ИИ, който мисли наум

Dimiter Dobrev

Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, *d@dobrev.com*

AGI трябва да разбира света. За целта ни е нужен модел на света. За да намерим този модел ни е нужен език за описание на светове. Ще вземем света на играта шах и ще опишем този свят. Това вече сме го правили в предишна статия, но тогава агентът виждаше таблото, а сега той ще играе blind. Когато не виждате таблото, задачата е по-сложна и изисква добавянето на абстрактни събитийни модели. Резултатът ще е модел на света, чрез който AGI ще може да мисли наум и да планира действията си.

1. Въведение

За да играе шах AGI трябва да разбере света. Трябва да разбере състоянието на света (позицията на таблото) и правилата на играта. В този свят има и противник, който играе срещу агента. Моделът на света се състои от множеството от вътрешните състояния и една функция, която описва правилата на играта и противника.

В статията [2] ние вече направихме модел на играта шах. Този модел беше създаден на базата на събитийните модели (ED models). Тук няма да се занимаваме с въпроса какво е събитие и какво е събитиен модел. На тези въпроси сме дали отговор в [1], макар че има още какво да се желае. Тук няма да даваме точна дефиниция на тези понятия, а само ще дадем примери, с които ще илюстрираме идеята.

Новото в тази статия е, че разделяме събитийните модели на два вида – реални и абстрактни. Пример за реален събитиен модел това са дните от седмицата. Това е модел със седем състояния и събитието, което ги превключва е „полунощ“. Това е реален модел, защото състоянията са нещо реално и във всеки момент светът е в някое от тези състояния (в някой от дните от седмицата).

Пример за абстрактен събитиен модел това са остатъците при деление на седем (т.е. числата от 0 до 6). Пак имаме модел със седем състояния. Събитието, което ги превключва ще бъде „следващ“. Двата модела много си приличат, но вторият е абстрактен, защото състоянията и събитието, което ги превключва са абстрактни.

В [2] ние описахме шахматната дъска като използвахме реални събитийни модели. Тук играем blind и затова ще се наложи да използваме абстрактни модели.

Ще разгледаме три варианта на играта шах. Първият вариант ще бъде по-прост, защото агентът ще играе сам като обръща таблото и играе последователно и с бели и с черните. Този вариант ще е по-прост, защото в света няма да има друг агент (противник).

Във втория вариант ще имаме двама играчи, които след всяка партия си сменят местата. В третия вариант агентът ще играе винаги с белите. Третият вариант ще е най-труден за разбиране, защото там агентът ще трябва да разбере за съществуването на черните фигури, въпреки че никога не се е докосвал до тях.

Защо е важно разбирането (моделът на света)? Големите езикови модели (LLM) не могат да играят шах, защото не могат да разберат позицията на таблото. Това не трябва да ни учудва. LLM не могат да направят дори и нещо съвсем елементарно като събирането на две произволни числа. Тоест, без разбиране няма мислене, а само имитация на мислене. (Много са хората, които подържат тази теза. Например прочетете мнението на Gary Marcus [3].)

Освен разбирането ни е нужно още и мислене наум. За да съберем две произволни числа трябва да изпълним алгоритъм. Когато изпълняваме алгоритъм ние не знаем колко време ще отнеме това. Може времето да е дълго, а може дори алгоритъмът никога да не завърши. Затова мисленето наум трябва да е асинхронно, за да не блокира работата на нашия AGI.

2. Модел на света

Моделът на света, който ще търсим, ще има две особености. Първо той ще включва полуразрешими предикати и второ, той ще е недетерминиран.

1. Можем да си мислим, че множеството от вътрешните състояния е $\mathbb{N}$ (или че можем да ги кодираме в $\mathbb{N}$). Можем да си мислим, че функцията на модела е от $\mathbb{N}$ в $\mathbb{N}$ (тук пак използваме кодиране, за да добавим действията и наблюденията).

Тоест въпросът за описанието на света се свежда до това да опишем функциите от $\mathbb{N}$ в $\mathbb{N}$. Тези функции са континуум много и затова не могат да бъдат описани точно. Нужно е да направим компромис и да търсим в по-малко множество от функции. Повечето автори правят голям компромис и предполагат, че търсената функция е рекурсивна (тоест тотална и изчислима). Това се прави например в статиите [4-7]. Ние тук ще направим по-малък компромис и ще предполагаме, че търсим частично рекурсивна функция (тоест изчислима без да е задължително тотална).

Защо няма да държим на това описанието на света да е тотална функция? Ако искаме да емулираме света, тогава функцията трябва да е тотално изчислима и дори трябва да е изчислима за разумно време. Да, но ние не искаме да емулираме света, а само да го опишем, за да можем да планираме бъдещите си действия. Затова в описанието на света можем да използваме полуразрешими предикати. Например в реалния свят ние често използваме правилото: „Ако има доказателство, значи е вярно.“ Предикатът „има доказателство“ е полуразрешим, но въпреки това ние не се притесняваме да го използваме в описанието на реалния свят.

