

به نام خدا



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

## پاسخنامه تمرین سری اول مبانی و کاربردهای هوش مصنوعی

### «فصل اول»

نیم سال دوم ۹۹-۱۳۹۸

۱. به نظر شما بین استفاده از captcha و Turing test ارتباطی وجود دارد؟ توضیح دهید.

تکنولوژی captcha، یک آزمون ساده تصویری است که جهت تشخیص انسان یا ماشین بودن کاربرها استفاده می‌شود و هدف آن، جلوگیری از نفوذ spammer ها به سیستم‌ها است. captcha، مبتنی بر تست تورینگ و مخفف عبارت زیر است :

Completely Automated Public Turing test to tell Computers and Humans Apart

آزمون تورینگ – که توسط آلن تورینگ معرفی شده – راهی است جهت تشخیص این که آیا ماشین‌ها می‌توانند مانند انسان‌ها عمل کنند یا خیر. در این آزمون، دو عامل شرکت می‌کنند که یکی از آن‌ها انسان و دیگری ماشین می‌باشد و یک بازپرس داریم که از پیش نمی‌داند کدام عامل ماشین و کدامیک انسان می‌باشد و تنها باید با پرسیدن تعدادی سوال از آن‌ها، بتواند ماشین را تشخیص دهد. اگر بازپرس قادر نباشد با طرح سوال تشخیص دهد کدام عامل ماشین و کدامیک انسان است، ماشین مورد نظر تست تورینگ را پاس کرده است.

هدف آزمون captcha نیز همانطور که گفته شد، تشخیص کاربرهای انسانی از کدهای خودکار می‌باشد و باید به نوعی باشد که که انسان‌ها به سادگی بتوانند از آن عبور کنند ولی عبور از آن برای ماشین‌ها ناممکن باشد (و یا هزینه محاسباتی زیادی نیاز داشته باشد). از آنجایی که تکنولوژی‌های امروزی هنوز به سادگی توانایی برابری با توانایی بینایی و تشخیص تصاویر انسان‌ها را ندارند، در بیشتر آزمون‌های captcha از تصاویر واقعی و طرح سوالاتی بر اساس آن تصاویر، استفاده می‌شود. با وجود اینکه تکنولوژی‌ها و الگوریتم‌های جدید هوش مصنوعی توانایی حل بسیاری از این سوالات را پیدا کرده‌اند، حل آن‌ها همچنان برای کامپیوترها از لحاظ محاسباتی بسیار هزینه بر می‌باشند و بنابراین، تکنولوژی captcha همچنان برای جلوگیری از نفوذ spammer ها موثر است.

آزمون captcha توسط کامپیوتر مشخص می‌شود درحالی که آزمون تورینگ استاندارد توسط انسان مشخص می‌گردد.

۲. وارد وبسایت [www.rong-chang.com/tutor\\_mike.htm](http://www.rong-chang.com/tutor_mike.htm) شده و بر روی "Tutor Mike" کلیک کرده تا با یک چت بات

مکالمه کنید. حداقل ۱۰ سوال از آن پرسیده و براساس آن ها توضیح دهید که تست تورینگ پاس و یا رد می شود. ( کل مکالمه خود را کپی کرده و بخشی از آن را که باعث نتیجه گیری شده است توضیح دهید. )

### My Conversation

**YOU: hi, what is your name?**

**Mike: My name is Mike. How are you doing today?**

**YOU: I'm doing great.**

**Mike: That's great to hear!**

**YOU: do you have any siblings?**

**Mike: Yes, I have one brother and one sister.**

**YOU: can you tell me their names, please?**

**Mike: They are Michelle, Steve, and Tony.**

**YOU: who is Steve?**

**Mike: STEVE IS A COMMON NAME.**

**YOU: what is your brother's name?**

**Mike: Robert.**

**YOU: who is Robert?**

**Mike: ROBERT IS A COMMON NAME.**

**YOU: do you have any brother named Robert?**

**Mike: I occasionally have, but not so often.**

**YOU: don't you wanna ask me my name?**

**Mike: Yes, I want to.**

**YOU: so?**

**Mike: Yes?**

**YOU: nothing , you are just a terrible machine !**

**Mike: There must be something.**

بر اساس قسمت های آبی ، جواب های بی ربط داده شده و عدم ارتباط جواب ها با هم ، تست تورینگ رد می شود.

۳. برای هریک از ادعاهای زیر در صورت صحیح بودن توجیه منطقی و در صورت غلط بودن مثال نقض آورید .

