

language Robot {

```
attributes int *.inx; int *.iny;  
           int *.outx; int *.outy;
```

Attribute definition

lexicon {

```
  ReservedWord left | right | up | down | begin | end | drek  
  ignore [ \0x0D\0x0A\ ]
```

Lexical part of language specifications defined with named regular expressions (regular definitions). From that part LISA generates Scanner (Java source code).

rule start {

```
  START ::= begin COMMANDS end com  
  START.outx = COMMANDS.outx;  
  START.outy = COMMANDS.outy;  
  COMMANDS.inx =  
  COMMANDS.iny = 0;  
};
```

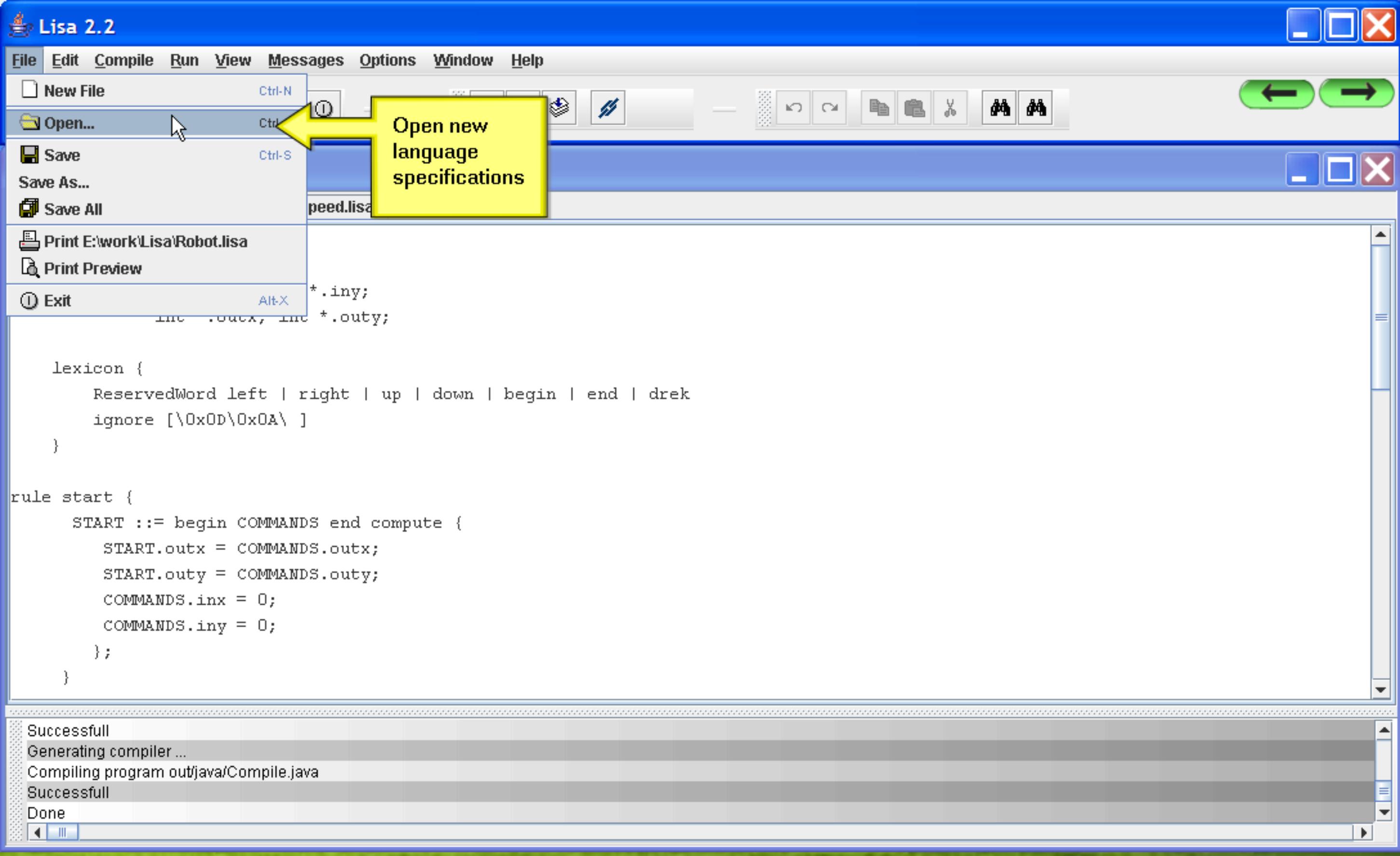
Syntax (BNF) and semantics specifications (attribute grammars).

