# Assignment 5 Written Solutions

## Partial Derivations

This is a maximal partial derivation for `if true then 1 + 3 else x / false`:

$$\dfrac{\dfrac{}{\varnothing \vdash \texttt{true} : \texttt{bool}}\ \text{trueLit} \qquad \dfrac{\dfrac{}{\varnothing \vdash \texttt{1} : \texttt{int}}\ \text{intLit} \quad \dfrac{}{\varnothing \vdash \texttt{3} : \texttt{int}}\ \text{intLit}}{\varnothing \vdash \texttt{1 + 3} : \texttt{int}}\ \text{intAdd} \qquad \dfrac{\boxed{\varnothing \vdash \texttt{x} : \texttt{int}} \qquad \boxed{\varnothing \vdash \texttt{false} : \texttt{int}}}{\varnothing \vdash \texttt{x / false} : \texttt{int}}\ \text{intDiv}}{\varnothing \vdash \texttt{if true then 1 + 3 else x / false} : \texttt{int}}\ \text{if}$$

The highlighted judgments cannot be derived by any rule from any judgments. The error message:

<div align="center">

Error:  Unbound value x

</div>

This is the maximal partial derivation for `if true then 1 + 3 else false / x`:

$$\dfrac{\dfrac{}{\varnothing \vdash \texttt{true} : \texttt{bool}}\ \text{trueLit} \qquad \dfrac{\dfrac{}{\varnothing \vdash \texttt{1} : \texttt{int}}\ \text{intLit} \quad \dfrac{}{\varnothing \vdash \texttt{3} : \texttt{int}}\ \text{intLit}}{\varnothing \vdash \texttt{1 + 3} : \texttt{int}}\ \text{intAdd} \qquad \dfrac{\boxed{\varnothing \vdash \texttt{false} : \texttt{int}} \qquad \boxed{\varnothing \vdash \texttt{x} : \texttt{int}}}{\varnothing \vdash \texttt{false / x} : \texttt{int}}\ \text{intDiv}}{\varnothing \vdash \texttt{if true then 1 + 3 else false / x} : \texttt{int}}\ \text{if}$$

The error message:

<div align="center">

Error:  This expression has type bool but an expression was expected of type int

</div>

This tells us that OCaml type-checks the arguments of division from left to right.

## Typing Derivation

$\mathcal{D}_1$:

$$\dfrac{\dfrac{}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list},\texttt{h}:\texttt{int},\texttt{t}:\texttt{int list}\}\vdash\texttt{sum}:\texttt{int list}\to\texttt{int}}\ \text{var}\qquad\dfrac{}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list},\texttt{h}:\texttt{int},\texttt{t}:\texttt{int list}\}\vdash\texttt{t}:\texttt{int list}}\ \text{var}}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list},\texttt{h}:\texttt{int},\texttt{t}:\texttt{int list}\}\vdash\texttt{sum t}:\texttt{int}}$$

$\mathcal{D}_2$:

$$\dfrac{\dfrac{}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list},\texttt{h}:\texttt{int},\texttt{t}:\texttt{int list}\}\vdash\texttt{h}:\texttt{int}}\ \text{var}\qquad\mathcal{D}_1}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list},\texttt{h}:\texttt{int},\texttt{t}:\texttt{int list}\}\vdash\texttt{h + sum t}:\texttt{int}}\ \text{intList}$$

$\mathcal{D}_3$:

$$\dfrac{\dfrac{}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list}\}\vdash\texttt{l}:\texttt{int list}}\ \text{var}\qquad\dfrac{}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list}\}\vdash\texttt{0}:\texttt{int}}\ \text{intLit}\qquad\mathcal{D}_2}{\{\texttt{sum}:\texttt{int list}\to\texttt{int},\texttt{l}:\texttt{int list}\}\vdash\texttt{match l with | [] -> 0 | h :: t -> h + sum t}:\texttt{int}}\ \text{listMatch}$$

Final derivation:

$$\dfrac{\mathcal{D}_3\qquad\dfrac{}{\{\texttt{sum}:\texttt{int list}\to\texttt{int}\}\vdash\texttt{sum}:\texttt{int list}\to\texttt{int}}\ \text{var}}{\varnothing\vdash\texttt{let rec sum l = match l with | [] -> 0 | h :: t -> h + sum t in sum}:\texttt{int list}\to\texttt{int}}\ \text{letRec}$$

# Semantic Derivation

$$\dfrac{\dfrac{\dfrac{}{1 \Downarrow 1}\ \text{intLitEval} \qquad \dfrac{\dfrac{}{2 \Downarrow 2}\ \text{intLitEval} \qquad \dfrac{}{[] \Downarrow \varnothing}\ \text{nilEval}}{2 \,::\, [] \Downarrow [2]}\ \text{consEval}}{1 \,::\, 2 \,::\, [] \Downarrow [1,2]}\ \text{consEval} \qquad \dfrac{}{\texttt{false} \Downarrow \bot}\ \text{falseEval}}{\texttt{match 1 :: 2 :: [] with | [] → true | h :: t → false} \Downarrow \bot}\ \text{matchListEvalCons}$$