Дали светът е тотално изчислим? Това е философски въпрос, но дори и ако предположим, че това е така, пак ще се нуждаем от полуразрешими предикати. Причината е в това, че ние не търсим функцията f, която описва света, а някаква функция g, която дава една добра стратегия за агента (вижте [8]). Естествено е, когато решаваме дали да тръгнем по някакъв път, да си зададем въпроса дали този път е краен или безкраен, а това е полуразрешим въпрос. Тоест дори и функцията f да е тотално изчислима, търсената функция g вероятно няма да е такава.

2. Ето и втората особеност. Няма да търсим детерминистична функция, защото ще предположим, че в света има случайност и още агенти, чието поведение не може да бъде предсказано напълно. Да търсим детерминистичен модел на света означава, да се опитваме да разберем света напълно и да искаме да кажем точно какво ще се случи. Например в играта шах можем да си мислим, че противникът е една конкретна компютърна програма, да намерим коя точно е тази програма и тогова ще можем да предскажем с абсолютна точност кой ще е следващият ход на противника. Ако имаме случайност, може да предположим, че това не е случайност, а псевдо-случайност, да намерим функцията, която генерира тази псевдо-случайност и тогава ще можем да предскажем точно кое ще е следващото псевдо-случайно число.

Дори и светът да е детерминиран, пак е по-добре да опростим неговото описание като представим агентите като черни кутии. Тоест ще направим някакви предположения за тяхното поведение без да се опитваме да разберем как точно тези агенти работят. Ще

представим тези черни кутии като оракули и моделът на света ще бъде изчислима функция, която използва оракули.

3. Действия и наблюдения

Ще опишем света на играта шах, когато агентът играе blind. Действието на агента ще се състои от координатите на две квадратчета:

$$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle$$

Тези две квадратчета описват хода на агента. (Фигурата, която е на $\langle x_1, y_1 \rangle$ я местим на $\langle x_2, y_2 \rangle$.)

Наблюдението ще има вида:

$$\langle x_3, y_3 \rangle, \langle x_4, y_4 \rangle, Result$$

Това е ходът на противника и резултатът от двата хода. *Result* ще има 8 възможни стойности *{win, loss, draw, correct move, bad 1, bad 2, bad 3, bad 4}*. Тук *win* означава победа за белите, а *loss* означава победа за черните. Защо за некоректен ход сме сложили 4 възможности? Защото искаме да опростим задачата на агента и да му подскажем защо ходът му не е коректен.

bad 1 – в колоната x_1 няма бяла фигура.

bad 2 – на координатите $\langle x_1, y_1 \rangle$ няма бяла фигура.

bad 3 – $\forall y$ ходът $\langle x_1, y_1 \rangle, \langle x_2, y \rangle$ не е коректен.

bad 4 – ходът $\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle$ не е коректен.

(Тук, ако агентът играе с черните, заменяме бяла с черна. Ако са верни няколко от горните твърдения, спираме на първото. Тоест за *bad* избираме най-малкото възможно.)

За да дадем смисъл на този свят, ще въведем награда, която ще зависи само от резултата.

$$reward(R) = \begin{cases} 10, & R = win \\ -10, & R = loss \\ 0, & R = draw \\ 0, & R = correct\ move \\ -400, & R = bad\ 1 \\ -300, & R = bad\ 2 \\ -200, & R = bad\ 3 \\ -100, & R = bad\ 4 \end{cases}$$

При тази дефиниция агентът ще може да провери дали на координати $\langle x_1, y_1 \rangle$ има бяла фигура. Ще извърши действието $\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle$, където x_2 и y_2 са произволни. Ще получи някакъв *Result* и ако *rewards(Result)> -300*, това ще означава, че на координати $\langle x_1, y_1 \rangle$ има бяла фигура.

Това ще бъде едно проверимо събитие. Тоест събитие, което не се наблюдава директно, но може да бъде направен някакъв тест, който да го провери. В случая тестът е да се опитаме да преместим тази бяла фигура.

Това събитие има параметри и затова ще го наричаме предикат. Събитието зависи от това дали агентът играе с белите или с черните фигури. Когато играе с черните, същият

предикат познава дали в квадратчето има черна фигура. Ще представим събитието с предиката:

my_figure(x, y)

Искаме да определим още един предикат:

move(Color, ⟨x_1, y_1⟩, ⟨x_2, y_2⟩)

Този предикат ще е истина, ако можем да преместим фигурата с цвят *Color* от квадрат ⟨x_1, y_1⟩ на квадрат ⟨x_2, y_2⟩ по правилата на играта шах. Това, че можем да преместим фигурата не означава, че ходът ще е коректен, защото имаме още едно правило, което казва, че след преместването не трябва да сме шах.

