Introduction to AI: Tutorial 2

Abductive Inference and EC Abductive Planning

Alessandra Russo

The aim of this tutorial is to practice with some questions on abductive inference and Event Calculus Abductive planning. The material related to this tutorial is included in Unit 3 and Unit 4. To answer questions, you can use the abductive engine available in CATE. Note that the file in CATE has extension .pdf. It is not a .pdf file, you need to remove the extension .pdf and call the file "abduction.pl". For information on how to use the abductive engine, you can read the user guide also available in CATE. Note that it is important that you know how to construct an abductive proof. So please look at the proof derivations and proof trees that will be given in the model answer for some examples.

Question 1

Consider the biological process of lactose metabolism by the bacterium E. Coli. The background knowledge includes the following pieces of information:

> E. coli can feed on the sugar lactose (*lact*) if it makes two enzymes permease (*perm*) and galactosidase (*gala*). Enzymes (E) are made if they are coded by a gene (G) that is expressed. Enzyme permease is coded by gene *lac(y)* and enzyme galactosidase is coded by gene *lac(z)*. These genes are part of a cluster of genes (*lac(X)*), that is expressed when the amounts of glucose (*gluc*) are low (*lo*) and lactose (*lact*) are high (*hi*).

1. Define an abductive inference task that explains the lactose metabolism (*feed(lact)*) of E. Coli, by addressing the following three points:
   - a Using the signature $\mathcal{L}$ given below, model in Prolog the above pieces of information as background knowledge KB:
     $\mathcal{L} = \{feed/1, make/1, code/2, express/1, amt/2, code/2, sugar/1\}$.
   - b Define the set A of abducibles as ground instances of predicates *amt* and *sugar*.
   - c Define two integrity constraints that state respectively that the amount of a substance (S) may not be both high and low; and that the amount of a substance can only be known if the substance is a sugar.
2. Compute an abductive solution for the task given in part (1), with goal (*feed(lact)*). Use the abductive engine provided in CATE.
3. Draw the abductive proof derivation of the solution generated in part (2).

Question 2

Consider the following legal reasoning extract. The background knowledge includes the following pieces of information:

> People *born in* USA are USA *citizen*. People *born outside* USA but *resident of* USA and *naturalised*, are USA *citizen*. People *born outside* USA but *registered* in USA and with USA *citizen* mother, are USA *citizen*. Mary is John's *mother*. Mary is USA *citizen*.

1. Define an abductive inference task that explains the possibilities of *John be USA citizen*, by addressing the following three points:

   a Using the signature $\mathcal{L}$ given below, model in Prolog the above pieces of information as background knowledge KB:
   $\mathcal{L} = \{citizen/2, mother/2, bornIn//2, bornOut/2, resident/2,$
   $naturalised/2, registered/2\}.$

   b Define the set A of abducibles as ground instances of the predicates *bornIn*, *bornOut*, *resident*, *naturalised*, *registered*.

   c Define the integrity constraint that states that John is not USA *resident*.

2. Compute all abductive solutions for the task given in part (1), with goal (*citizen(john, usa)*). Use the abductive engine provided in CATE. State which one are minimal solutions.

3. Draw the abductive proof tree of all solutions generated in part (2).

Question 3

Consider the following logical circuit for adding three binary digits (see Figure 1). This is an extension of the logical circuit given in Slide 29 of Unit3 on abduction.

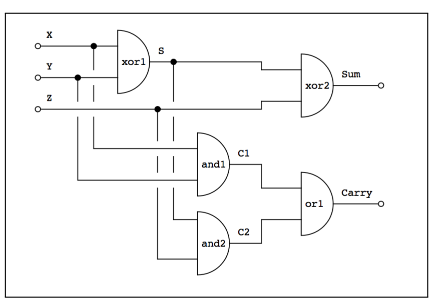

Figure 1: Logical Circuit for adding 3 binary digits

1. Building upon the example given in Slide 29 of Unit3, define an abductive inference task that computes diagnosis of fault operations of the circuit.

