Homework # 6 **SOLUTION**
due April 1, 13:00

As usual, turn in your answers to written problems (§3,4) on paper in lecture, and send your SASyLF proofs (§2) as a file `sumprod.slf` to `scmalte@inf.ethz.ch`.

1 Reading

Please read Chapter 11 in your textbook.

2 Proofs

Modify the proofs of type soundness of the simply-typed lambda calculus with "unit" types to add sums (See Figure 11-9). A skeleton file is available which gives the syntax. Your evaluation and types rules should follow the book as closely as possible.

3 Problems

Please do the following problems

1. Enums can be seen as a variation on "variants". Write new evaluation and typing rules (on paper, not SASyLF) for a type τ written like `[red,green,blue]` with new syntax:

$$\begin{array}{rcl} \tau & ::= & \ldots \\ & | & \texttt{[}n_1\texttt{,}\ldots\texttt{,}n_m\texttt{]} \end{array}$$

$$\begin{array}{rcl} t & ::= & \ldots \\ & | & n_i \texttt{ as [}n_1\texttt{,}\ldots\texttt{,}n_m\texttt{]} \quad \text{(a new kind of value)}, \\ & | & \texttt{case } t \texttt{ of } n_1 \texttt{ -> } t_1 \texttt{ | } \ldots \texttt{ | } n_m \texttt{ -> } t_m. \end{array}$$

   The identifier n_i by itself is *not* a legal term.

   Syntax:

```
T ::= ...
   | [n1, ..., n_m]

t ::= ...
   | ni as [n1, ..., n_m]
   | case t of n1 => t_1 | ... | n_m => t_m

v ::= ...
   | ni as [n1, ..., n_m]
```

Evaluation rules:

E-Case

$$\frac{t \to t'}{\texttt{case}\ t\ \texttt{of}\ n_1\ \texttt{=>}\ t_1\ \texttt{|}\ \ldots\ \texttt{|}\ n_m\ \texttt{=>}\ t_m \to \texttt{case}\ t'\ \texttt{of}\ n_1\ \texttt{=>}\ t_1\ \texttt{|}\ \ldots\ \texttt{|}\ n_m\ \texttt{=>}\ t_m}$$

E-CaseNum

$$\frac{i \in \{1, \ldots, m\}}{\texttt{case}\ n_i\ \texttt{of}\ n_1\ \texttt{=>}\ t_1\ \texttt{|}\ \ldots\ \texttt{|}\ n_m\ \texttt{=>}\ t_m \to t_i}$$

Types rules:

T-Enum

$$\frac{i \in \{1, \ldots, m\}}{\Gamma \vdash n_i\ \texttt{as}\ [n_1, \ldots, n_m]\ :\ [n_1, \ldots, n_m]}$$

T-Case

$$\frac{\Gamma \vdash t_0 : [n_1, \ldots, n_m] \qquad \Gamma \vdash t_1 : T \qquad \ldots \qquad \Gamma \vdash t_m : T}{\Gamma \vdash \texttt{case}\ t_0\ \texttt{of}\ n_1\ \texttt{=>}\ t_1\ \texttt{|}\ \ldots\ \texttt{|}\ n_m\ \texttt{=>}\ t_m\ : T}$$

These type rules do not allow permutation of labels, and also do not check uniqueness of labels. The latter means we don't have deterministic evaluation (Why?) but doesn't impact soundness.

2. Explain informally why these type rules should be sound. (You might wish to write the SASyLF proofs for sum types first; otherwise your explanations may not be convincing!)

Progress: for T-Enum, the term is already a value. For T-Case, we use induction on t_0 to either get a value, which by the canonical forms lemma (we need a new case) must be an enum value, and we can use E-CaseNum, or if t_0 evaluates further, we can use E-Case.

Substitution: by induction, simpler than with sum types because we don't introduce any new names in this "case" construct.

Preservation: For E-Case, we use induction and T-Case. For E-CaseEnum, we invert T-Case, to get preservation for the branch chosen.

4 Literature Survey

Look at *two* recent[1] papers published in prestiguous venues[2] (if possible selected from those used in Homework #4) and indicate which of the extensions from Chapter 11 (1,2,3,4,5,6,7,8,9,10,11,12) or minor variants are used in the languages. What other extensions are not covered by this list? Skim ahead in the book to see topics that may be relevant to these other extensions.

[1] Within last 5 years

[2] Top journal or conference; ask if not sure