Следователно събитието *move* няма да е проверимо. Нека агентът играе с цвят *Color* и преместването да може да бъде направено. Тогава почти винаги *rewards(Result)> -100,* но не винаги, защото може след хода да сме шах. Тоест резултатът от теста няма да е сигурен.

Събитие, което не се наблюдава винаги, но се наблюдава често ще наричаме вероятностно събитие. Следователно събитието *move* ще бъде вероятностно проверимо.

Можеше да помогнем още малко на агента и да добавим *bad 5*, като по този начин направим събитието *move* проверимо. Все пак, достатъчно му помогнахме. Нека агентът да може да открива и вероятностни събития.

4. Играе сам

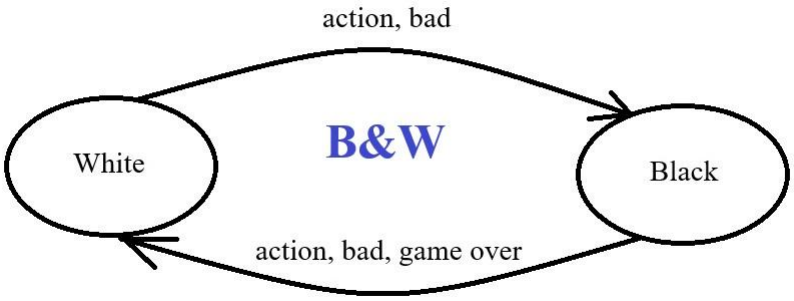
Когато играем blind ще се наложи да описваме света с абстрактни събитийни модели. Ще започнем с един по-прост вариант на играта шах, в чието описание освен абстрактните събитийни модели, ще има и един реален събитиен модел, който ще наречем **B&W** (фигура 1). В този свят агентът ще играе сам срещу себе си. Първо ще играе с белите, после ще обърне дъската и ще играе с черните и т.н.

В този свят агентът, когато играе с белите, трябва да се стреми те да победят и обраното. Това предполага известно раздвоение на личността. Агентът трябва да има две съзнания, защото той трябва да има нещо наум и това нещо трябва да се променя, когато смени фигурите. Агентът не може на всяка стъпка наново да решава накъде да тръгне. Той трябва да си е избрал междинни цели и да е започнал изпълнението на някакви алгоритми. Пример за междинна цел е „размяна на цариците“. Пример за изпълнение на алгоритъм е да се движим към изпълнение на междинната цел. Съзнанието трябва да помни какви междинни цели си е избрал агентът и накъде е тръгнал.

Знаете историята за Буридановото магаре, което стояло между две купчини сено и не могло да избере. Ако магарето няма съзнание, то би могло да блуждае между двете купи, защото на всяка стъпка ще преосмисля избора си и ще сменя посоката. Ние предполагаме, че имаме едно съзнателно магаре, което избира една от двете купчини и стартира алгоритъма за доближаване до избраната купчина. Съзнателното магаре също може да преосмисли избора си, но това няма да се случва на всяка стъпка.

Тук, за да опростим нещата, ще предполагаме, че *rewards* връща нула за *win* и *loss*. Тоест ще предполагаме, че агентът не търси победа, а само се стреми да играе коректни ходове. В този свят няма да има втори агент (опонент) и затова ще предполагаме, че наблюдението се състои само от *Result*.

За да опишем света ще започнем с модела **B&W**, който ни казва с кои фигури играе агентът (фигура 1).



Фигура 1

Тук *action* означава произволно действие на агента, *bad* означава някой от четирите некоректни хода, а *game over* означава *win, loss* или *draw*.

След всяко действие на агента състоянието на модела се превключва и ако ходът е некоректен се превключва повторно и в резултат състоянието си остава същото. Тоест за една стъпка състоянието може да се превключи два пъти. Веднъж се превключва след действието и още веднъж след наблюдението. Не беше ли по-добре състоянието да се превключва само ако действието е коректен ход? Имаме правилото, че ходът е некоректен, ако след него ще сме шах. Заради това правило е добре да можем да си представим, че сме изиграли некоректния ход и ако сме шах, да се върнем обратно.

Имаме още една стрелка по *game over*. Тя е сложена заради това, че след края на играта новата партия трябва да започне с ход на белите.

