Quick Review

Get paper and pencil out.
Not graded just for review.

Questions

Write a derivation tree for
\[ \text{pred}(\text{succ}(\text{zero})) \]

Is the rule below derivable, admissible, or neither?
\[ \text{eq}(\text{pred}(\text{succ}(\text{X})), \text{X}) \]

Is the rule below derivable, admissible, or neither?
\[ \text{succ}(\text{pred}(\text{X})) \]

Is the rule below derivable, admissible, or neither?
\[ \text{pred}(\text{succ}(\text{X})) \]
Is the rule below derivable, admissible, or neither?

\[ \text{pred}(\text{succ}(X)) \rightarrow \text{pred}(X) \rightarrow \text{int} \]

Prove the following

If \( X \) nat then \( X \) int

(just sketch out the structure)

What is the principle of rule induction look like for the rules above?

Answers

Write a derivation tree for

\[ \text{succ} (\text{zero}) \rightarrow \text{int} \]

\[ \text{pred} (\text{succ} (\text{zero})) \rightarrow \text{int} \]
Is the rule below derivable, admissible, or neither?
\[
\frac{X: \text{int}}{\text{succ}(X): \text{int}}^R_0
eq (\text{pred}(\text{succ}(X)), X)
\]

The rule below is derivable.
\[
\frac{X: \text{int}}{\text{succ}(X): \text{int}}^R_2
\]

Is the rule below derivable, admissible, or neither?
\[
\frac{X: \text{int}}{\text{pred}(X): \text{int}}^R_0
eq (\text{pred}(\text{succ}(X)), X)
\]

The rule below is admissible.
\[
\frac{X: \text{int}}{\text{succ}(X): \text{int}}^R_1
\]
Is the rule below derivable, admissible, or neither?

\[
\frac{\text{pred}(X) \text{ int}}{\text{pred(succ}(X)) \text{ int}}
\]

The rule below is admissible.

\[
\frac{\text{pred}(X) \text{ int}}{\text{pred(succ}(X)) \text{ int}}
\]

Is the rule below derivable, admissible, or neither?

\[
\frac{\text{succ}(X) \text{ int}}{\text{pred(succ}(X)) \text{ int}}
\]

The rule below is derivable.

\[
\frac{\text{succ}(X) \text{ int}}{\text{pred(succ}(X)) \text{ int}}
\]

Proof Sketch

By induction on \( X \text{ nat} \)

\( \text{IH}(x) = \text{ If } x \text{ nat then } x \text{ int} \)

Subgoal1: \( \text{IH}(\text{zero}) \)

Subgoal2: If \( \text{IH}(X') \text{ then } \text{IH}(\text{succ}(X')) \)

Prove the following

If \( X \text{ nat} \text{ then } X \text{ int} \)

(just sketch out the structure, i.e.)
Rule Induction Principle

If \( X \in \text{int} \),

\begin{align*}
    & P(\text{zero}), \\
    & \text{if } P(X) \text{ then } P(\text{succ}(X)), \text{ and} \\
    & \text{if } P(Y) \text{ then } P(\text{pred}(Y)), \\
    & \text{then } P(X)
\end{align*}

What is the principle of rule induction look like for the rules above?

Did You Ace The Quiz?

- If so great!
- If not go through the notes and the slides from lecture 1
- Still stuck talk to me or the TA
- Did the entire class ace the quiz?
  - Probably not you are not the only one who is confused!

Inductively Defined Functions and Standard ML

COS 441
Princeton University
Fall 2004

Assignment 1

- Handout today due back next Wednesday
- Requires ML programming an a few simple proofs
- Make sure you're all set up to use your CS account and program in ML
- Details and updates available through the course web

Relations Review

- A relation is set of tuples
  - \( \text{Odd} = \{1, 3, 5, \ldots\} \)
  - \( \text{Line} = \{(0,0), (1.5,1.5), (x,x), \ldots\} \)
  - \( \text{Circle} = \{(x,y) \mid x^2 + y^2 = 1.0\} \)
- \( \text{Odd} \) is a predicate on natural numbers
- \( \text{Line}, \text{Circle}, \) and \( \text{Sphere} \) are relations on real numbers
- \( \text{Line} \) is a function
Functions and Their Graphs

- The graph of a function \( f(x) \) is the unique relation \( \{(x,y) \mid f(x) = y\} \)
- We can uniquely specify a function by defining its graph as a relation
- Not all relations specify valid functions!

