Sequent Calculus

Robert Keller

February 2013

Sequent Calculus (SC)

- The **sequent calculus** was used by Gentzen (he called it "System S") to derive results about Natural Deduction.
- SC allows **sets** of formulas on **both** sides of |–.
- In the SC, |– becomes an **object-language** symbol, rather than a meta-language symbol as we have been using it. However, some authors use ⇒ or → instead of |– for SC.
- It is easier to **implement in software** a proof generator based on SC than it is one based on ND.

Meaning of $\vdash$ in SC

- The ***intuitive*** meaning of

  $A_1, \ldots, A_m \vdash B_1, \ldots, B_n$

  "Using hypotheses in the set $\{A_1, \ldots, A_m\}$ one can derive **at least one** member of $\{B_1, \ldots, B_n\}$".

  So the left side is like **and** and the right side is like **or**.

- If n = 1, we have the kind of sequent seen before.

- If n = 0, it's as if the right-hand side is $\bot$.

Sequent Calculus Rules

- Rather than have a sequence of formulas as in ND, all relevant formulas are represented in the sequent.
- A proof progresses from one sequent to another.
- It is easiest to start with a goal sequent and **work backward ("upward")**, even though the rules are cast as if we were working downward.
- In some cases, more than one sequent is required to prove a lower sequent.

Convention

- In what follows, ___ and … stand for ***sets* of formulas**.
- Often these sets are shown as Γ and Δ, but our notation seems less cluttered.
- Also ___, F, G
means ___ $\cup$ {F, G}.

L and R Rules

- For each connective, there is a rule for introducing the connective on the **left** and one for introducing it on the **right**.
- These correspond to **elimination** and **introduction** rules, respectively, in natural deduction.

SC Rules for ∧

- Recall the natural deduction (ND) rules ∧E:

$$\frac{F \land G}{F} \qquad \frac{F \land G}{G}$$

- In SC, the ∧E rule corresponds to ∧L (L for "left"):

$$\frac{\_\_\_, F, G \vdash \ldots}{\_\_\_, F \land G \vdash \ldots} \quad \land L$$

Interpretation of ∧L

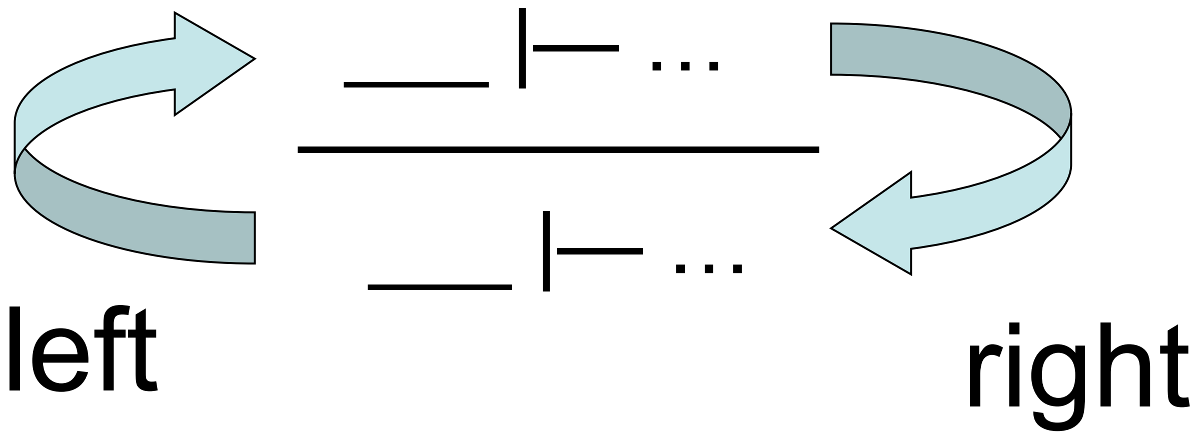
$$\frac{\_\_\_,\ F,\ G \vdash \ldots}{\_\_\_,\ F \land G \vdash \ldots}\ \land L$$

Meaning: **If** we can deduce … from ___, F, G

then we can also deduce … from ___, F ∧ G.

This is indeed what ∧E in ND tells us.

Informal: “Information Flow” in an SC Proof



assuming the SC proof is constructed bottom to top

Information Flow in ∧L

∧L

From F ∧ G we can derive both F and G (in ND).