Целта ни не е просто да опишем играта шах, а да я опишем така, че описанието да може да бъде намерено автоматично. Как може да бъде намерен моделът **B&W**? Събитийните модели се описват с някакви събития, които превключват състоянията им. Освен тези събития имаме нещо, което се нарича *следа* и това са събития, които ни дават възможност да различим състоянията на модела. Ако състоянията на модела бяха еднакви, то този модел нямаше да има никакъв смисъл. Името следата дава смисъла на модела. Това са събития, които се случват в едни състояния, а в други състояния не се случват. Следата може да бъде постоянна и подвижна. Тоест съществуващата особеност може да се появява и изчезва или да се мести. Дали следата е постоянна или подвижна не е толкова важно. Важното е да има някаква следа.

Следата, която ще ни позволи да открием модела **B&W**, е предиката *my_figure(1, 1)*. Този предикат ще е истина докато не направим някакъв коректен ход. После ще е лъжа докато пак не направим коректен ход и така нататък. Тази следа ще е подвижна, защото като преместим белия топ, тя ще изчезне. Важното е, че тази следа, макар и подвижна, ще ни позволи да намерим модела **B&W**.

Разбира се, модел с толкова много стрелки е труден за намиране, но може да забележим, че *my_figure(1, 1)* периодично променя стойността си. След като направим това наблюдение може да започнем да търсим събитията, които превключват това. Не е нужно да намерим наведнъж всички стрелки. Може да ги добавяме постепенно. Тоест може да започнем от един опростен модел и да го усъвършенстваме докато не получим търсеното.

5. Абстрактни събитийни модели

Вече имаме предиката *my_figure* и модела **B&W**. С тяхна помощ можем да направим предикатите *white_figure* и *black_figure*. Например предиката *white_figure* можем да го проверим, когато моделът **B&W** е в състояние *White*. Тоест това е проверим

предикат. (Не е нужно тестът да може да бъде направен по всяко време. Достатъчно е, ако може да бъде направен при определени обстоятелства.)

Вече имаме идея за белите и черните фигури. Сега трябва по някакъв начин да си представим дъската, върху която тези фигури са подредени. Това ще е един абстрактен събитиен модел с 64 състояния. Защо абстрактен, а не реален събитиен модел? Защото състоянията на света няма да са разделени на 64 непресичащи се подмножества. За агента няма да има едно специално квадратче, което да е активното (в което той се намира, което той наблюдава или за което той си мисли).

Състоянията на абстрактния модел ще са абстрактни, но той ще има реална следа. Тоест квадратчето $\langle 1, 1 \rangle$ ще е нещо абстрактно, но там ще има реална фигура. Тази фигура може да се премести, което означава, че говорим за подвижна следа.

Най-естественият начин да получим този абстрактен модел е да вземем дефиниционната област на предиката *my_figure(* x_1, y_1*)*. Това е декартовото произведение на две множества с по 8 елемента. Тук се възползваме от това, че светът ни е зададен по много прост и естествен начин (действието е декартово произведение на 4 множества с по 8 елемента). Бихме могли да предположим, че входът и изходът са кодирани така, че е много трудно да бъдат разбрани (декодирани). Ние ще приемем, че светът е достатъчно сложен и няма нужда да го усложняваме допълнително. Затова ще предположим, че входът и изходът са дадени по възможно най-простия и естествен начин. Затова, когато търсим обяснение, ще избираме най-простото възможно обяснение.

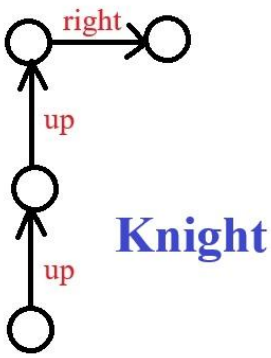
6. Движение на фигурите

Представихме дъската като декартово произведение на две множества с по 8 състояния. Това може да ни опише къде са фигурите, но за да ги местим трябва да въведем абстрактни събития, чрез които да се движим по дъската. Ще предположим, че x_1 и y_1 са описани с числата от 1 до 8. Тук естествената операция е *плюс едно*. Тази операция за x_1 ще наречем *дясно*, а нейната обратна ще наречем *ляво*. За y_1 съответно това ще са *нагоре* и *надолу*.

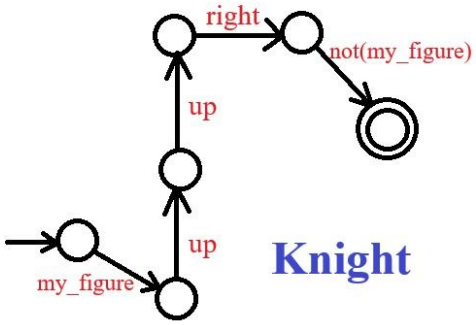
Отново предполагаме, че светът не е излишно усложнен. Можеше да разбъркаме числата от 1 до 8 с някоя произволна пермутация и тогава щеше да е трудно да определим *дясно* и другите абстрактни събитията.

