Propositional Logic: Gentzen System, $\mathcal{G}$

CS402, Spring 2017

Shin Yoo

Quiz on Thursday, 6th April: 15 minutes, two questions.

Sequent Calculus in $\mathcal{G}$

In Natural Deduction, each line in the proof consists of exactly one proposition. That is, $A_1, A_2, \ldots, A_n \vdash B$.

In Sequent calculus, each line in the proof consists of zero or more propositions. That is, $A_1, A_2, \ldots, A_n \vdash B_1, B_2, \ldots, B_k$. The standard semantic is, "whenever every A_i is true, at least one B_j will also be true".

Axioms in $\mathcal{G}$

Definition 1 (3.2, Ben-Ari)

An axiom of $\mathcal{G}$ is a set of literals U containing a complementary pair.

Note that sets in $\mathcal{G}$ are implicitly disjunctive. For example, $\{\neg p, q, p\}$ is an axiom, i.e. $\vdash \neg p, q, p$ in $\mathcal{G}$.

Inference Rules in $\mathcal{G}$

Definition 2 (3.2, Ben-Ari)

There are two types of inference rules, defined with reference to tables below:

- Let $\{\alpha_1, \alpha_2\} \subseteq U_1$ and let $U_1' = U_1 - \{\alpha_1, \alpha_2\}$. Then $U = U_1' \cup \{\alpha\}$ can be inferred.
- Let $\{\beta_1\} \subseteq U_1, \{\beta_2\} \subseteq U_2$ and let $U_1' = U_1 - \{\beta_1\}, U_2' = U_2 - \{\beta_2\}$. Then $U = U_1' \cup U_2' \cup \{\beta\}$ can be inferred.

Inference Rules in $\mathcal{G}$

$$\frac{\vdash U_1' \cup \{\alpha_1, \alpha_2\}}{\vdash U_1' \cup \{\alpha\}} \; \alpha$$

$$\frac{\vdash U_1' \cup \{\beta_1\} \quad \vdash U_2' \cup \{\beta_2\}}{\vdash U_1' \cup U_2' \cup \{\beta\}} \; \beta$$

α	α_1	α_2
$\neg\neg\alpha$	α	
$\neg(\alpha_1 \wedge \alpha_2)$	$\neg\alpha_1$	$\neg\alpha_2$
$\alpha_1 \vee \alpha_2$	α_1	α_2
$\alpha_1 \rightarrow \alpha_2$	$\neg\alpha_1$	α_2
$\alpha_1 \uparrow \alpha_2$	$\neg\alpha_1$	$\neg\alpha_2$
$\neg(\alpha_1 \downarrow \alpha_2)$	α_1	α_2
$\neg(\alpha_1 \leftrightarrow \alpha_2)$	$\neg(\alpha_1 \rightarrow \alpha_2)$	$\neg(\alpha_2 \rightarrow \alpha_1)$
$\alpha_1 \oplus \alpha_2$	$\neg(\alpha_1 \rightarrow \alpha_2)$	$\neg(\alpha_2 \rightarrow \alpha_1)$

That is, $\alpha-$rules build up disjunctions.

β	β_1	β_2
$\beta_1 \wedge \beta_2$	β_1	β_2
$\neg(\beta_1 \vee \beta_2)$	$\neg\beta_1$	$\neg\beta_2$
$\neg(\beta_1 \rightarrow \beta_2)$	β_1	$\neg\beta_2$
$\neg(\beta_1 \uparrow \beta_2)$	β_1	β_2
$\beta_1 \downarrow \beta_2$	$\neg\beta_1$	$\neg\beta_2$
$\beta_1 \leftrightarrow \beta_2$	$\beta_1 \rightarrow \beta_2$	$\beta_2 \rightarrow \beta_1$
$\neg(\beta_1 \oplus \beta_2)$	$\beta_1 \rightarrow \beta_2$	$\beta_2 \rightarrow \beta_1$

That is, $\beta-$rules build up conjuntions (consider $(a \vee b) \wedge (c \vee d) \models a \vee c \vee (b \wedge d)$).

Example Proof

Prove that $\vdash p \vee (q \wedge r) \rightarrow (p \vee q) \wedge (p \vee r)$ in $\mathcal{G}$.

1.	$\vdash \neg p, p, q$	Axiom
2.	$\vdash \neg p, (p \vee q)$	$\alpha\vee$, 1
3.	$\vdash \neg p, p, r$	Axiom
4.	$\vdash \neg p, (p \vee r)$	$\alpha\vee$, 3
5.	$\vdash \neg p, (p \vee q) \wedge (p \vee r)$	$\beta\wedge$, 2, 4
6.	$\vdash \neg q, \neg r, p, q$	Axiom
7.	$\vdash \neg q, \neg r, (p \vee q)$	$\alpha\vee$, 6
8.	$\vdash \neg q, \neg r, p, r$	Axiom
9.	$\vdash \neg q, \neg r, (p \vee r)$	$\alpha\vee$, 8
10.	$\vdash \neg q, \neg r, (p \vee q) \wedge (p \vee r)$	$\beta\wedge$, 7, 9
11.	$\vdash \neg(q \wedge r), (p \vee q) \wedge (p \vee r)$	$\alpha\wedge$, 10
12.	$\vdash \neg(p \vee (q \wedge r)), (p \vee q) \wedge (p \vee r)$	$\beta\vee$, 5, 11
13.	$\vdash p \vee (q \wedge r) \rightarrow (p \vee q) \wedge (p \vee r)$	$\alpha \rightarrow$, 12

Wait...

