به نام خدا

تکلیف اول درس سیستم‌های چندعاملی

مرضیه علیدادی ۸۱۰۱۰۱۲۳۶

۱ سوال اول

۱.۱ **ubiquity** (حضور در همه جا در یک زمان)

با گذشت زمان، هزینه‌های قدرت محاسباتی کاهش یافته‌است. این مسئله موجب آسان‌تر شدن ترکیب قابلیت‌های پردازش در دستگاه‌ها و مکان‌هایی که قبلاً غیرعملی یا شنیده نشده بود، شده‌است. با ادامه‌ی این روند، دسترسی به قدرت پردازش و هوشمندی، گسترده‌تر و رایج‌تر خواهد شد. به علاوه، افزایش همه جانبه‌ی قابلیت‌های پردازشی موجب به وجود آمدن سطح مشخصی از هوش در بسیاری از جنبه‌های زندگی می‌شود.

۲.۱ **interconnection** (ارتباط متقابل)

سیستم‌های کامپیوتری در ابتدا یک سری موجودیت مجزا بودند. سپس دچار تکامل شدند و با هم ارتباط متقابل برقرار کردند و بخشی از سیستم‌های توزیع شده‌ی بزرگ را تشکیل دادند. اینترنت یک نمونه‌ی بارز این ارتباط متقابل است که یافتن کامپیوترهای بدون دسترسی به اینترنت نادر شده‌است. همچنین، در گذشته، سیستم‌های توزیع‌شده و همروند، پیچیده دیده می‌شدند و به همین دلیل از آن‌ها اجتناب می‌شد. با این حال، رشد اینترنت این تصور را تغییر داد و استفاده از سیستم‌های توزیع‌شده و همروند در محاسبات تجاری و صنعتی مرسوم شد.

۳.۱ **intelligence** (هوش)

پیشرفت‌های فناوری به ما این امکان را داده‌است که کارهای پیچیده‌تر را به طور خودکار به کامپیوترها واگذار کنیم. درک ما از سیستم‌های کامپیوتری بهبود یافته‌است، و ما توان انجام کارهایی را که قبلا غیرقابل تصور تلقی می‌شدند، پیدا کرده‌ایم.

۴.۱ **delegation** (واگذاری)

روند رو به رشدی به سمت واگذاری وظایف به سیستم‌های کامپیوتری، حتی در زمینه‌های حیاتی مانند خلبانی هواپیما، وجود دارد. این بدان معناست که وظایفی که به طور سنتی توسط انسان‌ها انجام می‌شد، مانند خلبانان با تجربه، اکنون به برنامه‌های کامپیوتری سپرده می‌شود. مفهوم واگذاری اختیار در اینجا به واگذاری قابلیت‌های کنترل و تصمیم‌گیری به سیستم‌های کامپیوتری اشاره دارد.

۵.۱ **human-orientation** (انسان‌محوری)

روند تعامل انسان-کامپیوتر، از دیدگاه‌های ماشین‌محور به سمت مفاهیمی که بیشتر با نحوه‌ی درک انسان از جهان مطابقت دارد، پیش رفته‌است. در ابتدا، کاربران مجبور بودند سوئیچ‌های دستگاه‌ها را به صورت دستی تغییر دهند و عملکرد داخلی آن‌ها را درک کنند. این مسئله با معرفی رابط‌های خط فرمان تغییر کرد، جایی که کاربران می‌توانستند با صدور دستورالعمل‌ها با کامپیوترها تعامل داشته باشند. سپس این رابط‌ها با رابط‌های گرافیکی جایگزین شدند، جایی که کاربران می‌توانستند مستقیماً آیکون‌های روی دسکتاپ را برای کنترل دستگاه تغییر دهند.