(آ) عاملی که در هر حالت تنها بخشی از percept ها را توسط سنسورهایش دریافت می کند ، نمی تواند کاملاً عقلانی باشد.

غلط . عقلانی بودن یک عامل وابسته به معیار کارایی، ادراکات او تا این لحظه، اعمالی که می تواند انجام دهد و دانش قبلی اش از محیط است. برای مثال عامل جاروبرقی که وضعیت تمیز و یا کثیف بودن خانه مجاورش را حس نمی کند می تواند وابسته به معیار کارایی و ... عملکرد عقلانی داشته باشد.

(ب) محیط‌هایی وجود دارند که در آن هیچ عامل واکنشی ساده‌ای نمی‌تواند عملکرد منطقی داشته باشد.

صحیح . هیچ عاملی واکنشی ساده‌ای در محیط‌هایی که در آن‌ها انجام عمل عقلانی نیاز به حافظه داخلی دارد، نمی‌تواند عملکرد منطقی داشته باشد. (معمولاً در محیط های نیمه مشاهده پذیر اتفاق می افتد).

(پ) محیط‌هایی وجود دارند که در آن‌ها هر عاملی عقلانی است .

صحیح . برای مثال محیطی با تنها یک حالت را در نظر بگیرید که تمام عمل ها دارای پاداش یکسانی هستند. بنابراین انتخاب عمل در میزان پاداش هیچ تاثیری ندارد.

(ت) یک عامل کاملاً عقلانی در یک بازی با نتیجه تنها برد و یا باخت ، هیچ وقت نمی‌بازد .

غلط . در آن بازی دو عامل از این نوع را در مقابل هم قرار می‌دهیم . به اجبار یکی از آن‌ها برنده و دیگری می‌بازد !

۴. در مورد هر یک از مسائل زیر ، ابتدا PEAS را مشخص کرده و سپس محیط ها را نظر ویژگی های " کاملاً مشاهده پذیر / نیمه مشاهده پذیر " ، " قطعی / تصادفی " ، " مرحله ای / ترتیبی " ، " تک عاملی / چند عاملی " ، " ایستا / پویا " ، " گسسته / پیوسته " ، " شناخته شده / شناخته نشده " بررسی کنید.

|                                  | Performance measurement                             | Environment  | Actuators  | Sensors  |
|----------------------------------|---|--|--|--|
| سیستم تشخیص چهره (apple face ID) | تشخیص صحیح افراد مجاز                               | سیستم عاملی که برنامه روی آن نصب شده است ، فرد مورد آزمایش | انجام الگوریتم تشخیص چهره روی تصویر گرفته شده از دوربین    | دوربین برای عکس برداری از صورت   |
| سیستم autopilot هواپیما          | هدایت هواپیما با امنیت کامل طبق دستورالعمل داده شده | هواپیما و سخت افزار کامپیوتری آن ، آسمان پرواز ، خلبان     | تنظیم جهت و ارتفاع پرواز ، تنظیم سرعت پرواز                | حس گر اندازه گیری ارتفاع و فشار هوا ، مکان یاب ، رادار                   |
| ربات امدادگر                     | تشخیص افراد زنده در محل ، جابه جایی آوار            | مکان تخریب شده   | چرخ برای حرکت ، بازو برای جابه جایی آوار                   | دوربین مادون قرمز برای تشخیص فرد زنده                                    |
| تمرین tennis در مقابل دیوار      | افزایش آمادگی بدنی و تجربه و حرفه بیشتر             | محل تمرین (سالن ورزشی)                                     | راکت برای ضربه زدن به توپ ، محرک برای جابه جایی به سمت توپ | بینایی برای دیدن محل توپ   |
| بازی mench                       | برنده شدن در بازی                                   | صفحه بازی ، بقیه بازی کتان ، تاس و مهره های بازی           | تاس ریزی ، حرکت دادن مهره ها                               | بینایی برای تشخیص عدد تاس ، بینایی برای مشاهده محل مهره های خود و حریفان |

|                                  | observable | agents | Deterministic | Episodic   | Static      | Discrete   | Known   |
|----------------------------------|------------|--------|---------------|------------|-------------|------------|---------|
| سیستم تشخیص چهره (apple face ID) | Fully      | Single | Deterministic | Episodic   | semiDynamic | Continuous | Known   |
| سیستم autopilot هواپیما          | Partially  | Single | Stochastic    | Sequential | Dynamic     | Continuous | Unknown |
| ربات امدادگر                     | Partially  | Single | Stochastic    | Sequential | Dynamic     | Continuous | Unknown |

|                                |       |        |            |            |         |            |       |
|--------------------------------|-------|--------|------------|------------|---------|------------|-------|
| تمرین tennis<br>در مقابل دیوار | Fully | Single | Stochastic | Sequential | dynamic | Continuous | Known |
| بازی mench                     | Fully | Multi  | Stochastic | Sequential | Static  | Discrete   | Known |

توجه : با توجه به نگرش شما در مورد محیط وظیفه هر یک از عامل ها ، پاسخ این سوال می تواند متفاوت باشد.

