Consider the language of *propositional formulae* formed from variables \((a, b, c, \ldots)\), negation \((\neg)\), conjunction \((\land)\), and disjunction \((\lor)\), according to the following abstract syntax:

\[
\phi ::= a \\
| \neg \phi_1 \\
| \phi_1 \land \phi_2 \\
| \phi_1 \lor \phi_2
\]

We can represent propositional formulae in SML using the following datatype:

```sml
datatype prop = Var of string \\
| Not of prop \\
| And of prop * prop \\
| Or of prop * prop
```

For example, the formula \(a \land \neg(b \lor \neg c)\) is represented as the SML value

\[
\text{And}(\text{Var "a"}, \text{Not(Or(Var "b", \text{Not(Var "c")}))})
\]

We define the language of *disjunctive normal forms* as

\[
D ::= C \\
| A \lor D \\

C ::= A \\
| A \land C \\

A ::= a \\
| \neg a
\]

This language can be represented as the following SML datatype:

```sml
datatype dnf = Or of conjunct list \\
and conjunct = And of atom list \\
and atom = Var of string \\
| Not of string
```

Because we have used the same constructor names, we must put the *prop* and *dnf* types in separate modules:

```sml
structure Prop = 
struct 
  datatype prop = ...
end
```
One can convert an arbitrary formula to DNF by using the following rewrite rules:

\[
\begin{align*}
\neg(\neg \phi) & \Rightarrow \phi \\
\neg(\phi_1 \land \phi_2) & \Rightarrow \neg \phi_1 \lor \neg \phi_2 \\
\neg(\phi_1 \lor \phi_2) & \Rightarrow \neg \phi_1 \land \neg \phi_2 \\
\phi_1 \land (\phi_2 \lor \phi_3) & \Rightarrow (\phi_1 \land \phi_2) \lor (\phi_1 \land \phi_3) \\
(\phi_1 \lor \phi_2) \land \phi_3 & \Rightarrow (\phi_1 \land \phi_3) \lor (\phi_2 \land \phi_3)
\end{align*}
\]

Your assignment is to write an SML function (toDNF) that converts propositional formulae to their equivalent DNF. It should have the following signature:

\[
\text{val toDNF : Prop.prop -> DNF.dnf}
\]

You solution should consist of four files: prop.sml (holding the module Prop), dnf.sml (holding the module DNF), convert.sml (holding the Convert module, which contains the toDNF function), and hw1.cm (containing the CM specification). Please ensure that your name appears in a comment at the beginning of each file.

The CM specification should be as follows:

\[
\textbf{Library}\n\begin{align*}
\text{structure Prop} \\
\text{structure DNF} \\
\text{structure Convert}
\end{align*}
\]

\[
\text{is} \\
$/$basis.cm \\
prop.sml \\
dnf.sml \\
convert.sml
\]