From ___, F, G we can derive … .

Rules for ∧R

- Recall the natural deduction (ND) rule ∧I:

$$\frac{F \qquad G}{F \wedge G}$$

In SC, the ∧I rule corresponds to

$$\frac{\_\_\_ \vdash F, \ldots \qquad\qquad \_\_\_ \vdash G, \ldots}{\_\_\_ \vdash F \wedge G, \ldots} \qquad \begin{array}{l}\text{(2 sequents)}\\ \wedge R\end{array}$$

Here there two sequents above the line, and **each** must be proved. The information flow combines F and G. Bottom up, one sequent "**splits**" into two.

∧ Rule Summary

$$\frac{\_\_\_, A, B \vdash \ldots}{\_\_\_, A \wedge B \vdash \ldots} \quad \wedge\mathbf{L}$$

"To ***use*** A ∧ B, you can use both A and B."

$$\frac{\_\_\_ \vdash A, \ldots \qquad \_\_\_ \vdash B, \ldots}{\_\_\_ \vdash A \wedge B, \ldots} \quad \wedge\mathbf{R}$$

"To ***derive*** A ∧ B, derive each of A and B separately."

SC "Axiom"

- In SC, there is one axiom
  (rule with no antecedent):

$$\frac{}{\_\_\_,\ P \vdash P,\ \ldots}\quad \text{Axiom}$$

- Here ___ and … represent sets of formulas as usual.

- The meaning in this case is that any formula can be derived from itself.

- A full derivation tree will have an **axiom at each leaf**.

∨ Rule Summary

$$\frac{\_\_\_ \vdash A, B, \ldots}{\_\_\_ \vdash A \lor B, \ldots} \ \lor\mathbf{R}$$

"To ***derive*** A ∨ B, it suffices to derive either."

(like ∨ Introduction in ND)

$$\frac{\_\_\_, A \vdash \ldots \qquad \_\_\_, B \vdash \ldots}{\_\_\_, A \lor B \vdash \ldots} \ \lor\mathbf{L}$$

"To ***use*** A ∨ B, we need separate derivations **using** A and B."

(similar to ∨ Elmination in ND, but cleaner).

→ Rules

$$\frac{\_\_\_,\ A \vdash B,\ \ldots}{\_\_\_ \vdash A \to B,\ \ldots} \quad {\to}R$$

"To **derive** A → B, it suffices to derive B with A as an added assumption." (similar to → Introduction in ND).

$$\frac{\_\_\_ \vdash A,\ \ldots \qquad \_\_\_,\ B \vdash \ldots}{\_\_\_,\ A \to B \vdash \ldots} \quad {\to}L$$

"If A can be proved, and we can prove … based on B as an added assumption, then … can be proved from A → B."

(similar to → Elimination in ND).

¬ Rules

___ ,A |– …

________________ ¬R

___ |– ¬A, …

"To **derive** ¬A, … from ___, it suffices to **use** ___, A to get … ." (similar to ¬ Introduction in ND)

___ |– A, …

_________________ ¬L

___ ,¬A |– …

"To **use** ___ ,¬A to derive … it suffices to derive A, … from ___."

(similar to RAA in ND)

A way to remember → Rules

- Recall that F→G is like ¬F∨G.
- So the effect of the → can be achieved (bottom up) by using the appropriate ∨ rule followed by a ¬rule.

- In particular, →L splits as does ∨L. But once the split is done, the F in ¬F can be "flipped" to the other side.

SC Rule Summary

	Introduce on Left	Introduce on Right
∧	___, A, B |– … ―――――――― ___, A ∧ B |– …	___ |– A, … ___ |– B, … ―――――――――――――― ___ |– A ∧ B, …
∨	___, A |– … ___, B |– … ―――――――――――――― ___, A ∨ B |– …	___ |– A, B, … ―――――――― ___ |– A ∨ B, …
→	___ |– A, … ___, B |– … ―――――――――――――― ___, A → B |– …	___, A |– B, … ―――――――― ___ |– A → B, …
¬	___ |– A, … ―――――――― ___ ,¬A |– …	___ ,A |– … ―――――――― ___ |– ¬A, …

Assuming "introduce" means from top down, even though we are typically working bottom up.

