Practical Examples Using Semantic Simulation With Reinforcement Learning

D. Gueorguiev 12/4/2022

The Game Addition

Let us consider the game $Addition$ described in *Blackwell's Theory of Games and Statistical Decisions* (Blackwell & Girshik, 1978, p. 14):

I and II alternatively choose integers, each choice being one of the integers $1, \dots, k$ and each choice made with the knowledge of all preceding choices. As soon as the sum of the chosen integers exceeds N, the last player to choose pays his opponent one unit.
The situation at which player I finds himself at his rth move is described by a sequence $s_r = (i_1, i_2, \dots, i_{2r-2})$ with each i_j being one of the integers $1, \dots, k$ and

$\sum_{j=1}^{2r-2} i_j \leq N$

Denote by S_r the set of possible sequences s_r where $r = 2, \dots, \left[\frac{N}{2}\right] + 1$ and $[z]$ denotes the closest integer which does not exceed z. A strategy x for I consists of a set of $\left[\frac{N}{2}\right] + 1$ functions $f_1, \dots, f_{\left[\frac{N}{2}\right]+1}$, where f_r is a function defined on S_r assuming only values $1, 2, \dots, k$: f_r specifies I's rth move when the previous history of the play is s_r. Similarly, a strategy y for II is a set of $\left[\frac{(N+1)}{2}\right]$ functions $g_1, \dots, g_{\left[\frac{(N+1)}{2}\right]}$, where g_r is defined for the set T_r of all sequences $t_r = (i_1, \dots, i_{2r-1})$ with each i_j being one of the integers 1,2, …, k and

$\sum_{j=1}^{2r-1} i_j \leq N$

Define $i_1(x, y) = f_1$ and inductively for $j > 0$,
$i_{2j}(x, y) = g_j\left(i_1(x, y), \dots, i_{2j-1}(x, y)\right)$
$i_{2j+1}(x, y) = f_{j+1}\left(i_1(x, y), \dots, i_{2j}(x, y)\right)$
(this induction describes the manner in which a referee would carry out the instructions of the players) and let $j^*(x, y)$ be the largest j for which $i_j(x, y)$ is defined. Then

$$M(x, y) = \begin{cases} 1 \text{ if } j^*(x, y) \text{ is even} \\ -1 \text{ if } j^*(x, y) \text{ is odd} \end{cases}$$

Constructing semantic universe for the game $Addition$

Let us consider the following thought experiment – we have two players playing the $Addition$ game described earlier. Each player is represented by semantic simulation which has its own set of semantic structures and semantic template which recognizes the rules of the game. Let us start our experiment by looking in the semantic template which recognizes the rules of the game which we will name *semantic* *recognizer*. That is - we are interested in what the semantic recognizer might be taking as an input and producing as an output and how the semantic recognizer template would be interacting with the rest of the semantic structures running in the simulation.

Let us assume that the semantic simulation corresponding to each of the two players I and II is limited to the simply connected regions R_1 and R_2 in semantic space. Additionally, we introduce an Arbiter which will be assigned its own simply connected region R_3 in semantic space. Let $\dim(R_1) = \dim(R_2) = \dim(R_3) = L$. Let us assume that $R_1 \cap R_2 \cap R_3 = C$ where C is finite, closed and simply connected region of semantic space with the same number of dimensions L. We will denote C as the *common simulation region*.