۵. عامل جاروبرقی را در یک جدول  $n \times n$  در نظر بگیرید که هر خانه میتواند خیلی کثیف ، کثیف و یا تمیز باشد . عامل در لحظه اولیه می تواند در هریک از این خانه ها قرار داشته باشد . فرض کنید که عامل در هر لحظه از زمان می تواند تنها وضعیت کثیف و یا تمیز بودن خانه خود و مکان فعلی خود را مشاهده کند . action های عامل به صورت " رفتن به بالا" ، " رفتن به پایین" ، " رفتن به چپ" ، " رفتن به راست" و " مکش" می باشد. در صورتی که عمل مکش روی خانه خیلی کثیف اعمال شود ، آن خانه تبدیل به خانه کثیف شده و با اعمال مکش روی خانه کثیف ، آن را به خانه تمیز تبدیل می کند .

آ) در صورتی که هدف عامل تمیز کردن کل جدول باشد ، آیا عاملی که به صورت واکنشی مبتنی بر مدل عمل می کند ، می تواند عملکرد کاملاً منطقی داشته باشد ؟ اگر بله چگونه و اگر خیر چرا ؟

بله . از آنجایی که معیار کارایی تنها تمیز کردن کل جدول است ، می توان ۲ state برای عامل تعریف کرد که در هر کدام از این حالات طبق منطق جداگانه ای عمل کند : state اولیه و state ثانویه . به طوری که وقتی عامل شروع به کار می کند در state اولیه باشد و وقتی به خانه (1 , 1) رسید حالتش را به ثانویه تغییر دهد .

در حالت اولیه عامل سعی می کند به خانه (1 , 1) برود به طوری که اگر location فعلی آن را (i , j) در نظر بگیریم ، در صورتی که i بزرگتر از 1 باشد به چپ رفته و در صورتی که j بزرگتر از ۱ باشد به پایین می رود و زمانی که i و j هر دو برابر 1 شدند حالتش را به ثانویه تغییر می دهد .

در حالت ثانویه عامل شروع به تمیز کردن کل جدول می کند به طوری که تا اگر locarion فعلی آن (i , j) باشد ، تا تمیز شدن کامل آن ، عملیات تمیز کردن را انجام داده و سپس اگر j فرد باشد تا زمانی که  $i \neq n$  است به راست رفته و سپس یک واحد به بالا می رود ، اگر j زوج باشد تا زمانی که  $i \neq 1$  است به چپ رفته و سپس یک واحد به بالا می رود و در نهایت پس از رسیدن به خانه (n , n) اتمام عملیات خود را اعلام کند.

ب) حال فرض کنید بین هر دو خانه مجاور می تواند دیوار وجود داشته باشد که در اینصورت نمی توان از یکی به دیگری رفت . در صورتی که هدف عامل تمیز کردن جدول باشد و انجام هر action یک امتیاز منفی برای عامل داشته باشد ، یک مدل منطقی برای این منظور طراحی کرده و نوع عامل را از جهت رفتار ( مثلاً واکنشی ساده یا مبتنی بر مدل ) مشخص نمایید.

بهترین مدل برای این عامل ، مبتنی بر یادگیری است . باتوجه به تعریف می‌دانیم که عامل باید کل جدول را با انجام کمترین عمل تمیز کند. بر این مبنا اگر عامل در خانه خیلی کثیف باشد آن را ۲ بار تمیز کرده و اگر در خانه کثیف باشد آن را یک بار تمیز می‌کند و همچنین باید کل جدول را با کوتاه‌ترین مسیر بپیماید . بدین منظور عامل نقشه ای از جدول ساخته و با حرکت روی آن ، نقاط پیموده شده و دیوارها را بر روی نقشه خود اضافه می‌کند . چون عامل مبتنی بر یادگیری است پس از برخورد با چندین دیوار می‌تواند و همچنین پس از پیمودن مسیری می‌تواند احتمال وجود دیوار در خانه‌های دیگر و همچنین کوتاه‌ترین مسیر احتمالی را حدس زده و در خانه‌های ناشناخته action محتمل تری برای عدم برخورد به دیوارها و همچنین پیمایش کل جدول انتخاب کند.