Other Rules not dependent on connectives

$$\frac{\_\_\_ \vdash \ldots}{\_\_\_, A \vdash \ldots}$$

Expansion

"Formulas can be added arbitrarily on the left." (going downward)

$$\frac{\_\_\_ \vdash \ldots}{\_\_\_ \vdash A, \ldots}$$

Contraction

"Formulas can be added arbitrarily on the right."(going downward)

Sequent Calculus Strategy Summary

Working **backward/upward** from the desired sequent:

Situation	Action
A on left and right	Axiom.
¬A in a right formula	Flip A to the left.
¬A in a left formula	Flip A to the right.
A∧B in a right formula	**Split** right set into A and B versions.
A∧B in a left formula	Replace the formula with A, B.
A∨B in a right formula	Replace the formula with A, B.
A∨B in a left formula	**Split** left set into A and B versions.
A→B in a right formula	Replace with A on the left, B on the right.
A→B in a left formula	**Split** into two versions with A on the right in one, B on the left in the other.

Example Sequent Calculus Proof

Constructed from bottom to top:

$P \vee Q, \neg P \vdash Q$

Example Sequent Calculus Proof

Constructed from bottom to top:

$$\frac{P \lor Q \ \vdash P, Q}{P \lor Q, \neg P \vdash Q} \quad \neg L \text{ rule}$$

Example Sequent Calculus Proof

Constructed from bottom to top:

$$\dfrac{\dfrac{P \vdash P, Q \qquad Q, \vdash P, Q}{P \vee Q \vdash P, Q}}{P \vee Q, \neg P \vdash Q}$$

vL rule

¬L rule

Example Sequent Calculus Proof

Constructed from bottom to top:

Axiom — Axiom

P |– P, Q — Q, |– P, Q — vL rule

P v Q |– P, Q — ¬L rule

P v Q, ¬P |–Q

Corresponding Natural Deduction Tree:

$[P]_1$ ¬P

⊥ — ¬E

P v Q — Q — $[Q]_2$ — ⊥E

Q — $vE_{1,2}$

Rough Correspondence: SC vs. ND

- The last sequent of an SC derivation will always correspond to the overall sequent derived in ND.
- Other sequents may correspond to subproofs.
- Moving from **bottom to top** in an SC proof is like working **outside-in** in a ND proof.
- Consider LHS of a sequent to be *all* operative hypotheses, including assumptions in sub-proofs.
- Consider RHS to be goal (as a disjunction).

Conversion Between SC and ND

- http://twelf.plparty.org/wiki/POPL_Tutorial/Sequent_vs_Natural_Deduction
- http://www.ags.uni-sb.de/~chris/papers/2002-pisa.pdf

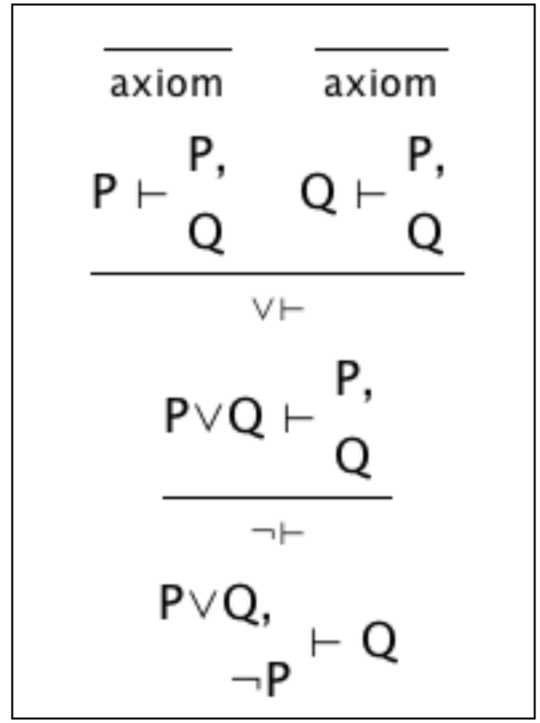
Compare JAPE SC Version (MCS)

________ Axiom ________ Axiom

P |–P, Q Q, |– P, Q v |– rule

P v Q |– P, Q ¬ |– rule

P v Q, ¬P |–Q



MCS = Multi-Conclusion (classical)
SCS = Single Conclusion (intuitionistic)

