



Knowledge Technology

CLIPS PROJECT - JOBEX

The Problem

- ▶ Using CLIPS 6.3, I implemented an intelligent system that acts as a career advisor. For every user it produces a set of jobs that are best suited for him/her, based on the answers he/she will give to a questionnaire;
- ▶ Every answer generates 1 or more facts. Every fact has an associated numeric value between -1 and 1 called certainty factor (CF);

```
(deftemplate a-fact
  (slot name)
  (slot cf (default 0)))
```

- ▶ Through a repeated application of modus ponens the system infers the best set of jobs, also with an associated certainty factor (CF).

Certainty Theory

- ▶ An uncertainty methodology has been implemented to account for the inherent unreliability of the data, since it is generated from answers given by a human, thus affected by human factors such as level of attention, degree of honesty, lack of knowledge etc.
- ▶ Certainty Theory is a system that works with judgmental measures of belief and associates to every fact a certainty factor (CF).

Certainty Factor

- Certainty Factor (CF) = Belief ($0 \leq CF \leq 1$) – Disbelief ($-1 \leq CF \leq 0$)

Meaning	Value
Definitely not	-1
Almost certainly not	-0.8
Probably not	-0.6
Maybe not	-0.4
Unknown	-0.2 to 0.2
Maybe	0.4
Probably	0.6
Almost certainly	0.8
Definitely	1

Certainty Factor Propagation #1

- ▶ A certainty factor can be associated to a fact axiomatically;
- ▶ **Basic rule of derivation:** IF E THEN H $CF(Rule) - CF(H, E) = CF(E) * CF(Rule)$;
- ▶ **Conjunction:** IF $(E1 \cap E2 \cap \dots)$ THEN H $CF(Rule) - CF(H, E1 \cap E2 \cap \dots) = \min\{CF(Ei)\} * CF(Rule)$;
- ▶ **Disjunction:** IF $(E1 \cup E2 \cup \dots)$ THEN H $CF(Rule) - CF(H, E1 \cup E2 \cup \dots) = \max\{CF(Ei)\} * CF(Rule)$;
- ▶ **Negation:** $CF(!E) = -CF(E)$;
- ▶ **Mix:** IF E1 AND E2 OR E3 AND E4 THEN H - $CF(H) = \text{Max}=\{\min(E1, E2), \min(E3, E4)\} * CF(Rule)$;

Certainty Factor Propagation #2

- ▶ When more rules conclude the same hypothesis, there is a way to combine the values of belief/disbelief;
- ▶ Both > 0 : $CF(CF1, CF2) = CF1 + CF2 * (1 - CF1)$;
- ▶ One < 0 : $CF(CF1, CF2) = (CF1 + CF2) / (1 - \min\{|CF1|, |CF2|\})$;
- ▶ Both < 0 : $CF(CF1, CF2) = CF1 + CF2 * (1 + CF1)$.

Certainty Factor Propagation CLIPS #1

- In CLIPS IF (E1 U E2 U ...) THEN H CF(Rule) - CF(H, E1 U E2 U ...) = $\max\{CF(E_i)\} * CF(Rule)$ is implemented as follows:

```
(defrule rule1
  (or
    (a-fact (name q1) (cf ?cf1))
    (a-fact (name likes-sports) (cf ?cf1)))
  =>
  (if (>= ?cf1 0.4)
    then
      (assert (a-fact (name is-social) (cf (* 0.8 ?cf1))))
      (assert (a-fact (name is-active) (cf (* 0.6 ?cf1))))
      (assert (a-fact (name likes-outdoor-activities) (cf (* 0.8 ?cf1))))
      (assert (a-fact (name likes-spending) (cf (* 0.8 ?cf1))))
    else (if (<= ?cf1 -0.4)
      then (assert (a-fact (name is-introvert) (cf (* 0.6 ?cf1))))
      (assert (a-fact (name likes-indoor-activities) (cf (* 0.9 ?cf1)))))))
```

