CS 245

Resolution provides the basis for a powerful proof system for propositional logic.

However, resolution proofs may not have the structure and intuitive markers that computer scientists, engineers and mathematicians find helpful in constructing and understanding proofs.

As an alternative proof technique we will study natural deduction, an example of a formal deductive proof system designed to incorporate informal but structured and sound reasoning strategies.

Similar to proofs in resolution, proofs in natural deduction (or ND) consist of a sequence of steps, each step labeled either as a premise or by a justification rule.

To distinguish resolution proofs from natural deduction proofs we use

$\Sigma \vdash_{ND} A$ for a proof in natural deduction or

$\Sigma \vdash_{RES} A$ for a proof in resolution.

When clear from context, we just write $\Sigma^t \vdash A$.

Intuition for ND proofs can be seen in the following example of tautological consequence:

$$A \to B, \; B \to C \models A \to C$$

Suppose we know that
if A then B,
and we know that
if B then C.
Then we can reason that
if A then C,
as follows.

Suppose
if A then B
and
if B then C
both hold but
if A then C
does not hold.
Given that if A then C
does not hold then
A is true but C is not.

Since if A then B holds and A is true then B is true.

Since if B then C holds and B is true then C is true, a contradiction.

Therefore
if A then C holds.

Given a set $\Sigma \subseteq \mathrm{Form}(\mathcal{L}^p)$,

if $\Sigma = \{A_1, \ldots, A_n\}$

we may write

$\Sigma \vdash A$

or

$A_1, \ldots, A_n \vdash A$

to denote an ND proof of A from Σ.

Similarly if $\Sigma = \Sigma' \cup \{B\}$
we may write
$\Sigma \models A$
or equivalently

$\Sigma', B \models A$.

This is generalized by
if $\Sigma = \Sigma_1 \cup \Sigma_2$ then
we may write $\Sigma \models A$
or $\Sigma_1, \Sigma_2 \models A$.

Some ND rules (of inference):

(Ref) $A \vdash A$ reflexivity

(+) If $\Sigma \vdash A$ addition of premises

then $\Sigma, \Sigma' \vdash A$

$(\to -)$ If $\Sigma \vdash A \to B$

$\Sigma \vdash A$

then $\Sigma \vdash B$ $\quad$ ($\to$ elimination)

$(\to +)$ If $\Sigma, A \vdash B$

then $\Sigma \vdash A \to B$ $\quad$ ($\to$ introduction)

Example:

Let $\Sigma \subseteq Form(L^p)$ where $A \in \Sigma$ and $\Sigma' = \Sigma \cup \{A\}$.

1. $A \vdash A$ (Ref)
2. $A, \Sigma' \vdash A$ (+), 1.

Since $A, \Sigma' \vdash A$ then this may also be written $\Sigma \vdash A$ since $\Sigma = \Sigma' \cup \{A\}$.

Since this last example happens frequently in ND proofs it is encoded as the rule $(\in)$, that is

$(\in)$ if $A \in \Sigma$ then $\Sigma \vdash A$

Example:

1. $A \to B, B \to C, A \vdash A$ $(\in)$
2. $A \to B, B \to C, A \vdash A \to B$ $(\in)$
3. $A \to B, B \to C, A \vdash B$ $(\to -)$ 1,2
4. $A \to B, B \to C, A \vdash B \to C$ $(\in)$
5. $A \to B, B \to C, A \vdash C$ $(\to -)$ 3,4
6. $A \to B, B \to C \vdash A \to C$ $(\to +)$ 5

So the above steps show:

$$A \to B, B \to C \vdash A \to C$$

Notice that the rule (Ref) is applicable without reference to any preceding rule applications.

Rules $(\to-)$ and $(\to+)$ depend on previously established facts.

The rules for $\wedge$, $\vee$, $\leftrightarrow$ and $\neg$ also depend on previosly established facts.

Notice that in the rule for $(\rightarrow +)$

If $\Sigma, A \vdash B$

then $\Sigma \vdash A \rightarrow B$

Σ is any set of formulae in Form(LP) and A, B are two arbitrary formulae in Form(LP).

So the rules are purely syntactic, as will be the case with additional rules.

This means that the rule applications could be mechanically checked.

ND Rules

$(\neg -)$ If $\Sigma, \neg A \vdash B$

$\Sigma, \neg A \vdash \neg B$

then $\Sigma \vdash A$.

($\neg$ elimination)

Notice that $(\neg -)$ is designed to express proof by contradiction.

