Lecture 21: Logic, part II

To Truth through Proof

Prof. Clarkson

Fall 2014

Today’s music: "The Devil went down to Georgia"
by The Charlie Daniels Band
Review

Current topic:
• How to reason about correctness of code
• Last week: informal arguments

Today:
• Logic, part II
• Upgrade from *propositional* logic to *predicate* logic
Question #1

How much of PS5 have you finished?

A. None
B. About 25%
C. About 50%
D. About 75%
E. I’m done!!!
Review: A biased perspective on logic

• A logic is a programming language for expressing reasoning about evidence

• Like any PL, a logic has
  – syntax
  – dynamic semantics (evaluation rules) --omitted here
  – static semantics (type checking)
Review: IPC

IPC = Intuitionistic Propositional Calculus

Syntax:

\[ f ::= P \mid f_1 \land f_2 \mid f_1 \lor f_2 \mid f_1 \rightarrow f_2 \mid \neg f \]
\[ P ::= \text{true} \mid \text{false} \mid \ldots \]
## Review: Proof rules so far

<table>
<thead>
<tr>
<th>Rule name</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\land) intro</td>
<td>if (F \vdash f_1) and (F \vdash f_2) then (F \vdash f_1 \land f_2)</td>
</tr>
<tr>
<td>(\land) elim L</td>
<td>if (F \vdash f_1 \land f_2) then (F \vdash f_1)</td>
</tr>
<tr>
<td>(\land) elim R</td>
<td>if (F \vdash f_1 \land f_2) then (F \vdash f_2)</td>
</tr>
<tr>
<td>=&gt; elim</td>
<td>if (F \vdash f) and (F \vdash f \Rightarrow g) then (F \vdash g)</td>
</tr>
<tr>
<td>=&gt; intro</td>
<td>if (F, f \vdash g) then (F \vdash f \Rightarrow g)</td>
</tr>
<tr>
<td>assump</td>
<td>(f \vdash f)</td>
</tr>
<tr>
<td>weak</td>
<td>if (F \vdash f) then (F, g \vdash f)</td>
</tr>
<tr>
<td>set assump</td>
<td>(F, f \vdash f)</td>
</tr>
</tbody>
</table>
Evidence for true and false

Q: What constitutes evidence for true?
A: We don't need any; true trivially holds

Q: What constitutes evidence for false?
A: Nothing; false can never hold.

If we ever did somehow have evidence for false, then we'd be in a contradictory situation, and all reason has broken down.
Proof rules for true and false

• $F \vdash true$
  – only an introduction rule, no elimination
  – another axiom
  – intuition: we can always give evidence for true

• if $F \vdash false$ then $F \vdash f$
  – ex falso quodlibet: "from false follows whatever you please"
  – Principle of Explosion
  – only an elimination rule, no introduction
  – intuition: we can never give evidence for false; but once we can conclude false, we can conclude anything
Evidence for ~

Q: What constitutes evidence for ~\(f\)?

A: Since ~\(f\) really means \(f \Rightarrow \text{false}\), it would be a way of transforming evidence for \(f\) into evidence for false. That is, a way of reaching a contradiction.
Proof rules for ~

Negation is just syntactic sugar, so free to convert between those two forms:

• if $F \vdash f \Rightarrow \text{false}$ then $F \vdash \sim f$
  – intuition: if there's a way to transform evidence for $f$ into evidence for $\text{false}$, then you have evidence for $\sim f$

• if $F \vdash \sim f$ then $F \vdash f \Rightarrow \text{false}$
  – intuition: if you have evidence for $\sim f$, then you have a way of transforming evidence for $f$ into evidence for $\text{false}$
Evidence for \(/

Q: What constitutes evidence for \(f_1 / f_2\)?

A: Evidence for either \(f_1\) or for \(f_2\), tagged to indicate which one it's evidence for.

