pvs2C functions:

```
defmethod pvs2C* ((expr number-expr) bindings livevars result

defmacro pvs2C_tuple (args)

defmethod pvs2C* ((expr tuple-expr) bindings livevars result)

defmethod pvs2C* ((expr record-expr) bindings livevars result)

defmethod pvs2C* ((expr projection-application) bindings livevars result)

defmethod pvs2C*  ((expr field-application) bindings livevars result)

defmethod pvs2C* ((expr list) bindings livevars result)

defmethod pvs2C* ((expr application) bindings livevars result)

defun pvs2C-primitive-app (expr bindings livevars)
defun pvs2C-defn-application (expr bindings livevars)
defun pvs2C-resolution (op)
defun pvs2C-declaration (op-decl)
defun pvs2C-resolution-nondestructive (op-decl formals body range-type)
defun pvs2C-resolution-destructive (op-decl formals body range-type)

defmethod pvs2C* ((expr name-expr) bindings livevars result)

defun pvs2C-constant (expr op-decl bindings livevars)

defun pvs2C-lambda (bind-decls expr bindings)
defmethod pvs2C* ((expr lambda-expr) bindings livevars result)

defmethod pvs2C* ((expr cases-expr) bindings livevars result)
defun pvs2C-cases (selections else-part bindings livevars)

defmethod pvs2C* ((expr update-expr) bindings livevars result)

defmethod pvs2C-type ((type recordtype) &optional tbindings)
defmethod pvs2C-type ((type tupletype) &optional tbindings)
defmethod pvs2C-type ((type funtype) &optional tbindings)
defmethod pvs2C-type ((type subtype) &optional tbindings)
defmethod pvs2C-type ((type type-name) &optional tbindings)


defun pvs2C-theory (theory)
defun pvs2C-datatype (dt)
defun pvs2C-datatype-formal (formal dt)
```

```
defun generate-C-for-pvs-file (filename &optional force?)
```