Understanding Resolution

0. Useful Definitions

Atom/Atomic Proposition/Propositional Variable

Propositional variable (the same as atomic proposition, and often referred to as atom) is the simplest 'bit' of the propositional formula. Most often, it will be a letter like A, B, C... . It can be symbols, words or more letters (for example, below, to represent 'goose in the boat' I use the atom 'GB'). The defining property is not the symbol used, but the fact that the atom cannot be decomposed further. For example, this term will not refer to an expression like $\neg A$, as it is a combination of the propositional variable A with the negation operator.

Valuation

A valuation is an assignment of truth values to propositional atoms. For example, the expression $A \vee B$ has four distinct valuations: A-true, B-true; A-true, B-false; A-false, B-true; A-false, B-false. Here, first three valuations make the expression true, the last valuation makes the expression false.

Valuations are often represented with a letter V, for example if we have an expression composed of atoms A, B, C and we call it α, we can present a particular valuation like so: $\mathbf{V}(\alpha)$: A-T, B-F, C-F.

Satisfiability

An expression is satisfiable iff there exists a valuation that makes that expression true.

Why should we care?

In real-life situations, the expression in question will represent a problem we want to solve. Once we formulate the problem as a logical expression, we can apply procedures like resolution to it to determine its satisfiability. If we find the expression is unsatisfiable, it means that the problem cannot be solved. If we find that it is possible to satisfy the expression, we can look for the satisfying valuation, which will correspond to the solution of our problem.

As an example, take this very simple problem:
We can place a goose in the boat (represented by the propositional variable GB), on the west side of the river (GW) or on the east side of the river (GE). Let us say we do not want to place the goose on either side of the river (for example because there

is corn growing on each riverbank and we do not want the goose to eat it). Where should we place the goose?

Expressed as a boolean formula, our problem becomes:

$$(GB \vee GW \vee GE) \wedge \neg GW \wedge \neg GE$$

Resolution will lead you to derivation of GB (not deriving the empty clause means that expression is satisfiable, so the problem can be solved). With a refutation tree you will find that valuation which makes the expression true is: GB true, GW false, GE false. Meaning that the goose has to be in the boat, and not on the west side on the river, and not on the east side of the river. Put the goose in the boat – problem solved, corn is safe!

Note that the example above is extremely simple and you could guess very easily that the goose should be placed in the boat – however the procedure you performed in your mind to do it (if goose has to be in either of these three places, and it cannot be on the west side, and it cannot be in the east side, then it has to be in the boat) is exactly what resolution is. Normally we are faced with problems far more complicated and harder to solve in your mind. That is why we formally define rules of logic, so that given a boolean expression, computers can determine its satisfiability for us.

1. Purpose of Resolution

The purpose of resolution is to find out whether a given expression is satisfiable (whether it has a valuation that makes it true) or not, to see if the problem corresponding to the expression is solvable or not.

2. Resolution Mechanism

Resolution works to make it easier for us to determine whether a given expression is satisfiable by identifying the elements of the original expression on which the satisfiability depends. Take the most general example which shows the satisfiability dependencies of the expression $(X \vee Y) \wedge (\neg X \vee Z)$:

$$\frac{\{X, Y\} \qquad \{\neg X, Z\}}{\{Y, Z\}}$$

The most important observation is that we cannot use the variable X to guarantee truthfulness of $(X \vee Y) \wedge (\neg X \vee Z)$. If we make X true, $(X \vee Y)$ will be made true, but $(\neg X \vee Z)$ truthfulness will depend on the truth value of Z. Conversely, if we make X false, $(\neg X \vee Z)$ will be true, but then $(X \vee Y)$ can only be made true if Y is true, and will be false if Y is false. Because X and its negation appears in two different clauses, regardless of the truth value of X, the truthfulness of the entire expression will depend on the remaining variables, here, $(Y \vee Z)$.

In other words, if we found that $(Y \vee Z)$ is false (we cannot satisfy it), we would not be able to satisfy the original expression. Also, note that in order to make the original expression true, the derived clause $(Y \vee Z)$ must be made true.

All resolution steps rely on this principle. That is why if we derive an empty clause (which is always false and there is no way to satisfy it), we can deduce that the original expression is not satisfiable. That is also why if after resolving each possible clause, empty clause is not derived, we can derive a satisfying valuation by picking variable values which satisfy all clauses (and the simplest way to do this is by starting from the last step of resolution and going back; see Example 4.3).

