Joshua

LosAngeles971, 2021

Joshua

> I am not afraid of how much a machine can look like a human being, but how much a human being can be assimilated to a complex machine.
>
> — @Bluepulsar971

Joshua is a software. It was born from an inspiration taken from a book, and its purpose is solving generics problems using a starting knowledge and state. The knowledge consists of events and relationship between events. An event is something which can happen under certain circumstances, and it can be a concept (Fred is a frog) or an action (the farmer crosses the river). A relationship ties two events under the principle of cause-effect. Finally the state is a set of variables in a specific point of time. Variables can determine if an event can happen, as well as the happening of an event can change the state.

Joshua can work as a “business rules manager” o it can solve generic problems.

1. Introduction

TBD

2. How Joshua reasons

From Joshua's perspective, reasoning means understanding if one event (the solution of a problem) can be the final effect of a chain of occurrences of cause-effect relationships. To verify the existance of such type of chain, Joshue leverage on a predefined knowledge and a state.

A knowledge includes a set of events, and a set of cause-effect relationships. A state is just a set of variables.

Even if Joshua is able to increase the knowledge deriving new cause-effect relationships, it needs a starting knowledge, since is not able to build one from scratch. Building a knowledge requires the ability to observe (by senses) and analyze (by reasoning) events, in order to produce new cause-effect relationships.

2.1. Events

An event is a phenomenon which can always happen or it can happen under certain circumstances. Technically, an event is defined by:

- a human readable description of the event itself
- an **optional** list of condition, where each condition is a mathematical equation
- an **optional** list of assignments, where each assigment change the value of a specific variable

2.2. Cause-effect relationships

A cause-effect relationship ties two events, defining that the happening of one event (cause) "may" trigger the happening of another one.

For Joshua there are two different type of relationships:

- **always true relationships** declares that the happening of event X **always** trigger the happening of event Y
- **possible relationship** declares that the happening of event X **may** trigger the happening of event Y

Thus every relationship has a weight, and the latter is a number into the interval [0,1] such that:

- 0 means that the event X **never** triggers the event Y
- 1 means that the event X **always** triggers the event Y
- any value in (0,1) means that the event X **may** trigger the event Y

2.3. State

The state is a set of variables, whose values can determine if an event can happen, as well as the values can be altered by the occurrence of events.

Managing a state is important to solve some type of problems, where the solution is determinated by reaching specific values of some variables.

2.4. Solving problems

When Joshua loads a knowledge file and transforms it into a directed, weighted graph, where events are the nodes and relationships are the arcs.

NOTE	Givan a graph, solving a problems means asking the question: **is there a concatenation of cause-effect relationships that terminate into the desidered effect?**

When a problem is only related to "always true relationships", then finding out a solution means finding out a path between a cause and an effect into the knowledge's graph. In this case, we say that Joshua is operating like a "rules manager".

Differently, when the occurrence of events are bound to a specific state, then the reasoning is harder. Joshua treats the graph as a kind of state machine, the path from the source event to the desired effect can be really long, and the same cause-effect relationship may occur multiple times.

2.5. Be able to describe the problem

An effective description of the problem is everything (Joshua is not able to do that by itself).

Looking from Joshua's eyes, describing a problems involves an iterative process of four steps:

1. Which event represents the problem's solution?
2. Which others events are possible?
3. Which cause-effect relationship has the problem's solution as effect?
4. Which others relationships are possible?

If some events can occur only if a specific state is reached, then it is necessary to define the state too.

Next paragraph explain the process of describing a problem throughout an example.

2.6. The farmer, the wolf, the goat and the cabbage

> Once upon a time a farmer went to a market and purchased a wolf, a goat, and a cabbage. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and a single one of his purchases: the wolf, the goat, or the cabbage. If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage. The farmer's challenge was to carry himself and his purchases to the far bank of the river, leaving each purchase intact. How did he do it?
>
> — https://en.wikipedia.org/wiki/Wolf, _goat_and_cabbage_problem

Here a problem you solve by attempts until you find out the solution, a kind of brutal force approac. Joshua approaches this kind of problems in the same way, throughout an iterative application of well-known rulesm, and there is no learning in terms of knowledge expansion.

