Artificial intelligence resolution proof

Name: - Bhavesh Sharma CS-B EN18CS301067

Propositional Resolution works only on expressions in *clausal form*. Before the rule can be applied, the premises and conclusions must be converted to this form. Fortunately, as we shall see, there is a simple procedure for making this conversion.

A *literal* is either an atomic sentence or a negation of an atomic sentence. For example, if *p* is a logical constant, the following sentences are both literals.

*p*  
*¬p*

A *clausal sentence* is either a literal or a disjunction of literals. If *p* and *q* are logical constants, then the following are clausal sentences.

*p*  
¬*p*  
*¬p* ∨ *q*

A *clause* is the set of literals in a clausal sentence. For example, the following sets are the clauses corresponding to the clausal sentences above.

{*p*}  
{¬*p*}  
{*¬p*, *q*}

Note that the empty set {} is also a clause. It is equivalent to an empty disjunction and, therefore, is unsatisfiable. As we shall see, it is a particularly important special case.

As mentioned earlier, there is a simple procedure for converting an arbitrary set of Propositional Logic sentences to an equivalent set of clauses. The conversion rules are summarized below and should be applied in order.

1. Implications (I):

|  |  |  |  |
| --- | --- | --- | --- |
|  | φ ⇒ ψ | → | ¬φ ∨ ψ |
|  | φ ⇐ ψ | → | φ ∨ ¬ψ |
|  | φ ⇔ ψ | → | (¬φ ∨ ψ) ∧ (φ ∨ ¬ψ) |

2. Negations (N):

|  |  |  |
| --- | --- | --- |
| ¬¬φ | → | φ |
| ¬(φ ∧ ψ) | → | ¬φ ∨ ¬ψ |
| ¬(φ ∨ ψ) | → | ¬φ ∧ ¬ψ |

3. Distribution (D):

|  |  |  |
| --- | --- | --- |
| φ ∨ (ψ ∧ χ) | → | (φ ∨ ψ) ∧ (φ ∨ χ) |
| (φ ∧ ψ) ∨ χ | → | (φ ∨ χ) ∧ (ψ ∨ χ) |
| φ ∨ (φ1 ∨ ... ∨ φ*n*) | → | φ ∨ φ1 ∨ ... ∨ φ*n* |
| (φ1 ∨ ... ∨ φ*n*) ∨ φ | → | φ1 ∨ ... ∨ φ*n* ∨ φ |
| φ ∧ (φ1 ∧ ... ∧ φ*n*) | → | φ ∧ φ1 ∧ ... ∧ φ*n* |
| (φ1 ∧ ... ∧ φ*n*) ∧ φ | → | φ1 ∧ ... ∧ φ*n* ∧ φ |

4. Operators (O):

|  |  |  |  |
| --- | --- | --- | --- |
|  | φ1 ∨ ... ∨ φ*n* | → | {φ1, ... , φ*n*} |
|  | φ1 ∧ ... ∧ φ*n* | → | {φ1}, ... , {φ*n*} |

As an example, consider the job of converting the sentence (*g* ∧ (*r* ⇒ *f*)) to clausal form. The conversion process is shown below.

|  |  |
| --- | --- |
|  | *g* ∧ (*r* ⇒ *f*) |
| I | *g* ∧ (¬*r* ∨ *f*) |
| N | *g* ∧ (¬*r* ∨ *f*) |
| D | *g* ∧ (¬*r* ∨ *f*) |
| O | {*g*} |
|  | {¬*r*, *f*} |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_