More Sequent Calculus Examples
(constructed working backward/upward)

$$\vdash ((P \rightarrow Q) \wedge (Q \rightarrow R)) \rightarrow (P \rightarrow R)$$

More Sequent Calculus Examples

$$\frac{(P \rightarrow Q) \wedge (Q \rightarrow R) \quad \vdash \quad (P \rightarrow R)}{\vdash ((P \rightarrow Q) \wedge (Q \rightarrow R)) \rightarrow (P \rightarrow R)} \quad \vdash \rightarrow$$

More Sequent Calculus Examples

$\underline{(P \rightarrow Q), (Q \rightarrow R) \;\vdash\; (P \rightarrow R)}$ $\quad \wedge \vdash$

$\underline{(P \rightarrow Q) \wedge (Q \rightarrow R) \;\vdash\; (P \rightarrow R)}$ $\quad \vdash \rightarrow$

$\vdash ((P \rightarrow Q) \wedge (Q \rightarrow R)) \rightarrow (P \rightarrow R)$

More Sequent Calculus Examples

$\underline{(P \to Q), (Q \to R), P \;\vdash\; R}$	$\vdash \to$
$\underline{(P \to Q), (Q \to R) \;\vdash\; (P \to R)}$	$\wedge \vdash$
$\underline{(P \to Q) \wedge (Q \to R) \;\vdash\; (P \to R)}$	$\vdash \to$
$\vdash ((P \to Q) \wedge (Q \to R)) \to (P \to R)$	

More Sequent Calculus Examples

$$\frac{\dfrac{\dfrac{\dfrac{(P \to Q), R, P \vdash R \qquad (P \to Q), P \vdash Q, R}{(P \to Q), (Q \to R), P \vdash R}\ \to\vdash}{(P \to Q), (Q \to R) \vdash (P \to R)}\ \vdash\to}{(P \to Q) \wedge (Q \to R) \vdash (P \to R)}\ \wedge\vdash}{\vdash ((P \to Q) \wedge (Q \to R)) \to (P \to R)}\ \vdash\to$$

More Sequent Calculus Examples

$$\frac{}{(P \to Q), R, P \vdash R}\,\text{Ax} \qquad \frac{P \vdash Q, R, P \qquad Q, P \vdash Q, R}{(P \to Q), P \vdash Q, R}\,\to\vdash$$

$$\frac{(P \to Q), R, P \vdash R \qquad (P \to Q), P \vdash Q, R}{(P \to Q), (Q \to R), P \vdash R}\,\to\vdash$$

$$\frac{(P \to Q), (Q \to R), P \vdash R}{(P \to Q), (Q \to R) \vdash (P \to R)}\,\vdash\to$$

$$\frac{(P \to Q), (Q \to R) \vdash (P \to R)}{(P \to Q) \wedge (Q \to R) \vdash (P \to R)}\,\wedge\vdash$$

$$\frac{(P \to Q) \wedge (Q \to R) \vdash (P \to R)}{\vdash ((P \to Q) \wedge (Q \to R)) \to (P \to R)}\,\vdash\to$$

More Sequent Calculus Examples

$$\dfrac{\dfrac{}{(P \to Q), R, P \vdash R}\ \text{Ax} \qquad \dfrac{\dfrac{}{P \vdash Q, R, P}\ \text{Ax} \qquad \dfrac{}{Q, P \vdash Q, R}\ \text{Ax}}{(P \to Q), P \vdash Q, R}\ {\to}\vdash}{\dfrac{(P \to Q), (Q \to R), P \vdash R}{\dfrac{(P \to Q), (Q \to R) \vdash (P \to R)}{\dfrac{(P \to Q) \wedge (Q \to R) \vdash (P \to R)}{\vdash ((P \to Q) \wedge (Q \to R)) \to (P \to R)}\ \vdash{\to}}\ \wedge\vdash}\ \vdash{\to}}\ {\to}\vdash$$

