Ling 5701: Lecture Notes 10
Hierarchic Sequential Prediction

Contents

10.1 Complex events 1
10.2 Recognition of complex events using event fragments 2
10.3 Recognition Model 2
10.4 Example recognition by hierarchic sequential prediction 4
10.5 Practice 6

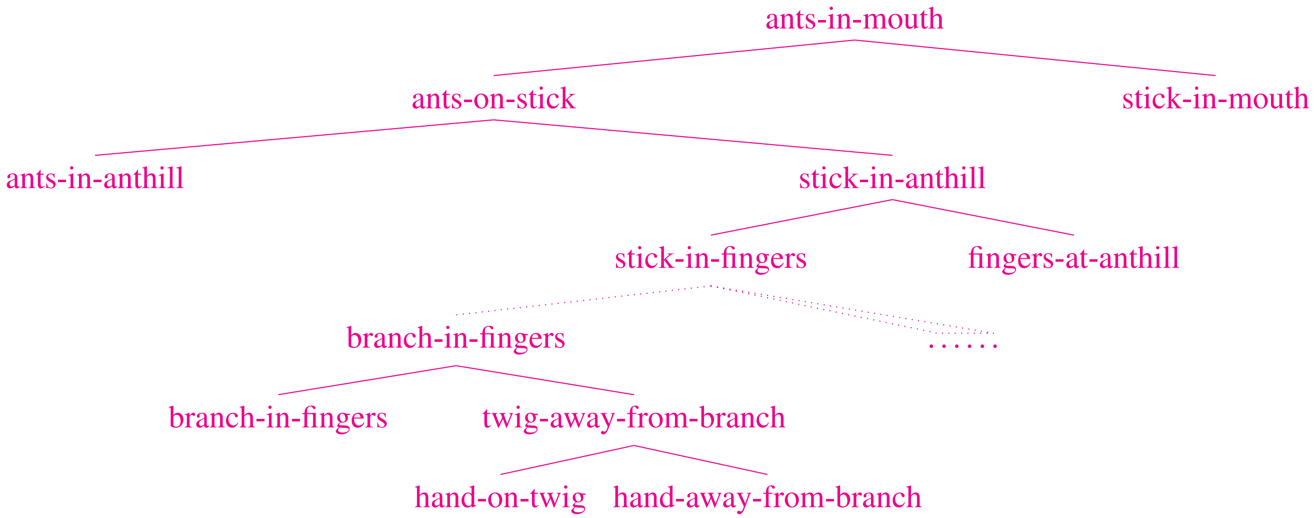
Language processing may be based on domain-general complex event prediction.

This uses memory and generalization (learning) to recognize complex events (plans).

(Recall that events may be represented in the brain as elementary predications. We will assume events are also connected by elementary predications of causation.)

10.1 Complex events

Events can contain hierarchies of subevents, especially complex plans (complex ideas):



Sub-events are related to parent events by ‘cause’ elementary predications.

When similar (recognition) operations are nested inside other operations, a process is called **recursive**.

10.2 Recognition of complex events using event fragments

Humans and (some) animals can recognize and re-create complex hierarchic events.

[Fuster, 1990, Botvinick, 2007]

Partial sequences of events can be grouped and stored as **event fragments** a/b, where:

- a is a **predicted whole** ‘apex’ top-level event or sub-event,
- b is an **expected part** ‘base’ sub-event / observed event yet to come, which completes the whole.

E.g. ants-in-anthill can be accounted as ants-on-stick/stick-in-anthill.

Use cued association (‘A’) to directly link an individual expectation b to a supported prediction a.

Near-complete sub-events can be chained together to save memory:

E.g. ants-on-stick/stick-in-anthill and stick-in-fingers form ants-on-stick/fingers-at-anthill.

When a recent event fragment is completed, it can be added to an earlier event fragment.