$(\wedge -)$ If $\Sigma \vdash A \wedge B$

then $\Sigma \vdash A$

$\Sigma \vdash B$ ($\wedge$-elimination)

$(\wedge +)$ If $\Sigma \vdash A$

$\Sigma \vdash B$

then $\Sigma \vdash A \wedge B$

$(\vee-)$ If $\Sigma, A \vdash C$

$\Sigma, B \vdash C$

then $\Sigma, A \vee B \vdash C$ (∨- elimination)

$(\vee+)$ If $\Sigma \vdash A$

then $\Sigma \vdash A \vee B$

$\Sigma \vdash B \vee A$ (∨-introduction)

$(\leftrightarrow -)$ If $\Sigma \vdash A \leftrightarrow B$

$\Sigma \vdash A$

then $\Sigma \vdash B$

If $\Sigma \vdash A \leftrightarrow B$

$\Sigma \vdash B$

then $\Sigma \vdash A$ $(\leftrightarrow$ elimination$)$

$(\leftrightarrow +)$ If $\Sigma, A \vdash B$

$\Sigma, B \vdash A$

then $\Sigma \vdash A \leftrightarrow B$ $(\leftrightarrow)$ introduction$)$

Proofs using the $(\vee-)$ are a form of case analysis.

If C follows from A, and

If C follows from B, then

C follows from A or B.

Def. (Formal deducibility)

A is deducible from Σ, written $\Sigma \vdash A$ if $\Sigma \vdash A$ is generated by a finite number of applications of the rules of formal deduction.

So $\Sigma \vdash A$ holds if we can establish

$\Sigma_1 \vdash A_1$ and $\Sigma_2 \vdash A_2, \ldots,$

and $\Sigma_n \vdash A_n$

where each of the

$\Sigma_i \vdash A_i$

for $i \in [1..n]$ is generated by one of the previously given ND rules (based on the $\Sigma_j \vdash A_j$ for $1 \leq j < i$) and $\Sigma_n \vdash A_n$ is $\Sigma \vdash A$.

For example, if $\Sigma_k \vdash A_k$ is generated from

$$\Sigma_1 \vdash A_1, \ldots, \Sigma_{k-1} \vdash A_{k-1}$$

by the application of $(\neg -)$ then there must exist

$$\Sigma_i \vdash A_i \text{ and } \Sigma_j \vdash A_j$$

where $A_j = \neg A_i$, $\Sigma_i = \Sigma_i' \cup \{\neg A_k\}$ and $\Sigma_j = \Sigma_j' \cup \{\neg A_k\}$, $\Sigma_i' = \Sigma_j'$ and $\Sigma_k = \Sigma_i'$. for $i, j \in [1..k-1]$.

We write $\Sigma \nvdash A$ as a short hand for it is not the case that $\Sigma \vdash A$.

Let $\Sigma' \subseteq Form(L^p)$.

Then $\Sigma \vdash \Sigma'$ is a notation for:

for all $A \in \Sigma'$ it is the case that $\Sigma \vdash A$.

Fact: If $\Sigma \vdash \Sigma'$
and $\Sigma' \vdash A$
then $\Sigma \vdash A$.

Proof idea:

1. $\Sigma' \vdash A$ premise
2. $A_1, \ldots, A_n \vdash A$ If Σ' is finite then $\Sigma' = \{A_1, \ldots, A_n\}$
3. $A_1, \ldots, A_{n-1} \vdash A_n \to A$ $(\to +)$ 2
4. $\vdash A_1 \to (\ldots (A_n \to A))$ $(\to +)$ 3
5. $\Sigma \vdash A_1 \to (\ldots (A_n \to A))$ $(+)$ 4
6. $\Sigma \vdash A_1$ premise, $A_1 \in \Sigma'$

7. $\Sigma \vdash A_2 \to (\ldots (A_n \to A))$ $(\to -)$ 5, 6

8. $\Sigma \vdash A_2$ premise, $A_2 \in \Sigma'$

9. $\Sigma \vdash A_3 \to (\ldots (A_n \to A))$ $(\to -)$ 8, 9

10. $\Sigma \vdash A_n \to A$ similar to 9

11. $\Sigma \vdash A_n$ premise, $A_n \in \Sigma'$

12. $\Sigma \vdash A$ $(\to -)$ 10, 11

This is known as the transitivity of deducibility and may be used as a derived rule (Tr) in ND proofs.

For $A, B \in Form(L^p)$ we write

$$A \dashv\vdash B$$

as a shorthand for

$$A \vdash B$$

and

$$B \vdash A.$$