Some “graphs” of Relations

- Below are some plotted graphs of the relations Circle and Line

![Circle and Line graphs](image)

- For a relation to be a valid graph of a function each unique input has a unique output

Defining the Function add

- We want to define a function \( \text{add}(m,n) \)
- To do this first specify a relation that defines its graph \( A(m,n,p) \) inductively
- Next show that for any unique pair of \( m \) and \( n \) there is a unique \( p \) such that \( A(m,n,p) \)

Defining the Graph of add

\[
\frac{X \text{ nat}}{A(X, \text{zero}, X)} \quad \frac{A(X,Y,Z)}{A(X, \text{succ}(Y), \text{succ}(Z))}
\]

Avoiding Clutter

Alternative definition that is equivalent to our previous one but its more cluttered since we have redundant premises

\[
\frac{X \text{ nat}}{A(X, \text{zero}, X)} \quad \frac{A(X,Y,Z)}{A(X, \text{succ}(Y), \text{succ}(Z))}
\]

Why are the all those extra premises not needed?

Defining the Graph of add

\[
\frac{X \text{ nat}}{A(X, \text{zero}, X)} \quad \frac{A(X,Y,Z)}{A(X, \text{succ}(Y), \text{succ}(Z))}
\]

The definition above immediately entails the following rules

\[
\frac{A(X,Y,Z)}{A(X, \text{succ}(Y), \text{succ}(Z))}
\]

Why?
Defining the Graph of \textbf{add}

\[
\frac{X \text{ nat}}{A(X, \text{zero}, X)} \quad \frac{A(X, Y, Z)}{A(X, \text{succ}(Y), \text{succ}(Z))}
\]

The definition above immediately entails the following rules

\[
\frac{A(X, Y, Z)}{X \text{ nat}} \quad \frac{A(X, Y, Z)}{A(Y, \text{nat}}} \quad \frac{A(X, Y, Z)}{A(Z, \text{nat}}
\]

They can be shown to be admissible with the principle of rule induction for derivations of \(A\) and the rules \(Z\) and \(S\).

---

Proving \(A\) is a Function Graph

If \(A(X, Y, Z)\), \(X\) unique, and \(Y\) unique then \(Z\) is unique.

\[\text{Proof: } \text{By } \approx\]

---

Proving \(A\) is a Function Graph

If \(A(X, Y, Z)\), \(X\) unique, and \(Y\) unique then \(Z\) is unique.

\[\text{Proof: } \text{By rule induction for } A(X, Y, Z)\]

---

Proving \(A\) is a Function Graph

If \(A(X, Y, Z)\), \(X\) unique, and \(Y\) unique then \(Z\) is unique.

\[\text{Proof: } \text{By rule induction for } A(X, Y, Z)\]

\[
\begin{align*}
\text{If } A(X', Y, Z) & \text{ then } A(X', \text{succ}(Y), \text{succ}(Z)) \\
\text{then } A(X, Y, Z).
\end{align*}
\]

---

Proving \(A\) is a Function Graph

If \(A(X, Y, Z)\),

\[
\begin{align*}
\text{case } A-Z: & \text{ If } X' \text{ nat then } \\
& \text{IH}(X', \text{zero}, X'), \\
\text{case } A-S: & \text{ If } \text{IH}(X', Y', Z') \text{ then } \\
& \text{IH}(X', \text{succ}(Y'), \text{succ}(Z')) \\
& \text{then } \text{IH}(X, Y, Z). \\
& \text{IH}(x, y, z) = \approx
\end{align*}
\]
Proving \( A \) is a Function Graph

If \( A(X,Y,Z) \),
  case A-Z: If \( X' \) nat then
    \( \text{IH}(X,\text{zero},X') \),
  case A-S: If \( \text{IH}(X',Y',Z') \) then
    \( \text{IH}(X,\text{succ}(Y'),\text{succ}(Z')) \)
then \( \text{IH}(X,Y,Z) \).
\( \text{IH}(x,y,z) = \) If \( A(x,y,z) \), \( x \) unique, and \( y \) unique then \( z \) is unique.

Proving \( A \) is a Function Graph

case A-Z: If \( X' \) nat then \( \text{IH}(X',\text{zero},X') \).
\( \text{IH}(x,y,z) = \) If \( A(x,y,z) \), \( x \) unique, and \( y \) unique then \( z \) is unique.

Proving \( A \) is a Function Graph

case A-Z: If \( X' \) nat then \( \text{IH}(X',\text{zero},X') \), \( X' \) unique, and \( \text{zero} \) unique then \( X' \) is unique.

1. \( X' \) nat by assumption
2. \( A(X',\text{zero},X') \), \( X' \) unique, and \( \text{zero} \) unique by assumption

Proving \( A \) is a Function Graph

case A-Z: ...