За да опишем движението на фигурите, трябва да забележим, че $\langle x_1, y_1 \rangle$ и $\langle x_2, y_2 \rangle$ описват състоянията на един и същи абстрактен събитиен модел и съответствието е същото (тоест едни и същи координати при $\langle x_1, y_1 \rangle$ и при $\langle x_2, y_2 \rangle$ сочат към едно и също квадратче). За тази цел ще ни помогне наблюдението, че ако преместим успешно бяла фигура от $\langle x_1, y_1 \rangle$ на $\langle x_2, y_2 \rangle$, то тогава на $\langle x_2, y_2 \rangle$ ще има бяла фигура (не винаги, но почти винаги).

Искаме да определим предиката *move*, който ще ни показва коректните ходове (пак почти винаги). Това ще е предикат, който свързва състоянията на абстрактния събитиен модел (квадратчетата). Ще го опишем с алгоритми, които казват как от едно квадратче можем да отидем до друго. Ще започнем с алгоритъма на коня, който се движи под формата на буквата L (фигура 2). Този алгоритъм можем да го запишем като $up^2.right$. Същата работа би ни свършил алгоритъмът $right.up^2$. Почти същата работа би свършил и алгоритъмът $up^3.right.down$, но този алгоритъм ще е по-слаб, защото понякога ще излиза извън таблото и тогава няма да работи.



Фигура 2

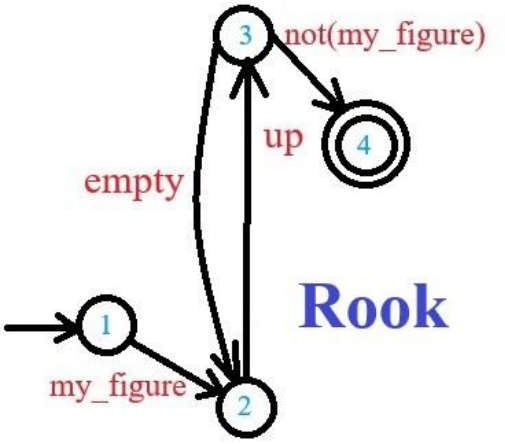


Фигура 3

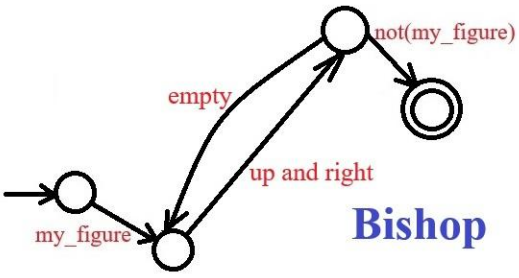
Описваме алгоритъм, който е от типа *going home* (виж [2]). Това означава, че ако изпълним алгоритъма, ще стигнем до целта, но нищо не ни задължава да минем точно по този път. В [2] се разглеждаха алгоритми от вида *railway*. Там релсата се определяше от света, който не позволяваше да излизаме извън пътя на алгоритъма. В [2] дори имаше възможност да се движим напред-назад (без да излизаме от релсите). Тук няма релси и можем да се движим както си искаме, но алгоритъмът описва един път и той е възможно най-простият.

Към алгоритъма на коня трябва да добавим това, че можем да местим само наши фигури и че нашите фигури не можем да ги взимаме. Това, което се получава, е алгоритъм използващ следата (фигура 3). Тук имаме абстрактен събитиен модел с реална следа. Би могло един абстрактен събитиен модел да има много реални следи. Например би могло да играем blind на 10 дъски едновременно. Тогава дъската (която е абстрактен събитиен модел) ще има 10 следи, които се превключват последователно.

Разбира се, движението на коня се описва с 8 алгоритъма подобни на този от фигура 3. Можем да кажем, че движението на коня е дизюнкцията на тези 8 алгоритъма. Когато направим дизюнкция на два алгоритъма получаваме недетерминистичен алгоритъм. Нека разгледаме алгоритъма на топа, който също е недетерминистичен (фигура 4).



Фигура 4



Фигура 5

Тук *empty* означава *not(white_figure) and not(black_figure)*. Това е алгоритъмът, който описва движението на топа напред. Той е недетерминистичен, защото от състояние 3, когато имаме празно квадратче можем недетерминистично да преминем към състояние 2 или към състояние 4.

На фигура 5 е показан алгоритъма на офицера, когато се движи напред и надясно. Тук интересното е, че можем да се движим по диагонал, тоест абстрактните събития *up* и *right* могат да бъдат извършени едновременно. В [2] това бяха реални събития и светът не позволяваше те да се случат едновременно. Затова тук алгоритъмът на офицера е по-прост от този, който е даден в [2].