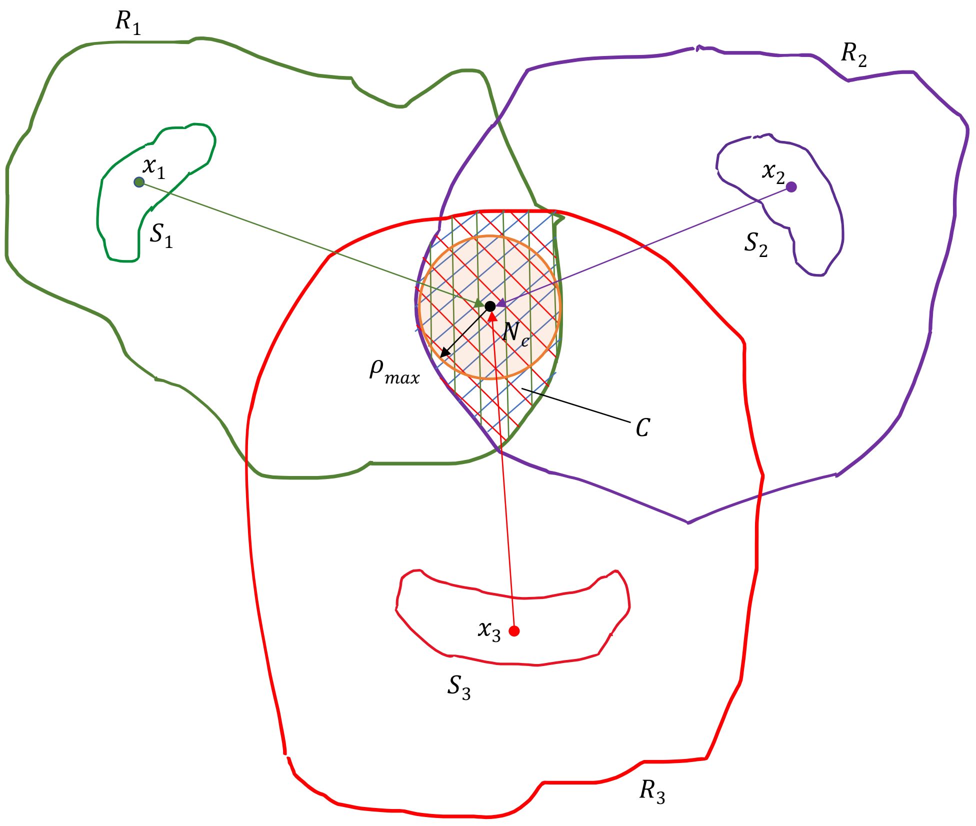

Figure 1: Layout of the simulation space in Blackwell's game *Addition*

Definition: *neutral point* of a simply connected region in metric space
Let C is a simply connected region in some L dimensional metric space. Then the point N_c is a neutral point *iff* it is the center of the largest L dimensional sphere which can fit entirely in the simply connected region C without including any points outside of C. Formally,
$\exists\, N_c \in C \therefore\ \rho_{max} = \max_{\rho} |N_c - x| \leq \rho \ \forall\, x \in C$

With N_c we denote *the neutral point* of the common simulation region C. The neutral point will be the attraction center for all outputs from player I and II's as well as the arbiter simulations. Both players I and II as well as the Arbiter will produce an output which will be a semantic particle starting its existence at a point inside their respective regions S_1, S_2, S_3 shown on Figure 1.

A couple notational conventions which will simplify the discussion:
Let us have a template T_s defined over the semantic region C.

Region over which a template is defined
In the future we will denote the region over which the template is defined with the appropriate symbol denoting the region in parentheses; Thus $T_s(C)$ indicates that T_s is defined over C.

Trajectory of semantic particle
A particle p_s having trajectory intersecting with specific region C will be denoted with the following notation $p_s \rightsquigarrow C$.

Template match
We denote a template match, that is the template T_s has matched the input represented by p_s, with the following symbolic notation $T_s(p_s \rightsquigarrow C) \uparrow$.

Chaining of template actions
The notation $P_1 \uparrow \overset{p}{\Rightarrow} P_2 \uparrow$ (P_1, P_2 are predicates) indicates that triggering P_1 causes particle p to be emitted which if matched will trigger the predicate P_1.
Let us have the following templates $T_s(C)$ and $T_{1,0}(S_1)$ and $p_{s,1}$ is semantic particle.
Then the notation
$T_s(p_s \rightsquigarrow C) \uparrow \overset{p_{s,1}}{\Longrightarrow} T_{1,0}\left(p_{s,1} \rightsquigarrow S_1\right) \uparrow$
indicates that the particle p_s being present in C triggers T_s which in turn emits a new particle $p_{s,1}$ which if present in S_1 will trigger a *chained template* $T_{1,0}$.

Here is how the game simulation will proceed:
For simplicity let us assume that the game parameter N defined earlier is given and it is known by the two players and the Arbiter. Also, we will assume that the Arbiter will make decision who will be the first of the two players to play; for simplicity the decision-making process of the Arbiter will be omitted from the discussion. Let us represent this decision-making process of the Arbiter by the semantic template T_s (s for start of the game). The template T_s accepts an input indicating the start of the game.

The input indicating the start of the game will be represented as a particle with specific signature which we will denote with p_s. As soon as the arbiter template T_s detects that the signature of p_s is present in C it sends either a particle $p_{s,1}$ to region S_1 or $p_{s,2}$ to region S_2.