The first step of problem's description is defining the solution's event:

1. The farmer, the wolf, the goat and the cabbage are on the bank B of the river

Other possible events are:

- The farmer brings the cabbage on the bank B of the river
- The farmer brings the cabbage on the bank A of the river
- The farmer brings the goat on the bank B of the river
- The farmer brings the goat on the bank A of the river
- The farmer brings the wolf on the bank B of the river
- The farmer brings the wolf on the bank A of the river
- The farmer goes to the bank A of the river
- The farmer goes to the bank B of the river

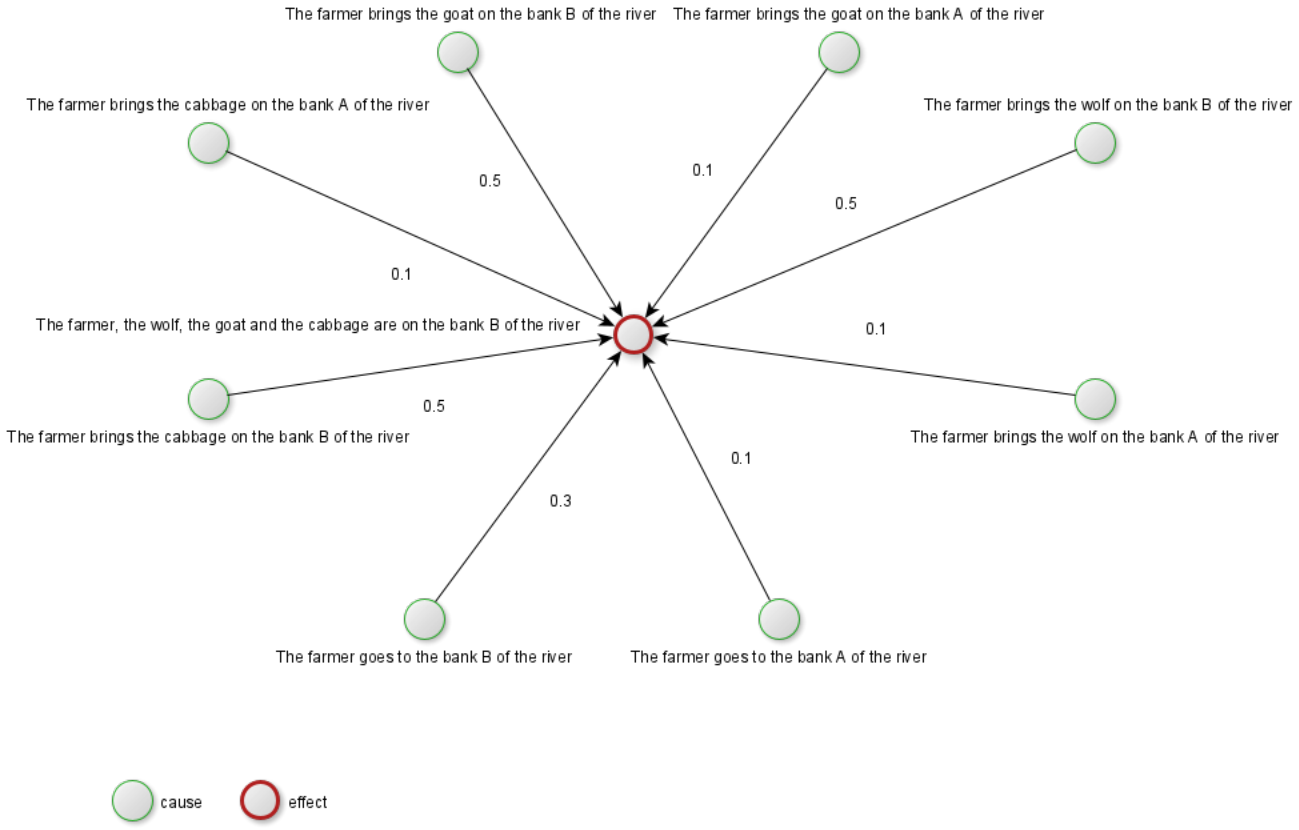
Now e have the events, we can have the cause-effect relationships.

