



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Práctico de IBC

Federico De Rocco

Director: Vanina Martinez
Buenos Aires, 2020

Índice general

| | |
|--|---|
| 1.. Práctico Tema 1 | 1 |
| 2.. Práctico Tema 2: Agentes | 4 |

1. PRÁCTICO TEMA 1

1. Suponiendo la premisa como cierta, se puede asumir que programas triviales (como por ejemplo el “Hola Mundo”) no presentan un comportamiento que podríamos llamar inteligente, ya que este es altamente determinístico. Pero este comportamiento no siempre se puede analizar de esta forma. El programa podría ser muy complejo, de tal manera que su determinismo no estuviera claro, ni siquiera para los programadores. Pero todo esto es suponiendo que este sea determinístico, uno podría escribir una pieza de software que tuviera comportamiento no determinístico. Un ejemplo de esto sería que el programa seleccionara entre varios comportamientos distintos en base a valores generados aleatoriamente. Deberíamos preguntarnos también, si el hecho de tener comportamiento que este influenciado cualquier forma implica no tener inteligencia, lo cual es lo que asume esta frase. A mi entender esto no es cierto, y considero que los seres humanos poseamos algunas formas de este.

2. Existen experimentos como la caja de Skinner, que han demostrado que, sometiendo a estímulos apropiados se puede lograr que un animal desarrolle un comportamiento determinado. Pero esto no significa que Skinner u otros científicos tengan la capacidad de predecir todos los elementos de como los animales actúan. Usar a los “genes”, como una posible forma de insistir en que los animales no son inteligentes es ridículo porque, de primeras los seres humanos seríamos en el mejor caso, versiones más complejas de ellos y considero que decir que los seres humanos no podemos ser inteligentes sería dejar de lado la metáfora. Pero en segundo lugar, al igual que las computadoras, lo innato en los animales podría sugerir comportamiento que fuera muy difícil o casi imposible de determinar.

3. No solo el sistema puede reconocer y entender información tal como el tráfico aéreo, sino que puede reaccionar a las situaciones que esta información indica. Su comportamiento queda determinado por decisiones tomadas por él mismo usando la información que recibe. Esto muestra un programa capaz de adaptarse y tomar decisiones con criterios determinados. Si se está dispuesto a clasificar este tipo de comportamiento como inteligente, entonces estamos ante un sistema que lo es.

En la naturaleza, las aves voladoras poseen características similares, ellas no siguen una ruta fija para moverse, cazar, migrar, etc. Divergen en ellas por decisiones tomadas de acuerdo con los elementos en el entorno que pueden percibir, el viento y el clima por ejemplo. Pero estas también pueden enfrentarse a situaciones de las cuales no tengan una respuesta optima o no sepan percibirla, por ejemplo si un avión vuela contra ellas o si un cazador les dispara en el aire.

4. Dado que existe un problema que no puede ser resuelto utilizando software, debo entender que la única razón por la que esto puede impedir el desarrollo de la IA es que esta no podría resolver estos problemas. Si entendemos que la IA debe ser capaz de replicar cualquier comportamiento inteligente que un ser humano o animal puede realizar, entonces efectivamente podemos decir que esta sería imposible porque las piezas de software no podrían resolver estos problemas. Pero esto solo tiene sentido si tenemos este como

la definición de inteligencia artificial, si concedimos que una de estas puede dedicarse a resolver problemas específicos, como la navegación en el avión del ejercicio anterior, entonces no tenemos este problema. Además la mayoría de definiciones de IA implican que el programa debe operar bajo una metáfora del pensamiento, la cual no tiene porque ser biologicamente observable, para considerarse IA. Noten que esto implica la noción de inteligencia, no como su totalidad, sino como posibles formas de funcionamiento que pueda argumentarse que esta pueda operar. Sugiriendo que la idea es que exhiba comportamiento inteligente según esta metáfora, no que sea irreconocible de las capacidades humanas. Incluso podríamos decir que si abstraemos todos los elementos de inteligencia que no son suficientes para resolver estos problemas no computables, y creamos una IA a partir de estos, tendríamos un aproximado a una IA general.

