User-defined primitives

- Enumerations allow programmers to create their own primitive (non-decomposable) types:
  - C++
    ```c++
    enum day {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday};
    Monday prints as 1
    ```
  - Java
    ```java
    enum Day {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}
    Monday prints as Monday
    methods can be attached to enumerands
    ```
Composite types

• Cartesian products
  – tuples, records, structures

• disjoint unions
  – union, discriminated or variant records

• mappings
  – arrays, functions

• recursive types
  – lists, trees
Cartesian products

- $S \times T = \{(x,y) \mid x \text{ in } S, y \text{ in } T\}$
- Basic operations:
  - construction of a pair
  - selection of a component
- Cardinality ($#$) of a Cartesian product:
  - $\#(S \times T) = \#S \times \#T$
- Can be extended naturally beyond pairs to general n-tuples
C structures

enum Month {jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec};

struct Date {
    Month m;
    int d;
};

Date today = {mar, 6};
Pascal records

\[ \text{month} = (\text{jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec}); \]
\[ \text{date} = \text{record} \]
\[ \quad \text{m: month}; \]
\[ \quad \text{d: 1..31}; \]
\[ \end{\text{end}}; \]
\[ \text{today} : \text{date}; \]
\[ \text{today.m} := \text{mar}; \]
\[ \text{today.d} := 6; \]
ML records

Form of record:

\{ <label>:<type>, <label>:<type>, ... <label>:<type> \}

Accessing a member: 

#<label>(<record>)

datatype month = jan | feb | mar | apr | may | jun | jul | aug | sep | oct | nov | dec;
val today = \{ m=mar, d=6 \};
ML tuples

• A tuple in ML is a special type of record, whose field labels are integers starting from 1.
• ("fred",false) is a tuple whose type is reported as string * bool. This is underlyingly the record { 1="fred", 2=false }, but the (...) are special syntactic sugar.
Tuples and records in ML

- `val a = ("fred", false)` has type `string*bool`.
- `val b = {1="fred", 2=false}` has type `string*bool`, and is printed as ("fred",false).
- `val c = {one="fred", two=false}` has type `{one:string, two:bool}` and is printed as `{one="fred", two=false}`.
Making a date in ML

datatype month = jan | feb | mar | apr | may | jun | jul | aug | sep | oct | nov | dec;
datatype date = Date of month * int;

val today = Date(mar,6);

Date is a new type which is built using a tuple, a special type of record.
Accessing members of a datatype

val today = Date(mar,6);
val Date(x,y) = today;

This binds x to mar and y to 6!
ML interaction

• Showed how to define a record
• Showed that a tuple is a record with special field labels
• Showed how record member access works with #{<label>}(<record>)
• Defined datatype for month
• Showed use of datatype constructor
Homogeneous tuples

• What are they?
  – \( S^2 = S \times S \)
  – \( S^n = S \times S \times \ldots \times S \)

• Cardinalities
  – \( #(S^2) = (#S)^2 \)
  – \( #(S^n) = (#S)^n \)
  – \( #(S^0) = (#S)^0 = 1 \)

• What is \( S^0 \)?
  – It is the type Unit, whose value is ()
  – Unit corresponds to void in C, C++ and Java.