Table 1. Cause-effect relationships

Cause	Effect	Weight
The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	0.5
The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	0.5
The farmer brings the wolf on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	0.5

Cause	Effect	Weight
The farmer brings the cabbage on the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	0.1
The farmer brings the goat on the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	0.1
The farmer brings the wolf on the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	0.1 The farmer goes to the bank A of the river
The farmer, the wolf, the goat and the cabbage are on the bank B of the river	0.3 The farmer goes to the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river

Having defined the problem in this way, the resulting knowledge's graph include several "possible" cause-effect relationships, all of them with the same effect (the problem's solution event).



NOTE

The weights only affect how joshua gives precedence to relationships, when he has to choose from a plurality of "possible" relationships. Therefore if all weights were randomly chosen from the interval (0,1), Joshua would reach the solution anyway (maybe slower, maybe faster).

You can easily see that the events can occur only if a specific state is reached. Indeed, the event "The

farmer brings the cabbage on the bank B of the river" can only happen if the farmer and the cabbage are on the bank A of the river, such type of condition comes from a state.

We define out state identifying 8 variables.

NOTE The way we are describing the problem likely is not the only one possible.

Table 2. Variables of the state

Variable	Initial value	Meaning
FarmerA	1	1 if the farmer is on river's bank A, 0 otherwise
FarmerB	0	1 if the farmer is on river's bank B, 0 otherwise
WolfA	1	1 if the wolf is on river's bank A, 0 otherwise
WolfB	0	1 if the wolf is on river's bank B, 0 otherwise
GoatA	1	1 if the goat is on river's bank A, 0 otherwise
GoatB	0	1 if the goat is on river's bank B, 0 otherwise
CabbageA	1	1 if the cabbage is on river's bank A, 0 otherwise
CabbageB	0	1 if the cabbage is on river's bank B, 0 otherwise

NOTE The state may determine the conditions for an event to occur, as well as the occurrence of events may change the state.

Most of the defined events have conditions. For example, below the complete definition of the event "The farmer brings the cabbage on the bank B of the river".

Table 3. Definition of the event: "The farmer brings the cabbage on the bank B of the river"

Condition/assignment	Expression
Condition	FarmerA == 1
Condition	CabbageA == 1
Condition	WolfA == 0) && (GoatA == 1
	WolfA == 1) && (GoatA == 0
Assignment	FarmerB = 1
Assignment	FarmerA = 0
Assignment	CabbageA = 0
Assignment	CabbageB = 1

Conditions of events arise from the problem's nature, thus the event "The farmer brings the cabbage on the bank B of the river" can happen only if:

- The farmer and the cabbage are on the bank A of the river
- The wolf and the g\oat are not in the same place (the wolk would eat the goat)

If the event "The farmer brings the cabbage on the bank B of the river" occurs, its assignments change the state accordingly (the farmer and the cabbage will be on the bank B of the river).

You may read the complete YAML files (knowledge and initial state) at the appendix.

2.7. Reasoning

Aking Joshua to find out the solution of the problem "The farmer, the wolf, the goat and the cabbage" produces 24 steps.

Table 4. Solving "The farmer, the wolf, the goat and the cabbage"

Step	Cause	effect	Outcome	State changed?
0	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
1	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
2	The farmer brings the wolf on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
3	The farmer goes to the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
4	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
5	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
6	The farmer goes to the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true

Step	Cause	effect	Outcome	State changed?
7	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
8	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
9	The farmer brings the wolf on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
10	The farmer goes to the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
11	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
12	The farmer goes to the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
13	The farmer brings the cabbage on the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	true but loop	true
14	The farmer brings the goat on the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
15	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
16	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	true but loop	true

Step	Cause	effect	Outcome	State changed?
17	The farmer brings the wolf on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
18	The farmer goes to the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
19	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
20	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
21	The farmer brings the wolf on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
22	The farmer goes to the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
23	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	cause not happened	false
24	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	true	true

Above table shows the cause-relationships that Joshua tried to reach the desired effect.

The brutal force approach is recognizable by all attempts with the outcome "cause not happened". Indeed, the outcome "cause not happened" means that a cause-effect relationship cannot happen, since its cause cannot occur, usually because the cause requires a state that is not there. For example the very first attempt is the cause-effect "The farmer brings the cabbage on the bank B of the river", the latter cannot happen, because the farmer would left the wolf and the goat alone.

