Midterm II Exam

15-317/657 Constructive Logic
André Platzer

April 2, 2020

Name: André Platzer

Andrew ID: aplatzer

Instructions

- **For fairness reasons all answers must be typed on a computer in text editors/text-processing (e.g. LaTeX) and submitted as PDF. Besides PDF viewers, no other software is allowed and no handwritten answers/scans are accepted. You can use scratch paper but not hand it in.**
- **Only** the following resources can be used during this exam:
  1. 15317 lecture and recitation notes
  2. editors or text-processing software
  3. email with course staff or **private** Piazza posts with course staff

  All other communications with anyone about the exam or this course during the exam period constitute an academic integrity violation.
- You have 24 hours from when the exam was available to complete it.
- There are 3 problems on 9 pages.
- **Submit** on GradeScope → Midterm 2 → Submit assignment

	Max	Score
New Connections	70	
Theorem Proving	30	
Prolog Principles	50	
Total:	150	

This is a sample solution*, not a model solution. Problems admit multiple correct answers, and the answer the instructor thought of may not necessarily be the best or most elegant.*

1 New Connections (70 points)

The sequent calculus $\Gamma \Longrightarrow C$ came from natural deduction. Consider the new connective $@(A,B,D)$ that is given meaning by the following natural deduction introduction rule:

$$\dfrac{\dfrac{}{B\ \mathit{true}}\,u \quad \dfrac{}{A\ \mathit{true}}\,w \\ \vdots \\ D\ \mathit{true}}{@(A,B,D)\ \mathit{true}}\ @I^{u,w}$$

This question uses the **flat notation** $A_1, A_2, \ldots, A_n \vdash A$ to indicate that A *true* is provable in the natural deduction calculus from the assumptions A_1 *true* and A_2 *true* and $\ldots A_n$ *true*.

2 **Task 1** Rewrite the $@I^{u,w}$ natural deduction rule in its flat notation $\Gamma \vdash A$.

Solution:

$$\frac{\Gamma, B, A \vdash D}{\Gamma \vdash @(A,B,D)}\ @I$$

5 **Task 2** Give a set of elimination rules that harmoniously fit to $@I$ (you can use flat notation):

Solution:

$$\frac{\Gamma \vdash @(A,B,D) \quad \Gamma \vdash B \quad \Gamma \vdash A}{\Gamma \vdash D}\ @E$$

5 **Task 3** Give corresponding sequent calculus rules for $@(A,B,D)$ (in the original $\Longrightarrow$ calculus):

Solution:

$$\frac{\Gamma, B, A \Longrightarrow D}{\Gamma \Longrightarrow @(A,B,D)}\ @R \qquad \frac{\Gamma, @(A,B,D) \Longrightarrow B \quad \Gamma, @(A,B,D) \Longrightarrow A \quad \Gamma, @(A,B,D), D \Longrightarrow C}{\Gamma, @(A,B,D) \Longrightarrow C}\ @L$$

You can assume without proof the usual theorems to hold after adding the @ connective:

Weaken: If $\Gamma \vdash C$ then $\Gamma, A \vdash C$.

Substitution: If $\Gamma \vdash A$ and $\Gamma, A \vdash C$ then $\Gamma \vdash C$.

Weakening: If $\Gamma \Longrightarrow C$ then $\Gamma, A \Longrightarrow C$.

Identity: $\Gamma, A \Longrightarrow A$.

Cut: If $\Gamma \Longrightarrow A$ and $\Gamma, A \Longrightarrow C$ then $\Gamma \Longrightarrow C$.

$\boxed{20}$ **Task 4** Prove that the sequent calculus is sound w.r.t. natural deduction, i.e., $\Gamma \Longrightarrow A$ implies $\Gamma \vdash A$, for the new cases for $@(A,B,D)$.