2. Define a query that accepts an empty abductive solution. Would such a query accept also non empty abductive solutions? Explain why and give an example of a non empty abductive solution.

3. Using the abductive engine, define a query that requires a double fault in the circuit as possible explanation, compute all abductive solutions and state which one are minimal diagnostic outcomes.

4. Consider the query $G = [add(add, 0, 0, 1, 0, 1)]$. Using the abductive engine, compute all abductive solutions, list them, state which one are minimal solutions and which one is the most cost effective to check first for repair.

Question 4

This exercise is about abduction-based diagnosis to find faults in power supply networks. Consider a geographical region with a power supply control centre that gets informed as soon as any village in the region is without electricity power. The centre checks, using mobile phones, whether the remaining villages have electricity power. On the basis of this information the centre has to take the decision of where to send the repair teams. The diagnostic system receives as observations the information of which villages have or have not electricity power, and have to give suggestions of which wires or power plants might be down, and therefore candidates for repair. A wire can be in one of two states, $state(w, up)$ or $state(w, down)$ and similarly for a power plant (denoted as *pp*). The electricity power network is depicted in Figure 2, where pp is the only power plant in the region, $w1w9$ are the wires, $n1n4$ are nodes in which a number of wires are connected, and $v1v5$ are the villages.

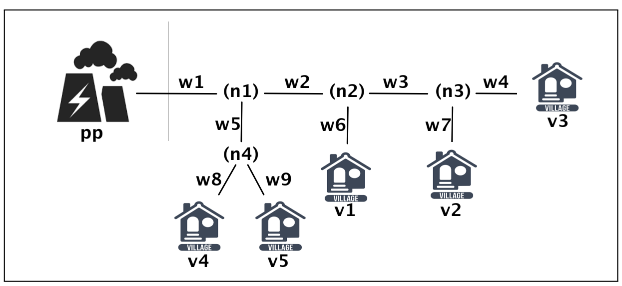

Figure 2: Power Supply Network

1. Define an abductive inference task for fault diagnosis of the above power supply network, by addressing the following three points:

   a Using the signature $\mathcal{L}$ given below, model in Prolog the above power supply network as background knowledge KB. Note that you need to explicitly define when a node or a village has no power. This is in order to be able to abductively diagnose which wire is up and which is down.

   $\mathcal{L} = \{has\_power/2, has\_no\_power/2, state/2\}$.

b Define the set A of abducibles as ground instances of the predicate $state/2$.

c Define an integrity constraint that forces a *state* to have only one of the two possible values *up* and *down*.

2. Using the abductive engine, compute the minimal abductive solutions for each of the following three sets of observations:

$$\begin{array}{l} observe\_all\_butV3\text{:-} \\ \quad has\_power(v1), has\_power(v2), has\_no\_power(v3), \\ \quad has\_power(v4), has\_power(v5). \end{array}$$

$$\begin{array}{l} observe\_all\_butV1V3\text{:-} \\ \quad has\_no\_power(v1), has\_power(v2), has\_no\_power(v3), \\ \quad has\_power(v4), has\_power(v5). \end{array}$$

$$\begin{array}{l} observe\_V4V5\text{:-} \\ \quad has\_no\_power(v1), has\_no\_power(v2), has\_no\_power(v3), \\ \quad has\_power(v4), has\_power(v5). \end{array}$$

Question 5

Consider the simple abductive planning problem described in Slide 30 of Unit 3, but extended with the following additional information.

- The domain includes four items *car*, *tv*, *home*, *pc* and *money*.
- You can $buy(.)$ any item, except for money, if you $have(money)$.
- You can *hire* any item, except for money and houses, if you $have(money)$.
- You can hire a car if you own a driving license.
- Buying leaves you with no money.
- You can borrow anything, except for money and houses.

1. Formalise it as an Event Calculus abductive planning task. Assume 5 time points (hint: define a $time(T)$ predicate using the range operator in Prolog) and that the only integrity constraint is that given in slide 18 of Unit 4.