۶. با توجه به تعریف تابع عامل و برنامه عامل به سوالات زیر پاسخ کامل بدهید و دلایل را ذکر کنید.

(آ) آیا ممکن است بیش از یک برنامه عامل برای پیاده‌سازی یک تابع عامل وجود داشته باشد؟

همانطور که می‌دانیم برنامه عامل یک پیاده‌سازی درونی از تابع عامل است. از آن جا که می‌توان یک الگوریتم را به چندین صورت متفاوت پیاده‌سازی کرد ، یک تابع عامل می‌تواند بیش از یک برنامه عامل داشته باشد . همچنین برای پیاده‌سازی یک تابع عامل برای معماری های مختلف ممکن است نیاز به برنامه‌های عامل مختلف داشته باشیم.

(ب) آیا تابع عاملی وجود دارد که نتوان آن را با هیچ برنامه عاملی پیاده سازی کرد ؟

بله . برای مثال عاملی را فرض کنید که هر بار یک بیت گرفته و بر اساس عددی که بیت های گرفته شده از ابتدا تا کنون نشان می‌دهند یک عدد صحیح بر می‌گرداند . بنابراین این عامل در نهایت با کمبود حافظه مواجه شده و در نتیجه برنامه عامل آن با شکست مواجه می‌شود.

(پ) با داشتن معماری ثابت عامل، آیا هر برنامه عامل دقیقاً یک تابع عامل مشخص را پیاده‌سازی می‌کند؟

هر عامل شامل یک معماری و برنامه عامل است. فرض کنید یک عامل دارای معماری ثابتی باشد ، در این صورت عامل به ازای هر ورودی از دنباله مشاهدات، رفتار معینی را نشان خواهد داد. در نتیجه می‌توان گفت هر برنامه عامل یک تابع عامل مشخص را پیاده‌سازی می‌کند.

(ت) فرض کنید معماری عامل ، شامل  $n$  بیت حافظه باشد . این عامل حداکثر چند برنامه عامل می‌تواند داشته باشد؟

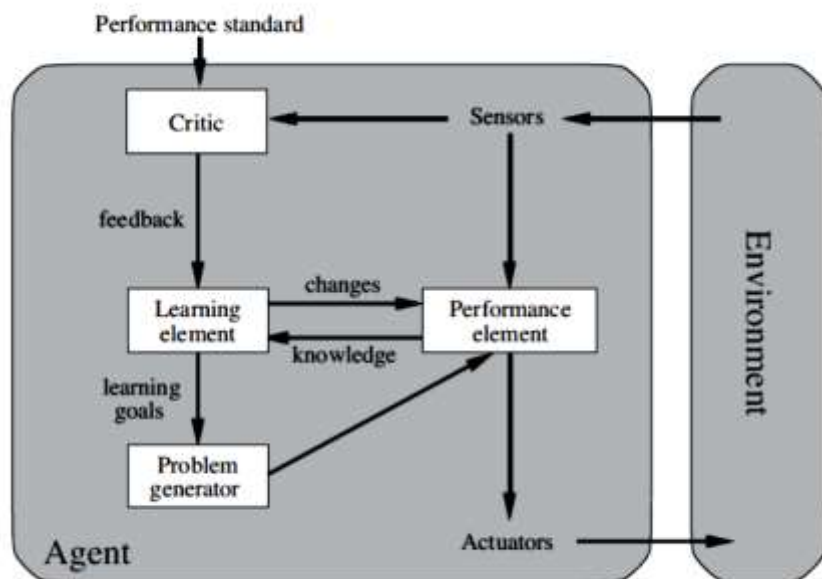
( تعداد کل action هایی که عامل می‌تواند انجام دهد را  $a$  در نظر بگیرید . )

اگر حافظه شامل  $n$  بیت باشد بنابراین  $2^n$  حالت داخلی می‌توان داشت که در هر کدام  $a$  عمل قابل انجام است. بنابراین  $a^{2^n}$  برنامه عامل می‌توان داشت.