3. Example: Unsolvable Problem

3.1. Defining the problem

Problem:
We can place the goose on the riverbank or in a boat. There is corn growing on the riverbank. The boat is also full of corn (let's say we cannot get it out). If we put the goose in the same place as the corn, the goose will eat it. We want to prevent that from happening as the goose is allergic to corn. Where should we put the goose?

Corresponding boolean expression is:

$$(GRB \vee GB) \wedge CRB \wedge CB \wedge \neg(CRB \wedge GRB) \wedge \neg(CB \wedge GB)$$

where:
GRB – goose is on the riverbank
CRB – corn is on the riverbank
GB – goose is in the boat
CB – corn is in the boat

$(GRB \vee GB)$ – goose is on the riverbank or in the boat
$\neg(CRB \wedge GRB)$ – goose and corn cannot be together on the riverbank
$\neg(CB \wedge GB)$ – goose and corn cannot be together in the boat

Deriving an equivalent CNF expression by applying de Morgan law $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$ to the original expression:

$$(GRB \vee GB) \wedge CRB \wedge CB \wedge (\neg CRB \vee \neg GRB) \wedge (\neg CB \vee \neg GB)$$

3.2. Applying Resolution

Note that the order of the variables to which we apply resolution does not matter for the validity of the resolution. Here the clause that we currently resolve on is the name of the column, and indices next to clauses indicate which step the clause was used in.

	CB	GB	GRB	CRB
$\{CB\}^1$	$\{\neg GB\}^2$	$\{GRB\}^3$	$\{\neg CRB\}^4$	$\{\}$
$\{CRB\}^4$				
$\{GB, GRB\}^2$				
$\{\neg GB, \neg CB\}^1$				
$\{\neg GRB, \neg CRB\}^3$				

1. We may start by resolving on CB. Since corn is in the boat (CB) and we need to make it true that goose is not in the boat or corn is not in the boat ($\neg GB \vee \neg CB$), we must make it true that goose is not in the boat ($\neg GB$).
2. Here I chose to resolve on GB next. We know that goose cannot be in the boat (derived in the previous step) and that it has to be in the boat or on the riverbank ($GB \vee GRB$). We deduce it must be on the riverbank (GRB).
3. Resolving on GRB, since we must make it true that GRB (goose on the riverbank) and ($\neg GRB \vee \neg CRB$) (goose or corn must not be on the riverbank, since they must be separate), we have to make it true that corn is not on the riverbank ($\neg CRB$).
4. Last two clauses that were not used are CRB and $\neg CRB$. We find that to satisfy the expression, we must make it true that corn is on the riverbank (as was defined in the problem) and that corn is not on the riverbank (as we deduced from all the other clauses in previous steps). This is impossible and so leads to empty clause. Empty clause is always false and shows us that the original expression is not satisfiable.

Remember that in cases where there is more than two clauses containing the literal on which you resolve, you have to resolve every combination of each clause which contains the positive literal with each clause that contains its negation (in this case it is ok – even necessary – to use the same clause several times). Only after that you can mark the clauses involved as 'used'.

Since the expression defining the problem is unsatisfiable, the problem is unsolvable – there is no way to place a goose in either boat or riverbank and have it not eat corn if the corn is in the boat and growing on the riverbank.

4. Example: Solvable Problem

4.1. Defining the Problem

Problem:
There is a goose and a wolf. We can place each in a boat or on the riverbank. If we

put the goose and the wolf in the same place, the wolf will eat the goose. How should we place the goose and the wolf to prevent this?

Corresponding boolean expression:

$$(GRB \lor GB) \land (WRB \lor WB) \land \neg(GRB \land WRB) \land \neg(GB \land WB)$$

where:

GRB – goose is on the riverbank
WRB – wolf is on the riverbank
GB – goose is in the boat
WB – wolf is in the boat

$(GRB \lor GB)$ – goose is on the riverbank or in the boat
$(WRB \lor WB)$ – wolf is on the riverbank or in the boat
$\neg(GRB \land WRB)$ – goose and wolf cannot be together on the riverbank
$\neg(GB \land WB)$ – goose and wolf cannot be together in the boat

Deriving an equivalent CNF expression by applying de Morgan law $\neg(A \land B) \equiv (\neg A \lor \neg B)$ to the original expression:

$$(GRB \lor GB) \land (WRB \lor WB) \land (\neg GRB \lor \neg WRB) \land (\neg GB \lor \neg WB)$$

4.2. Applying Resolution

Note that the order of the variables to which we apply resolution does not matter for the validity of the resolution. Here the clause that we currently resolve on is the name of the column, and indices next to clauses indicate which step the clause was used in.

