Рассмотрим задачу:

Пуля массой 9г, летевшая со скоростью 600м/с, попадает в деревянную стену и проникает в нее на глубину 20 см. Определить среднюю силу сопротивления движению пули.

Метаграф для данной задачи может быть представлен следующим образом:



В текстовом представлении:

Metavertex(Name=Bullet,

Vertex(Name:=mass, Value:=0.009),

Vertex(Name:=speed, Value:=600),

Vertex(Name:=energy),

Edge(Name:=e1, V1:=speed, V2:=mass, V3:=energy), \\* здесь и далее V1 – выходная вершина, V2 и V3 – входные \*/

Edge(Name:=e2, V1:=mass, V2:=speed, V3:=energy),

Edge(Name:=e3, V1:=energy, V2:=speed, V3:=mass)

)

Metavertex(Name=Wall,

Vertex(Name:=energy,

Vertex(Name:=depth, Value:=0.02),

Vertex(Name:=resistance\_force),

Edge(Name:=e4, V1:= energy, V2:= depth, V3:= resistance\_force),

Edge(Name:=e4, V1:= depth, V2:= energy, V3:= resistance\_force),

Edge(Name:=e4, V1:= resistance\_force, V2:= energy, V3:= depth)

)

Metaedge(Name=to\_slow\_down

Vertex(Name:=end\_speed, Value:=0)

)

Metaedge(Name=to\_penetrate

Vertex(Name:=end\_speed, Value:=0)

)

Связи могут быть функциями на любом языке или целыми программами, реализация которых хранится отдельно.

Например, для нашей задачи, связи, записанные на языке Python, будут выглядеть следующим образом:

import math

#метавершина Пуля

def e1(mass, energy):

return math.sqrt( 2 \* energy / mass )

def e2(speed, energy):

return 2 \* energy / ( speed \*\* 2 )

def e3(speed, mass):

return mass \* ( speed \*\* 2 ) / 2

/\* Конструктор как-бы моделирует связь между вершинами и метавершинами \*/

class Bullet:

def \_\_init\_\_( self, mass, energy, speed );

self.mass=mass

self.energy=energy

self.speed=speed

def e4 (depth, resistance\_force):

return depth \* resistance\_force

def e5(energy, resistance\_force):

return energe / resistance\_force

def e6(energy, depth):

return energy / depth

class Wall:

def \_\_init\_\_(self, depth, energy, resistance\_force):

self.depth = depth

self.resistance\_force=resistance\_force

self.energy=energy

/\* возможно, end\_speed стоит обыграть через декоратор \*/

def to\_slow\_down(bullet, wall, end\_speed):

wall.energy = bullet.energy - bullet.mass \* (end\_speed \*\* 2) / 2

def to\_penetrate(bullet, wall, end\_speed):

bullet.energy = wall.energy + bullet.mass \* (end\_speed \*\* 2) / 2

Теперь можем применить миварный подход.

Пусть известны m правил и n переменных (входящих в правила либо в качестве исходных, активизирующих их, либо в качестве получаемых, т.е. выходных). Тогда в матрице V (m × n), каждая строка которой соответствует одному из правил и содержит информацию об используемых в правиле переменных, могут быть представлены все взаимосвязи между правилами и переменными. При этом в каждой строке все входные переменные этого правила на соответствующих позициях матрицы помечаются символом x, все выходные – y; все переменные, которые уже получили в процессе вывода или задания исходных данных некоторое конкретное значение, – z; а все искомые (выходные) переменные, т.е. те, которые необходимо "вывести" из исходных (входных) данных, - w. Кроме того, добавим в матрицу V одну строку и один столбец для хранения в них служебной информации.

В начальный момент имеем:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Применяемая функция | Пуля | Пуля.масса | Пуля.энергия | Пуля.скорость | Конечная скорость | Стена | Стена.энергия | Стена.глубина | Стена.сила сопротивления |
| служебная информация | | z |  | z | z |  |  | z | w |
| e1 |  | x | x | y |  |  |  |  |  |
| e2 |  | y | x | x |  |  |  |  |  |
| e3 |  | x | y | x |  |  |  |  |  |
| e4 |  |  |  |  |  |  | y | x | x |
| e5 |  |  |  |  |  |  | x | y | x |
| e6 |  |  |  |  |  |  | x | x | y |
| to\_slow\_down | x |  |  |  | x |  | y |  |  |
| to\_penetrate |  | x | y |  | x | x |  |  |  |
| Bullet | y | x | x | x |  |  |  |  |  |
| Wall |  |  |  |  |  | y |  |  |  |

Применяем связи, для которых известны все входные данные:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Применяемая функция | Пуля | Пуля.масса | Пуля.энергия | Пуля.скорость | Конечная скорость | Стена | Стена.энергия | Стена.глубина | Стена.сила сопротивления |
| служебная информация | | z |  | z | z |  |  | z | w |
| Wall |  |  |  |  |  | y |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Применяемая функция | Пуля | Пуля.масса | Пуля.энергия | Пуля.скорость | Конечная скорость | Стена | Стена.энергия | Стена.глубина | Стена.сила сопротивления |
| служебная информация | | z |  | z | z | z |  | z | w |
| e3 |  | x | y | x |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Применяемая функция | Пуля | Пуля.масса | Пуля.энергия | Пуля.скорость | Конечная скорость | Стена | Стена.энергия | Стена.глубина | Стена.сила сопротивления |
| служебная информация | | z | z | z | z | z |  | z | w |
| Bullet | y | x | x | x |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Применяемая функция | Пуля | Пуля.масса | Пуля.энергия | Пуля.скорость | Конечная скорость | Стена | Стена.энергия | Стена.глубина | Стена.сила сопротивления |
| служебная информация | z | z | z | z | z | z |  | z | w |
| to\_slow\_down | x |  |  |  | x |  | y |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Применяемая функция | Пуля | Пуля.масса | Пуля.энергия | Пуля.скорость | Конечная скорость | Стена | Стена.энергия | Стена.глубина | Стена.сила сопротивления |
| служебная информация | z | z | z | z | z | z | z | z | w |
| e6 |  |  |  |  |  |  | x | x | y |

Таким образом, мы нашли цепочку функций, которая ведет к решению. После оптимизации и нахождения кратчайшей цепочки решений, можем посчитать ответ на задачу.

С помощью данного алгоритма можно решать задачи из относительно узкой области, например, школьной физики.

В перспективе, алгоритм может быть соединен с программами, осуществляющими перевод естественного языка в модель метаграфа и метаграфа в модель естественного языка. Данная объединенная система будет представлять собой слабый искусственный интеллект в некоторой предметной области.

Построение метаграфа для решения школьных задач.

Применение метаграфовой модели к решению школьных задач по физике.

Построение модели предметной области

Алгоритм логического вывода на метаграфовой моделе с применением миварного подхода.

Модуль логического вывода для решения школьных задач.