Design and Recognition

Two different tasks use assumption-based reasoning:

- **Design** The aim is to design an artifact or plan. The designer can select whichever design they like that satisfies the design criteria.
- **Recognition** or **Diagnosis** The aim is to find out what is true based on observations. If there are a number of possibilities, the recognizer can't select the one they like best. The underlying reality is fixed; the aim is to find out what it is.

Compare: Recognizing a disease with designing a treatment.

The Assumption-based Framework

The assumption-based framework is defined in terms of two sets of formulae:

- F is a set of closed formula called the facts.
  These are formulae that are given as true in the world.
  We assume F are Horn clauses.
- H is a set of atoms called the possible hypotheses or assumables.
  The possible hypotheses can be assumed if consistent with F.

Making Assumptions

- A **scenario** of $\langle F, H\rangle$ is a set D of atoms in H such that $F \cup D$ is satisfiable.
  This happens if no subset of H is a *conflict* of F.
- An **explanation** of g from $\langle F, H\rangle$ is a scenario that, together with F, implies g.
  D is an explanation of g if $F \cup D \models g$ and $F \cup D \not\models \mathit{false}$.
  A **minimal explanation** is an explanation such that no strict subset is also an explanation.

Example

$a \leftarrow b \wedge c.$
$b \leftarrow e.$
$b \leftarrow h.$
$c \leftarrow g.$
$c \leftarrow f.$
$d \leftarrow g.$
$false \leftarrow e \wedge d.$
$f \leftarrow h \wedge m.$
assumable e, h, g, m, n.

- $\{e, m, n\}$ is a scenario.
- $\{e, g, m\}$ is not a scenario.
- $\{h, m\}$ is an explanation for a.
- $\{e, h, m\}$ is an explanation for a.
- $\{e, g, h, m\}$ isn't an explanation.

Default Reasoning and Abduction

There are two strategies for using the assumption-based framework:

- Default reasoning Where the truth of g is unknown and is to be determined.
  An explanation for g corresponds to an argument for g.
- Abduction Where g is given, and we are interested in explaining it. g could be an observation in a recognition task or a design goal in a design task.

Diagnosis

- Determining what is going on inside a system based on observations about the behavior is the problem of diagnosis or recognition.
- In abductive diagnosis, the agent hypothesizes diseases and malfunctions, as well as that some parts are working normally, to explain the observed symptoms.
- This differs from consistency-based diagnosis in that the designer models faulty behavior in addition to normal behavior, and the observations are explained rather than added to the knowledge base.
- Abductive diagnosis requires more detailed modeling and gives more detailed diagnoses, because the knowledge base has to be able to actually prove the observations. It also allows an agent to diagnose systems in which there is no normal behavior.

Example: diagnostic assistant

bronchitis $\leftarrow$ influenza.
bronchitis $\leftarrow$ smokes.
coughing $\leftarrow$ bronchitis.
wheezing $\leftarrow$ bronchitis.
fever $\leftarrow$ influenza.
soreThroat $\leftarrow$ influenza.
false $\leftarrow$ smokes $\wedge$ nonsmoker.
assumable: smokes, nonsmoker, influenza.

- If wheezing is observed, two minimal explanations: {influenza} and {smokes}.

  These explanations imply bronchitis and coughing.

- If wheezing $\wedge$ fever is observed, one minimal explanation: {influenza}.

- If wheezing $\wedge$ nonsmoker was observed instead, there is one minimal explanation: {influenza, nonsmoker}

  The other explanation of wheezing is inconsistent with being a non-smoker.

Design

- Abduction can also be used for design, in which what is to be explained is a design goal and the assumables are the building blocks of the designs.
- The explanation is the design.
- Consistency means that the design is possible.
- The implication of the design goal means that the design provably achieved the design goal.

Design

- There are two minimal explanations of lit-l1: {live-outside, ok-cb1, ok-l1, ok-s1, ok-s2, up-s1, up-s2}. {down-s1, down-s2, live-outside, ok-cb1, ok-l1, ok-s1, ok-s2}. This could be seen in design terms as a way to make sure the light is on: put both switches up or both switches down, and ensure the switches all work. It could also be seen as a way to determine what is going on if the agent observed that l1 is lit; one of these two scenarios must hold. There are ten minimal explanations of dark-l1: {broken-l1} {broken-s2} {down-s1, up-s2} {broken-s1, up-s2} {broken-cb1, up-s1, up-s2} {outside-power-down, up-s1, up-s2} {down-s2, up-s1} {broken-s1, down-s2} {broken-cb1, down-s1, down-s2} {down-s1, down-s2, outside-power-down} There are six minimal explanations of dark-l1 ?lit-l2: {broken-l1, live-outside, ok-cb1, ok-l2, ok-s3, up-s3} {broken-s2, live-outside, ok-cb1, ok-l2, ok-s3, up-s3} {down-s1, live-outside, ok-cb1, ok-l2, ok-s3, up-s2, up-s3} {broken-s1, live-outside, ok-cb1, ok-l2, ok-s3, up-s2, up-s3} {down-s2, live-outside, ok-cb1, ok-l2, ok-s3, up-s1, up-s3} {broken-s1, down-s2, live-outside, ok-cb1, ok-l2, ok-s3, up-s3} Notice how the explanations cannot include outside-power-down or broken-cb1 because they are inconsistent with the explanation of l2 being lit.

Computing Explanations

To find assumables to imply the query $?q_1 \wedge \ldots \wedge q_k$:

> $ac :=$ "$yes \leftarrow q_1 \wedge \ldots \wedge q_k$"
> **repeat**
>   **select** non-assumable atom a_i from the body of ac;
>   **choose** clause C from KB with a_i as head;
>   replace a_i in the body of ac by the body of C
> **until** all atoms in the body of ac are assumable.

To find an explanation of query $?q_1 \wedge \ldots \wedge q_k$:

- find assumables to imply $?q_1 \wedge \ldots \wedge q_k$
- ensure that no subset of the assumables found implies *false*