1. \( X' \) nat by assumption
2. \( A(X',\text{zero},X'), X' \) unique, and \( \text{zero} \) unique by assumption
3. \( X' \) unique by (2)
Proving $A$ is a Function Graph

Case A-S: If $IH(X', Y', Z')$ then

$IH(X', \text{succ}(Y'), \text{succ}(Z'))$

$IH(x, y, z) = \text{If } A(x, y, z), x$ unique, and $y$ unique then $z$ is unique.

Proving $A$ is a Function Graph

Case A-S: ... then

If $A(X', \text{succ}(Y'), \text{succ}(Z'))$, $X$ unique, and $\text{succ}(Y')$ unique

then $\text{succ}(Z')$ is unique

1. $IH(X, Y, Z)$ by assumption

$IH(x, y, z) = \text{If } A(x, y, z), x$ unique, and $y$ unique then $z$ is unique.

Proving $A$ is a Function Graph

Case A-S: ... then $\text{succ}(Z')$ is unique

1. $IH(X, Y, Z)$ by assumption

2. $A(X', \text{succ}(Y'), \text{succ}(Z'))$, $X$ unique, and $\text{succ}(Y')$ unique by assumption

3. $A(X', Y', Z')$ by ??

$IH(x, y, z) = \text{If } A(x, y, z), x$ unique, and $y$ unique then $z$ is unique.

Proving $A$ is a Function Graph

Case A-S: ... then $\text{succ}(Z')$ is unique

1. $IH(X, Y, Z)$ by assumption

2. $A(X', \text{succ}(Y'), \text{succ}(Z'))$, $X$ unique, and $\text{succ}(Y')$ unique by assumption

3. $A(X', Y', Z')$ by ??

4. $Y'$ unique

$IH(x, y, z) = \text{If } A(x, y, z), x$ unique, and $y$ unique then $z$ is unique.
Proving $A$ is a Function Graph

case $A$: ... then $\text{succ}(Z)$ is unique
1. $\text{IH}(X,Y,Z)$ by assumption
2. $A(X',\text{succ}(Y'),\text{succ}(Z'))$, $X'$ unique, and
   $\text{succ}(Y')$ unique by assumption
3. $A(X',Y,Z)$ by (2) and invert-$A$-S
4. $Y$ unique by ??

$\text{IH}(x,y,z) = \text{If } A(x,y,z), x \text{ unique, and } y \text{ unique then } z \text{ is unique.}$

Proving $A$ is a Function Graph

case $A$: ... then $\text{succ}(Z)$ is unique
1. $\text{IH}(X,Y,Z)$ by assumption
2. $A(X',\text{succ}(Y'),\text{succ}(Z'))$, $X'$ unique, and
   $\text{succ}(Y')$ unique by assumption
3. $A(X',Y,Z)$ by (2) and invert-$A$-S
4. $Y$ unique by ??
5. $Z$ unique by ??

$\text{IH}(x,y,z) = \text{If } A(x,y,z), x \text{ unique, and } y \text{ unique then } z \text{ is unique.}$

Proving $A$ is a Function Graph

case $A$: ... then $\text{succ}(Z)$ is unique
1. $\text{IH}(X,Y,Z)$ by assumption
2. $A(X',\text{succ}(Y'),\text{succ}(Z'))$, $X'$ unique, and
   $\text{succ}(Y')$ unique by assumption
3. $A(X',Y,Z)$ by (2) and invert-$A$-S
4. $Y$ unique by ??
5. $Z$ unique by ??
6. $\text{succ}(Z)$ unique by ??

$\text{IH}(x,y,z) = \text{If } A(x,y,z), x \text{ unique, and } y \text{ unique then } z \text{ is unique.}$

Some Missing Pieces

$A(X,\text{succ}(Y),\text{succ}(Z))$ invert-$A$-S

• We need to assume the following
  - zero is unique
  - if $X \text{ nat}$ and $X$ unique then $\text{succ}(X)$ is unique
  - if $X \text{ nat}$ and $\text{succ}(X)$ unique then $X$ is unique
• It is okay to assume ‘obvious’ things just be explicit about what you assume in proofs

A Function as Recursive Equations

• We often use different notations to defined the graph of a functions
  $\text{add}(M,\text{zero}) \equiv M$
  $\text{add}(M,\text{succ}(N)) \equiv \text{succ}(\text{add}(M,N))$
• The equations define a relation implicitly that relation must still be shown to be the graph of a valid function
Example: Fibonacci Function

\[ \text{fib}(\text{zero}) \equiv \text{succ}(\text{zero}) \]
\[ \text{fib}(\text{succ}(\text{zero})) \equiv \text{succ}(\text{zero}) \]
\[ \text{fib}(\text{succ}(\text{succ}(N))) \equiv \text{add}(\text{fib}(\text{succ}(N)), \text{fib}(N)) \]

What are the rules for the relation being implicitly defined by the equations above?