An SC Prover in Prolog

```
% If L |- R, where L and R are lists of formula,
% prove(L, R, Proof) will produce a sequent calculus proof tree.
% Using test(L, R) will display the proof in line-numbered fashion.
% If the sequent is not provable, the prove predicate will fail.

prove(L, R, axiom(L, R)) :- member(F, L), member(F, R).

prove(L, R, notLeft(L, R, Proof)) :-
            select(not(A), L, Rest),
            prove(Rest, [A | R], Proof).

prove(L, R, notRight(L, R, Proof)) :-
            select(not(A), R, Rest),
            prove([A | L], Rest, Proof).

prove(L, R, andLeft(L, R, Proof)) :-
               select(and(A, B), L, Rest),
            prove([A, B | Rest], R, Proof).

prove(L, R, orRight(L, R, Proof)) :-
            select(or(A, B), R, Rest),
            prove(L, [A, B | Rest], Proof).

prove(L, R, impliesRight(L, R, Proof)) :-
            select(implies(A, B), R, Rest),
            prove([A | L], [B | Rest], Proof).

prove(L, R, andRight(L, R, ProofL, ProofR)) :-
               select(and(A, B), R, Rest),
            prove(L, [A | Rest], ProofL),
            prove(L, [B | Rest], ProofR).

prove(L, R, orLeft(L, R, ProofL, ProofR)) :-
               select(or(A, B), L, Rest),
            prove([A | Rest], R, ProofL),
            prove([B | Rest], R, ProofR).

prove(L, R, impliesLeft(L, R, ProofL, ProofR)) :-
               select(implies(A, B), L, Rest),
            prove(Rest, [A | R], ProofL),
            prove([B | Rest], R, ProofR).
```

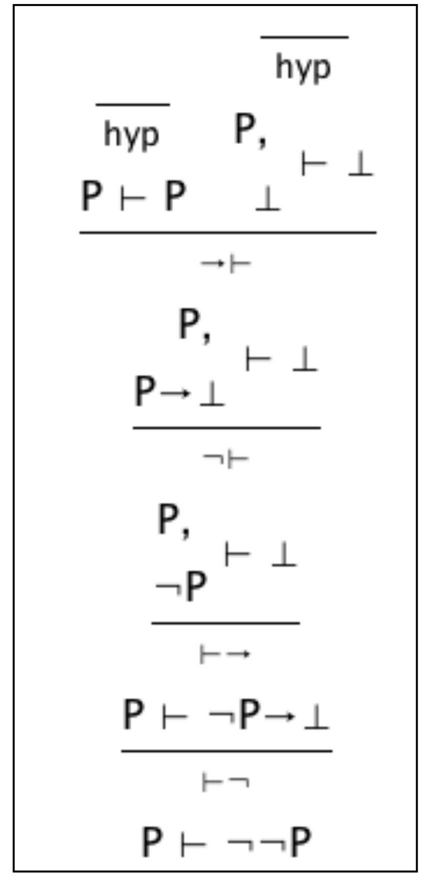
Formatted Output of the Prolog SC Prover

```
/* Example output from test([not(or(a, b))], [and(not(a), not(b))]).
 *
 *   Proof of sequent or(not(a),not(b)) |- not(and(a,b)):
 *   1: a, b |- a                              -- Axiom
 *   2: not(a), a, b |-                        -- not L, line 1
 *   3: a, b |- b                              -- Axiom
 *   4: not(b), a, b |-                        -- not L, line 3
 *   5: a, b, or(not(a),not(b)) |-             -- or L, lines 2 and 4
 *   6: and(a,b), or(not(a),not(b)) |-         -- and L, line 5
 *   7: or(not(a),not(b)) |- not(and(a,b))  -- not R, line 6
 */
```

We use line numbers to display the tree linearly.

JAPE SCS vs. MCS

- SCS: Single-conclusion:
  - Will only prove intuitionistic sequents.
  - The RHS is always a single formula.
  - For some reason
    $\neg\varphi$ converts to $\varphi\to\bot$ first.



This is ¬¬ Introduction.
¬¬ Elimination cannot be proved intuitionistically.

SC vs. Tableaux

- Tableaux proofs were introduced previously.
- A correspondence can be established between an SC proof and a **block tableau** proof T.
- Tableau is constructed top-down; SC is bottom-up.
- The conclusion of SC is negated in T.
- The premises of SC, if any, are unnegated in T.
- There is a correspondence between rules of the two systems.
- Splitting in SC is like splitting in the block tableau.
- Axioms of SC correspond to path closure.