	GRB	GB	WB
$\{GRB, GB\}^1$	$\{GB, \neg WRB\}^2$	$\{\neg WB, \neg WRB\}^3$	$\{WRB, \neg WRB\}$
$\{WRB, WB\}^3$			
$\{\neg GRB, \neg WRB\}^1$			
$\{\neg GB, \neg WB\}^2$			

After three steps of resolution, nothing more can be resolved and we did not derive an empty clause, meaning the original expression was not inconsistent at any point - it can be satisfied.

4.3. Deriving a Satisfying Valuation

The last step of resolution informs us that to satisfy the original expression, it must be true that wolf is on the riverbank or that the wolf is not on the riverbank ($WRB \lor$

$\neg WRB$). We are free to choose which part of the clause we want to make true:

a) choose WRB to be true. Once this is done, we move to the last but one step: $(\neg WB \vee \neg WRB)$. Since WRB is true, we have to make $(\neg WB)$ true - since the wolf is on the riverbank, it cannot be on the boat. Next clause we have to make true is $(GB \vee \neg WRB)$. As WRB is true, we have to make GB true in order to satisfy this clause - since wolf is on the riverbank, the goose is in the boat. Finally we need to take a look at the original clauses and see if they are all made true by our current valuation. The only exception is $(\neg GRB \vee \neg WRB)$ since $\neg WRB$ is false and we have not defined $\neg GRB$. Making GRB false will make this clause true and complete our valuation.

Our derived valid valuation is: WRB-true, WB-false, GB-true, GRB-false.

b) choose WRB to be false. Once this is done, we move to the last but one step: $(\neg WB \vee \neg WRB)$. Since WRB is false, this clause is already made true. We are not forced to make WB true or false to make it true, so for now we will refrain from picking a value for it (that is necessary as you might find you will have to make one of the remaining clauses true by picking a particular value for WB). Next clause we have to make true is $(GB \vee \neg WRB)$. This is also made true by WRB being false, so we move on. In the original clauses, since the false WRB appears in $(WRB \vee WB)$, we have to make WB true (since the wolf is not on the riverbank, it must be in the boat). Negation of WB appears in $(\neg GB \vee \neg WB)$, so we have to make $\neg GB$ true: since it is false that the wolf is not in the boat (the wolf is in the boat), we have to make it true that the goose is not in the boat. Finally, since we made GB false, $(GRB \vee GB)$ can only be made true by making $\neg GRB$ true (the goose is not in the boat so it must be on the riverbank).

Our derived valid valuation is: WRB-false, WB-true, GB-false, GRB-true.

In other words, our problem has two solutions. We either put the wolf on the riverbank and the goose in the boat (valuation a), or we put the wolf in the boat and the goose on the riverbank (valuation b).

This example shows that you can make a different decision at the beginning of assigning values (choose WRB true or false) to satisfy clauses at the last step of resolution (here, $WRB \vee \neg WRB$) and as long as you carefully assign values to remaining clauses, you will be able to satisfy all clauses. Indeed, that is the guarantee that the resolution principle gives you - if you can satisfy the simpler clause you derived, you are able to satisfy the more complicated original expression.

For the Record

Note that if you were to model scenarios like the above, normally you would need to account for the constraint that one entity cannot be in two places at the same time (in other words, use **xor** instead of **or** in clauses like $(GRB \vee GB)$). Examples presented here do not need that as the combination of constraints always implies it. They were intentionally chosen as using **or** makes it easier to follow the argument.

For example, in the last problem we account for the fact that goose/wolf cannot be in two places at the same time through a combination of the four constraints. If goose is on the riverbank, then we will have to place the wolf in the boat, and they cannot be together at the same time, so we cannot place the goose in the boat as well (there is only two entities and two locations, each entity has to be in one of the two locations, and no two entities can be in one location at the same time - therefore no entity can be in both of these places).

These notes were created by Dagmara Niklasiewicz and revised by Michael Fourman. Please send any questions, comments and feedback to Dagmara Niklasiewicz at `s1349662@sms.ed.ac.uk`