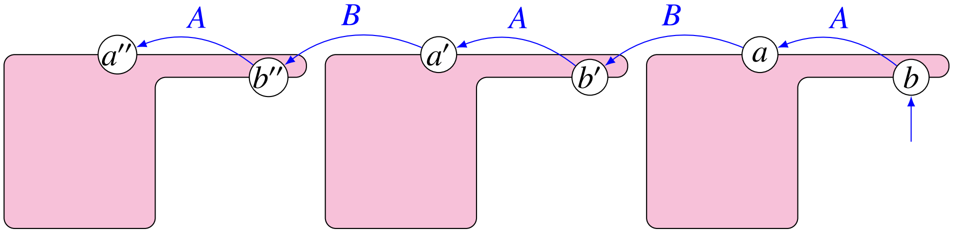
E.g. if stick-in-fingers is complete, it can satisfy stick-in-anthill with fingers-at-anthill expected.

Use cued association (‘B’) to directly link an individual prediction a to a preceding expectation b.

Uncertainty about events may be modeled using superposed activation vectors, described earlier.

10.3 Recognition Model

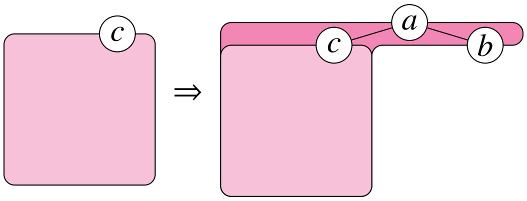
This model maintains a sequence of event fragments accessible from the current expectation b:



E.g. a'/b' is ants-on-stick/stick-in-anthill, a/b is stick-in-fingers/twig-away-from-branch.

Crucially, this store can only be a few elements long before interference causes trouble.

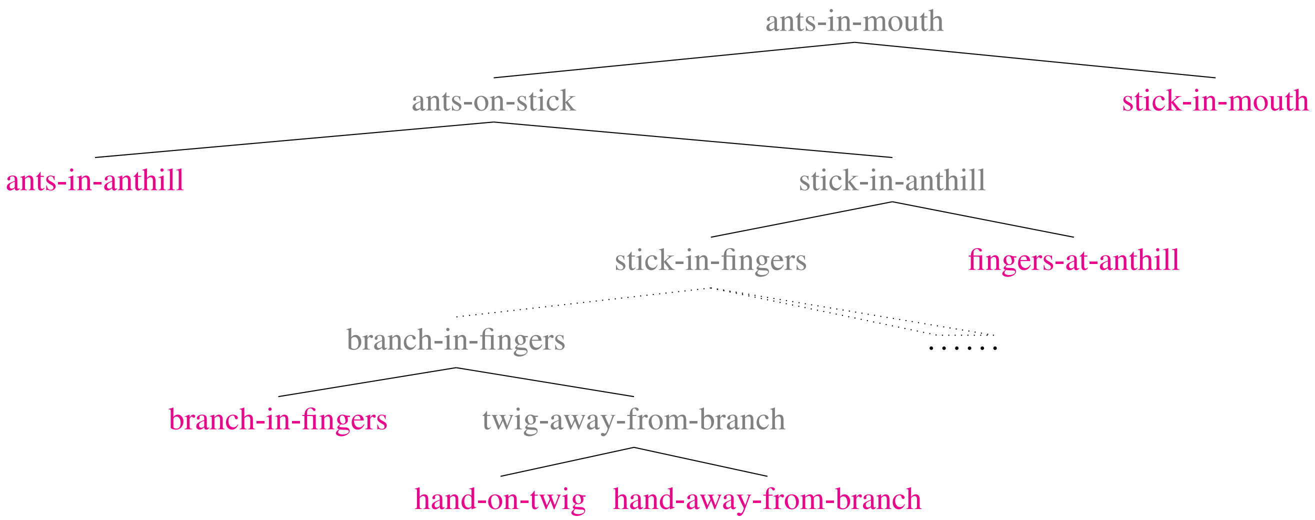
The model also assumes a set of learned **prediction rules**:



E.g. ants-on-stick (*a*) is composed of ants-in-anthill (*c*) followed by stick-in-anthill (*b*).