Example: Fibonacci Function

\[ \text{F}(\text{zero}, \text{succ}(\text{zero})) \]
\[ \text{F}(\text{succ}(\text{zero}), \text{succ}(\text{zero})) \]
\[ \text{F}(\text{succ}(\text{succ}(N)), Z) \]
\[ \text{F}(\text{succ}(\text{succ}(N)), X) \]
\[ \text{F}(\text{succ}(\text{succ}(N)), Y) \]

Does the relation \( F \) define the graph of a function?
Why?

Summary of Definitions

\[ \text{zero} \]
\[ \text{succ}(X) \]
\[ \text{add}(X, Y) \]
\[ \text{A}(X, Y, Z) \]
\[ \text{A}(X, Z, Y) \]
\[ \text{A}(Y, Z, X) \]
Some Derivable Judgments

\[ F(\text{succ}(\text{succ}(\text{zero}))), \text{succ}(\text{succ}(\text{zero}))) \]
\[ F(\text{succ}(\text{succ}(\text{succ}(\text{zero})))), \text{succ}(\text{succ}(\text{succ}(\text{zero})))) \]
\[ F(\text{succ}(\text{succ}(\text{succ}(\text{succ}(\text{zero})))), \text{succ}(\text{succ}(\text{succ}(\text{succ}(\text{zero}))))) \] ....

From Relations to SML

- Deriving the judgments by hand is tedious!
- We can use SML as a calculator of sorts to directly express the function we defined as an SML function
- To do this first we have to separate our functions from our data

Separating Functions From Data

- The \text{nat} predicate defines an \textit{abstract syntax tree}
  - More about this next lecture
- We can express the remaining relations as recursive equations that define a function
  - We need to verify that the equations do define well defined functions
  - But we've done that already for these two relations in this lecture!

Separating Functions From Data

\[
\begin{align*}
\text{nat} &::= \text{zero} \mid \text{succ}(n) \\
A(X,\text{zero},X) &\quad A(X,\text{succ}(Y),\text{succ}(Z)) \\
\text{F}(\text{zero},\text{succ}(\text{zero})) &\quad \text{F}(\text{succ}(\text{zero}),\text{succ}(\text{zero})) \\
\text{F}(\text{succ}(N),X) &\quad \text{F}(\text{succ}(N),Z) \\
\text{F}(\text{A}(X,Y,Z)) &\quad \text{F}(\text{A}(X,Y,Z)) \\
\text{A}(\text{X},\text{zero},X) &\quad A(X,\text{succ}(Y),\text{succ}(Z)) \\
\text{F}(\text{zero},\text{succ}(\text{zero})) &\quad \text{F}(\text{succ}(\text{zero}),\text{succ}(\text{zero}))
\end{align*}
\]
Separating Functions From Data

- $$\text{nat } n ::= \text{zero } | \text{succ}(n)$$
- $$\text{add}(M, \text{zero}) \equiv M$$
- $$\text{add}(M, \text{succ}(N)) \equiv \text{succ}(\text{add}(M, N))$$
- $$\text{fib}(\text{zero}) \equiv \text{succ}(\text{zero})$$
- $$\text{fib}(\text{succ}(\text{zero})) \equiv \text{succ}(\text{zero})$$
- $$\text{fib}(\text{succ}(\text{succ}(N))) \equiv \text{add}(\text{add}(\text{succ}(N)), \text{fib}(N))$$

From Relations to SML (cont.)

- We can convert the abstract syntax tree into and ML \texttt{datatype} declarations
- The recursive equations we can write down as ML functions
  - What if our recursive equations didn’t actually define a function but we translated it naively anyway?

\begin{verbatim}
datatype nat = zero | succ of nat

fun add(m, zero)       = m
| add(m, succ(n))   = succ(add(m, n))

fun fib(zero)         = succ(zero)
| fib(succ(zero))   = succ(zero)
| fib(succ(succ(n))) = add(fib(succ(n)),
                      fib(n))
\end{verbatim}
Lessons Learned

• We can define functions by inductively specifying a relation that defines it graph
• Recursive equations can be used to specify relations that define functions
  – We must verify that the function is well defined
• Many recursive equations can be turned directly into SML code
  – The reason we use SML in this course

Next Lecture

• Lexical analysis and parsing along with other things you will not learn about in detail from this course
  – We’ll talk about them to understand why they are “uninteresting”
• Abstract Syntax