So evidence for \(f_1 / f_2\) is really a value of a datatype:

```haskell
type ('a,'b) sum =
    Left of 'a | Right of 'b
```
Proof rules for \(/

- if $F \models f_1$ then $F \models f_1 \lor f_2$
- if $F \models f_2$ then $F \models f_1 \lor f_2$

  - intuition: if you have evidence for $f_1$, then you have evidence for $f_1 \lor f_2$

  - further intuition: these rules are really just constructor application
Proof rules for $\lor$

- if $F \vdash f_1 \lor f_2$ and $F \vdash f_1 \Rightarrow g$ and $F \vdash f_2 \Rightarrow g$ then $F \vdash g$
  
  – intuition: if you have evidence for $f_1 \lor f_2$, and if you have a way of transforming evidence for $f_1$ into evidence for $g$, as well as for $f_2$ into $g$, then you can obtain evidence for $g$

  – further intuition: this rule is really just pattern matching!

    match s with
    
    Left $f_1$ -> $e_1$

    | Right $f_2$ -> $e_2$
Proof with 

Let's show \( |- (P \lor Q) \rightarrow (Q \lor P) \)

1. \( P \lor Q |- P \lor Q \) byassump
2. \( P |- P \) by assump
3. \( P |- Q \lor P \) by (2) and \( \lor \) intro R
4. \( |- P \rightarrow Q \lor P \) by (3) and \( \rightarrow \) intro
5. \( P \lor Q |- P \rightarrow Q \lor P \) by (4) and weak.
6. \( Q |- Q \) by assump
7. \( Q |- Q \lor P \) by (6) and \( \lor \) intro L
8. \( |- Q \rightarrow Q \lor P \) by (7) and \( \rightarrow \) intro
9. \( P \lor Q |- Q \rightarrow Q \lor P \) by (8) and weak.
10. \( P \lor Q |- Q \lor P \) by (1), (5), (9) and \( \lor \) elim
11. \( |- (P \lor Q) \rightarrow (Q \lor P) \) by \( \rightarrow \) intro
Tree form

\[ \begin{array}{c}
| - (P \lor Q) \Rightarrow (Q \lor P) \\
\end{array} \]
Tree form

\[
P \lor Q \vdash Q \lor P
\]

\[
\vdash (P \lor Q) \Rightarrow (Q \lor P)
\]

=> intro

\[
\vdash (P \lor Q) \Rightarrow (Q \lor P)
\]
Tree form

\[ P \lor Q \vdash P \lor Q \quad P \lor Q \vdash P \Rightarrow (Q \lor P) \quad P \lor Q \vdash Q \Rightarrow (Q \lor P) \]

\[ \lor \text{elim} \]

\[ P \lor Q \vdash Q \lor P \quad \Rightarrow \text{intro} \]

\[ \vdash (P \lor Q) \Rightarrow (Q \lor P) \]
Tree form

\[
\frac{
\begin{array}{c}
\text{assump} \\
P \lor Q \vdash P \lor Q
\end{array}
}{
\begin{array}{c}
P \lor Q \vdash P \Rightarrow (Q \lor P) \\
P \lor Q \vdash Q \Rightarrow (Q \lor P)
\end{array}
}{P \lor Q \vdash Q \lor P}
\]

\[
\frac{
\begin{array}{c}
P \lor Q \vdash Q \lor P
\end{array}
}{
\begin{array}{c}
| - (P \lor Q) \Rightarrow (Q \lor P)
\end{array}
}
\]
Tree form

\[ \frac{\text{assump}}{P \lor Q \vdash P \lor Q} \]

\[ \frac{P \lor Q, P \vdash Q \lor P}{\implies \text{intro}} \]

\[ \frac{P \lor Q \vdash P \implies (Q \lor P)}{P \lor Q \vdash Q \implies (Q \lor P)} \]

\[ \frac{P \lor Q \vdash Q \lor P}{\lor \text{elim}} \]

\[ \frac{\vdash (P \lor Q) \implies (Q \lor P)}{\implies \text{intro}} \]

\[ \vdash (P \lor Q) \implies (Q \lor P) \]
Tree form

\[ \begin{align*}
&\frac{\frac{}{P \lor Q \vdash P \lor Q}}{} \\
&\frac{\frac{P \lor Q, P \vdash Q \lor P}{\frac{}{P \lor Q \vdash P \rightarrow (Q \lor P)}}{} \\
&\frac{\frac{}{P \lor Q \vdash Q \rightarrow (Q \lor P)}}{} \\
&\frac{}{P \lor Q \vdash (P \lor Q) \rightarrow (Q \lor P)}
\end{align*} \]
Tree form

