KRR – Backward-Forward Chaining & Fuzzy-Control

Author: Marinescu Alexandru

Group: 407

# Table of Contents:

## Backward-Forward Chaining

### Backward Chaining

1. Forward Chaining
2. Personal Example
3. Applications and Implementation
4. Fuzzy-Control

### Vagueness

1. Personal Example
2. Applications and Implementation
3. Bibliography
4. Code Implementations

### Backward-Forward Chaining

1. Fuzzy-Control

# Backward-Forward Chaining

## Backward Chaining

SLD (selective linear definite clauses) resolution is a type of resolution that benefits of the Horn representation of elements in a KB. It comes from:

* **S** – **Selective**: At each step, it selects a single goal (subgoal) to resolve
* **L** – **Linear**: Each new resolvent is derived from the previous one and a clause from the program
* **D** – **Definite Clauses**: It is used for definite logic programs, which are composed of Horn clauses

An **SLD derivation** is a sequence of clauses

Backward chaining is a **goal-directed** search strategy that systematically attempts to prove a query (goal) by working backwards from the goal to the facts in the KB. It works in depth-first method and if in the end, there is no query left in the goal list, it means KB can entail the goal, otherwise no. The derivation chain isn’t unique, as it depends on order of clauses the algorithm can go through.

## Forward Chaining

Forward chaining is a **data-driven** inference method that systematically applies rules to known facts to derive new conclusions until the goal is reached. Unlike backward chaining, which works from goals to facts, forward chaining works from facts to goals. This is similar to a breath-first approach, but it is best suited to propositional logic.

But in FOL, facts and rules contain variables which means infinite possibilities, unlike the ground cases of propositional logic. Terms can be infinitely instantiated and that can make forward and backward chaining to never terminate.

## Personal Example

I have the following rules as an example:

1. If a politician is of high rank and is receiving donations more than 1 million $, then he is influential.  
   [influence, n(high\_rank), n(donations\_more\_than\_one\_million)]
2. If a politician is influential and has authored more than 3 bills, then he is a key policymaker.  
   [key\_policymaker, n(influence), n(authored\_bills\_more\_than\_three)]
3. If a politician is a key policymaker and is part of an old national family then he gains high lobbying power.  
   [high\_lobbying\_power, n(key\_policymaker), n(old\_family)].
4. If a politician has high lobbying power and over 10 years of activity then he is an establishment figure.  
   [establishment\_figure, n(high\_lobbying\_power), n(active\_years\_more\_than\_10)]

I want to derive if a politician is an establishment figure.

The resolution result chain is of length 4, but the goal is only in the last one.

## Applications and Implementation

As mentioned, forward chaining is data driven, so it seems to be better suited for **real time decision making**, or **monitoring**. In [2], they use the method in their corn crop diagnosis pipeline with an addition: priority values for each fact in the conflict (goal) set -> mentioned to represent highest probability of disease/pest, but have to look more into it.

Backwards chaining is goal oriented approach, so it seems better suited for particularization, proofs, etc. One research [3] was using a model based on this for learning difficulties on special needs children. If a fact matches a rule, that rule is executed, and the corresponding hypothesis (possible cause) is added to the database. If there is no match, the system looks for a subgoal and continues the process until the learning difficulty is identified or no information can be gathered.

# Fuzzy-Control

## Vagueness

Vagueness is typically represented in fuzzy logic, where predicates (like "tall," "short," "hot," "cold," etc.) are associated with membership functions that assign a degree of truth (or membership) to a statement. These degrees range from 0 (completely false) to 1 (completely true), and values between 0 and 1 represent partial membership. For example, someone who is 1.75 meters tall might have a membership of 0.7 for the predicate "tall." The relationship between the vague predicate and its base function is encapsulated in a **degree curve**.

Aggregation (and here are multiple options: weighted average, min/max aggregation) is used to obtain a final curve for the goal predicate, but in order to obtain a single value, we would apply methods like getting the centroid of the curve.

While reading back the course, started to ask how ignorance (lack of knowledge) can affect degree curve estimation. Age is also quite vague, but we could represent in as an interval and base on existing data, build a degree curve.

## Personal Example

The rules I’ve chose are:

1. If the mood is grumpy or the gift is shabby, then the reward is coal.  
   [or, [mood/grumpy, gift/shabby], [reward/coal]].
2. If the mood is cheerful, then the reward is moderate.  
   [and, [mood/cheerful], [reward/moderate]].
3. If the mood is jolly or the gift is amazing, then the reward is generous.  
   [or, [mood/jolly, gift/amazing], [reward/generous]]

And the degree curve functions for the needed predicates are: santa\_mood, gift\_quality and reward. Maximum aggregation was used and the final score was decided weighted average (centroid method).

Santa's Mood Functions (Domain: X∈[0,10])



Gift Quality Functions (Domain: X∈[0,10])



Reward Functions (Domain: X∈[0,20])



## 

## Applications and Implementation

Since I’m renovating an apartment, when I searched for applications of fuzzy logic, I stumbled upon [4]. In this paper Mohamed Samad explores how fuzzy logic enhances consumer electronics by enabling devices to mimic human-like decision-making. Unlike traditional binary logic, fuzzy logic operates on a continuum between 0 and 1, allowing smart devices to process imprecise or vague inputs. Thermostats are the most obvious between them all, but there are others as well. It makes decision-making actions much more concise, without the need to choose from a finite set of rigid facts.

Another example is load frequency control (LFC). It helps balance generation and demand, preventing power fluctuations that can lead to blackouts. Traditional controllers struggle with uncertainties in load variations, but fuzzy logic-based controllers provide a more adaptive solution. It is described that fuzzy logic-based LFC system is designed with inputs such as frequency deviation (Δf), rate of change of frequency (dΔf/dt) and power imbalance (ΔP).

Bibliography

1. **Radke, P. (2025, January 23). *Forward chaining vs. backward chaining in artificial intelligence*. Built In.** [**https://builtin.com/artificial-intelligence/forward-chaining-vs-backward-chaining**](https://builtin.com/artificial-intelligence/forward-chaining-vs-backward-chaining)
2. **Nurlaeli, & Subiyanto. (2017). Forward chaining method on diagnosis of diseases and pests corn crop. *AIP Conference Proceedings, 1818*(1), 020038.** [**https://doi.org/10.1063/1.4976902**](https://doi.org/10.1063/1.4976902)
3. **Kurniati, D., & [Second Author's Name]. (2022). Backward chaining model for identifying learning difficulties experienced by special needs children. *Scientific Journal of Informatics, 9*(1).** [**https://doi.org/10.15294/sji.v9i1.33872**](https://doi.org/10.15294/sji.v9i1.33872)
4. **Samad, M. (2023). Enhancing consumer electronics with fuzzy logic: Smart devices that understand user preferences. *Global Journal of Technology and Optimization, 14*(4).** [**https://doi.org/10.37421/2229-8711.2023.14.347**](https://doi.org/10.37421/2229-8711.2023.14.347)
5. **5th and 9th courses of Knowledge Representation and Reasoning, 2024-2025, University of Bucharest, Faculty of Mathematics and Informatics, prof. M. Cidota**

Code Implementations

Backward-Forward Chaining



## Fuzzy-Control

## 