



CSC384: Intro to Artificial Intelligence Knowledge Representation IV

- Answer Extraction
- Factoring (optional, not on Test 2)
- Prolog (not on Test 2)
- Review: One more example (do it yourself)

Answer Extraction.

- The previous example shows how we can answer true–false questions. With a bit more effort we can also answer “fill–in–the–blanks” questions (e.g., what is wrong with the car?).
- As in Prolog we use free variables in the query where we want the fill in the blanks. We simply need to keep track of the binding that these variables received in proving the query.
 - `parent(art, jon)` –is art one of jon’s parents?
 - `parent(X, jon)` –who is one of jon’s parents?

Answer Extraction.

- A simple bookkeeping device is to use an predicate symbol **answer(X,Y,...)** to keep track of the bindings automatically.
- To answer the query `parent(X,jon)`, we construct the clause
 $(\neg \text{parent}(X,\text{jon}), \text{answer}(X))$
- Now we perform resolution until we obtain a clause consisting of only answer literals (previously we stopped at empty clauses).

Answer Extraction: Example 1

1. $\text{father}(\text{art}, \text{jon})$
2. $\text{father}(\text{bob}, \text{kim})$
3. $(\neg \text{father}(Y, Z), \text{parent}(Y, Z))$
i.e. *all fathers are parents*
4. $(\neg \text{parent}(X, \text{jon}), \text{answer}(X))$
i.e. the query is: who is parent of jon?

Here is a resolution proof:

5. $R[4, 3b]\{Y=X, Z=\text{jon}\}$
 $(\neg \text{father}(X, \text{jon}), \text{answer}(X))$
6. $R[5, 1]\{X=\text{art}\}$ **answer(art)**
so art is parent of jon

Answer Extraction: Example 2

1. $(\text{father}(\text{art}, \text{jon}), \text{father}(\text{bob}, \text{jon}))$ //either bob or art is parent of jon
2. $\text{father}(\text{bob}, \text{kim})$
3. $(\neg \text{father}(Y, Z), \text{parent}(Y, Z))$ //i.e. all fathers are parents
4. $(\neg \text{parent}(X, \text{jon}), \text{answer}(X))$ //i.e. query is $\text{parent}(X, \text{jon})$

Here is a resolution proof:

5. $R[4, 3b]\{Y=X, Z=\text{jon}\} (\neg \text{father}(X, \text{jon}), \text{answer}(X))$
6. $R[5, 1a]\{X=\text{art}\} (\text{father}(\text{bob}, \text{jon}), \text{answer}(\text{art}))$
7. $R[6, 3b] \{Y=\text{bob}, Z=\text{jon}\}$
 $(\text{parent}(\text{bob}, \text{jon}), \text{answer}(\text{art}))$
8. $R[7, 4] \{X=\text{bob}\} (\text{answer}(\text{bob}), \text{answer}(\text{art}))$

A disjunctive answer: either bob or art is parent of jon.

Factoring (optional)

1. $(p(X), p(Y))$ // $\forall X. \forall Y. \neg p(X) \rightarrow p(Y)$
2. $(\neg p(V), \neg p(W))$ // $\forall V. \forall W. p(V) \rightarrow \neg p(W)$

- These clauses are intuitively contradictory, but following the strict rules of resolution only we obtain:

3. $R[1a, 2a](X=V) (p(Y), \neg p(W))$

Renaming variables: $(p(Q), \neg p(Z))$

4. $R[3b, 1a](X=Z) (p(Y), p(Q))$

No way of generating empty clause!

Factoring is needed to make resolution complete, without it resolution is incomplete!

Factoring (optional).

- If two or more literals of a clause C have an mgu θ , then $C\theta$ with all duplicate literals removed is called a factor of C .
- $C = (p(X), p(f(Y)), \neg q(X))$
 $\theta = \{X=f(Y)\}$
 $C\theta = (p(f(Y)), p(f(Y)), \neg q(f(Y))) \rightarrow (p(f(Y)), \neg q(f(Y)))$ is a factor

Adding a factor of a clause can be a step of proof:

1. $(p(X), p(Y))$
2. $(\neg p(V), \neg p(W))$
3. $f[1ab]\{X=Y\} p(Y)$
4. $f[2ab]\{V=W\} \neg p(W)$
5. $R[3,4]\{Y=W\} ()$.

