1 Alice automatas

In this section we must present a evolution automatas using a technique we call sequential contextualization.

1.1 Alice elements

The basic Alice system is composed by the following elements:

- Symbol: A symbol is a element that represents an interaction. In Alice, this interaction can be a pure interaction with the environment or a internal interaction.
- Step: A step is a partial state representation that represents the sequential operation of the automata.
- Link: A link is a directional relation between steps that represents the temporal sequence of steps.
- Layer: A layer is a set of steps related between them by links and containing only one active step at each time.

1.1.1 Symbol

A symbol represents an interaction. There are two basic types of symbols: internal and external symbols. External symbols represents interaction with environment and internal symbols represents internal interactions. Any special internal symbol represented by ϕ we call start symbol, is required and commented later.

1.1.2 Step

A step is one entity that produce a symbol when activated (output interaction or action) or reversely can be activated by the reception of a symbol (input interaction or context trailing). Each new external symbol introduced must lead to the creation of one and only one step for that symbol. The internal level symbol ϕ has a associated step called **start step** that is created whith each level instantiation.

1.1.3 Link

A link is a directional association between previous step and next step inside a layer. A link represents any feasible case of temporal sequence of step activations inside a layer. Any step can have many links pointing to activate-just-after-me steps and reversely, any step can have many activated-just-before-me links. Also, **a link is a internal symbol**, that can be produced by a inner layer or by the step-container layer.

1.1.4 Layer

A layer is a set of steps related by links that only can contain an active step. Several layers can exist. A layer is down layer if the symbols interacting with the layer are representing interaction with the environment. Other layers are

internal layers and these kind of layers deal with internal symbols. On the present basic model, layers are stacked one over other. The state of the automata is the product of the state of each layer. At each layer always exist one and only one active step.

2 Alice dialog

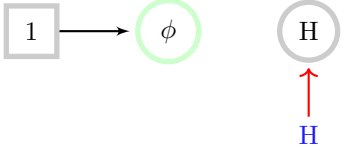
Alice dialog is a sequence of environmental symbols that can be splitted in subsequences wich are in turn received and produced by Alice. The subsequences received by Alice causes the automata to follow it using prediction based on previous data or change his structure for being able to predict those sequence. Both cases changes the internal state, but the later also changes the internal structure. Now we will illustrate the contextualization process. In the following representation we show:

- layers : Each layer is represented by small square with a number inside. Each layer can have a upper-layer connected by an arrow from the former (down-layer) to the upper-layer. Numbers start at cero and increase by one at each stage upwards.
- steps : Each step is represented by a small green or black circle, depending on if the step is active or not. Remember that only one step can be active at each level.
- links : Links between steps would be represented by arrows between steps using red or blue arrows. Red links would be used for representing the first created link from one step to anoter step (also refered as "natural action"), while the others are represented in blue colour.
- symbols : For this representation we represent external and special symbol ϕ by blue letters. Each symbol can have a double arrow to the center of any step that represents that symbol.

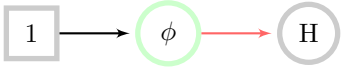
We would illustrate the contextualization process of the phrase "Hello Alice, are you doing well?"

- **start** The system starts with a single layer containing a special internal symbol ϕ and start-step

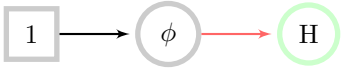
- **H** Now, the letter H enters as a external symbol to the system. We do the following steps:
  - create a step for the symbol.



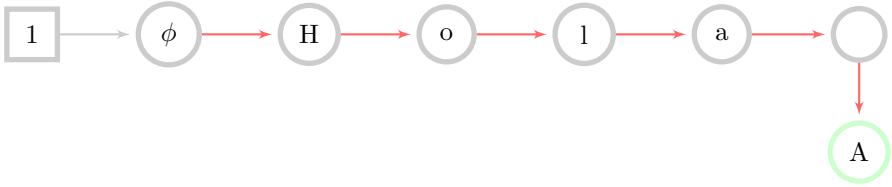
– create a link from the previous active step to the next one.



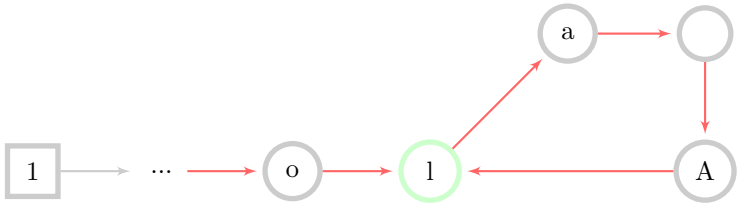
– activate the next step.