7. Обекти

Алгоритмите, които могат да се приложат към една фигура, това са нейните свойства. Тези алгоритми определят коя е фигурата. Например движението на топа се определя от 4 алгоритъма, движението на офицера от други 4 алгоритъма. Движението на царицата се определя от алгоритмите на топа и на офицера. Важно е една фигура какви свойства има и какви свойства няма. Тоест царицата няма да е офицер. Тя има всички негови свойства, но има и свойства, които офицерът няма.

Ще въведем абстракцията обект. Ще предполагаме, че действието на агента мести обект от квадрат $\langle x_1, y_1 \rangle$ на квадрат $\langle x_2, y_2 \rangle$. Можеше да си мислим, че се преместват само свойствата от единия квадрат в другия, но светът ще е по-лесен за разбиране, ако предположим, че има обекти и свойствата се пакетират в обекти.

Представата ни за света ще бъде дъската (абстрактен събитиен модел с 64 състояния) и реални обекти, които са следата на този модел (фигурите, които са на дъската). Ще си представим конкретната позиция на дъската и ще видим как тази позиция се променя във времето.

Когато имаме подвижна следа, естествените операции са добавяне на обект към някое състояние (квадратче), премахване на обект и преместване на обект. Това, което се случва, когато извършим действието $\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle$, може да бъде обяснено по следния естествен начин: „Обектът от $\langle x_1, y_1 \rangle$ се премества в $\langle x_2, y_2 \rangle$.“

Друга сравнително естествена операция е да се върнем назад (да отменим последната операция). Ще предположим, че след наблюдението *bad* се отменя последната операция и дъската се връща в предишната си позиция.

Единствената операция, която е по-сложна и неестествена е появата на началната позиция на дъската след наблюдението *game over*.

8. Не можем да сме шах

За да не играе агентът некоректни ходове, трябва той да може да предскаже кога ходът ще бъде некоректен. Основното правило казва, че ако ходът не може да бъде направен по правилата на играта шах, то той е некоректен. (Тук *My_Color* е състоянието на модела **B&W**.)

$$not(\ move(My\_Color,\ \langle x_1, y_1\rangle,\ \langle x_2, y_2\rangle\ )\ ),$$
$$action(\ \langle x_1, y_1\rangle,\ \langle x_2, y_2\rangle\ )$$
$$\Rightarrow bad$$

Тук използваме полуразрешим предикат, дори използваме отрицание на полуразрешим предикат. Твърдението е, че съществува изпълнение на алгоритъма *move*. В нашия конкретен случай този предикат е разрешим, защото в играта шах има краен брой фигури и въпросът дали съществува ход е разрешим. Въпреки това, в общия случай съществуването на алгоритъм ще е полуразрешим въпрос, но ние казахме, че при описанието на света ще използваме полуразрешими предикати и дори и техните отрицания, защото това е коректно описание на света.

Има още един случай когато ходът е некоректен. Това е когато на следващият ход могат да ни вземат царя (когато след хода сме шах). Това може да се опише така:

$$action(\ \langle x_1, y_1\rangle,\ \langle x_2, y_2\rangle\ ),$$
$$possible(\ move(My\_Color,\ \langle x_3, y_3\rangle,\ \langle x_4, y_4\rangle\ ),\ king(\ \langle x_4, y_4\rangle\ )\ )$$
$$\Rightarrow bad$$

Тук *possible* означава, че съществува изпълнение на алгоритъм. По-точно, че съществуват някакви $\langle x_3, y_3\rangle$, $\langle x_4, y_4\rangle$, за които алгоритъмът може да се изпълни. При това изпълнение *My_Color* се е обърнал, защото мислим наум и мислено правим действие, което означава, че моделът **B&W** е превключил състоянието си.

Когато мислим наум трябва да приемем, че след нашето мислено действие настъпват същите последствия каквито биха настъпили, ако действието беше реално извършено. Когато мислено отваряме чадъра си, трябва да приемем, че в нашите мисли дъждът престава да ни мокри. Трябва да правим разлика между мислено и реално действие. Мислено може да сме отворили чадъра си, но реално чадъра да не е отворен и дъждът да продължава да ни мокри.

Когато правим действие ние променяме представата си за състоянието на света и така имаме реална представа за това какво е текущото състояние. Когато мислим наум, ние мислено променяме състоянието на света, но това не се отразява на реалната ни представа.

