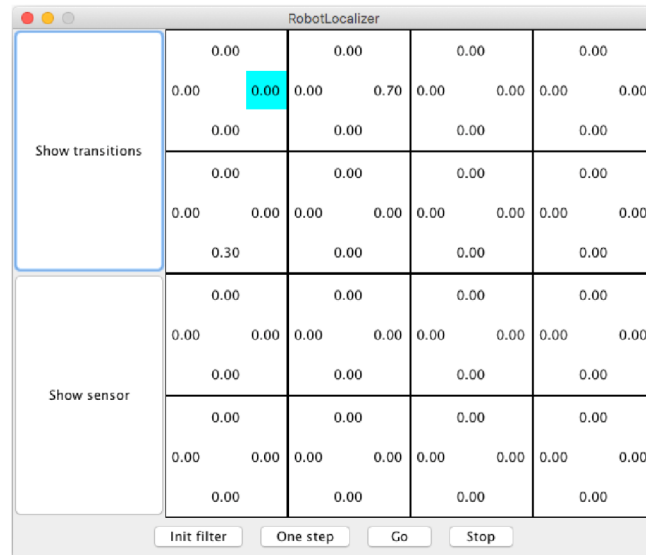
حالت شروع

Viewer با یک جدول خالی شروع به کار می‌کند. شما باید یک instance از Localizer خودتان در تابع control.Main.java بسازید. (یا Main خودتان را بنویسید!) تصویر زیر viewer را به ازای یک جدول $4 * 4$ نشان می‌دهد. دقت داشته باشید که این ابعاد تنها به عنوان مثال هستند و باید ابعاد بزرگ‌تری را در پیاده‌سازی خود پوشش دهید.

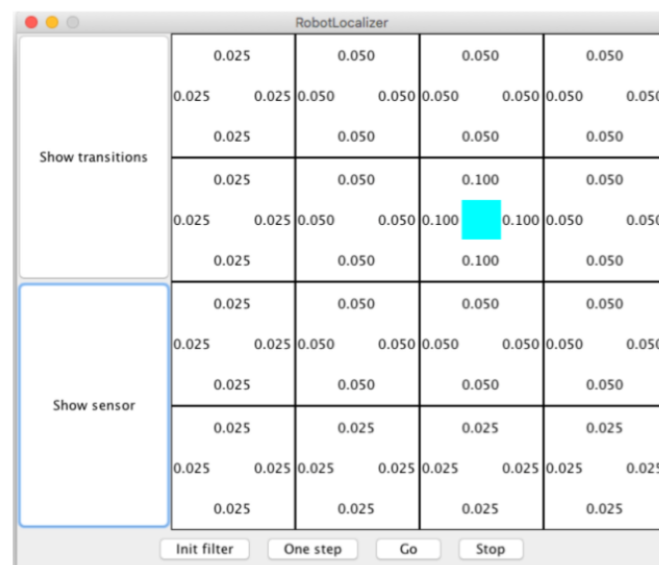


نمایش ماتریس‌ها

کلیک روی Show Transitions احتمال گذار به هر حالت از حالت فعلی (با آبی روشن مشخص شده است) را نشان می‌دهد. با هر کلیک، یک گام جلوتر می‌رویم. تصویر زیر احتمال گذار از حالت $(0, 0, EAST)$ به حالات دیگر را نمایش می‌دهد. همانطور که مشخص است در این تصویر تنها دو حالت برای مکان بعدی متصور هستیم.

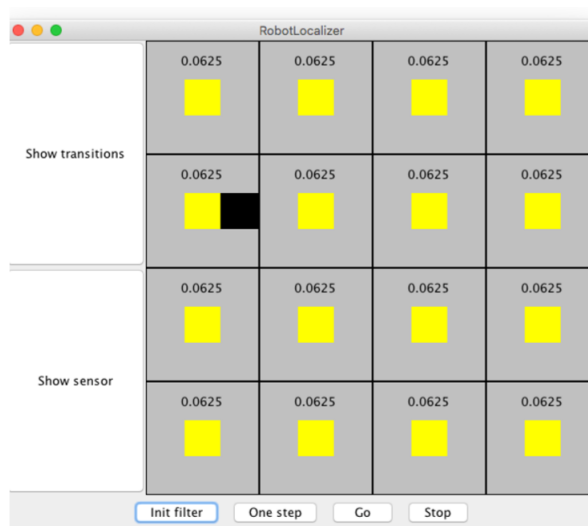


کلیک روی **Show Sensor** احتمال آن را نشان می‌دهد که سنسور خانه‌ی فعلی (با آبی روشن) یا **nothing** (همه خانه‌ها سفید) را گزارش کند، در حالی ربات در خانه‌های دیگر است. شکل زیر احتمال گزارش خانه (1, 2) را نشان می‌دهد.



نمایش گام‌های **filtering** و نتایج آن

کلیک روی **init filter** نمونه‌های **viewer** و **localizer** را مقداردهی اولیه می‌کند. این کار برای آن‌که گام‌های بعدی به درستی کار کنند، ضروری است. تصویر زیر حالت شروع (سیاه) را در (1, 0, 1) نشان می‌دهد. احتمال حضور در هر خانه در ابتدا برابر است. خانه‌ای که با خاکستری مشخص شده است، بیشترین احتمال را دارد.



با کلیک روی **One step** یک گام **filtering** انجام می‌شود. رنگ سیاه حالت واقعی ربات را نشان می‌دهد. رنگ آبی روشن حالتی را نشان می‌دهد که سنسور گزارش می‌کند. رنگ خاکستری محتمل‌ترین خانه و رنگ سفید خانه‌هایی را نشان می‌دهد که احتمال حضور در آن‌ها صفر است. به همین ترتیب رنگ زرد خانه‌ها با احتمال کم (بین 0 و 0.1)، رنگ نارنجی خانه‌ها با احتمال بیشتر (بین 0.1 و 0.3) و رنگ قرمز خانه‌ها با احتمال بیشتر (بیشتر از 0.3) را نمایش می‌دهد. با کلیک روی **Go** گام‌ها به صورت مداوم اجرا می‌شوند. **Stop** این حلقه را متوقف می‌کند و پس از آن می‌توان گام به گام یا مداوم پیش رفت.



موفق باشید (:)