- **o,l,a, ,A, l** we continue the process until we reach the next problem. The structure before the new "l" become introduced is as follows:

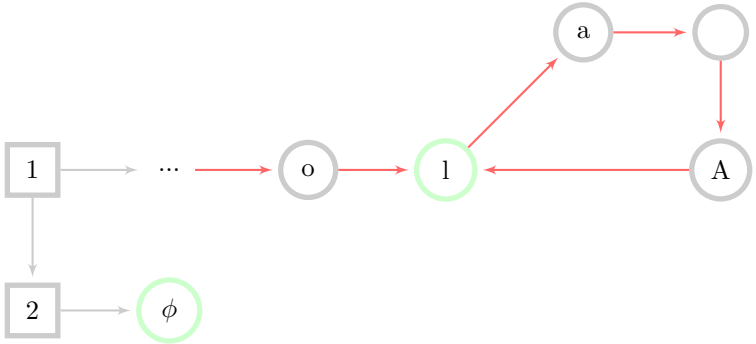


so now, after receiving the input "l", the structure will become (omitting some steps):

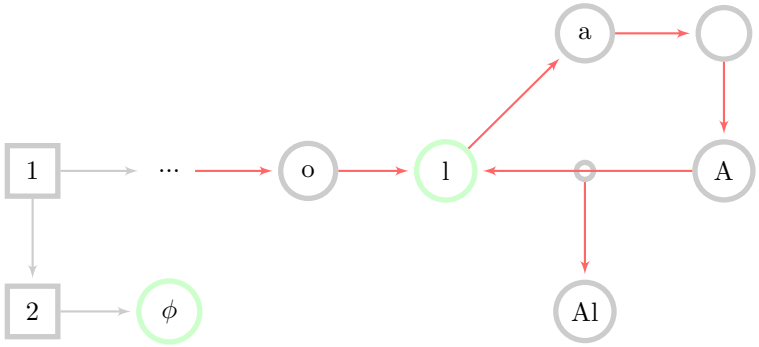


Why this is problematic? This is problematic because we lose information. If we only see that "l" step is active, what was the previous active step? The "o" or the "A"?. The automata has losed information about what happened before. The solution at this case is to remember that each link can be considered as a internal node so we act as follows

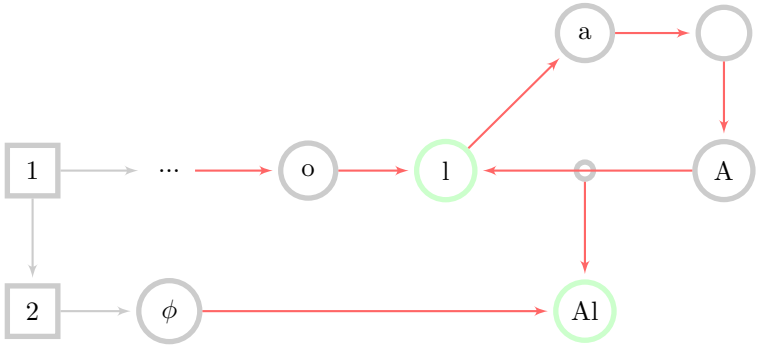
– if don't exist an upper level, create it.



– use the link from A to l as internal symbol for the upper level

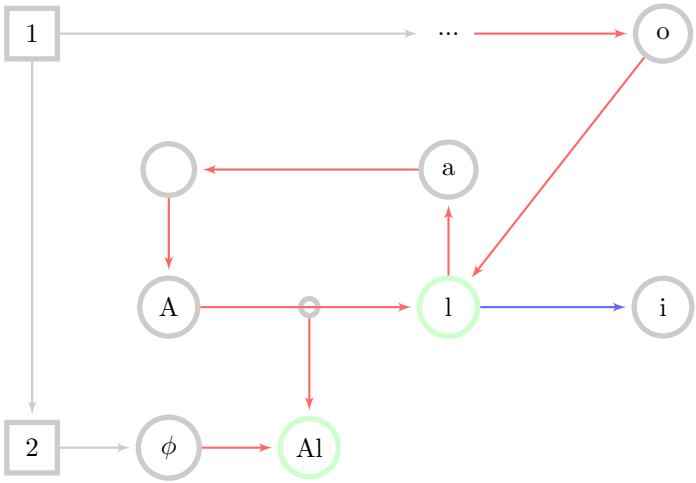


– create a link into the super level from the active node and activate this super step.