In case of $p_{s,1} \rightsquigarrow S_1$ a template $T_{1,0}$ which belongs to Player I will recognize the signature of $p_{s,1}$ that is $T_{1,0}$ will be triggered: $T_{1,0}(p_{s,1} \rightsquigarrow S_1) \upharpoonleft$. On a match $T_{1,0}$ will send a messenger particle $m_{1,0}$ to another template of Player I - $T_{1,1}$. In turn the inference structure of $T_{1,1}$ sends an information particle $\langle x_1|k_1\rangle$ toward N_c in C. The information particle is a composite semantic particle and contains two sub-particles:

- sub-particle x_1 conveying the information that it has been created by a template which belongs to Player I
- sub-particle k_1 conveying the information that Player I has chosen the number k_1 on his current move

As $\langle x_1|k_1\rangle$ is sent toward N_c a template A_0 which belong to the Arbiter is looking for specific patterns. A_0 is the so called *end-of-the-game recognizer*. This template will create different response depending on the pattern it detects. One of the patterns A_0 recognizes is single $\langle x_1|k_1\rangle$ particle in C. We can write this sequence as:

$$\text{Start} \overset{p_s}{\Rightarrow} T_s(p_s \rightsquigarrow C) \upharpoonleft \overset{p_{s,1}}{\Longrightarrow} T_{1,0}(p_{s,1} \rightsquigarrow S_1) \upharpoonleft \overset{m_{1,0}}{\Longrightarrow} T_{1,1}(m_{1,0} \rightsquigarrow S_1) \upharpoonleft \xRightarrow{\langle x_1|k_1\rangle} A_0(\langle x_1|k_1\rangle \rightsquigarrow C) \qquad (1a)$$

We can simplify the notation above, writing short hand:

$$\text{Start} \overset{p_s}{\Rightarrow} T_s(C) \overset{p_{s,1}}{\Longrightarrow} T_{1,0}(S_1) \overset{m_{1,0}}{\Longrightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_1\rangle} A_0(C) \qquad (1b)$$

In the case when $p_{s,2}$ is sent to S_2 a template T_2 which belongs to Player II will recognize the signature of $p_{s,2}$. In this case we write

$$\text{Start} \overset{p_s}{\Rightarrow} T_s(p_s \rightsquigarrow C) \upharpoonleft \overset{p_{s,2}}{\Longrightarrow} T_{2,0}(p_{s,2} \rightsquigarrow S_2) \upharpoonleft \overset{m_{2,0}}{\Longrightarrow} T_{2,1}(m_{2,0} \rightsquigarrow S_2) \upharpoonleft \xRightarrow{\langle x_2|k_1\rangle} A_0(\langle x_2|k_1\rangle \rightsquigarrow C) \qquad (2a)$$

Similar to (1b) we write short hand:

$$\text{Start} \overset{p_s}{\Rightarrow} T_s(C) \overset{p_{s,2}}{\Longrightarrow} T_{2,0}(S_2) \overset{m_{2,0}}{\Longrightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_1\rangle} A_0(C) \qquad (2b)$$

In case of (1) the Inference Structure of A_0 will create particle p_2 sent toward S_2. Alternatively, in case of (2) the Inference Structure of A_0 will create particle p_1 sent toward S_1.

So, the sequence (1b) is extended as:

$$\text{Start} \overset{p_s}{\Rightarrow} T_s(C) \overset{p_{s,1}}{\Longrightarrow} T_{1,0}(S_1) \overset{m_{1,0}}{\Longrightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_1\rangle} A_0(C) \overset{p_2}{\Rightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_2\rangle} A_0(C)$$

And the ball is one more time in the field of the *end-of-the-game recognizer* A_0.
Seeing $\langle x_2|k_2\rangle$ moving toward the neutral point of C, A_0 will either send p_1 towards S_1 or issue a signal for the end of the game.

Thus we will end with one of the four sequences:

Start $\overset{p_s}{\Rightarrow}$ $T_s(C) \overset{p_{s,1}}{\Longrightarrow} T_{1,0}(S_1) \overset{m_{1,0}}{\Longrightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_1\rangle} A_0(C) \overset{p_2}{\Rightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_2\rangle} A_0(C) \overset{p_1}{\Rightarrow} \cdots$
$\overset{p_2}{\Rightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_n\rangle}$End of Game
or
Start $\overset{p_s}{\Rightarrow}$ $T_s(C) \overset{p_{s,1}}{\Longrightarrow} T_{1,0}(S_1) \overset{m_{1,0}}{\Longrightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_1\rangle} A_0(C) \overset{p_2}{\Rightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_2\rangle} A_0(C) \overset{p_1}{\Rightarrow} \cdots$
$\overset{p_1}{\Rightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_n\rangle}$End of Game
or
Start $\overset{p_s}{\Rightarrow}$ $T_s(C) \overset{p_{s,2}}{\Longrightarrow} T_{2,0}(S_2) \overset{m_{2,0}}{\Longrightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_1\rangle} A_0(C) \overset{p_1}{\Rightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_2\rangle} A_0(C) \overset{p_2}{\Rightarrow} \cdots$
$\overset{p_1}{\Rightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_n\rangle}$End of Game
or
Start $\overset{p_s}{\Rightarrow}$ $T_s(C) \overset{p_{s,2}}{\Longrightarrow} T_{2,0}(S_2) \overset{m_{2,0}}{\Longrightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_1\rangle} A_0(C) \overset{p_1}{\Rightarrow} T_{1,1}(S_1) \xRightarrow{\langle x_1|k_2\rangle} A_0(C) \overset{p_2}{\Rightarrow} \cdots$
$\overset{p_2}{\Rightarrow} T_{2,1}(S_2) \xRightarrow{\langle x_2|k_n\rangle}$End of Game

In either case the end of the game is recognized with a pattern, similar to the one shown on Figure 2 below.