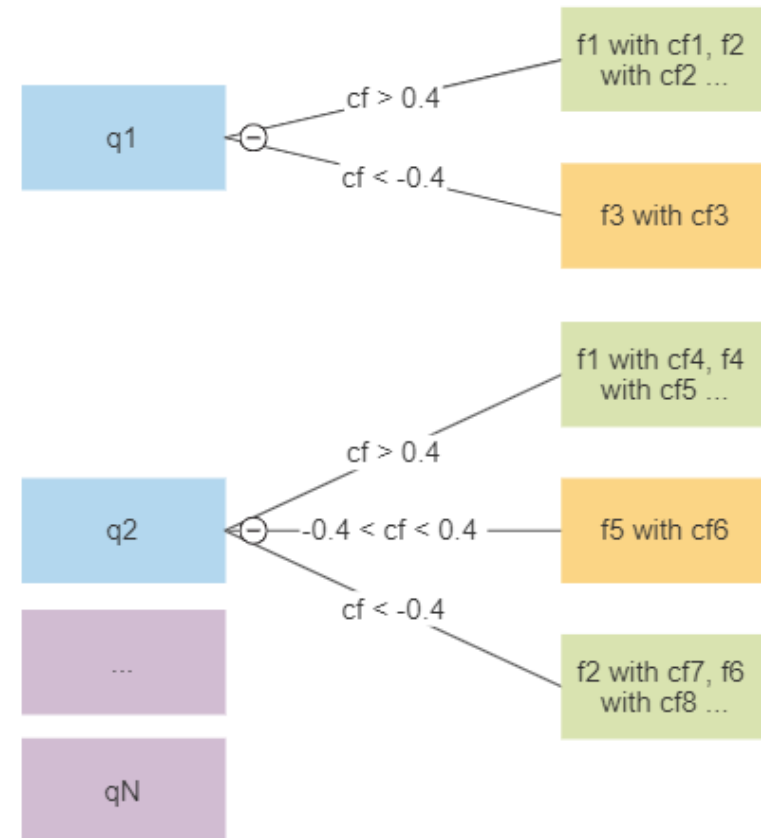
Certainty Factor Propagation CLIPS #2

- In CLIPS $CF(CF1, CF2) = (CF1 + CF2) / (1 - \min\{|CF1|, |CF2|\})$ is implemented as follows:

```
(defrule combine-certainties-3 (declare (salience 100)(auto-focus TRUE))
  ?fact1 <- (a-fact (name ?id) (cf ?cf1))
  ?fact2 <- (a-fact (name ?id) (cf ?cf2))
  (test (neq ?fact1 ?fact2))
  (test (> ?cf1 0))
  (test (< ?cf2 0))
  =>
  (retract ?fact1)
  (modify ?fact2 (cf (/ (+ ?cf1 ?cf2) (- 1 (min (abs ?cf1) (abs ?cf2)))))))
```


Information Flow #1

- ▶ q_i = questionnaire questions;
- ▶ The answer to a q_i is: $-1 < CF < 1$;
- ▶ Depending on the value of the answer, a set of facts are asserted. The certainty factor propagates from the answer to the facts according to the rules of propagation;
- ▶ Different q_i s can infer the same fact, with different or equal certainty factors. The certainty factor of this fact is resolved into one according the rules of propagation.