Sequent Calculus vs. Tableaux

Sequent Calculus	**Tableau**
Constructed bottom-up.	Constructed top-down.
Proves $A_1, \ldots, A_m \mid\!- B_1, \ldots, B_n$ Equivalent to proving $(\neg A_1 \vee \ldots \vee \neg A_m \vee B_1 \vee \ldots \vee B_n)$	Negated conclusion (typically m = 0, n = 1) $\neg(\neg A_1 \vee \ldots \vee \neg A_m \vee B_1 \vee \ldots \vee B_n)$ $\equiv A_1 \wedge \ldots \wedge A_m \wedge \neg B_1 \wedge \ldots \wedge \neg B_n$
Formulas on the right	Originally negated formulas
Formulas on the left	Originally un-negated formulas
Axiom …, p, … |- __, p, __	Closure p, ¬p

Tableau vs. SC Example

negated formulas in tableau go on the right of |–

unnegated formulas in tableau go on the right of |–

Block Tableau
{ ¬((p → q) → (¬q → ¬p)) }

SC "upside-down"
|– (p → q) → (¬q → ¬p)

Tableau vs. SC Example

Block Tableau
$\{ \neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)) \}$
$\{ (p \rightarrow q), \neg(\neg q \rightarrow \neg p) \}$

SC “upside-down”
$\vdash (p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$
$(p \rightarrow q) \vdash (\neg q \rightarrow \neg p)$

Tableau vs. SC Example

Block Tableau
{ ¬((p → q) → (¬q → ¬p)) }
{ (p → q), ¬(¬q → ¬p) }
{ (p → q), ¬q, ¬¬p }

SC "upside-down"
⊢ (p → q) → (¬q → ¬p)
(p → q) ⊢ (¬q → ¬p)
¬q, (p → q) ⊢ ¬p

Tableau vs. SC Example

Block Tableau	**SC “upside-down”**
$\{ \neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)) \}$	$\vdash (p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$
$\{ (p \rightarrow q), \neg(\neg q \rightarrow \neg p) \}$	$(p \rightarrow q) \vdash (\neg q \rightarrow \neg p)$
$\{ (p \rightarrow q), \neg q, \neg\neg p \}$	$\neg q, (p \rightarrow q) \vdash \neg p$
$\{ (p \rightarrow q), \neg q, p \}$	$\neg q, (p \rightarrow q), p \vdash$

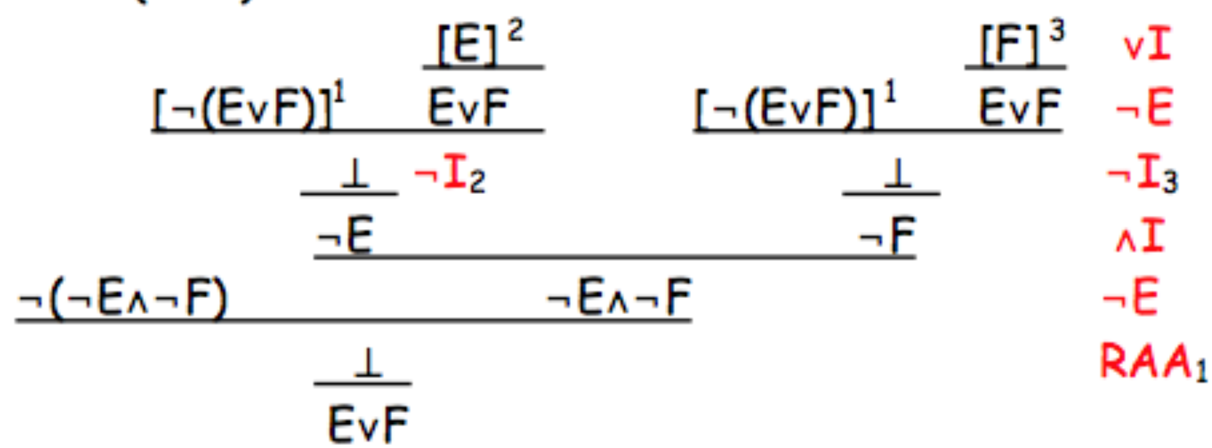
Tableau vs. SC Example

Block Tableau
$\{ \neg((p \to q) \to (\neg q \to \neg p)) \}$
$\{ (p \to q), \neg(\neg q \to \neg p) \}$
$\{ (p \to q), \neg q, \neg\neg p \}$
$\{ (p \to q), \neg q, p \}$
$\{\neg p, \neg q, p \}$X $\{ q, \neg q, p \}$X