به طور مشابه، در برنامه نویسی، برنامه نویسان اولیه مجبور بودند با کد ماشین خام کار کنند، که نیاز به درک عمیق ساختار داخلی کامپیوتر داشت. با گذشت زمان، زبان‌های برنامه‌نویسی سطح بالاتر مانند زبان‌های اسمبلر، رویه‌ای، انواع داده‌های انتزاعی و objectها پدیدار شدند. این پیشرفت‌ها به برنامه‌نویسان اجازه می‌دهد تا با انتزاع‌های انسان‌محور بیشتری کار کنند و توسعه‌ی نرم‌افزارها آسان‌تر شود.

به طور کلی، روند برنامه‌نویسی و تعامل انسان-کامپیوتر تکامل یافته تا رابط‌ها و ابزارهایی را فراهم کنند که با درک انسان همسو می‌شوند و استفاده از کامپیوتر را بصری‌تر و سازنده‌تر می‌کنند.

۲ سوال دوم

	شباهت‌ها	تفاوت‌ها
سیستم‌های موازی	۱. سیستم‌های چندعاملی یک زیرمجموعه از سیستم‌های موازی هستند. ۲. در پیاده‌سازی هر دو نوع این سیستم‌ها باید مسائلی مانند انحصار متقابل، بن‌بست و ... را درنظر داشت.	۱. سیستم‌های موازی برخلاف سیستم‌های چندعاملی، خودمختاری ندارند. ۲. سیستم‌های موازی برخلاف سیستم‌های چندعاملی، self-interested نیستند.
هوش مصنوعی	۱. هر دو دارای هوشمندی هستند.	۱. سیستم‌های چندعاملی در هر زمان فقط یک کار خاص را می‌خواهد انجام دهد. ۲. هوش مصنوعی کلاسیک برخلاف سیستم‌های چندعاملی، دارای جنبه‌های اجتماعی نیست.
تئوری بازی‌ها	۱. ابزارها و تکنیک‌های موجود در تئوری بازی‌ها کاربرد زیادی در سیستم‌های چندعاملی دارند.	۱. مفهوم منطق و عامل منطقی در این دو مطابقت ندارد. ۲. در سیستم‌های چندعاملی بر خلاف تئوری بازی‌ها، بهینگی زمان و هزینه‌ی کارها بررسی شده و اهمیت دارد.

	شباهت‌ها	تفاوت‌ها
علوم اجتماعی	۱. دانشمندان هر دو حوزه، جوامع انسانی را برای مدل‌سازی‌ها و بررسی‌های خود در نظر می‌گیرند.	۱. برای سیستم‌های چندعاملی، بهترین کار استفاده از جوامع انسانی نیست. ۲. سیستم‌های چندعاملی ابزاری قدرتمند برای مدل‌سازی و درک جوامع ارائه می‌کنند، در حالی که علوم اجتماعی مخزن غنی از مفاهیم برای درک و ساختن سیستم‌های چندعاملی را نشان می‌دهند.

۳ سوال سوم

۱.۳ قابل دسترس (**accessible**) / غیر قابل دسترس (**inaccessible**)

یک محیط قابل دسترس محیطی است که در آن عامل می‌تواند اطلاعات کامل، دقیق و به‌روزی در مورد وضعیت محیط بدست آورد. اکثر محیط‌های دنیای واقعی مثلا دنیای روزمره و اینترنت، با این تعریف قابل دسترس نیستند.
محیط غیر قابل دسترس هم دارای تعریفی برعکس است.

۲.۳ قطعی (**deterministic**) / غیر قطعی (**non-deterministic**)

یک محیط قطعی محیطی‌ست که در آن هر عمل دارای یک اثر تضمین‌شده‌ی واحد است. و هیچ ابهامی در مورد وضعیتی که پس از انجام یک عمل حاصل می‌شود، وجود ندارد.
محیط غیر قطعی هم دارای تعریفی برعکس است.