Information Flow #1 in CLIPS

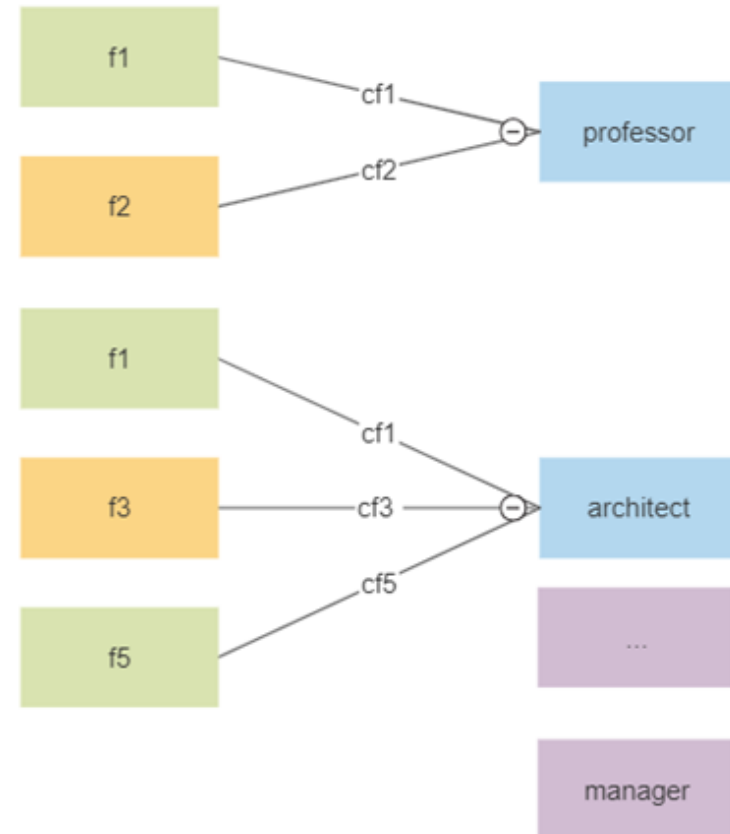
```
(defrule rule4
  (a-fact (name good-with-computers) (cf ?cf1))
  =>
  (if (>= ?cf1 0.4)
    then
      (assert (a-fact (name likes-indoor-activities) (cf (* 0.9 ?cf1))))
      (assert (a-fact (name is-follower) (cf (* 0.7 ?cf1))))
    else (if (<= ?cf1 -0.4)
      then (assert (a-fact (name is-active) (cf (* 0.6 ?cf1)))))))
```

Information Flow #2

- ▶ In the end after enough evidence has been collected is possible to infer a set of best jobs;
- ▶ Each job also has a CF associated to it. For example:

Mathematician with cf 0.6496

Construction Worker with cf 0.899397152038656



Information Flow #2 in CLIPS

```
(defrule job-selection-rule-2
  (finished)
  (a-fact (name is-social) (cf ?cf1))
  (a-fact (name is-leader) (cf ?cf2))
  (a-fact (name likes-teaching) (cf ?cf3))
  =>
  (if (>= (* 0.8 (min ?cf1 ?cf2 ?cf3)) 0.4)
    then
      (printout t "-----" crlf)
      (printout t "Professor with cf " (* 0.8 (min ?cf1 ?cf2 ?cf3)) crlf)
      (printout t "-----" crlf)
      (printout t crlf))))
```

Example #1

I like to work around targets, deadlines and routines for managing my life.
reply: 0.6

I am comfortable working with ambiguous or incomplete information/data and guessing its meaning.
reply: 1

I improvise based on my past experience rather than looking for theoretical understanding.
reply: 0.2

I improvise based on my past experience rather than looking for theoretical understanding.
reply: 0

I am very good with computers.
reply: 1

I would rather cook than go eat out in a restaurant.
reply: 1

Being around a group of people gives me energy.
reply: -1

I enjoy watching and playing team sports.
reply: 0.5

I enjoy watching and playing team sports.
reply: 0.6

I spend most of my spare time actively socialising with friends, attending parties, shopping, etc.
reply: 1

Example #2

The recommended jobs for you are:

Manager with cf 0.499130434782609

Architect with cf 0.570434782608696

Computer Programmer with cf 0.72

Mathematician with cf 0.499130434782609

How to run the program

- ▶ Download and install CLIPS from <http://www.clipsrules.net/>;
- ▶ Load jobex.clp, reset and run;



THANK YOU!