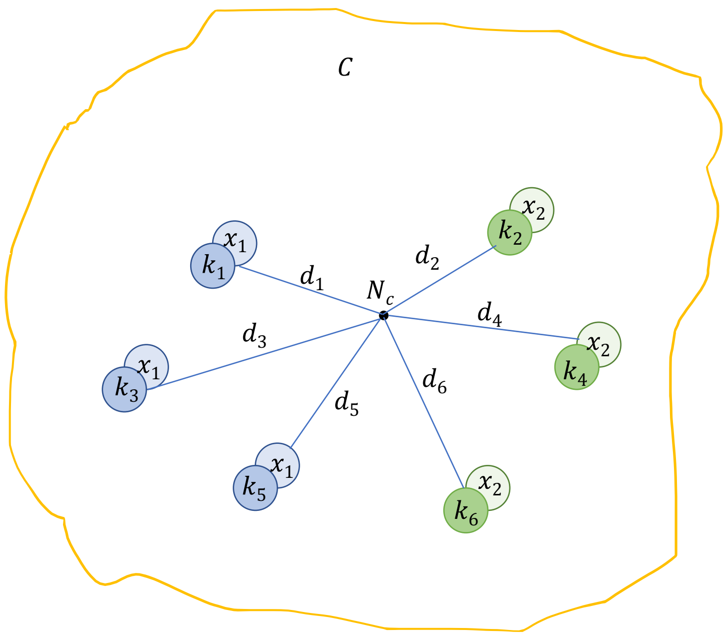

Figure 2: Possible final arrangement of the semantic particles produced by the two players at the end of a game of *Addition*

It is important to realize that the pattern on Figure 2 can be transformed by some known transformation to the path graph shown on Figure 3 below. This would be true with well chosen laws governing the motion of semantic particles. For details on the equations governing the positions and dynamics of semantic particles refer to documents Modeling Attractive and Repulsive Forces in Semantic Properties and On The Need Of Dynamic Simulation When Modeling Interactions of Semantic Particles.

.

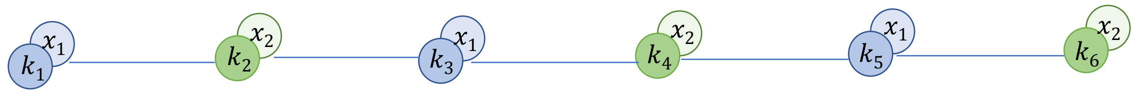

Figure 3: Semantic structure formed by the final arrangement of the output of the two players

Reinforcement Learning in the Blackwell's Game of Addition

//TODO: finish this

Bibliography

Bang-Jensen, J., & Gutin, G. (2007). *Diagraphs: Theory, Algorithms and Applications.* Odense, Denmark, London, UK: Springer-Verlag.

Bellman, R. (1972). *Dynamic Programming.* Princeton, NJ: Princeton University Press, Sixth Printing.

Blackwell, D. A., & Girshik, M. A. (1978). *Theory of Games and Statistical Decisions.* New York: Dover Publications; Illustrated edition.

Denardo, E. V. (1982). *Dynamic Programming: Models and Applications.* Englewood Cliffs, NJ: Prentice-Hall.

Francois-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An Introduction To Deep Reinforcement Learning. *Foundations and Trends in Machine Learning: Vol. 11, No. 3-4*.

Gosavi, A. (2022). Reinforcement Learning: Tutorial and Recent Advances. *INFORMS Journal on Computing*.

Harmon, M. E., & Harmon, S. S. (Jan 1997). *Reinforcement Learning: A Tutorial.* Wright Patterson AFB OH 45433: Avionics DIrectorate, Wright Laboratory, Air Force Materiel Command.

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020, Nov 1). *Offline Reinforcement Learning: Tutorial, Review, and Perespectives on Open Problems*. Retrieved from arxiv.org: https://arxiv.org/abs/2005.01643

Neapolitan, R. E. (2019). *Bayesian Networks.* Upper Saddle River, NJ: Prentice-Hall.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: Introduction, second edition.* Cambridge, Massachusetts: The MIT Press.