- How do you magically come up with the axioms $\{\neg p, p, q\}$, $\{\neg p, p, r\}$, $\{\neg q, \neg r, p, q\}$, and $\{\neg q, \neg r, p, r\}$?
- Haven't we seen something like this before?

$$\vdash (p \vee q) \rightarrow (q \vee p)$$

Proof in $\mathcal{G}$

$$\begin{array}{c}
\neg p, q, p \qquad \neg q, q, p \\
\neg(p \vee q), q, p \\
\neg(p \vee q), (q \vee p) \\
(p \vee q) \rightarrow (q \vee p)
\end{array}$$

$$\begin{array}{c}
\neg((p \vee q) \rightarrow (q \vee p)) \\
(p \vee q), \neg(q \vee p) \\
(p \vee q), \neg q, \neg p \\
p, \neg q, \neg p \qquad q, \neg q, \neg p \\
UNSAT \qquad UNSAT
\end{array}$$

Semantic Tableau
(Sets are conjunctive)

$\mathcal{G}$ and Semantic Tableau

Theorem 1 (3.6, Ben-Ari)

Let A be a formula in propositional logic. Then $\vdash A$ in $\mathcal{G}$ if and only if there is a closed semantic tableaux for $\neg A$.

Theorem 2 (3.7, Ben-Ari)

Let U be a set of formulas and let $\bar{U}$ be the set of complements of formulas in U. Then, $\vdash U$ in $\mathcal{G}$ if and only if there is a closed semantic tableau for $\bar{U}$.

$\mathcal{G}$ and Semantic Tableau

We prove that, if there exists a closed semantic tableau for $\bar{U}$, then $\vdash U$ in $\mathcal{G}$. The opposite direction is left for you.

Proof.

Let $\mathcal{T}$ be a closed semantic tableau for $\bar{U}$. We prove $\vdash U$ by induction on h, the height of $\mathcal{T}$.

- If $h = 0$, then $\mathcal{T}$ consists of a single node labeled by $\bar{U}$. By assumption, $\mathcal{T}$ is closed, so it contains a complementary pair of literals $\{p, \neg p\}$, that is, $\bar{U} = \bar{U}' \cup \{p, \neg p\}$. Obviously, $U = U' \cup \{\neg p, p\}$ is an axiom in $\mathcal{G}$, hence $\vdash U$.

$\mathcal{G}$ and Semantic Tableau

Proof. Cont.

- If $h > 0$, then some tableau rule was used on an α- or β-formula at the root of $\mathcal{T}$ on a formula $\bar{\phi} \in \bar{U}$, that is, $\bar{U} = \bar{U}' \cup \bar{\phi}$. The proof proceeds by cases, where you must be careful to distinguish between applications of the tableau rules and applications of the Gentzen rules of the same name.
  - Case 1: ϕ is an α-formula (such as) $\neg(A_1 \vee A_2)$. The tableau rule created a child node labeled by the set of formulas $\bar{U}' \cup \{\neg A_1, \neg A_2\}$. By assumption, the subtree rooted at this node is a closed tableau, so by the inductive hypothesis, $\vdash U' \cup \{A_1, A_2\}$. Using the appropriate rule of inference from $\mathcal{G}$, we obtain $\vdash U' \cup \{A_1 \vee A_2\}$, that is, $\vdash U' \cup \{\phi\}$, which is $\vdash U$.

Proof.

- If $h > 0$, then some tableau rule was used on an α- or β-formula at the root of $\mathcal{T}$ on a formula $\bar{\phi} \in \bar{U}$, that is, $\bar{U} = \bar{U}' \cup \bar{\phi}$. The proof proceeds by cases, where you must be careful to distinguish between applications of the tableau rules and applications of the Gentzen rules of the same name.
    - Case 2: ϕ is a β-formula (such as) $\neg(B_1 \wedge B_2)$. The tableau rule created two child nodes labeled by the sets of formulas $\bar{U}' \cup \{\neg B_1\}$ and $\bar{U}' \cup \{\neg B_2\}$. By assumption, the subtrees rooted at this node are closed, so by the inductive hypothesis $\vdash U' \cup \{B_1\}$ and $\vdash U' \cup \{B_2\}$. Using the appropriate rule of inference from $\mathcal{G}$, we obtain $\vdash U' \cup \{B_1 \wedge B_2\}$, that is, $\vdash U' \cup \{\phi\}$, which is $\vdash U$.

$\square$

Why $\mathcal{G}$ and not natural deduction?

Taste. Or, more appropriately, aesthetics.

Natural deduction feels more, umm, natural. It is also more simplistic; having multiple disjunct on the right hand side, in $\mathcal{G}$, is clearly cumbersome and adds complexity.

$\mathcal{G}$ shows the symmetric nature of negation more vividly.

$$A_1, \ldots, A_n \vdash B_1, \ldots, B_k$$

$$\vdash (A_1 \wedge \cdots \wedge A_n) \rightarrow (B_1 \vee \cdots \vee B_k)$$

$$\vdash \neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_n \vee B_1 \vee B_2 \vee \cdots \vee B_k$$

$$\vdash \neg(A_1 \wedge A_2 \wedge \cdots \wedge A_n \wedge \neg B_1 \wedge \neg B_2 \wedge \cdots \wedge \neg B_k)$$

Soundness and Completeness of $\mathcal{G}$

Theorem 3 (3.8 in Ben-Ari)

$\models A$ if and only if $\vdash A$ in $\mathcal{G}$.

Proof.

A is valid iff $\neg A$ is unsatisfiable iff there is a closed semanti tableau for $\neg A$ iff there is a proof of A in $\mathcal{G}$. $\square$

Exercises

Prove the following in $\mathcal{G}$:

- $\vdash (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$
- $\vdash (A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B)$
- $\vdash ((A \rightarrow B) \rightarrow A) \rightarrow A$