Here, *a*, *b*, and *c* might be connected by a 'cause' elementary predication (black lines).

First, distinguish terminal (simple, observed) and nonterminal (complex, hidden) events:



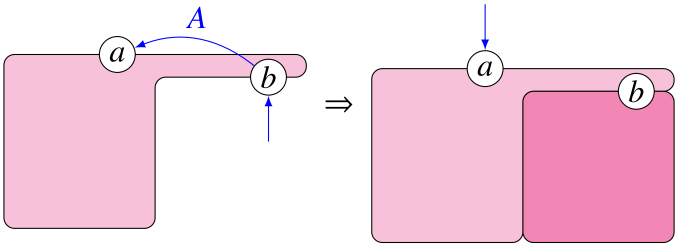
Note: in a binary-branching structure there are equal numbers of terminal and nonterminal events.

We can build this structure by adding one terminal and one nonterminal branch at every observation.

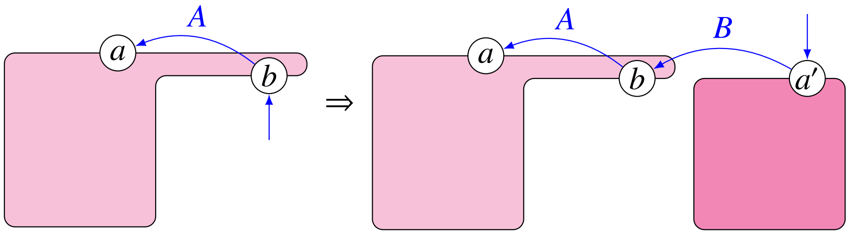
Complex ideas can now be assembled by connecting observed events to event fragments...

- <u>**Terminal**</u> decision (add observed event and connect to existing event fragment, or don't):

  Yes-match outcome (set current prediction):

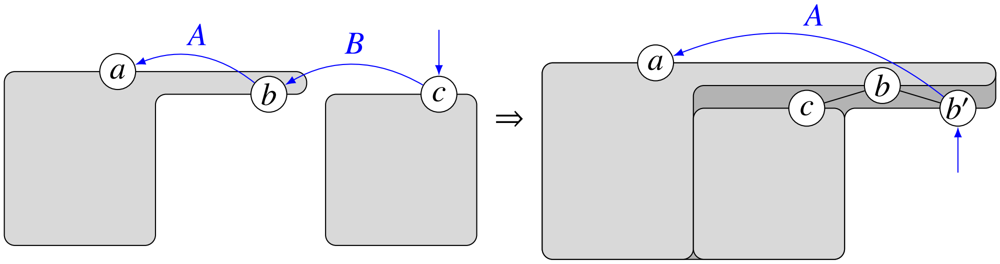


  No-match outcome (check types, store cued association from *a′* to *b*, set current prediction):

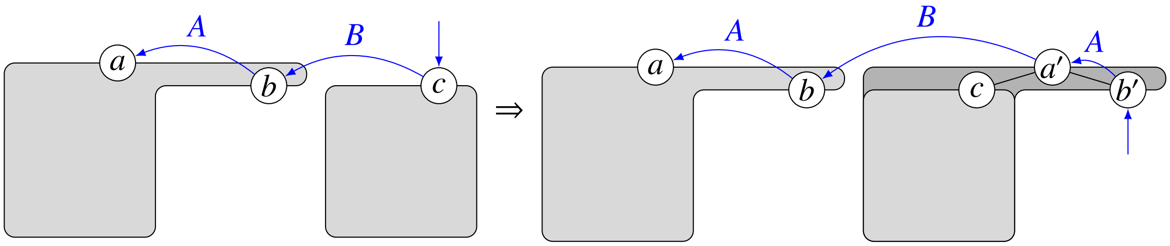


- Nonterminal decision (apply prediction rule and connect resulting event fragment, or don't):

