CS5860
Tue. Oct 4, 2011 First-Order Logic as a Programming Language continued

PLAN : 1. Tips for programming, 2. Boolean Logics revisited. 3. Some theory.

1 Hints and tips for writing programs

gather inputs into the hypothesis list as soon as possible in order to know what the data and preconditions provide for reaching the goals.

Note, assumptions such as $f: A \Rightarrow B$, $g: \forall x.A(x)$ tell you the type of the functions but nothing about their code. The data assumptions like $x: A \& B$, $x: \exists y B(y)$, $d: A \vee B$ tell you when you need operations such as spread and decide to access the data. But the ONLY thing you can do with a function, $f: A \Rightarrow B$ or $g: \forall x.A(x)$ is APPLY IT! You must use $ap(f;a)$, $ap(g;d)$ to get output values.

Suggested self assigned work. Write up notes about how to apply these programming tips to other programming tasks specified by types.

2. Boolean Logics revisited

We can explain Boolean logics computationally when the application area is defined using computational concepts. That means the domain D is a data type, the properties of D, say $P: D \to Prop$ are decidable, i.e. $\forall x.(P(x) \vee \sim P(x))$, relations $R^2: D \times D \to Prop$, $R^3: D \times D \times D \to Prop$ are decidable, i.e. $\forall x \forall y (R^2(x,y) \vee \sim R^2(x,y))$.

We also require that compound relations are decidable, e.g.

$$\exists x.P(x) \vee \sim \exists x.P(x), \quad \forall x \exists y R(x,y) \vee \sim \forall x \exists y R(x,y)$$

To program with Boolean logics, we need to make these assumptions explicit in the hypotheses as necessary. Next is an example.

Tue Oct 4, 2011 Programming in Boolean Logics – an example

Consider this task: write code for $\sim \forall x.\, B(x) \Rightarrow \exists x \sim B(x)$

In Boolean Logic where all properties and relations are decidable, we can use Tableau Rules (see Smullyan). Here is a proof.

1. F $(\sim \forall x.\, B(x) \Rightarrow \exists y.\sim B(y))$
2. T $\sim \forall x.\, B(x)$
3. F $\exists y \sim B(y)$
4. F $\forall x.\, B(x)$ from 2
5. F $B(x)$ from 4 new $x : D$
6. F $\sim B(x)$ from 3 x allowed for y (x can be any value)
7. T $B(x)$ from 6

✱ 5, 7

Computationally, this is much harder. Consider this approach:

$\vdash \sim \forall x.\, B(x) \Rightarrow \exists y \sim B(y)$ $\lambda(h. \_\_)$

$h : (\forall x.\, B(x)) \Rightarrow$ False $\vdash \exists y.\sim B(y)$ $ap(h; \_\_ ; v. \_\_)$

$v :$ False $\vdash \exists y.\sim B(y)$ by any(v)