Solution: Case $@(A,B,D)$ proved by $@R$ in the succedent:

$\Gamma \Longrightarrow @(A,B,D)$	1 proved by rule $@R$
$\Gamma, B, A \vdash D$	2 by IH on the premise of line 1
$\Gamma \vdash @(A,B,D)$	3 rule $@I$ on line 4

Case $@(A,B,D)$ is used by $@L$ as an antecedent:

$\Gamma, @(A,B,D) \Longrightarrow C$	1 proved by rule $@L$
$\Gamma, @(A,B,D) \vdash B$	2 by IH on premise 1 of line 1
$\Gamma, @(A,B,D) \vdash A$	3 by IH on premise 2 of line 1
$\Gamma, @(A,B,D), D \vdash C$	4 by IH on premise 3 of line 1
$\Gamma, @(A,B,D) \vdash @(A,B,D)$	5 by hyp
$\Gamma, @(A,B,D) \vdash D$	6 by $@E$ from line 5 and premises 2 and 3 of line 1
$\Gamma, @(A,B,D) \vdash C$	7 by substitution from line 6 and weakend line 4

20 **Task 5** Prove that the sequent calculus is complete w.r.t. natural deduction, i.e., $\Gamma \vdash A$ implies $\Gamma \Longrightarrow A$, for the new cases for $@(A,B,D)$.

Solution: Case $@(A,B,D)$ proved by $@I$:

$\Gamma \vdash @(A,B,D)$	1 proved by rule $@I$
$\Gamma, B, A \Longrightarrow D$	2 by IH on premise of line 1
$\Gamma \Longrightarrow @(A,B,D)$	3 by rule $@R$ on line 2

Case $@(A,B,D)$ used by $@E$:

$\Gamma \vdash D$	1 proved by rule $@E$
$\Gamma \Longrightarrow B$	2 by IH on premise 2 of line 1
$\Gamma \Longrightarrow A$	3 by IH on premise 3 of line 1
$\Gamma, D \Longrightarrow D$	4 by identity
$\Gamma, @(A,B,D) \Longrightarrow D$	5 by rule $@L$ on lines 2+3+4 with weakening
$\Gamma \Longrightarrow @(A,B,D)$	6 by IH on premise 1 of line 1
$\Gamma \Longrightarrow D$	7 by cut with D on lines 6+5

18 **Task 6** Prove the case of the cut theorem for sequent calculus showing that $\Gamma \Longrightarrow C$ with principal formula $@(A,B,D)$ in the deductions for $\Gamma \Longrightarrow @(A,B,D)$ and $\Gamma, @(A,B,D) \Longrightarrow C$. Explicitly indicate why the induction hypothesis is applicable in each of its uses.

Solution: Consider the case of proofs $\mathcal{D}$ and $\mathcal{E}$ as follows:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma, B, A \Longrightarrow D\end{array}}{\Gamma \Longrightarrow @(A,B,D)}\,@R \qquad \begin{array}{c}\mathcal{E}_1\\ \Gamma, @(A,B,D) \Longrightarrow B\end{array} \quad \begin{array}{c}\mathcal{E}_2\\ \Gamma, @(A,B,D) \Longrightarrow A\end{array} \quad \begin{array}{c}\mathcal{E}_3\\ \Gamma, @(A,B,D), D \Longrightarrow C\end{array}}{\Gamma, @(A,B,D) \Longrightarrow C}\,@L$$

$\Gamma \Longrightarrow B$	1 by IH on $@(A,B,D)$ and $\mathcal{D}, \mathcal{E}_1 \prec \mathcal{E}$
$\Gamma \Longrightarrow A$	2 by IH on $@(A,B,D)$ and $\mathcal{D}, \mathcal{E}_2 \prec \mathcal{E}$
$\Gamma, D \Longrightarrow C$	3 by IH on $@(A,B,D)$ and $\mathcal{D}, \mathcal{E}_3 \prec \mathcal{E}$
$\Gamma, A \Longrightarrow D$	4 by IH on $B \prec @(A,B,D)$ and line 1 and $\mathcal{D}_1$
$\Gamma \Longrightarrow D$	5 by IH on $A \prec @(A,B,D)$ and line 2+4
$\Gamma \Longrightarrow C$	6 by IH on $D \prec @(A,B,D)$ and line 5+3

2 Theorem Proving (30 points)

Dyckhoff's contraction-free sequent calculus is *sound*, *complete*, and has the *termination* property that all its premises are strictly smaller in a well-founded ordering. Each of the following tasks considers one change to our original contraction-free sequent calculus. Concisely but clearly **explain** whether these properties still hold when replacing only the indicated rule and **mark (s)** for sound, **(u)** for unsound, **(c)** for complete, **(i)** for incomplete, **(t)** for terminating, **(n)** for nonterminating. If they fail, also give an example sequent demonstrating the failure.