9. Играят двама

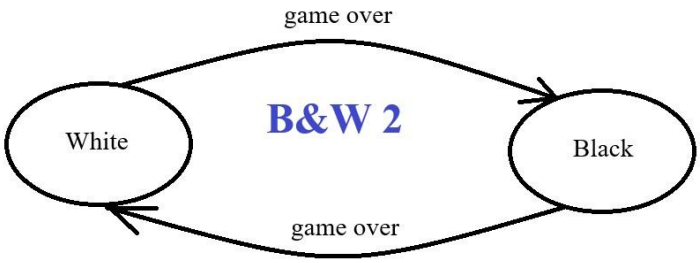
Описахме играта шах, когато агентът играе сам срещу себе си. Сега ще предположим, че в света има втори агент, който играе срещу нас. Допускането на съществуването на втори агент е важна абстракция, която ще ни помогне да разберем света. Ако бяхме се опитали да опишем света заедно с втория агент, то това би била една много трудна задача. Вторият агент може да е човек или някоя сложна компютърна програма. И в двата случая той е труден за разбиране (за точно описване). Затова ние ще

си представим втория агент като черна кутия. Няма да знаем как той мисли (или как работи). Ще направим само някои най-общи предположения. Например ще предполагаме, че той ни е враг и играе срещу нас.

Тоест идеята на многоагентния свят се свежда до това да представим другите агенти като черни кутии и по този начин да скрием тяхната вътрешна сложност и така да опростим описанието на света.

Нека предположим, че двамата играчи се редуват и след всяка партия сменят фигурите. В предишния вариант агентът сменяше фигурите след всеки ход и му беше полесно да добие представа за белите и черните фигури. Сега агентът ще ги сменя по-рядко, но пак ще може да добие представа за черните фигури, макар и по-трудно.

Сега пак ще има един реален събитиен модел и това ще бъде моделът **B&W 2** (фигура 6), който ще замени **B&W**.



Фигура 6

Преди предполагахме, че състоянието на света (позицията на таблото) се променя след действие (като се премества фигура). Сега ще предположим, че наблюдението също променя състоянието на света (също като мести фигура). Ще предположим, че действието мести бяла фигура, а наблюдението мести черна (това е, ако играем с белите).

Ние ще търсим обяснение за всичко. Ще предположим, че черните фигури не се движат сами, а има някой, който ги движи. Ще предположим, че този някой е злонамерен и иска ние да загубим. От друга страна ще предположим, че този някой не е всесилен и не може да мести фигурите както му скимне. Тоест ще предполагаме, че противникът е задължен да спазва същите правила, които и ние трябва да спазваме. Това ще го открием като забележим, че противникът играе само коректни ходове. Ще предположим, че противникът има известна свобода, но тази свобода е ограничена в някакви рамки.

Тук, за да използваме едни и същи алгоритми и за действието и за наблюдението ще предположим, че предикатът *my_figure* зависи от състоянието на **B&W 2** и от това кой играе (ние или противникът).

Ще предполагаме, че когато играем с черните *rewards* обръща стойностите за *win* и за *loss*. Тоест, когато играем с черните, ще искаме те да победят. Това няма да е раздвоение на личността, защото ще имаме едно съзнание, което знае какво иска, въпреки че междинната цел след всяка партия ще се променя от *win* в *loss* и обратно.

И сега пак ще ни трябва правило, което казва, че не можем да позволим да ни вземат царя. Това правило ще изглежда така:

$$action(\langle x_1, y_1\rangle, \langle x_2, y_2\rangle),$$
$$possible(observation(\langle x_3, y_3\rangle, \langle x_4, y_4\rangle, Result), king(\langle x_4, y_4\rangle)) \Rightarrow bad$$

Тук стойността на *Result* няма значение. Важното е, че ще можем да видим как ни взимат царя. Тук трябва да правим разлика между „възможно е“ и „разрешено е“. В случая противникът няма как да ни вземе царя, защото на нас не ни е позволено да направим

подобна тривиална грешка. Тоест това никога няма да се случи, но по принцип, ако ние играехме този ход, то противникът би могъл да ни вземе царя.

Горното правило ни трябва, за да знаем кои ходове са ни позволени. Що се отнася до противника, той също няма право да направи такава тривиална грешка. Тук нещата зависят от това какъв е противникът. Може той да мисли *n* хода напред и да не прави грешки, който се виждат за *n* хода. Няма как да разберем дали той не прави тези грешки, защото му е забранено или защото така е пожелал. Все едно е дали на противника му е забранено да прави тривиални грешки или той просто никога не ги прави. В двата случая имаме един и същи свят.

10. Играе само с белите

Нека сега да играят двама, но да не сменят фигурите. Нека агентът да играе винаги с белите, а противникът винаги да е с черните. При този вариант няма да имаме реалния събитиен модел **B&W**. Ако решим да имаме подобен модел, то би се получил модел с едно единствено състояние, а събитиен можел с едно състояние не е никакъв модел.

Този свят ще е по-труден за разбиране, защото ще е много трудно агентът да стигне до идеята за черните фигури. Имаме тест за откриване на бяла фигура, но с черните е по-сложно.