5.

a) Por si mismo es capaz de realizar varias actividades asociadas con el pensamiento; la de lectura y análisis de un mapa, el análisis de información dinámica como el tráfico, el catalogar y estimar tiempos de distintos medios de transporte y el comunicar esta información a pedido. Todo esto con el objetivo de obtener las mejores rutas a pedido. Es una instancia de IA en el sentido que pretende emular una actividad, hasta hace poco puramente humana, tal es como la planificación de ruta en un entorno de transporte humano.

b) Este sistema realiza la actividad de reconocer patrones simbólicos, mediante un escaner, traducirlos a un valor número y a este asociarlo a un producto con su precio y otros datos. Sin duda, la actividad que más resalta es la de reconocimiento simbólico; particularmente la asociación de un símbolo a un valor numérico o, extendiendo la metáfora, a una idea es una de las actitudes más comunes de la inteligencia, solo hay que ver a las letras de el lenguaje escrito que sea para reconocer esto. Incluso los números, que en diferentes lugares o tiempos se representan con símbolos distintos.

c) En este caso tenemos un sistema de reconocimiento de voz. Se podría decir que este es una instancia de IA en el sentido de que la actividad que realiza está asociada al lenguaje verbal, el cual es parte de la forma en la que los humanos piensan. Aunque a su vez se podría decir que también los animales reconocen la voz e incluso pueden seguir ordenes comunicadas con esta, aunque no todos los animales pueden. Hasta antes de un sistema de reconocimiento de voz, la actividad solo podía realizarse a travez de seres vivientes específicos, dadas sus capacidades para escuchar y reconocer diferencias en el sonido.

d) Siendo el lenguaje escrito un fenómeno, posiblemente accidental a diferencia del verbal pero, sin duda propio del pensamiento humano. Esta sería entonces una máquina sin conciencia y carente de las cualidades innatas de los seres humanos que puede reconocer patrones en el habla para asociarlos con instrucciones que normalmente debería introducirse manualmente.

e) Similar al sistema de GPS, la idea es poder encontrar el mejor recorrido, teniendo en cuenta la disponibilidad de los elementos que son parte de los caminos potenciales. Es cierto que el camino no se refiere a lo mismo, ya que son paquetes de datos y no una persona, disponibilidad de nodos y no tráfico o disponibilidad de transportes, en última

instancia es inteligente por razones similares.

2. PRÁCTICO TEMA 2: AGENTES

1.

```
1  function TreeSearch(Raiz, ArbolBusqueda, Estrategia) {
2      Frontera = []
3      Frontera.add(Raiz)
4      SelectElement = Raiz
5      PotencialesCaminos = [[Raiz]]
6
7      while (!esMeta(SelectElement)) {
8
9          Frontera = Estrategia.nuevaFrontera(ArbolBusqueda, Frontera,
10         SelectElement)
11
12         if (Frontera.lenght == 0) {
13             return failure
14         }
15
16         SelectElement = Estrategia.select(ArbolBusqueda, Frontera)
17
18         AgregarAPotencialesCaminos(ArbolBusqueda, PotencialesCaminos,
19         SelectElement)
20
21         AgregarAPotencialesCaminos(ArbolBusqueda, PotencialesCaminos,
22         SelectElement)
23
24         Camino = PotencialesCaminos.last()
25
26         return Camino
27     }
28
29     function AgregarAPotencialesCaminos(ArbolBusqueda, PotencialesCaminos,
30     Elemento) {
31
32         foreach PotencialCamino en PotencialesCaminos {
33             UltimoElementoDelCamino = PotencialCamino.lastElement()
34
35             if (Elemento en ArbolBusqueda.vecinos(UltimoElementoDelCamino)) {
36                 UltimoElementoDelCamino.add(Elemento)
37             }
38         }
39     }
40 }
```

2.

```
1  funcion PrimeroProfundidad(Raiz, ArbolBusqueda) {
2      Frontera = []
3      Frontera.add(Raiz)
4      SelectElement = Raiz
5      PotencialesCaminos = [[Raiz]]
6      Visitados = []
7
8
9      while (!esMeta(SelectElement)) {
10
11          Vecinos = ArbolBusqueda.vecinos(SelectElement)
12
13          Frontera.remove(SelectElement)
14
15          if (Vecinos != []) {
16              Frontera = Vecinos + Frontera
17          }
18
19          if (Frontera.lenght == 0) {
20              return failure
21          }
22
23          SelectElement = Frontera.first()
24
25          AgregarAPotencialesCaminos(ArbolBusqueda, PotencialesCaminos,
26          SelectElement)
27      }
28
29      AgregarAPotencialesCaminos(ArbolBusqueda, PotencialesCaminos,
30      SelectElement)
31
32      Camino = PotencialesCaminos.last()
33
34      return Camino
35  }
36
37  funcion PrimeroAncho(Raiz, ArbolBusqueda) {
38      Frontera = []
39      Frontera.add(Raiz)
40      SelectElement = Raiz
41      PotencialesCaminos = [[Raiz]]
42
43      while (!esMeta(SelectElement)) {
44
45          Vecinos = ArbolBusqueda.vecinos(SelectElement)
46
47          Frontera.remove(SelectElement)
48
49          Frontera = Frontera + Vecinos
50
51          if (Frontera.lenght == 0) {
52              return failure
53          }
54
55          SelectElement = Frontera.first()
56      }
```

```

57     AgregarAPotencialesCaminos(ArbolBusqueda, PotencialesCaminos,
58         SelectElement)
59     }
60     AgregarAPotencialesCaminos(ArbolBusqueda, PotencialesCaminos,
61         SelectElement)
62     Camino = PotencialesCaminos.last()
63
64     return Camino
65
66 }
67
68