**ث)** فرض کنید با ثابت نگه‌داشتن برنامه عامل، سرعت ماشین را دو برابر کنیم. آیا تابع عامل تغییر خواهد کرد؟ در صورتی که محیط ایستا باشد و برنامه به گذشت زمان توجهی نداشته باشد تابع عامل ثابت خواهد بود. در صورتی که محیط پویا باشد، بالا بردن سرعت دستگاه می‌تواند منجر به تصمیمات متفاوت (احتمالا بهتر) یا انجام سریع‌تر عکس‌العمل شود. پس تابع عامل تغییر می‌کند.

**۷.** معماری عامل یادگیرنده را برای عامل هوشمند ربات گارسون در نظر بگیرید و کلیه بخش‌ها آن شامل اجزای کارکردی و پیام‌های رد و بدل شده بین این اجزا را به طور خاص برای این عامل تعیین کنید. (ربات گارسون وظیفه سفارش‌گیری و تحویل آن به میزهای مختلف در مکان‌های مختلف را بر عهده دارد.)

(Environment, sensor, performance element, performance standard, critic, learning element, learning goals, knowledge, changes, actuators)



: Environment

فضای رستوران، آشپزخانه، میز و صندلی‌ها، مشتریان و سایر افراد درون محیط رستوران

: sensors



دوربین local روی خود ربات ، دوربین global محیط بر رستوران و یا موقعیت یاب برای مکان یابی ربات ، گیرنده صوتی ، صفحه لمسی برای سفارش گیری

: performance element

این واحد با توجه به feedback هایی که از واحدهای دیگر می گیرد ، در نهایت تصمیم می گیرد که عامل در هر لحظه چه فعالیتی انجام دهد . در این مثال ، این واحد بخشی از سیستم کامپیوتری و برنامه ی ربات است.

: performance standard

این بخش معیارهایی را برای کارایی مناسب سیستم مشخص می کند . در اینجا : سفارش گیری به موقع از مشتریانی که تازه وارد رستوران شده اند ، تحویل سریع غذای هر میز ، انتخاب بهترین مسیر برای جابه جایی ، عدم برخورد با موانع و مشتریان در حین حرکت

: critic

این عنصر با توجه به ادراکاتی که از سنسورها می گیرد و همچنین استاندارد کارایی ربات ، میزان موفقیت عامل را مشخص کرده و به واحد یادگیری گزارش می کند . برای مثال دیر رسیدن به مقصد را به علت موانع سر راه و انتخاب مسیر اشتباه توسط ربات به واحد یادگیری گزارش می شود.

: learning element

این واحد با توجه به feedback دریافتی از عنصر منتقد ، معیارهای عنصر کارایی سیستم که تصمیم گیرنده اعمال عامل است را تغییر می دهد . در این جا ، برای مثال با توجه به موانع سر راه ربات در طی مسیرش که می توانند به طور پویا تغییر کنند ( حرکت مشتریان در رستوران ) ، ربات بعد از یادگیری می تواند مسیر پیش فرضی که برای رسیدن به میزهای مختلف طی می کرده را تغییر داده تا معیار کارایی خود را افزایش داده و سریع تر وظایفش را انجام دهد .

: learning goals

اهدافی هستند که توسط واحد یادگیری به منتقد گزارش شده تا منتقد راهی برای آن پیدا کند . در اینجا ، برای مثال بهبود مسیر حرکت بین آشپزخانه و میز مشتری در زمان شلوغی رستوران

: knowledge

دانشی است که در حال حاضر نحوه تصمیم گیری عامل را مشخص می کند . واحد یادگیری این دانش را از عنصر کارایی دریافت کرده و براساس آن تصمیم می گیرد که چه تغییراتی بر روی آن ایجاد کند . در اینجا ، برای مثال ترتیب پیش فرضی که عامل برای سفارش گیری از چندین میز در نظر دارد.

: changes

تغییراتی است که توسط واحد یادگیری در واحد کارایی و برنامه عامل برای نتیجه بهتر اعمال می‌شود. در اینجا، برای مثال چگونگی و ترتیب سفارش‌گیری از چندین میز با هدف کمینه شدن زمان کل سفارش‌گیری

: problem generator

این واحد باتوجه به اهدافی که از واحد یادگیری دریافت می‌کند، اقدامات تازه‌ای برای شناخت جدید و عملکرد بهتر ربات به واحد کارایی پیشنهاد می‌کند. برای مثال، مسیر متفاوتی برای رفتن از میز  $n$  به میز  $m$  به امید زودتر رسیدن نسبت به مسیر قبلی پیشنهاد می‌کند.

: actuators

چرخ‌ها برای حرکت ربات، پخش‌کننده صدا، صفحه نمایش برای نشان دادن منو، بازو برای جابه‌جایی غذا