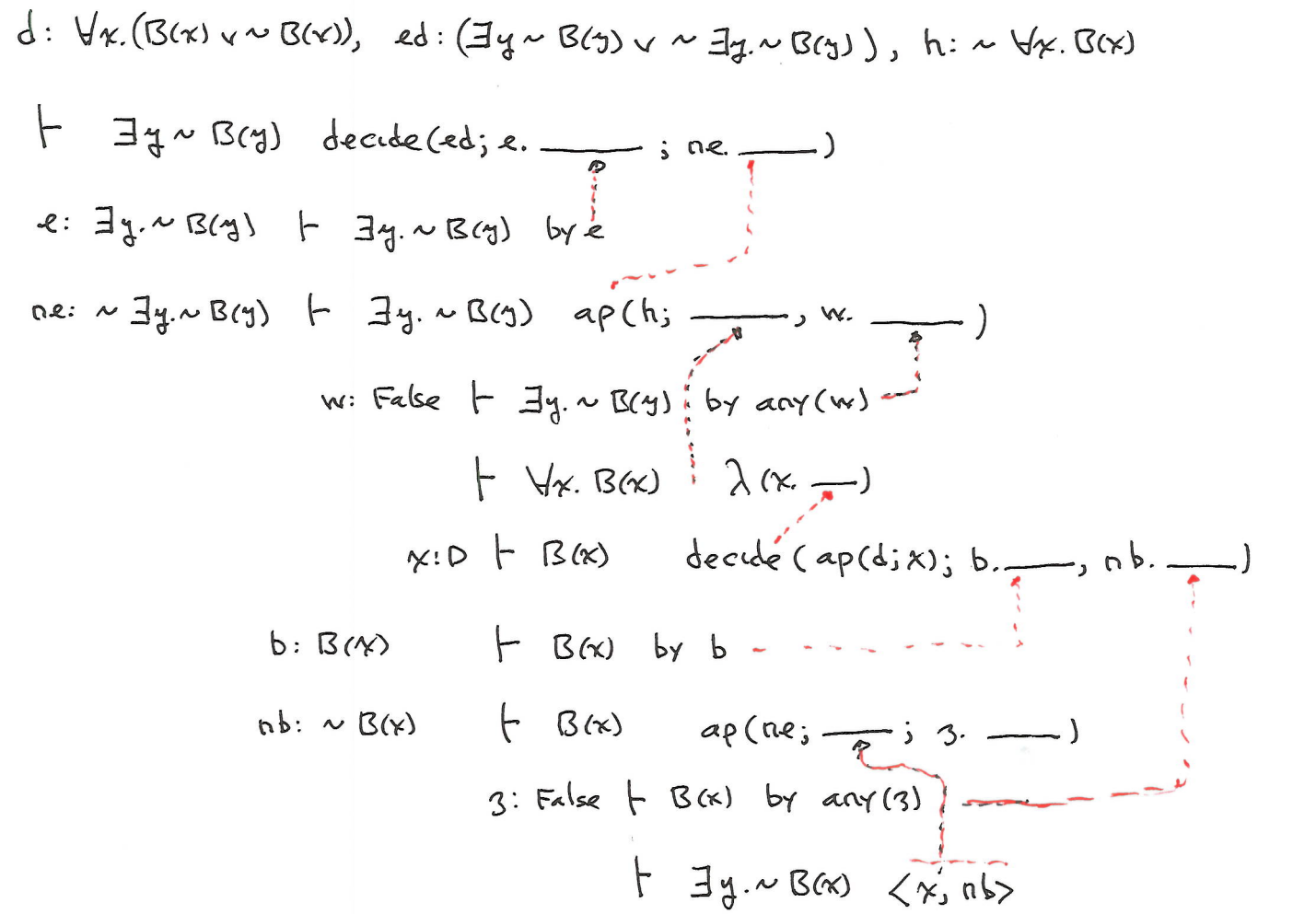
$\vdash \forall x.\, B(x)$ $\lambda(x. \_\_)$

$x : D \vdash B(x)$

STUCK!

We need to be explicit about some of the Boolean Logic assumptions, but which ones?

Is $\forall x.\, (B(x) \vee \sim B(x))$ enough?
Do we need to assume $\exists d.\, (B(d) \vee \sim B(d))$?
Here is a combination of assumptions that suffice.



λ(d,ed,h. decide(ed; e.e; ne. ap(h; λ(x. decide(ap(d;x);
b.b;
nb. any(ap(ne;<x,nb>)))))))

Extra Credit Exercise Can you find a simpler set of assumptions true in Boolean Models that gives this result? Can you use the Tableau Proof to find a simpler computational proof?

CS 5860

Tue Oct 4, 2011

Theory about First-Order Programming

We know some interesting facts about the FOL programming language. The results are expressed in logical terms.

1. Given a problem specification Spec, if we know ~Spec, i.e. Spec => False, then there is no program to meet the specification (i.e. solve the problem).
   We say that FOL is consistent.

2. If there is a program to solve a specification, then we can find it using the Refinement Rules.
   The programming rules are complete.

3. There is no way to automatically (mechanically) tell whether a programming problem is solvable.
   This is Church's Theorem.

4. We can define the Boolean Programming problems in the FOL language. (Gödel embedding)

CS 5860
Tue Oct 4

Tableau Form of Refinement Rules

$$\frac{T(X \& Y)}{\begin{array}{l} TX \\ TY \end{array}} \qquad \frac{F(X \& Y)}{FX \mid FY}$$

$$\frac{T(X \Rightarrow Y)}{TY \mid FX} \qquad \frac{F(X \Rightarrow Y)}{TX,\ FY}$$

$$\frac{T(X \vee Y)}{TX \mid TY} \qquad \frac{F(X \vee Y)}{FX} \text{ OR } \frac{F(X \vee Y)}{FY}$$

$$\frac{T(\text{False})}{\ast} \qquad \frac{F(\text{False})}{}$$

$$\frac{T(\forall x\, A(x))}{T\ A(a)} \quad \text{any } a \qquad \frac{F(\forall x.\, A(x))}{F\ A(x')} \quad \begin{array}{l} \text{new } x' \\ \text{(if needed, otherwise } x\text{)} \end{array}$$

$$\frac{T(\exists x\, A(x))}{T\ A(x')} \quad \text{new } x' \qquad \frac{F(\exists x\, A(x))}{F\ A(x)} \quad \text{any } x$$

Recall $\sim X$ is $(X \Rightarrow \text{False})$ Axioms are TX, FX.