```

3. El algoritmo de búsqueda Iterative deeping es completo. La razón es que al correr búsqueda en profundidad repetidamente con niveles de profundidades incrementales, se tiene efectivamente un algoritmo del cual el orden conmutativo de los nodos que visita es el de búsqueda en lo ancho. Este último es un algoritmo de búsqueda completo, con lo cual, Iterative deeping también lo es.

4.

Autonomía: La capacidad de elegir si actuar o no, y en que forma hacerlo.

Agente reflejo: Es un tipo de agente que observa y si ve algo determinado realiza una acción. Caso contrario, no hace nada o hace otra cosa. Esto en base a reglas de condición-acción. No posee un estado interno.

Agente basado en metas: Posee un estado interno que le permite determinar las consecuencias de realizar una acción. A su vez posee metas a alcanzar. De todas las acciones posibles, descarta las que no tienen que ver con las metas que posee, sin importar cual de las que la lleva a la meta es la mejor.

Agente basado en utilidades: En lugar de metas se tiene una utilidad, en estos casos puede haber más de una acción que lleva a la meta. La utilidad sirve para que se tomen las “mejores” decisiones.

5.

```

1     funcion AgenteBasadoEnMetas(DatosDeSensores, ComoEvolucionaElMundo,
2     ConsecuenciasDeLasAcciones, Estado, Metas) : Accion {
3
4         Acciones = ObtenerPosiblesAcciones(DatosDeSensores,
5         ComoEvolucionaElMundo, ConsecuenciasDeLasAcciones, Estado)
6
7         foreach Accion en Acciones {
8             ConsecuenciasDeAccion = ObtenerConsecuenciasDeAccion(
9             ComoEvolucionaElMundo, ConsecuenciasDeLasAcciones, Estado, Accion)
10            if (EstaMasCercaOEsMeta(ConsecuenciasDeAccion, Metas, Accion)) {
11                return Accion
12            }
13        }
14    }

```



```
9     }
10
11     return AccionNula //No hace nada
12 }
13
14
15 funcion AgenteBasadoEnUtilidades(DatosDeSensores, ComoEvolucionaElMundo
16 , ConsecuenciasDeLasAcciones, Estado, Utilidades) : Accion {
17     Acciones = ObtenerPosiblesAcciones(DatosDeSensores,
18 ComoEvolucionaElMundo, ConsecuenciasDeLasAcciones, Estado)
19     AccionATomar = AccionNula // Tiene Valor 0
20
21     foreach Accion en Acciones {
22         ConsecuenciasDeAccionActual = ObtenerConsecuenciasDeAccion(
23 ComoEvolucionaElMundo, ConsecuenciasDeLasAcciones, Estado, AccionATomar
24 )
25
26         ConsecuenciasDeAccion = ObtenerConsecuenciasDeAccion(
27 ComoEvolucionaElMundo, ConsecuenciasDeLasAcciones, Estado, Accion)
28
29         ValorDeEstadoAlQueLlegoActual = ValorDeEstado(Utilidades,
30 ConsecuenciasDeAccionActual)
31
32         ValorDeEstadoAlQueLlego = ValorDeEstado(Utilidades,
33 ConsecuenciasDeAccion)
34
35         if (ValorDeEstadoAlQueLlegoActual < ValorDeEstadoAlQueLlego) {
36             AccionATomar = Accion
37         }
38     }
39
40     return AccionATomar
41 }
```

6. Es una instancia de agente reactivo con estado interno, solo tiene dos acciones (apagar y encender) y si bien pueden haber muchas temperaturas diferentes; solo le interesan las que están por debajo de tres grados de la temperatura seteada o por encima de esta. Se necesita un estado para la temperatura seteada y para el estado actual de prendido o apagado, ya que da a entender que la acción apagar solamente está para cuando se encuentra encendida y viceversa. No hay metas, ya que solamente va cambiando entre dos estados sin fin. Por lo tanto, tampoco puede ser de utilidad.