\[
\begin{align*}
P \mid - P \\
P \mid - Q \lor P & \quad \text{// intro-r} \\
\hline
P \\ Q, P \mid - Q \lor P & \quad \text{weak} \\
\hline
\text{assump} \\
\hline
P \\ Q \mid - P \Rightarrow (Q \lor P) & \quad \Rightarrow \text{intro} \\
\hline
P \\ Q \mid - Q \Rightarrow (Q \lor P) \\
\hline
\text{\textbackslash elim} \\
\hline
\text{assump} \\
\hline
P \\ Q \mid - Q \lor P & \quad \Rightarrow \text{intro} \\
\hline
\mid - (P \lor Q) \Rightarrow (Q \lor P)
\end{align*}
\]
Tree form

\[
\begin{align*}
\frac{P \lor Q, P}{P} & \quad \text{assump} \\
\frac{P \lor Q}{Q \lor P} & \quad \text{intro-r} \\
\frac{P \lor Q, Q}{Q} & \quad \text{assump} \\
\frac{Q \lor P}{Q} & \quad \text{intro-l} \\
\frac{P \lor Q}{Q \lor P} & \quad \text{weak} \\
\frac{Q \lor P}{Q} & \quad \text{weak} \\
\frac{P \lor Q, Q \lor P}{P \lor Q} & \quad \text{intro} \\
\frac{P \lor Q}{Q \lor P} & \quad \text{intro} \\
\frac{P \lor Q}{(Q \lor P)} & \quad \text{intro-r} \\
\frac{Q \lor P}{(Q \lor P)} & \quad \text{intro-l} \\
\frac{P \lor Q}{(Q \lor P)} & \quad \text{elim} \\
\frac{P \lor Q}{Q \lor P} & \quad \text{intro} \\
\frac{(P \lor Q) \Rightarrow (Q \lor P)}{} & \quad \text{intro} \\
\end{align*}
\]

Note: bad formatting! hard to fit on slide 😞
As an OCaml program

```
let or_comm (s: ('p,'q) sum) : ('q,'p) sum =
  match s with
  | Left p -> Right p
  | Right q -> Left q
```

How to think about this program:

`or_comm` is a function that takes in evidence for either 'p or 'q, and returns evidence for either 'q or 'p.
As an OCaml program

```ocaml
let or_comm (s: ('p,'q) sum) : ('q,'p) sum =
  match s with
  | Left p  -> Right p
  | Right q -> Left q
```

What is its type?

```ocaml
('p, 'q) sum  ->  ('q, 'p) sum
```

imagine we could write `sum` as infix `+`

```ocaml
'p + 'q  ->  'q + 'p
```

What is the formula we proved?

```latex
(P \lor Q) \Rightarrow (Q \lor P)
```
What about $P \lor (\neg P)$?

- aka excluded middle
- Many presentations of logic simply assume this holds for any proposition $P$
  - Indeed, for any formula $\mathcal{F}$
- **Cannot be proved in IPC**
- But we could add $\vdash P \lor (\neg P)$ to IPC to get a new logic, CPC
  - CPC has same **syntax** as IPC, but **type system** that's "bigger" by one rule
  - Then we'd be saying there's always a way to give evidence for either $P$, or for $P \Rightarrow \text{false}$.
  - But we wouldn't be saying what that evidence is...
The Devil’s Middle
Classical vs. constructive

• Without excluded middle we have *constructive logic*
  – Constructive ≅ intuitionistic
  – A *constructive* proof is an algorithm (cf. the programs we've been writing that correspond to proofs)

• With it, we have *classical logic*
  – CPC = classical propositional calculus

• Truth vs. proof
  – Truth:
    • Classical proofs are concerned with truth values
    • All propositions are either true or false
  – Proof:
    • Constructive proofs are concerned with evidence
    • Propositions don't have "truth values"; rather, their truth is unknown until can be (dis)proved
## Proof rules of IPC, part 1

<table>
<thead>
<tr>
<th>Rule name</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\land) intro</td>
<td>if (F \vdash f_1) and (F \vdash f_2) then (F \vdash f_1 \land f_2)</td>
</tr>
<tr>
<td>(\land) elim L</td>
<td>if (F \vdash f_1 \land f_2) then (F \vdash f_1)</td>
</tr>
<tr>
<td>(\land) elim R</td>
<td>if (F \vdash f_1 \land f_2) then (F \vdash f_2)</td>
</tr>
<tr>
<td>(\Rightarrow) elim</td>
<td>if (F \vdash f) and (F \vdash f \Rightarrow g) then (F \vdash g)</td>
</tr>
<tr>
<td>(\Rightarrow) intro</td>
<td>if (F, f \vdash g) then (F \vdash f \Rightarrow g)</td>
</tr>
<tr>
<td>assms</td>
<td>(f \vdash f)</td>
</tr>
<tr>
<td>weak</td>
<td>if (F \vdash f) then (F, g \vdash f)</td>
</tr>
<tr>
<td>set assms</td>
<td>(F,f \vdash f)</td>
</tr>
</tbody>
</table>
## Proof rules of IPC, part 2