2. Use the Event Calculus domain independent axioms given in Appendix. Consider the abductive solution
$\Delta = \{happens(borrow(car), 1), happens(borrow(money), 2), happens(buy(home), 3)\}$.
Draw the abductive proof derivation of Δ for this new task, with respect to the goal $G = \{holds(have(car), 2), holds(have(home), 4)\}$.

3. Assume now that *you cannot borrow car unless you have a driving license*. Express this information using an integrity constraint. State if the abductive task has a solution for the given goal $G = \{holds(have(car), 2), holds(have(home), 4)\}$. Modify the abductive proof derivation you have given in part (2) to show your answer. What would you add to the BK to guarantee that the given gold has a plan?

4. Using the abductive engine, explore what happens when your goal is not ground. What possible plans would lead you to a state where the goal $G = holds(have(X), 3)$, is satisfied?

5. Modify the task given in part (2) to express the precondition of $hire(car$ as an integrity constraint. Use the abductive engine to compute plans for the goal $G = holds(have(X), 3)$. Give an informal proof tree derivation of the abductive reasoning process that leads to the three solutions. Don't unfold the *clipped* literals and keep terms unground where you can.

Question 6

This is an adaptation of a planning problem proposed as part of the international planning competition. The planning domain is about a catering robot that has to prepare sandwiches for children in a school, where some children are allergic to gluten. The problem includes two actions for making sandwiches. The first action ($makeSand(Bread, SandFilling)$ makes a sandwich and the second action ($makeNoGlutenSand(Bread, SandFilling)$ makes a sandwich taking into account that all ingredients are gluten-free. There are also actions to *put a sandwich on a tray* ($putOnTray(Sandwich, Tray)$), *move a tray to the kitchen* from a storageRoom ($moveTrayKitchen(Tray)$), and *serve sandwiches* ($serveSandwich(Sandwich, Child)$). The goal consists of having a child served with a sandwich to which he/she is not allergic.

- The domain includes four ingredients $bread1$, $sandFilling1$ (both gluten-free), and $bread2$, $sandFilling2$ (both not gluten-free), two trays ($tray1$, $tray2$) and two children $john$ who is allergic to gluten and $mark$ who is not.

- The robot can serve to a child, who is allergic to gluten, a gluten-free sandwich that is on the tray, and to a child who is not allergic to gluten, a not gluten-free sandwich that is on the tray.

- The robot can put a sandwich on the tray, when both the sandwich and the tray are in the kitchen.

- A sandwich is in the kitchen only after it has been made by the robot.

- A gluten-free sandwich can be made using all gluten-free ingredient, whereas a non gluten-free sandwich can be made using non gluten-free bread and any ingredient.

- Initially the trays are in the storageRoom.

1. Formalise it as an Event Calculus abductive planning task. Assume 5 time points, and the only integrity constraint is that given in slide 18 of Unit 4.
2. Give the plans for the two goals $holds(sandserved(sand(B, C), john), 5)$ and $holds(sandserved(sand(B, C), mark), 5)$ respectively.
3. Using the abductive engine, verify your answers to part (3), computing all possible plans for the two respective individual goals:
   $eval([holds(sandserved(sand(B, C), john), 5)], Ans)$.
   $eval([holds(sandserved(sand(B, C), mark), 5)], Ans)$.

Appendix

Note that the range of the time point domain can change from question to question.

```
time(T) :- T in 0..4.

% Simplified Event Calculus domain independent axioms

holds(F, T) :-
time(T),
initially(F),
\+ clipped(0, F, T).

holds(F, T) :-
time(T1), time(T),
0 #< T1, T1 #< T,
happens(A, T1),
initiates(A, F, T1).
\+ clipped(T1, F, T).

clipped(T1, F, T) :-
time(T1), time(T), time(T2),
T1 #< T2, T2 #< T,
happens(A, T2),
terminates(A, F, T2).
```