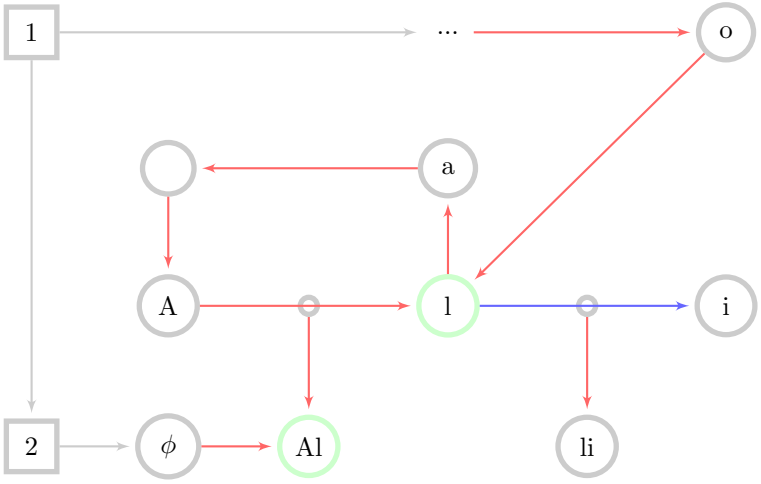
With this process we preserve the history. If both the step(l) and step(Al) are active, we know that the previous active was step(A) at level(1) and step(ϕ) at level(2). But the process of contextualization don't ends up there. This process of creation of upper level nodes can be seen as "reentering path/remember were" the next case that just complete the process is "exiting path/remember from".

- **i** entering just gives us another problem. We change a little the geometry for showing better the case. After entering the **i** and create a a link from the active level 1 node :

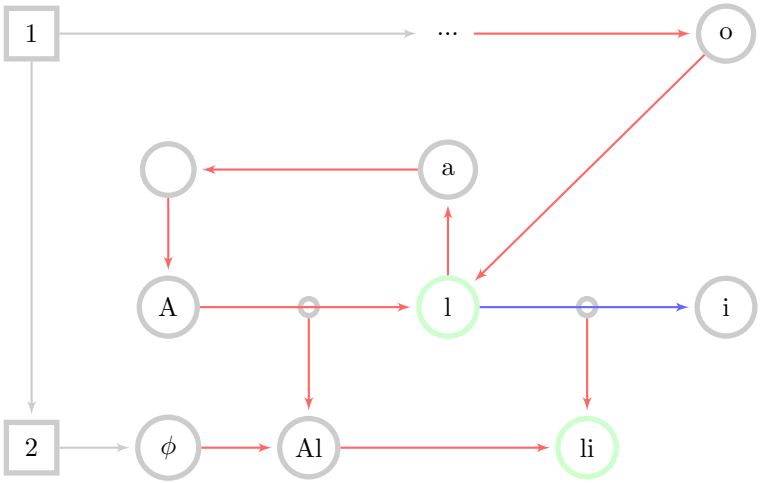


The problem now is that the link between the **l** and the **i** is not the first one from the former. The red links are natural actions in the expentance of that the more frequent can be the first to happen (we recall this later). The new link (blue color) is not a natural reponse so the automata can't decide by itself to choose between **a** or **i** as action (In fact it will decide the wrong **a** by being natural action). But this case can only happen after reentering the path, so the superlevel active step **Al** is enough to create a perfect context. So we proceed as follows:

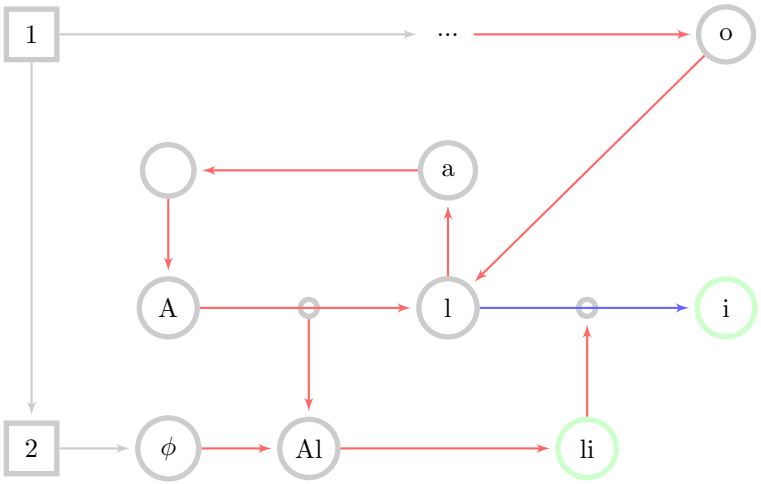
– use the new link as symbol for the superlevel and create a step for it.



– link the new superlevel step and activate it.



– Activate the new base leve symbol.



The natural action links (red) can inject information on the super level while normal links (blue) require information from the superlevel to be choosen as action.

The process can continue creating deeper levels, and the automata is able to make predictions based on his knowledge. When a prediction fails on base level for any already known symbol eiter a new link is created following the exposed procedure or the existent link is promoted to superlevel and recursively upwards until the exposed procedure can be done.

2.1 Pitails

One important limitation from the above procedure comes from the fact of that the first that happens is not always the more frequent, so natural actions can produce noise in the learning procedure and make it lasting. The quick solution for this is to maintain statistics and rebuild the system with prestablished natural links, This method has also the advantage of that when a level is statistically

mature, can be applied on upper levels only.