• Homework Q4 -
  you may include or exclude zero (I include it)

• Production 2 in CSX:
  memberdecls → fielddecl memberdecls

  need an alternative approach
  since memberdecls has no
  constructor with fielddecl parameter

solution - do some ast editing
You have two AST subtrees

One small AST for one FieldDecl

One big AST for all Decs to the right of the Field Decl

Connect them together

\[
\text{FieldDecl} \rightarrow \text{FieldDecs} \rightarrow \text{MemberDecs} \rightarrow \text{NewFieldDecs} \rightarrow \text{Method Decs}
\]
- Resolving IF-THEN/IF-THEN-ELSE Ambiguity

1. Rewrite CSX Grammar so that a stmt to the left of an else can't generate an if then

```
stmt
\_/\
|   |
| IF expr stmt else stmt |

Not allowed: ...
```
2. Use CUP's -EXPECT option to force a shift in a shift/reduce error.  
See modified BUILD.XMT target for CUP.
<target name="Cup" depends="checkCupNeeded" unless="CupnotNeeded">
  <mkdir dir="bin"/>
  <java classpath="src;bin:" classname="java_cup.Main" fork="true" input="csx.cup">
    <arg value="-expect"/>
    <arg value="1"/>
    </java>
  <move file="parser.java" todir="src"/>
  </move>
  <move file="sym.java" todir="src"/>
  </move>
  <javac includeantruntime="false" srcdir="src" destdir="bin"
    classpath="." includes="sym.java" debug="on"/>
  </javac>
</target>
PARSING EXAMPLE - CSX LITE

TOKEN STREAM:

... INT SUM ; ...

PARSER AUTOMATICALLY DECIDES WHICH PRODUCTION TO APPLY

STEP 1: MATCH TYPE -> INT
BUILD INTTYPE NODE
LINK NODE TO TYPE

STEP 2: MATCH IDENT -> IDENTIFIER
BUILD IDENT NODE ("SUM")
LINK TO IDENT

STEP 3: MATCH FIELDDECL -> TYPE IDENT ;
BUILD VARDECL NODE (INTTYPE NODE, IDENT NODE)
LINK TO FIELDDECL