Differently, the outcome "effect not happened" means that the cause of the cause-effect relationship happened, but not the effect. For example, at the cycle 1 the cause "The farmer brings the goat on

the bank B of the river" occurred, but the effect (the problem’s solution) did not yet (since the effect requires a specific state to occur).

If you clean the table of all foolish attempt, you have a clean path of cause-effect occurrences.

Table 5. Solving "The farmer, the wolf, the goat and the cabbage"

Step	**Cause**	**effect**	**Outcome**	**State changed?**
1	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
6	The farmer goes to the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
7	The farmer brings the cabbage on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
14	The farmer brings the goat on the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
17	The farmer brings the wolf on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
22	The farmer goes to the bank A of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	effect not happened	true
24	The farmer brings the goat on the bank B of the river	The farmer, the wolf, the goat and the cabbage are on the bank B of the river	true	true

3. Use Joshua in the financial sector

TBD

4. Evolution

TBD

5. User guide

Joshua is available for the following architectures:

- Windows amd64
- Linux amd64
- ARM amd64
- Mac OS/X

The software is available for download at the link:

?????

5.1. How to write a knowledge YAML file

5.2. How to write a problem YAML file

5.3. CLI usage

```
joshua.exe solve -k k_contadino.yml -p p_contadino.yml -m 50 --queuelog
$ ./joshua.exe solve -k ../resources/k_contadino.yml -p ../resources/p_contadino.yml
-m 50 --queuelog
```

6. Other problems

7. The heritage

> A person gets an heritage of 1024€, and it starts spending half of the amount each day. How many days does he take to stay with less than 1 euro?
>
> — anonymous

This simple mathematical problem is interesting because the representing graph is different from that of the problem "The farmer, the wolf, the goat and the cabbage".

Describing it is pretty fast:

- the solution’s event is: "He has less than 1 euro"
- the others events:
  - One day of spending
  - The inheritance is halved

Three cause-effect relationships are enough.

Table 6. Cause-effect relationships

Cause	Effect	Weight
One day of spending	The inheritance is halved	1.0
One day of spending	He has less than 1 euro	0.5
The inheritance is halved	He has less than 1 euro	0.5

Note how this problem:

- there is an always true relationship, indeed when he has one day of spending, then the inheritance is always halved
- there is one event that is the cause of multiple effects

Finally, for this problem the state contains only one variable:

- Inheritance = 1024

8. The water containers

> There are three containers A, B, C. A can hold a maximum of 3 litres of water, and it is empty. B can hold a maximum of 5 litres of water, and it is empty. C can hold a maximum of 8 litres of water, and it is full. There are no marks on the containers. You need to have at least one of B or C with 4 litres of water, how can you do that?
>
> — anonymous

This problem is really similar to "The farmer, the wolv, the goat and the cabbage", even if this one is more related to math and the state plays a crucial role. For this reason, it it may be better starting to define the state.

Table 7. Variables of the state

Variable	Initial value	Meaning
A	0	Amount of litres into container A
B	0	Amount of litres into container be
C	8	Amount of litres into container C
MaxA	3	Maximum amount of litres of the container A
MaxB	5	Maximum amount of litres of the container B
MaxC	8	Maximum amount of litres of the container C

The state may have additional variables, but now it is better to define the events:

- [solution’s event] B or C contains 4
- Empty C in A
- Empty C in B
- Empty B in A
- Empty B in C
- Empty A in B
- Empty A in C

Since the state plays a crucial role to solve this problem, the significant part is defining the assignments due to the events occurrence.

Below table shows the complete definition of the event "Empty C in A".

Table 8. Definition of the event: "Empty C in A"

Condition/assignment	**Expression**	**Condition**
C > 0	Assignment	DeltaA = MaxA - A
Assignment	SubC = min(C, DeltaA)	Assignment
C = C - SubC	Assignment	A = A + SubC

Note that the assignments are a little bit complex, because two temporary variables (DeltaA and SubC) are necessary to represent the final situation after the occurrence of the event.

Finally the definition of the possible relationships.

