32002: AI (Rule based system: Forward Chaining) **Spring 2014**

Lecture 13: January 30, 2014

Lecturer: K.R. Chowdhary *: Professor of CS (GF)*

13.1 Forward Chaining

In a forward chaining system you start with the initial facts, and keep using the rules to draw new conclusions (or take certain actions), given those facts. Forward chaining systems are primarily *data-driven*. Whenever an if pattern is observed to match an assertion, the antecedent is *satisfied*. When all the *if* patterns of a rule are satisfied, the rule is *triggered*. When a triggered rule establishes a new assertion or performs an action, it is *fired*. This procedure is repeated until no more rules are applicable (figure 13.1).

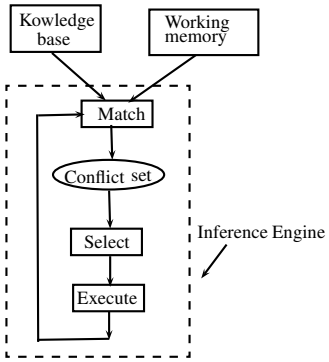

Figure 13.1: Inference Cycle.

The selection process is carried out in two steps:

1. *Pre-selection:* Determining the set of all executable rules, also called the *conflict-set*.

2. *Selection:* Selection of a rule from the conflict set by means of a conflict resolving strategy.

Forward chaining Algorithm: A simple forward chaining algorithm starts from the known facts in the knowledge-base, and triggers all the rules whose premises are the known facts, then adds the consequent for each into the knowledge-base. This process is repeated until the query is answered or until there is no conclusion generated to be added into the knowledge-base. The symbols θ, λ, γ represent unifiers, *unify* is a function which unifies $q^{'}$ and α, and returns a unifier λ if they are unified, else returns null.

Algorithm 1 Forward-chaining(Input: Γ, α) // α is a query, Γ is knowledge-base

```
1: while True do
2:    new = {}
3:    for each sentence s ∈ Γ do
4:       convert s into the format p_1 ∧ p_2 ∧ ··· ∧ p_n → q
5:       for each unifier θ such that (p_1 ∧ p_2 ∧ ..., ∧p_n)θ = (p_1' ∧ p_2' ∧ ..., ∧p_n')θ for some p_i' ∈ Γ do
6:          q' = qθ
7:          new = new ∪ {q'}
8:          λ = unify(q', α)
9:          if λ is not null then
10:            return λ
11:         end if
12:      end for
13:      Γ = Γ ∪ new
14:   end for
15: end while
16: Return Fail
```

The forward chaining algorithm 1 may fail to terminate in the case when query has no answer. For example, every natural number can be generated by recursively applying successor operation on a natural number, and assuming that 0 is natural number. This will lead to indefinite loop for very large numbers.

$$naturalnum(0)$$
$$\forall x[naturalnum(x) \rightarrow naturalnum(succ(x))]$$

Example 13.1 *Produce the inference, given the knowledge-base* $\Gamma = \{man(x) \rightarrow mortal(x), man(socrates)\}$ *and query* $\alpha = mortal(w)$*, i.e., "who is mortal?"*

Solution. We apply the algorithm 1 manually, and note that $p_1 \wedge \cdots \wedge p_n = p_1 = man(x)$, $p_1^{'} = man(socrates)$, and $\theta = \{man/x\}$. Also,

$$\begin{aligned} q^{'} &= q\theta \\ &= mortal(x)\{socrates/x\} \\ &= mortal(socrates) \end{aligned}$$

On unification of α (i.e, $mortal(w)$),

$$\begin{aligned} \lambda &= unify(q^{'}, \alpha) \\ &= unify(mortal(socrates), mortal(w)) \\ &= \{socrates/w\} \end{aligned}$$

the answer obtained is $w = socrates$. □

Complexity Issues: The complexity of the forward chaining algorithm is determined by the inner loop. It find all the possible premises such that the premises unifies with certain set of facts in the knowledge-base

Γ. The process is called pattern matching. This is tried for every rule, for every substitution (unification). Thus, if set of rules are P and there are n^k substitutions for each rule, assuming the worst case k arguments and n number of literals' assignments in the P. This makes the complexity of above algorithm as $|P|n^k$, which is exponential.

The most important conflict resolving strategies, which can also combined together are:

1. *Selection by order*
   - Apply the first matching rule (trivial strategy)
   - Apply the most current rule, i.e., the one whose precondition applied to the most recent entries in the KB.
2. *Selection according to syntactic structure of the rule*
   - Apply the most specific rule: A rule R_1 is more specific than R_2 if the precondition of R_1 (say, A, B, C) is superset of precondition of R_2 (say, A, B).
   - Apply the syntactically largest rule, i.e., the rule which contains the most propositions.
   - Solutions to subproblems are constructed incrementally, and are cached to avoid the recompilation.
3. *Selection by means of supplementary knowledge*
   - Apply the rule with the highest priority. For this purpose each rule must be given a priority, which may be represented, e.g., by a number.
   - Additional rules, called *meta-rules*, control the selection.

Indexing: Another way of improving over the brute-force algorithm is to index all the rules, so that each parameter produces a reference to its rule. When a parameter takes on a new value or changes its value, the reference enables the rules concerned to be found and checked efficiently.

Many conjunct ordering may be possible, for example, given,

$$\forall x \forall y [P(x)Q(x, y) \rightarrow R(x, a)]$$

in this case all the $P(x)$ may be ordered then find $Q(x, y)$ for each x and y. Alternatively, for each x find $P(x)$ and all $Q(x, y)$ for different y is possible. Which approach is better? Finding this is solution of a conjective-ordering problem, which is $NP - hard$. A heuristic can use most constrained, the one having fewest values. This show a close relationship between *CSP* (constraint satisfaction problem), discussed in next chapters, and this pattern matching problems. Due to above mentioned complexity issue the forward-chaining algorithm has *NP-hard* in its inner loop. However, since the size and arities are constant in the real-world, the expression $|P|n^k$ is a polynomial in nature.

References

[RN05] S. Russel and P. Norvig, "Artificial Intelligence - A Modern Approach," *2nd Edition, Pearson Education, India*, 2005, Chapter 9.

[GFL09] D. George F. Ludger, "Artificial Intelligence - Structures and Strategies for Complex Problem Solving," *5th Edition, Pearson Education, India*, 2009, Chapter 8.

[RIC02] Elan Rich and Kevin Knight, "Artificial Intelligence," *2nd Edition, Tata McGraw-Hill, India*, 2002, Chapter 6.

[NIL80] Nils J. Nilsson, "Principles of Artificial Intelligence," *Narosa Publishers, India*, 1980, Chapter 6.