<table>
<thead>
<tr>
<th>Rule name</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lor$ intro L</td>
<td>if $F \vdash f_1$ then $F \vdash f_1 \lor f_2$</td>
</tr>
<tr>
<td>$\lor$ intro R</td>
<td>if $F \vdash f_2$ then $F \vdash f_1 \lor f_2$</td>
</tr>
<tr>
<td>$\lor$ elim</td>
<td>if $F \vdash f_1 \lor f_2$ and $F \vdash f_1 \Rightarrow g$ and $F \vdash f_2 \Rightarrow g$ then $F \vdash g$</td>
</tr>
<tr>
<td>true intro</td>
<td>$F \vdash true$</td>
</tr>
<tr>
<td>false elim</td>
<td>if $F \vdash false$ then $F \vdash f$</td>
</tr>
<tr>
<td>$\neg$ intro</td>
<td>if $F \vdash f \Rightarrow false$ then $F \vdash \neg f$</td>
</tr>
<tr>
<td>$\neg$ elim</td>
<td>if $F \vdash \neg f$ then $F \vdash f \Rightarrow false$</td>
</tr>
</tbody>
</table>
Natural deduction

• Style of proof system we just gave is called *natural deduction*
  – Gentzen (1934), Prawitz (1965)
  – Very few axioms, mostly *inference rules*
  – With intro and elim rules for each connective

• Graphical notation for proof trees is considered a strength of this style
  – Even if it doesn't work well in slides! 😊
  – Even if it doesn't scale well to large proofs!

• In notes and in recitation: larger examples of proofs
Formalize this argument

• All squares are positive
• 9 is a square
• Therefore 9 is positive
Formalize this argument

• All squares are positive \( f \)
• 9 is a square \( g \)
• Therefore 9 is positive \( h \)

an attempt: \( f \land g \Rightarrow h \)

...but that's not a provable formula

...so we might have trouble proving that the return value of \textit{square} is positive!

...we need \textit{predicates}
Predicates

- *Predicates* aka *relations* upgrade propositions to have arguments:
  - `is_positive(x)`
  - `is_square(x)`
  - `equals(x, y)`, usually written `x=y`

- *Objects* (the variables above) are the atomic things we now talk about
  - might be integers, lists of strings, real numbers, etc.

- *Functions* map between objects
  - `square(3)`, which is 9

- *Quantifiers* let us talk about all objects at once:
  - "for all objects x, it holds that `P(x)` " (universal)
  - "there exists an object x, such that `P(x)` holds" (existential)
A new logic: IQC

Syntax:

\[ f ::= \mathcal{P}(t_1, \ldots, t_n) \]
\[ | f_1 \land f_2 \]
\[ | f_1 \lor f_2 \]
\[ | f_1 \Rightarrow f_2 \]
\[ | \sim f \]
\[ | \forall x, f \]
\[ | \exists x, f \]

\[ t ::= x \]
\[ | fn(t_1, \ldots, t_n) \]

- \( \mathcal{P} \) is a meta-variable for predicates/relations (incl. \textit{nullary} predicates \texttt{true} and \texttt{false})
- \( t \) is a meta-variable for \textit{terms}, including constants, variables, and functions \texttt{fn} applied to terms (including \textit{nullary} functions, i.e., constants)
IQC

• IQC = Intuitionistic Quantifier Calculus
• CQC = Classical Quantifier Calculus
  – equals IQC + excluded middle
• CQC aka
  – first order logic (FOL)
  – predicate logic
  – predicate calculus
Formalize this argument

• All squares are positive $\forall x, \text{is\_square}(x) \Rightarrow \text{is\_positive}(x)$
• 9 is a square $\text{is\_square}(9)$
• Therefore 9 is positive $\text{is\_positive}(9)$

$$((\forall x, \text{is\_square}(x) \Rightarrow \text{is\_positive}(x)) \land \text{is\_square}(9)) \Rightarrow \text{is\_positive}(9)$$
Proof rules for IQC

• All the rules of IPC, plus intro and elim for quantifiers

• New notation:
  – $f(x)$ means a formula $f$ that mentions a variable $x$
  – $f(t)$ means that same formula $f$, but with all mentions of $x$ replaced by term $t$
Q: What constitutes evidence for \( \text{forall } x, f(x) \)?