Може да открием черните фигури по аналогия. Имаме бели фигури. По аналогия ще имаме и черни и те ще са същите, но черни. Така учените откриват антиматерията. Принципът е: „Щом имаме материя, що да нямаме антиматерия? Тя ще е същата като материята, ама анти.“

По-логично е да открием черните фигури по това, че пречат на движението на белите. Например белите и черните фигури пречат на движението на белия топ, но пречат по различен начин. Бялата фигура го спира веднага, а черната го спира на следващата стъпка. Когато нещо спира движенето, може да предположим, че това нещо е там където действително е черната фигура, а може да предположим, че то е на следващото квадратче. Тоест откриването на черните фигури ще е трудна задача.

11. Варианти

Целта ни е да направим AGI, а не просто програма, която играе шах. Може би абстрактният събитиен модел, който използвахме, е твърде лесен за намиране. Вярно, че квадратчетата на таблото са нещо абстрактно, което не се вижда директно, но ние споменаваме тези квадратчета чрез техните координати. Освен това всяко квадратче си има име (координати). Може да си представим абстрактен събитиен модел, в който знаем имената само на част от състоянията. Например да вземем множеството на хората. Ние знаем имената само на част от тях и можем да споменем само тези, които знаем по име. Нека вземем релацията *родител*. Чрез тази релация може да дефинираме *брат* и *братовчед*.

Друг пример за абстрактен събитиен модел ще бъде, ако в играта шах не описваме ходовете чрез координатите на две квадратчета, а чрез фигурата, посоката, в която ще я придвижим и броят стъпки. Например *(rook, 2, up, 7)*. Тук 2 показва за кой топ става дума, защото имам два бели топа, 7 показва колко стъпки ще направим в тази посока (от 1 до 7). Друг пример *(pawn, 5, up&left, 1)*. Тук 5 показва коя пешка ще местим. Посоката е *up&left*, което означава, че ще взимаме по диагонал. Стъпката е 1, защото в тази посока не може повече.

В предишния свят фигурите бяха свойство на квадратчетата. Сега ще е обратното. При този вариант на света квадратчетата въобще не се споменават. Споменават се фигурите и техните координати са тяхно свойство, до което няма да е лесно да се стигне. Сега дъската ще е един абстрактен събитиен модел, чийто състояния въобще не се споменават.

Много трудно би било да се разбере така описаният свят, но никой не се учи да играе шах blind. Когато се учим ние използваме табло с фигури и учител, който ни обяснява как се движат фигурите. Започваме да играем blind чак когато вече знаем правилата. При хората трудността при играта blind идва от необходимостта да запомним позицията на таблото. Тук поставяме една много по-трудна задача и тя е да разберем правилата на играта на принципа проба и грешка. Решавайки тази по-трудна задача ние виждаме как можем да намираме абстрактни събитийни модели и да разбираме света.

12. Заключение

Тази статия ви казва как да създадете AGI, но не бързайте да го правите. ИИ не е поредната стъпка от пътя, а това е последната стъпка. Преди да направите тази последна стъпка и да скочите в бездната на неизвестността помислете добре какъв ИИ искате да създадете. Прочетете статии като [9], в които се дискутира този въпрос.

References

[1] Dobrev, D. (2022) Language for Description of Worlds. Part 1: Theoretical Foundation. *Serdica Journal of Computing 16(2), 2022, pp. 101-150.*

[2] Dobrev, D. (2023) Language for Description of Worlds. Part 2: The Sample World. *Serdica Journal of Computing 17(1), 2023, pp. 17-54.*

[3] Marcus, G. (2025) Game over for pure LLMs. Even Turing Award Winner Rich Sutton has gotten off the bus. https://garymarcus.substack.com/p/game-over-for-pure-llms-even-turing.

[4] Hutter, M. (2000). A Theory of Universal Artificial Intelligence based on Algorithmic Complexity. *arXiv:cs.AI/0004001 [cs.AI]*

[5] Hutter, M. (2007) UNIVERSAL ALGORITHMIC INTELLIGENCE A mathematical top→down approach. *In Artificial General Intelligence, 2007.*

[6] Hernández-Orallo, J., Minaya-Collado, N. (1998). A formal definition of intelligence based on an intensional variant of Kolmogorov complexity. *Proc. intl symposium of engineering of intelligent systems (EIS'98), February 1998, La Laguna, Spain (pp. 146–163). : ICSC Press.*

[7] Dobrev, D. (2019) The IQ of Artificial Intelligence. *Serdica Journal of Computing, Vol. 13, Number 1-2, 2019, pp.41-70.*

[8] Dobrev, D. (2024) Description of the Hidden State of the World. viXra:2404.0075.

[9] Dobrev, D., Ivanov, L., Popov, G., Tzanov, V. (2024) How Can We Make AI with a Nice Character? viXra:2408.0087.