۳.۳ ایستا (**static**) / پویا (**dynamic**)

محیط ایستا محیطی‌ست که اگر عملی توسط یک عامل در آن صورت نگیرد، می‌توان فرض کرد که بدون تغییر باقی می‌ماند.
در مقابل، یک محیط پویا محیطی‌ست که دارای فرآیندهای دیگری است که بر روی آن کار می‌کنند، و از این رو به شکلی خارج از کنترل عامل تغییر می‌کند. مثلا اینترنت محیطی پویا است.

۴.۳ گسسته (**discrete**) / پیوسته (**continuous**)

اگر تعداد ثابت و محدودی از اعمال و ادراک در محیط وجود داشته باشد، آن محیط گسسته است.
محیط پیوسته هم دارای تعریفی برعکس است.

۵.۳ اپیزودیک (**episodic**) / غیر اپیزودیک (**non-episodic**)

در یک محیط اپیزودیک، عملکرد یک عامل به تعدادی اپیزود گسسته وابسته است، بدون اینکه ارتباطی بین عملکرد عامل در اپیزودهای مختلف وجود داشته باشد.

۴ سوال چهارم

۱.۴ **reactivity** (واکنش‌پذیری)

عامل‌های هوشمند می‌توانند محیط خود را درک کنند و به تغییراتی که در آن رخ می‌دهد به موقع پاسخ دهند، تا اهداف طراحی خود را برآورده سازند.

۲.۴ **proactivity** (فعال بودن)

عامل‌های هوشمند می‌توانند با ابتکار عمل به منظور برآورده ساختن اهداف طراحی خود، رفتاری هدفمند از خود نشان دهند.

۳.۴ **social ability** (توانایی اجتماعی)

عامل‌های هوشمند توانایی تعامل با سایر عامل‌ها (و احتمالاً انسان‌ها) را دارند، تا اهداف طراحی خود را برآورده سازند.

۵ سوال پنجم

۱.۵ شباهت‌ها

شیء‌ها به عنوان موجودیت‌های محاسباتی تعریف می‌شوند، که به منظور کپسوله کردن برخی حالت‌ها به کار می‌روند. و می‌توانند actionها یا methodهایی را انجام دهند و همچنین با ارسال پیام ارتباط برقرار کنند.

۲.۵ تفاوت‌ها

۱.۲.۵ میزان استقلال

دلیل ایجاد شیءها و کپسوله کردن برخی قسمت‌ها این بود که هر شیء بتواند بر وضعیت داخلی خود کنترل داشته باشد. معمولا methodهای موجود در هر شیء می‌توانند به صورت عمومی یا خصوصی تعریف شوند. که این مسئله تعیین می‌کند که آیا فقط از داخل خود آن شیء قابل استفاده باشند یا از طریق شیءهای دیگر هم بتوان آن‌ها را صدا زد و از آن‌ها استفاده کرد. این بدین معناست که هر شیء بر وضعیت داخلی خود خودمختاری و کنترل دارد، ولی بر رفتار خود کنترلی ندارد. زمانی که یک شیء یک method از خود را عمومی کرد، دیگر هیچ کنترلی بر اجرا یا عدم اجرای آن ندارد.

۲.۲.۵ هدف

وقتی سیستمی را بسازیم، و شیءهایی را که در آن قرار می‌گیرند طراحی کنیم، معمولاً فرض می‌کنیم که آن‌ها یک "هدف مشترک" دارند. اما در بسیاری از انواع سیستم‌های چندعاملی، چنین هدف مشترکی را نمی‌توان در نظر گرفت.

۳.۲.۵ خودمختاری در تصمیم‌گیری

در مورد شیءها، یک شیء می‌تواند یک method با دسترسی عمومی از یک شیء دیگر را فراخوانی کند. و آن شیء دیگر، اختیاری از منظر اجازه/عدم اجازه‌ی فراخوانی آن method ندارد، هر چند که به ضررش باشد. اما درباره‌ی عامل‌ها، این مسئله به روش "درخواست کردن" رخ می‌دهد. یک عامل وقتی درخواست می‌کند که یک method از یک عامل دیگر را فراخوانی کند، آن عاملِ دیگر ممکن است آن method را اجرا کند یا نکند.
بنابراین، کنترل اجرای methodها در عامل‌ها و شیءها متفاوت است. در شیءها تصمیم‌گیری با شیء فراخوانی‌کننده است، ولی در عامل‌ها تصمیم‌گیری با عاملی‌ست که درخواست را دریافت می‌کند.