A: A way of producing evidence for \( f(x) \) out of an arbitrary object \( x \).

...That is, a way of transforming an object \( x \) into evidence of \( f(x) \)

(note the similarity to \( \Rightarrow \))
Proof rules for \texttt{forall}

- if $F \vdash f(x)$ and $F$ does not make any assumptions about $x$, then $F \vdash \forall x, f(x)$
  
  -$\text{introduction rule}$
  
  -$\text{intuition:}$ if you can construct evidence for $f(x)$ without making any assumptions about $x$, then you have a way of transforming $x$ into evidence for $f(x)$

...but what does "make assumptions about" mean"?
Free variables

*Free variables* are variables that aren't bound by any quantifier:

- \( P(x) \): \( x \) is free
- \( \forall x, P(x) \land Q(y) \): \( x \) is not free and \( y \) is free
- \( R(x) \Rightarrow (\forall x, P(x)) \): \( x \) is free in LHS of implication, but not in RHS

If \( x \) does not occur free in a formula, then the formula makes no assumptions about it. Likewise for a set of formulae.
Free variables (formal defn)

\[ FV(x) = \{x\} \]
\[ FV(f(t1, \ldots t_n)) = FV(t1) \cup \ldots \cup FV(t_n) \]
\[ FV(P(t1, \ldots t_n)) = FV(t1) \cup \ldots \cup FV(t_n) \]
\[ FV(f_1 / \backslash f_2) = FV(f_1) \cup FV(f_2) \]
\[ FV(f_1 \Rightarrow f_2) = FV(f_1) \cup FV(f_2) \]
\[ FV(f_1 \backslash / f_2) = FV(f_1) \cup FV(f_2) \]
\[ FV(\neg f) = FV(f) \]
\[ FV(\forall x, f) = FV(f) \setminus \{x\} \]
\[ FV(\exists x, f) = FV(f) \setminus \{x\} \]
Proof rules for forall

- if $F \vdash f(x)$ and $x$ does not occur free in $F$, then $F \vdash \forall x, f(x)$
  - introduction rule
  - "$x$ does not occur free in $F$" means $x$ not in $\text{FV}(f)$ for any $f$ in $F$
  - intuition: if you can construct evidence for $f(x)$ without making any assumptions about $x$, then you have a way of transforming $x$ into evidence for $f(x)$
Proof rules for \texttt{forall}

- if $F \vdash \texttt{forall } x, f(x)$, then $F \vdash f(t)$
  
  – elimination rule

  – \textbf{intuition:} if you have a way of transforming any $x$ into evidence for $f(x)$, then you can use that to produce evidence for $f(t)$ out of $t$
Proof with forall

Let's show \( |- (\forall x, R(x) \land Q(x)) \implies \equiv (\forall x, R(x)) \land (\forall x, Q(x)) \)

1. \( \forall x, R(x) \land Q(x) |- \forall x, R(x) \land Q(x) \) by assump.
2. \( \forall x, R(x) \land Q(x) |- R(x) \land Q(x) \) by (1) and \( \land \) elim.
3. \( \forall x, R(x) \land Q(x) |- R(x) \) by (2) and \( \land \) elim L.
4. \( \forall x, R(x) \land Q(x) |- \forall x, R(x) \) by (3) and \( \forall \) intro*.
5. \( \forall x, R(x) \land Q(x) |- Q(x) \) by (2) and \( \land \) elim R.
6. \( \forall x, R(x) \land Q(x) |- \forall x, Q(x) \) by (5) and \( \forall \) intro*.
7. \( \forall x, R(x) \land Q(x) |- (\forall x, R(x)) \land (\forall x, Q(x)) \) by (4), (6) and \( \land \) intro.
8. \( |- (\forall x, R(x) \land Q(x)) \implies \equiv (\forall x, R(x)) \land (\forall x, Q(x)) \) by (7) and => intro.