Prolog

- Prolog search mechanism (*without not and cut*) is simply an instance of resolution, except
 1. Clauses are Horn (only one positive literal)
 2. Prolog uses a specific depth first strategy when searching for a proof. (Rules are used first mentioned first used, literals are resolved away left to right).

Prolog

- Append:
 1. `append([], Z, Z)`
 2. `append([E1 | R1], Y, [E1 | Rest]) :-
append(R1, Y, Rest)`

Note:

- The second is actually the clause
`(append([E1 | R1], Y, [E1 | Rest]) , \neg append(R1, Y, Rest))`
- `[]` is a constant (the empty list)
- `[X | Y]` is `cons(X, Y)`.
- So `[a,b,c]` is short hand for `cons(a, cons(b, cons(c, [])))`

Prolog: Example of proof

- Try to prove : `append([a,b], [c,d], [a,b,c,d])`:
- 1. `append([], Z, Z)`
- 2. `(append([E1|R1], Y, [E1|Rest]),
¬append(R1,Y,Rest))`
- 3. `¬append([a,b], [c,d], [a,b,c,d])`
- 4. `R[3,2a]{E1 = a, R1 = [b], Y = [c,d], Rest = [b,c,d]}
¬append([b], [c,d], [b,c,d])`
- 5. `R[4,2a]{E1 = b, R1 = [], Y = [c,d], Rest = [c,d]}
¬append([], [c,d], [c,d])`
- 6. `R[5,1]{Z = [c,d]} ()`



Review: One Last Example!

Consider the following English description

- Whoever can read is literate.
 - Dolphins are not literate.
 - Flipper is an intelligent dolphin.
-
- Who is intelligent but cannot read.

Example: pick symbols & convert to first-order formula

- Whoever can read is literate.
 $\forall X. \text{read}(X) \rightarrow \text{lit}(X)$
- Dolphins are not literate.
 $\forall X. \text{dolp}(X) \rightarrow \neg \text{lit}(X)$
- Flipper is an intelligent dolphin
 $\text{dolp}(\text{flipper}) \wedge \text{intell}(\text{flipper})$
- Who is intelligent but cannot read?
 $\exists X. \text{intell}(X) \wedge \neg \text{read}(X).$

Example: convert to clausal form

- $\forall X. \text{read}(X) \rightarrow \text{lit}(X)$
 $(\neg \text{read}(X), \text{lit}(X))$
- Dolphins are not literate.
 $\forall X. \text{dolp}(X) \rightarrow \neg \text{lit}(X)$
 $(\neg \text{dolp}(X), \neg \text{lit}(X))$
- Flipper is an intelligent dolphin.
 $\text{dolp}(\text{flipper})$
 $\text{intell}(\text{flipper})$
- who are intelligent but cannot read?
 $\exists X. \text{intell}(X) \wedge \neg \text{read}(X).$
 $\rightarrow \forall X. \neg \text{intell}(X) \vee \text{read}(X)$
 $\rightarrow (\neg \text{intell}(X), \text{read}(X), \text{answer}(X))$

Example: do the resolution proof

1. $(\neg \text{read}(X), \text{lit}(X))$
 2. $(\neg \text{dolp}(X), \neg \text{lit}(X))$
 3. $\text{dolp}(\text{flip})$
 4. $\text{intell}(\text{flip})$
 5. $(\neg \text{intell}(X), \text{read}(X), \text{answer}(X))$

 6. R[5a,4] $X=\text{flip}$. $(\text{read}(\text{flip}), \text{answer}(\text{flip}))$
 7. R[6,1 a] $X=\text{flip}$. $(\text{lit}(\text{flip}), \text{answer}(\text{flip}))$
 8. R[7,2b] $X=\text{flip}$. $(\neg \text{dolp}(\text{flip}), \text{answer}(\text{flip}))$
 9. R[8,3] **$\text{answer}(\text{flip})$**
- so flip is intelligent but cannot read!