۴.۲.۵ خودمختاری در انعطاف‌پذیری رفتار

در مدل استاندارد شیءگرایی امکان پیاده‌سازی یکپارچه‌ی سیستم و انعطاف در رفتار (از منظر واکنش‌پذیری، فعال بودن یا توانایی اجتماعی) وجود ندارد.

۵.۲.۵ همزمانی

عامل‌ها هر کدام دارای رشته کنترل خود هستند، در صورتی که در مدل استاندارد شیءگرایی، یک رشته کنترل در سیستم وجود دارد.

۶ سوال ششم

یک باغ را به عنوان یک سیستم چندعاملی و افرادی با وظیفه‌ی میوه‌چینی را به عنوان عامل‌های این سیستم در نظر می‌گیریم.

عامل‌های موجود در این سیستم، افراد میوه‌چین هستند. این عامل‌ها مستقل خواهند بود؛ به این معنی که می‌توانند وظایف را بدون دخالت سایر عامل‌ها یا یک مقام مرکزی انجام دهند.

محیط شامل خود باغ و انواع درخت‌های موجود در آن و یک سیستم مدیریت متمرکز است که دستورالعمل‌ها را برای عامل‌ها ارسال می‌کند.

حالت‌های محیط شامل حالت فعلی هر درخت (اینکه کدام میوه رسیده و آماده چیدن است)، شرایط آب‌وهوایی (اینکه بارانی یا آفتابی است) و سلامت کلی باغ است.

اعمال قابل انجام توسط عامل‌ها شامل چیدن میوه از درخت، گزارش وضعیت یک درخت معین به سیستم مدیریتی و درخواست نگهداری یا تعمیر یک درخت معین در صورت لزوم است.

تابع انتقال حالت، وضعیت محیط (شامل وضعیت هر درخت، همراه با سلامت کلی باغ و هرگونه درخواست نگهداری لازم) بر اساس اقدامات انجام شده توسط عامل‌ها را به‌روزرسانی می‌کند. این تابع را می‌توان به عنوان مجموعه‌ای از قوانین انتزاع کرد که نحوه‌ی تغییر حالت محیط را بر اساس اقدامات انجام شده توسط عامل‌ها کنترل می‌کند. به عنوان مثال، اگر یک عامل تمام میوه‌های رسیده را از درخت بچیند، وضعیت درخت تغییر می‌کند و محصول کلی باغ کاهش می‌یابد.

حالت اولیه‌ی محیط می‌تواند یک باغ شامل درخت‌هایی سالم دارای تعداد میوه‌های مختلف با درجات مختلف رسیدگی باشد.

حالات پایانی می‌تواند یک باغ شامل درخت‌هایی باشد که میوه‌های آن‌ها کاملاً چیده شده‌است. و همچنین یک گزارش از محصول کلی باغ و هرگونه درخواست نگهداری یا تعمیر درخت‌ها از طرف عامل‌هاست.

تابع عامل را می‌توان به عنوان مجموعه‌ای از فعالیت‌های احتمالی شامل دریافت دستورالعمل از سیستم مدیریت و استفاده از حسگرها و ابزارها برای چیدن میوه و گزارش وضعیت درخت‌ها، انتزاع کرد.

