Tagged unions, *variant records*, *algebraic data types*.

exact unions which do not admit new members after initial definition, and *open unions*, which do.

```
type name =
    name1 [of type1]
  ...
  | nameN [of typeN]
```

Each `name`$_i$ is the *constructor* names and must begin with a capital letter.
The optional `type`$_i$ is its value.

With the syntax of function application give the constructor name followed by an instance of its value type, if applicable.

```
type 'a name =
  [> 'name1 [of type1]
  ...
  | 'nameN [of typeN] ] as 'a
```

It means the expression is an instance of a union type that contains the member `X` with value `Y`. More cases that are supported can be added using `|`.

... pattern matching.

```
type 'a list =
    []
  | :: of 'a * 'a list
```

`[]` is not a valid name and requires language support.

```
type 'a option =
    None
  | Some of 'a
```

Same as open unions, except with $<$ instead of $>$.