To get you started here's a simple example: Replacing $\wedge R$ by $P0$ would make it

$$\frac{\Gamma \longrightarrow A \quad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\ \wedge R \qquad \frac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \wedge B}\ P0$$

(u) because $\longrightarrow \top \wedge \bot$ proves by $P0 + \top R$ but is (constructively) false as it implies $\bot$ by $\wedge L$.
(c) every sequent provable by $\wedge R$ is provable by $P0$, which has a subset of the premises of $\wedge R$.
(t) the same ordering shows termination because $P0$ produces a subset of the premises of $\wedge R$.

$\boxed{10}$ **Task 1** What happens when we only replace rule $\bot{\supset}L$ by rule $P1$:

$$\frac{\Gamma \longrightarrow C}{\Gamma, \bot \supset B \longrightarrow C}\ \bot{\supset}L \qquad \frac{\Gamma, B \longrightarrow C}{\Gamma, \bot \supset B \longrightarrow C}\ P1$$

Solution:
(u) $\bot \supset \bot \longrightarrow \bot$ proves by $P1 + \bot L$ but is false, because it implies $\bot$ since $\bot \supset \bot$ by ${\supset}R + \bot L$.
(c) everything $\bot{\supset}L$ proves proves by $P1$ after weakening B.
(t) B in the premise is a (proper) subformula of $\bot \supset B$ in the conclusion.

$\boxed{10}$ **Task 2** What happens when we only replace rule $\wedge{\supset}L$ by rule $P2$:

$$\frac{\Gamma, A_1 \supset (A_2 \supset B) \longrightarrow C}{\Gamma, (A_1 \wedge A_2) \supset B \longrightarrow C}\ \wedge{\supset}L \qquad \frac{\Gamma, (A_1 \wedge A_2) \supset B \longrightarrow C}{\Gamma, A_1 \supset (A_2 \supset B) \longrightarrow C}\ P2$$

Solution:
(s) $A_1 \supset (A_2 \supset B)$ is propositionally equivalent to $(A_1 \wedge A_2) \supset B$ by (un)currying, which implies the soundness of $P2$.
(i) No rule in the contraction-free calculus now applies to $(\top \wedge \top) \supset P \longrightarrow P$, which is provable in the ordinary sequent calculus by ${\supset}L + \wedge R + \top R$.
(t) The contraction-free well-founded ordering considers $\supset$ smaller than $\wedge$, which no longer works. But considering $\wedge$ smaller than $\supset$ works for $P1$ and all remaining rules.

$\boxed{10}$ **Task 3** What happens when we only replace rule ${\supset}{\supset}L$ by rule $P3$ (no need to give examples):

$$\frac{\Gamma, E \supset B, D \longrightarrow E \quad \Gamma, B \longrightarrow C}{\Gamma, (D \supset E) \supset B \longrightarrow C}\ {\supset}{\supset}L \qquad \frac{\Gamma, D \longrightarrow E \quad \Gamma, B \longrightarrow C}{\Gamma, (D \supset E) \supset B \longrightarrow C}\ P3$$

Solution:
(s) the premises of $P3$ are subsets of the premises of $\supset\supset L$ so follow by weakening.
(i) the premise lacks implication $E \supset B$, which is the only case where an implication needs to be retained for completeness in the contraction-free calculus. Example: $\big(((A \supset C) \supset A) \supset A\big) \supset C \longrightarrow C$ proves by $\supset\supset L$ then $\supset L$ (or $\supset\supset L$ and $P{\supset}L$) and id:

$$\dfrac{\dfrac{\dfrac{}{A \supset C \longrightarrow A \supset C}\,id \quad \dfrac{}{A \longrightarrow A}\,id}{A \supset C, (A \supset C) \supset A \longrightarrow A}\,{\supset}L \quad \dfrac{}{C \longrightarrow C}\,id}{\big(((A \supset C) \supset A) \supset A\big) \supset C \longrightarrow C}\,{\supset\supset}L$$

(t) all premises are subsets of premises of $\supset\supset L$

3 Prolog Principles (50 points)

This question studies ways of computing the derivative of mathematical expressions in one variable with Prolog. Assume that these mathematical expressions are represented as a data structure of type `poly` built in an arbitrary shape from these constructors:

```
add(S,T) represents the sum of S and T
mul(S,T) represents the product of S and T
x        indicates the variable (only one variable occurs so no need for a name)
n(N)     represents the number literal N (as a built-in integer)
```

In this problem you will define a predicate `diff/2` to compute the derivative of an expression represented in this way. For example, the following query is expected to succeed:

```
?- diff(add(x,n(7)), add(n(1), n(0))).
```

Modes describe the intended ways of using a predicate. Mode `+poly` indicates an input argument that needs to be provided satisfying `poly/1`. Mode `-poly` indicates an output argument satisfying `poly/1` that will be computed by the predicate when all inputs are provided.