* \( x \) does not occur free in LHS
forall x, R(x) \& Q(x) |- forall x, R(x) \& Q(x)

forall x, R(x) \& Q(x) |- R(x) \& Q(x)

forall x, R(x) \& Q(x) |- R(x)

forall x, R(x) \& Q(x) |- (forall x, R(x))

forall x, R(x) \& Q(x) |- (forall x, Q(x)) -> intro

forall x, R(x) \& Q(x) -> (forall x, R(x)) \& (forall x, Q(x)) => intro

* x does not occur free in LHS

Note: bad formatting! hard to fit on slide 😐
As an OCaml program?

- OCaml's type system is not quite expressive enough to give a program whose type is that formula
  - In part, reason for that is to get good type inference
- Languages with richer type systems can do it
  - See CS 4110/6110
- Same will be true of existentials...
Evidence for exists

Q: What constitutes evidence for \( \exists x, f(x) \)?

A: A witness object \( w \), along with evidence for \( f(w) \).
Proof rules for exists

- if $F \vdash f(t)$ then $F \vdash \exists x, f(x)$
  - introduction rule
  - intuition: if you can construct evidence for $f(t)$ then $t$ is a witness.
Proof rules for \texttt{exists}

- If $F \vdash \texttt{exists } x, \ f(x)$ and $F \vdash f(x) \Rightarrow g$ and $x$ does not occur free in $F$ or $g$, then $F \vdash g$
  
  - elimination rule
  
  - \texttt{intuition:} if you have a witness $w$ for $f(w)$, and if you have a way of transforming evidence for $f(x)$ into evidence for $g$, and if there are no assumptions about $x$, then you can use $w$ in place of $x$ to get evidence for $g$. 
Proof with \(\text{exists}\)

Let's show \(\vdash \ (\text{exists } x, Q(x) \lor R(x)) \Rightarrow (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\)

1. \(Q(x) \vdash Q(x)\) by assump.
2. \(Q(x) \vdash \text{exists } x, Q(x)\) by (1) and \(\text{exists intro}\)
3. \(Q(x) \vdash (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by (2) and \(\lor\) intro L
4. \(\vdash Q(x) \Rightarrow (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by (3) and \(\Rightarrow\) intro
5. \(Q(x) \lor R(x) \vdash Q(x) \Rightarrow (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by (4) and weak.
6. \(Q(x) \lor R(x) \vdash R(x) \Rightarrow (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by \(\text{repeat (1—5)}\) with \(R\)
7. \(Q(x) \lor R(x) \vdash Q(x) \lor R(x)\) by assump.
8. \(Q(x) \lor R(x) \vdash (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by \(\lor\) elim using (7), (5), (6)
9. \(\vdash Q(x) \lor R(x) \Rightarrow (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by (8) and \(\Rightarrow\) intro
10. \(\text{exists } x, Q(x) \lor R(x) \vdash Q(x) \lor R(x) \Rightarrow (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by (9) and weak
11. \(\text{exists } x, Q(x) \lor R(x) \vdash \text{exists } x, Q(x) \lor R(x)\) by assump.
12. \(\text{exists } x, Q(x) \lor R(x) \vdash (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by \(\text{exists elim}\) using (11), (10), and \(x\) does not occur free in \((\text{exists } x, Q(x) \lor R(x))\) or in \((\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\)
13. \(\vdash (\text{exists } x, Q(x) \lor R(x)) \Rightarrow (\text{exists } x, Q(x)) \lor (\text{exists } x, R(x))\) by \(\Rightarrow\) intro

\(\text{tree form omitted; too big to fit on slides}\)
# Proof rules of IQC

<table>
<thead>
<tr>
<th>Rule name</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>All rules of IPC</td>
</tr>
<tr>
<td><strong>forall intro</strong></td>
<td>if $F</td>
</tr>
<tr>
<td><strong>forall elim</strong></td>
<td>if $F</td>
</tr>
<tr>
<td><strong>exists intro</strong></td>
<td>if $F</td>
</tr>
<tr>
<td><strong>exists elim</strong></td>
<td>if $F</td>
</tr>
</tbody>
</table>
Please hold still for 1 more minute

WRAP-UP FOR TODAY
Upcoming events

• PS5 due Thursday

This is logical.

THIS IS 3110