  Yes-match outcome (check types, apply rule, store cued association from b' to a):

  No-match outcome (apply rule, store cued association from b' to a' and a' to b:

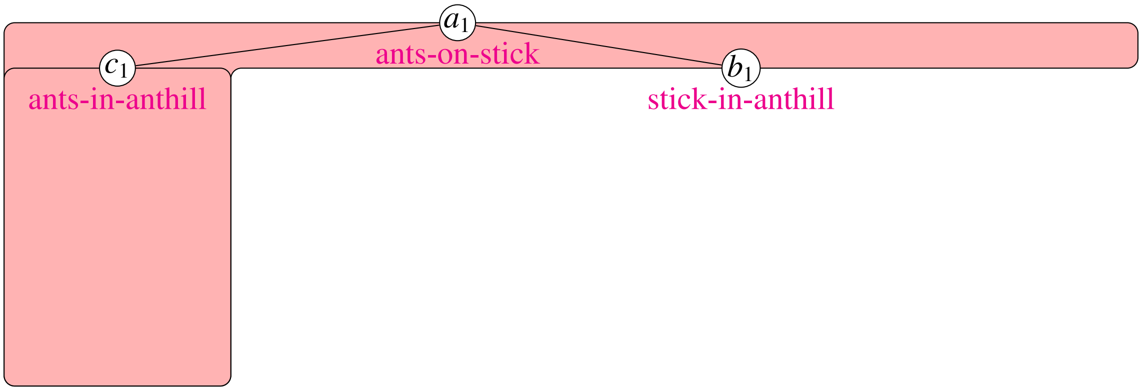
Matching can be implemented in simple neural networks, generalized by procedural learning.

These operations can recognize any branching event structure using a minimum amount of memory.

10.4 Example recognition by hierarchic sequential prediction

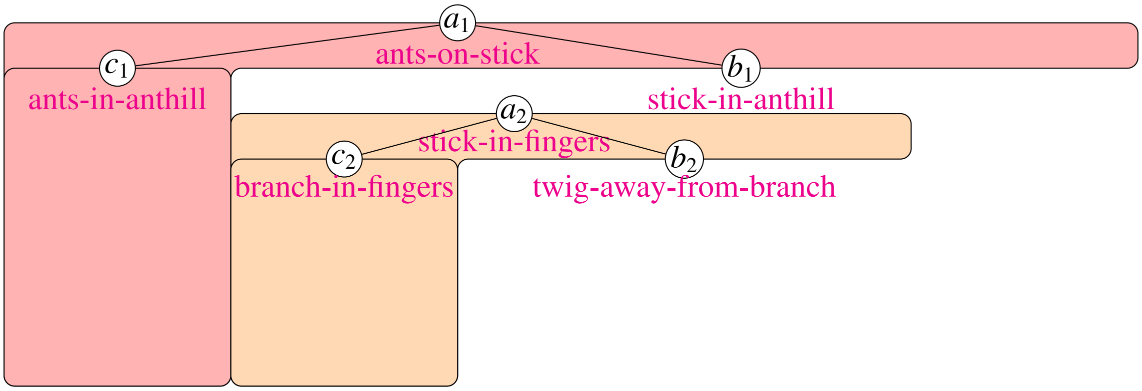
Here is an example of recognizing a complex plan from observations.