[2] **Task 1** Write a Prolog program defining the predicate `poly/1` that simply checks whether its argument is built solely from the above constructors in any order. The `integer/1` predicate checks if its argument is an integer.

Solution:

```
poly(x).
poly(n(N)) :- integer(N).
poly(add(S,T)) :- poly(S), poly(T).
poly(mul(S,T)) :- poly(S), poly(T).
```

[18] **Task 2** Write a Prolog program `diff(+poly,-poly)` that takes a `poly` as an input in the first argument and produces its derivative as an output in the second argument.

Solution:

```
diff(x,n(1)).
diff(n(N),n(0)).
diff(add(S,T), add(DS,DT)) :- diff(S,DS), diff(T,DT).
diff(mul(S,T), add(mul(S,DT),mul(DS,T))) :- diff(S,DS), diff(T,DT).
```

10 **Task 3** Prove that your implementation from Task 2 correctly satisfies the mode `diff(+poly,-poly)`, i.e., that, for any term `T` given as input as the first argument that satisfies `poly(T)`, the goal `diff(T,D)` succeeds and computes an output term `D` that satisfies `poly(D)`.

Solution: By induction on the rule. The first two clauses provide a ground term in the last argument, which satisfies poly.

Case `diff(add(S,T), add(DS,DT)) :- diff(S,DS), diff(T,DT).`

poly(add(S,T))	1 by assumption
poly(S), poly(T)	2 by inversion on line 1
poly(DS), poly(DT)	3 by induction hypothesis using on line 2
poly(add(DS,DT))	4 by clause poly for add on line 3

Case `diff(mul(S,T), add(mul(S,DT),mul(DS,T))) :- diff(S,DS), diff(T,DT).`

poly(mul(S,T))	1 by assumption
poly(S), poly(T)	2 by inversion on line 1
poly(DS), poly(DT)	3 by induction hypothesis using on line 2
poly(mul(S,DT))	4 by clause poly for mul on line 3
poly(mul(DS,T))	5 by clause poly for mul on line 3
poly(add(mul(S,DT),mul(DS,T))	6 by clause poly for add on line 4+5

5 **Task 4** Is the mode `diff(+poly,-poly)` for your implementation from Task 2 *total*? That is, does it always succeed at least once when given any `poly` as its first argument and a fresh variable as its second argument? Briefly explain why or why not.

Solution: Yes, because the proof for the mode from Task 3 yields an output for any input term that satisfies poly. All its cases `add/2,mul/2,x/0,n/1` are covered by that proof. Hence, the proof for the mode from Task 3 shows that it succeeds with an output for every input satisfying poly.

5 **Task 5** Is the mode `diff(+poly,-poly)` for your implementation from Task 2 *unique*? It is unique if, for any `poly` given as the first argument, `diff/2` can never succeed with two different results for the second argument. Briefly explain why or why not.

Solution: Yes, because the proof for the mode from Task 3 shows that none of its steps can succeed in more than one way as none of its clauses can apply to the same term and all of its clauses then succeed in only one way.

10 **Task 6** With mode `diff(+poly,-poly)`, the predicate from Task 2 computes a derivative. For arguments satisfying `poly`, carefully explain what exactly all other modes of `diff/2` do. If they do not do anything useful clearly explain why not.

Solution:
mode `diff(+poly,+poly)` checks whether the second output is the unique unsimplified derivative of the first argument. It does not correctly check whether the second argument is the antiderivative/indefinite integral if the first argument because different integration constants are not supported by the implementation of diff.

mode `diff(-poly,+poly)` **may seem to** compute an antiderivative/indefinite integral of the second argument, but it only succeeds on very few terms satisfying poly, namely the ones that directly correspond to an unsimplified derivative. For example, the following query does not succeed, because it does not unify with any shape of the second argument of the clauses of diff:

```
?- diff(N, add(mul(x,n(5)),mul(x,n(2)))).
```

This polynomial example still has an indefinite integral.

mode `diff(-poly,-poly)` would try enumerating some polynomial expressions and their unsimplified derivatives but will neither be exhaustive nor will it terminate. This behavior depends on the nonrecursive clauses coming before the recursive ones.