Table 9. Cause-effect relationships

Cause	**Effect**	**Weight**
Empty C in A	B or C contains 4	0.5
Empty C in B	B or C contains 4	0.5
Empty A in B	B or C contains 4	0.5
Empty A in C	B or C contains 4	0.5
Empty B in A	B or C contains 4	0.5
Empty B in C	B or C contains 4	0.5

Solving this requires a trial-and-error approach, as you can notice looking at the relationships, since the type of all of them is possibilistic.

9. Flights and pilots

> Carlo, Bruno and Alberto work in the same team. One is pilot, one is copilot, and one is an engineer. The copilot does not have sisters and brothers, and he earns the least of all. Carlo is the husband of the Bruno's sister, and he earns more than the pilot. Can you say the job of everyone?
>
> — anonymous

A logic problem is probably less intuitive to describe, but it is really interesting, since it highlights Joshua's ability to exploit the concatenation of cause-effect relationships.

TBD

Appendice

Il contadino, il lupo, la capra e il cavolo

The knowledge file.

```
---
  events:
    - id: The farmer, the wolf, the goat and the cabbage are on the bank B of the
river
      statements:
        - FarmerB == 1
        - WolfB == 1
        - GoatB == 1
        - CabbageB == 1

    - id: The farmer brings the cabbage on the bank B of the river
      statements:
        - FarmerA == 1
        - CabbageA == 1
        - ((WolfA < 1) && (GoatA == 1)) || ((WolfA == 1) && (GoatA < 1))
      assignments:
        - FarmerB = 1
        - CabbageB = 1
        - FarmerA = 0
        - CabbageA = 0

    - id: The farmer brings the cabbage on the bank A of the river
      statements:
        - FarmerB == 1
        - CabbageB == 1
        - ((WolfB < 1) && (GoatB == 1)) || ((WolfB == 1) && (GoatB < 1))
      assignments:
        - FarmerA = 1
        - CabbageA = 1
        - FarmerB = 0
        - CabbageB = 0

    - id: The farmer brings the goat on the bank B of the river
      statements:
        - FarmerA == 1
        - GoatA == 1
      assignments:
        - FarmerB = 1
        - GoatB = 1
        - FarmerA = 0
        - GoatA = 0

    - id: The farmer brings the goat on the bank A of the river
```

```
    statements:
      - FarmerB == 1
      - GoatB == 1
    assignments:
      - FarmerA = 1
      - GoatA = 1
      - FarmerB = 0
      - GoatB = 0

  - id: The farmer brings the wolf on the bank B of the river
    statements:
      - FarmerA == 1
      - WolfA == 1
      - ((CabbageA < 1) && (GoatA == 1)) || ((CabbageA == 1) && (GoatA < 1))
    assignments:
      - FarmerB = 1
      - WolfB = 1
      - FarmerA = 0
      - WolfA = 0

  - id: The farmer brings the wolf on the bank A of the river
    statements:
      - FarmerB == 1
      - WolfB == 1
      - ((CabbageB < 1) && (GoatB == 1)) || (CabbageB == 1) && (GoatB < 1))
    assignments:
      - FarmerA = 1
      - WolfA = 1
      - FarmerB = 0
      - WolfB = 0

  - id: The farmer goes to the bank A of the river
    statements:
      - FarmerB == 1
      - WolfB == 1 || CabbageA == 1 || GoatA == 1
      - CabbageB == 1 || WolfA == 1 || GoatA == 1
      - WolfA == 1 || CabbageA == 1 || GoatA == 1
    assignments:
      - FarmerA = 1
      - FarmerB = 0

  - id: The farmer goes to the bank B of the river
    statements:
      - FarmerA == 1
      - WolfA == 1 || CabbageB == 1 || GoatB == 1
      - CabbageA == 1 || WolfB == 1 || GoatB == 1
      - WolfB == 1 || CabbageB == 1 || GoatB == 1
    assignments:
      - FarmerB = 1
      - FarmerA = 0
```

```
  links:
    - type: 0.5
      cause:
        - The farmer brings the cabbage on the bank B of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river

    - type: 0.5
      cause:
        - The farmer brings the goat on the bank B of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river

    - type: 0.5
      cause:
        - The farmer brings the wolf on the bank B of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river

    - type: 0.1
      cause:
        - The farmer brings the cabbage on the bank A of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river

    - type: 0.1
      cause:
        - The farmer brings the goat on the bank A of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river

    - type: 0.1
      cause:
        - The farmer brings the wolf on the bank A of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river

    - type: 0.3
      cause:
        - The farmer goes to the bank A of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river

    - type: 0.5
```

```
      cause:
        - The farmer goes to the bank B of the river
      effect:
        - The farmer, the wolf, the goat and the cabbage are on the bank B of the
river
```