SC “upside-down”	
$\vdash (p \to q) \to (\neg q \to \neg p)$	
$(p \to q) \vdash (\neg q \to \neg p)$	
$\neg q, (p \to q) \vdash \neg p$	
$\neg q, (p \to q), p \vdash$	
$\neg q, q, p \vdash$	$q, p \vdash p$(ax)
$q, p \vdash q$ (ax)	

One Sequent $\neg(\neg E \wedge \neg F) \vdash E \vee F$ Proved Five Ways

Natural Deduction (tree)



One Sequent $\neg(\neg E \wedge \neg F) \vdash E \vee F$ Proved Five Ways

Natural Deduction (Fitch diagram)

1	$\neg(\neg E \wedge \neg F)$	premise
2	$\neg(E \vee F)$	assumption
3	E	assumption
4	$E \vee F$	3, $\vee$I
5	$\bot$	2, 4, $\neg$E
6	$\neg E$	3-5, $\neg$I
7	F	assumption
8	$E \vee F$	7, $\vee$I
9	$\bot$	2, 8, $\neg$E
10	$\neg F$	7-9, $\neg$I
11	$\neg E \wedge \neg F$	6, 10, $\wedge$I
12	$\bot$	1, 11, $\neg$E
13	$E \vee F$	2-12, RAA

One Sequent ¬(¬E∧¬F) |– E∨F Proved Five Ways

Sequent Calculus (read bottom-up)

```
E |- E, F                F |- E, F         |-¬
  |- E, F, ¬E              |- E, F, ¬F     |-∧
            |- E, F, (¬E∧¬F)               ¬|-
¬(¬E∧¬F) |- E, F                           |-∨
¬(¬E∧¬F) |- E∨F
```

Here the correspondence with the Natural Deduction proof is not 1-1.

One Sequent $\neg(\neg E \wedge \neg F) \vdash E \vee F$ Proved Five Ways

Block Tableau

$\{\neg(\neg E \wedge \neg F),\ \neg(E \vee F)\}$

$\{\neg(\neg E \wedge \neg F),\ \neg E, \neg F\}$

$\{\neg\neg E \vee \neg\neg F,\ \neg E, \neg F\}$

$\{\neg\neg E, \neg E, \neg F\}$	$\{\neg\neg F, \neg E, \neg F\}$
$\{E, \neg E, \neg F\}$	$\{F, \neg E, \neg F\}$
X	X

The blocks could close one step earlier, due to $\neg\neg E$ and $\neg E$, etc.

One Sequent $\neg(\neg E \wedge \neg F) \vdash E \vee F$ Proved Five Ways

Tableau (tree form)

1. $\neg(\neg E \wedge \neg F)$	premise √		
2. $\neg(E \vee F)$	negated conclusion √		
3. $\neg E$	(2 stack)		
4. $\neg F$	(2 stack)		
5. $\neg\neg E$ (1 split) √		6. $\neg\neg F$ (1 split) √	
7. E (5 $\neg\neg$)		8. F (5$\neg\neg$)	
X (3, 7)		X (4, 8)	

The paths could close one step earlier, due to $\neg\neg E$ and $\neg E$, etc.

An Automated Sequent Calculus Prover:

Defunct? http://bach.istc.kobe-u.ac.jp/seqprover/

Please use for *checking*, not doing, homework!

Sequent Prover (seqprover)

Sequent: (p->q)/\(q->r) --> p->r [Prove]

Output style: Pretty form

[Top page]

```
Trying to prove with threshold = 0

----------- Ax  ----------- Ax   ----------- Ax  ----------- Ax
p --> q,p,r     p,r --> p,r      p,q --> q,r     p,q,r --> r
--------------------------- L->  --------------------------- L->
      p,q->r --> p,r                   p,q,q->r --> r
      ----------------------------------------------- L->
                     p,p->q,q->r --> r
                     ------------------ R->
                     p->q,q->r --> p->r
                     ----------------------- L/\
                     (p->q)/\(q->r) --> p->r

# Proved in 0 msec.
```

Tableau Prover

http://www.umsu.de/logik/trees/

Please use for *checking*, not doing, homework!