یک run دلخواه برای عامل: اجرای مورد نظر برای یک عامل را می‌توان به عنوان دنباله‌ای از اقدامات انجام شده توسط عامل، با دریافت دستورالعمل از سیستم مدیریت مرکزی، تصمیم‌گیری برای تشخیص میوه‌های رسیده روی درخت، تصمیم‌گیری برای استفاده از ابزارهای مکانیکی برای چیدن میوه، و جمع آوری داده‌ها در مورد شرایط و درخواست تعمیرات، انتزاع کرد. مثلاً عامل دستورالعمل‌هایی را برای چیدن میوه از یک مجموعه‌ی خاص از درخت‌ها دریافت می‌کند و تصمیم می‌گیرد این کار را انجام دهد. با نزدیک شدن به اولین درخت، عامل احساس می‌کند که یک میوه‌ی رسیده در دسترس است و دست خود را برای چیدن میوه دراز می‌کند. سپس میوه را در سبدی که حمل می‌کند قرار می‌دهد، و به سمت درخت بعدی در مجموعه حرکت می‌کند. هنگامی که میوه می‌چیند، وضعیت هر درخت را یادداشت می‌کند و گزارش‌هایی را به سیستم مدیریت ارسال می‌کند که کدام درخت‌ها بیشترین میوه را تولید می‌کنند، کدام درخت ممکن است نیاز به نگهداری داشته باشد و ... . هنگامی که مجموعه‌ی درخت‌ها به طور کامل برداشت شد، عامل به سیستم

اصلی بازمی‌گردد تا میوه‌های خود را سپرده و درخواست تعمیرات لازم را بدهد.

۷ سوال هفتم

۱.۷ الف)

سودمندی یک مقدار عددی است که نشان می‌دهد یک وضعیت چقدر خوب است. هر چه عدد مربوط به سودمندی بالاتر باشد، بهتر است. وظیفه‌ی عامل این است که به نوعی رفتار کند که سودمندی حداکثر شود. ما برای عامل مشخص نمی‌کنیم که چگونه این کار انجام شود؛ بلکه تابع سودمندی تعریف می‌شود.

دو نوع تابع سودمندی تعریف می‌شود:
۱. تابع سودمندی برای هر حالت از محیط: به حالت‌های مختلف محیط، سودمندی نسبت داده می‌شود. وظیفه‌ی عامل ایجاد حالت‌هایی است که سودمندی را به حداکثر می‌رسانند.
۲. تابع سودمندی برای هر run (هر دنباله‌ی ممکن از حالت‌ها و اقدامات): به جای تعیین سودمندی برای حالت‌های مختلف محیط، به runها سودمندی نسبت داده می‌شود.

۲.۷ ب)

۱. تابع سودمندی برای هر حالت از محیط: این تابع، سودمندی وضعیت فعلی باغ، از جمله عواملی مانند سلامت کلی درخت‌ها، در دسترس بودن میوه‌های رسیده و نیاز به نگهداری یا تعمیرات را ارزیابی می‌کند. این تابع سودمندی به هر حالت ممکن یک مقدار عددی به عنوان سودمندی اختصاص می‌دهد که نشان‌دهنده‌ی سودمندی کلی آن حالت برای عامل است. به عنوان مثال، حالتی که در آن همه‌ی درخت‌ها سالم هستند و میوه‌های رسیده‌ی فراوانی دارند، مقدار سودمندی بالایی دارد.
۲. تابع سودمندی برای هر run (هر دنباله‌ی ممکن از حالت‌ها و اقدامات): تابع سودمندی به هر دنباله از حالت‌ها و اقدامات ممکن با در نظر گرفتن تولید کلی میوه‌ها، میزان چیدن میوه، گزارش موفقیت‌آمیز وضعیت درخت‌ها و درخواست‌های نگهداری، یک مقدار اختصاص می‌دهد که نشان دهنده‌ی سودمندی کلی آن دنباله برای عامل است. به عنوان مثال، به دنباله‌ای از اقدامات که منجر به تولید میوه‌ی بالا، میزان زیاد چیدن میوه، گزارش دقیق وضعیت درخت‌ها و درخواست‌های درست نگهداری می‌شود، مقدار سودمندی بالایی نسبت داده می‌شود، که نشان می‌دهد دنباله‌ای سودمند برای اجرا توسط عامل‌هاست.