The state file.

```
---
  variables:
    - name: FarmerA
      value: 1.0
    - name: FarmerB
      value: 0.0
    - name: WolfA
      value: 1.0
    - name: WolfB
      value: 0.0
    - name: GoatA
      value: 1.0
    - name: GoatB
      value: 0.0
    - name: CabbageA
      value: 1.0
    - name: CabbageB
      value: 0.0
  success: "The farmer, the wolf, the goat and the cabbage are on the bank B of the
river"
```

The heritage

The knowledge file.

```
---
  events:
    - id: One day of spending
      statements:
        - Inheritance > 1.0
    - id: The inheritance is halved
      assignments:
        - Inheritance = Inheritance / 2
    - id: He has less than 1 euro
      statements:
        - Inheritance < 1.0

  links:
    - type: 1.0
      cause:
        - One day of spending
      effect:
        - The inheritance is halved

    - type: 0.5
      cause:
        - One day of spending
      effect:
        - He has less than 1 euro

    - type: 0.5
      cause:
        - The inheritance is halved
      effect:
        - He has less than 1 euro
```

The state file.

```
---
  variables:
    - name: Inheritance
      value: 1024
  success: "He has less than 1 euro"
```

The water containers

The knowledge file.

```
---
  events:
    - id: Empty C in A
      statements:
        - C > 0
      assignments:
        - DeltaA = MaxA - A
        - SubC = min(C, DeltaA)
        - C = C - SubC
        - A = A + SubC

    - id: Empty C in B
      statements:
        - C > 0
      assignments:
        - DeltaB = MaxB - B
        - SubC = min(C, DeltaB)
        - C = C - SubC
        - B = B + SubC

    - id: Empty A in B
      statements:
        - A > 0
      assignments:
        - DeltaB = MaxB - B
        - SubA = min(A, DeltaB)
        - A = A - SubA
        - B = B + SubA

    - id: Empty A in C
      statements:
        - A > 0
      assignments:
        - DeltaC = MaxC - C
        - SubA = min(A, DeltaC)
        - A = A - SubA
        - C = C + SubA

    - id: Empty B in A
      statements:
        - B > 0
      assignments:
        - DeltaA = MaxA - A
        - SubB = min(B, DeltaA)
        - B = B - SubB
        - A = A + SubB

    - id: Empty B in C
      statements:
        - B > 0
      assignments:
```

```
        - DeltaC = MaxC - C
        - SubB = min(B, DeltaC)
        - B = B - SubB
        - C = C + SubC

    - id: B o C contiene 4
      statements:
        - B == 4 || C == 4

  links:
    - type: 0.5
      cause:
        - Empty C in A
      effect:
        - B or C contains 4

    - type: 0.5
      cause:
        - Empty C in B
      effect:
        - B or C contains 4

    - type: 0.5
      cause:
        - Empty A in B
      effect:
        - B or C contains 4

    - type: 0.5
      cause:
        - Empty A in C
      effect:
        - B or C contains 4

    - type: 0.5
      cause:
        - Empty B in A
      effect:
        - B or C contains 4
    - type: 0.5

      cause:
        - Empty B in C
      effect:
        - B or C contains 4
```

The state file.

```
---
  variables:
    - name: A
      value: 0
    - name: B
      value: 0
    - name: C
      value: 8
    - name: MaxA
      value: 3
    - name: MaxB
      value: 5
    - name: MaxC
      value: 8
    - name: DeltaA
      value: 0
    - name: DeltaB
      value: 0
    - name: DeltaC
      value: 0
    - name: SubA
      value: 0
    - name: SubB
      value: 0
    - name: SubC
      value: 0
  success: "B or C contains 4"
```