Start with observation of anthill c_1, predict ants on stick a_1, and expect a stick in the anthill b_1:

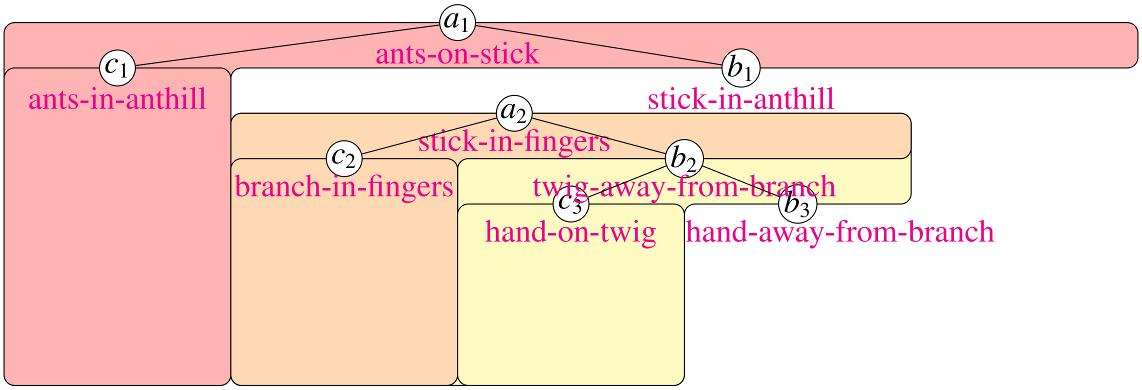


Perhaps other predictions and expectations, like pushing over the anthill, are also superposed at b_1.

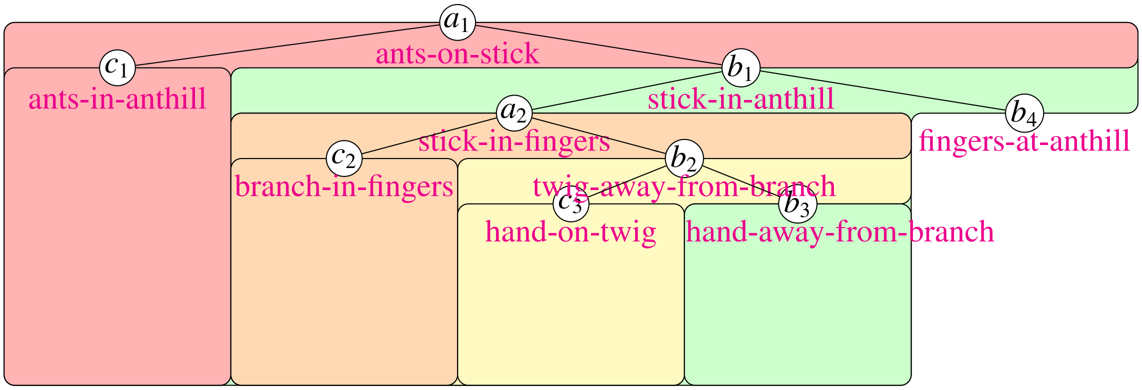
Then don't match observation of branch c_2, don't match prediction stick a_2, leaving new event fragment:



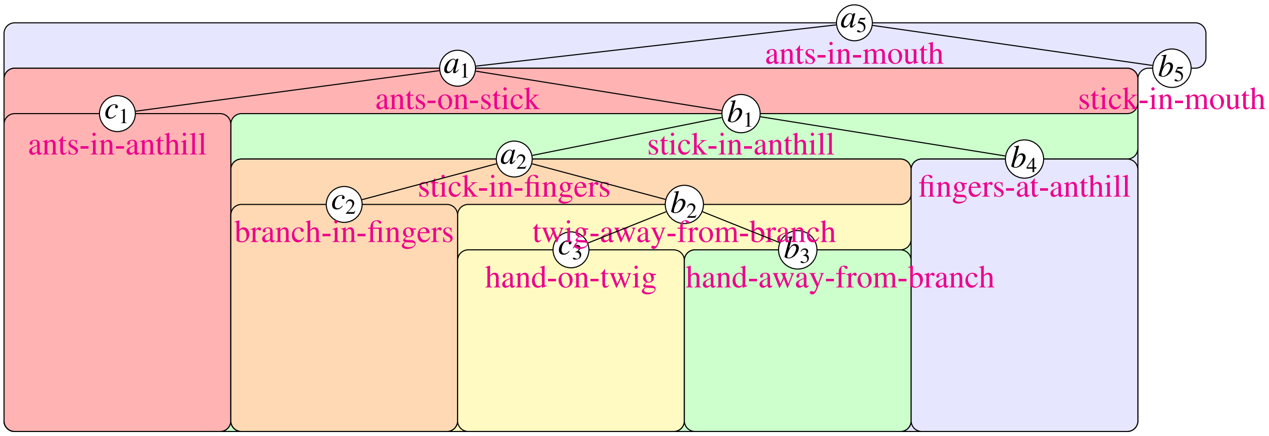
Then match observation about grabbing twig c_3, match prediction to b_2 in previous event fragment:



Then match observation b_3 to complete a_2, match prediction to b_1 in the previous event fragment, leaving only one event fragment:



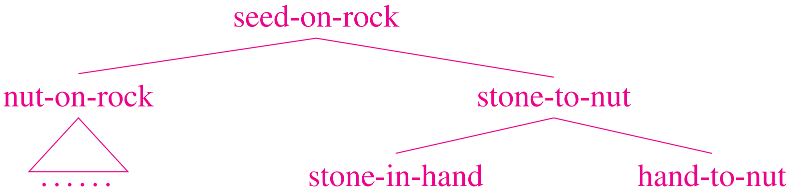
Then match fingers-at-anthill b_4, predict ants-in-mouth a_5, expect stick-in-mouth b_5:



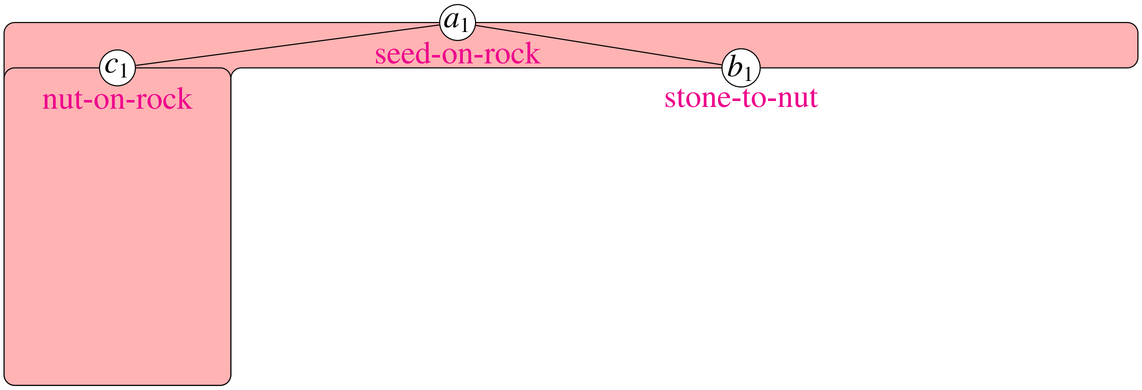
The structure of rule applications over time can be drawn as a tree:

10.5 Practice

Assume the following complex event is being recognized:



and the following event fragments exist after the observation of nut-on-rock:



Draw the event fragments that would exist immediately after observing stone-in-hand. Specifically:

1. What will be the terminal decision outcome, and what event will exist afterward?
2. What will be the non-terminal decision outcome, and what event fragment will exist afterward?

References

[Botvinick, 2007] Botvinick, M. (2007). Multilevel structure in behavior and in the brain: a computational model of Fuster's hierarchy. *Philosophical Transactions of the Royal Society, Series B: Biological Sciences*, 362:1615–1626.

[Fuster, 1990] Fuster, J. M. (1990). Behavioral electrophysiology of the prefrontal cortex of the primate. *Progress in Brain Research*, 85:313–324.