LISA is a compiler-compiler, or a system that generates automatically a compiler/interpreter from formal attribute grammar based language specifications. LISA tool produces highly efficient source code of scanner/parser/interpreter/compiler in Java. Lexical and syntax part of specifications uses well known formal methods: regular expressions and BNF. Semantics is further defined with attribute grammars.

Autocomplete capable editor (Ctrl + space -> code insight)

```
Successfull  
Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Done
```

Error reporting + info

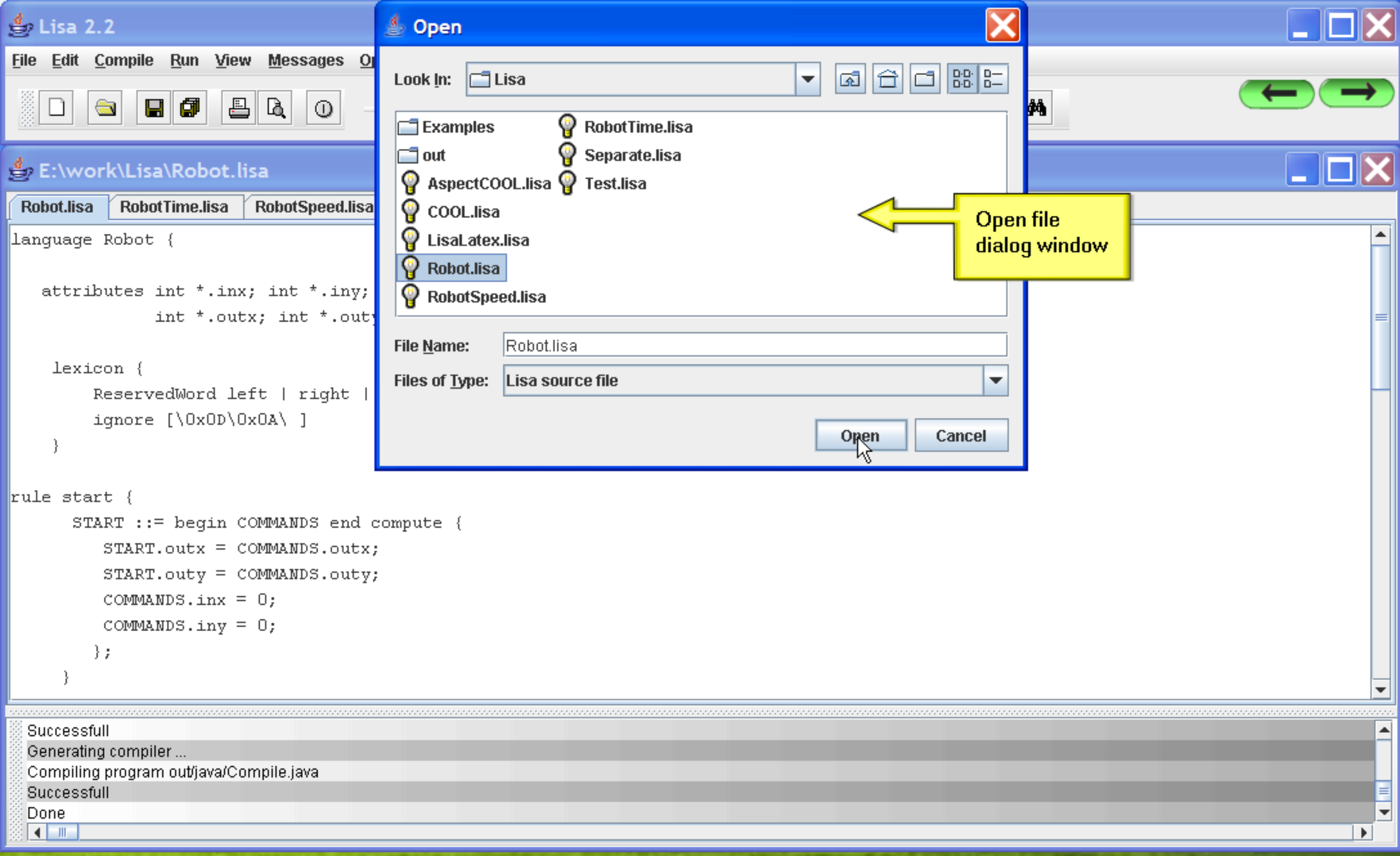


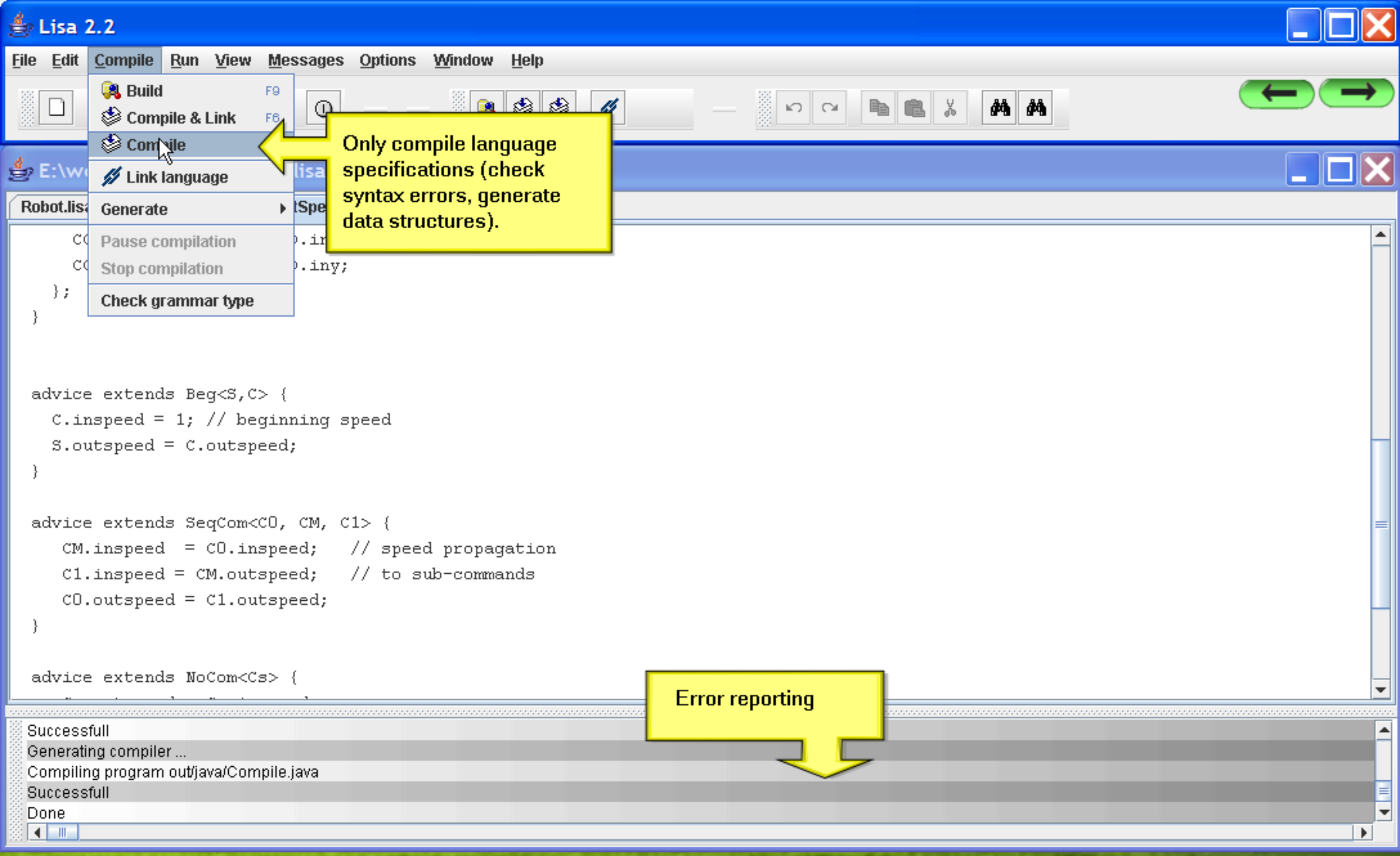
Open new language specifications

- New File (Ctrl-N)
- Open... (Ctrl-O)
- Save (Ctrl-S)
- Save As...
- Save All
- Print E:\work\Lisa\Robot.lisa
- Print Preview
- Exit (Alt-X)

```
*.iny;  
inc .outx, inc *.outy;  
  
lexicon {  
  ReservedWord left | right | up | down | begin | end | drek  
  ignore [\0x0D\0x0A\ ]  
}  
  
rule start {  
  START ::= begin COMMANDS end compute {  
    START.outx = COMMANDS.outx;  
    START.outy = COMMANDS.outy;  
    COMMANDS.inx = 0;  
    COMMANDS.iny = 0;  
  };  
}
```

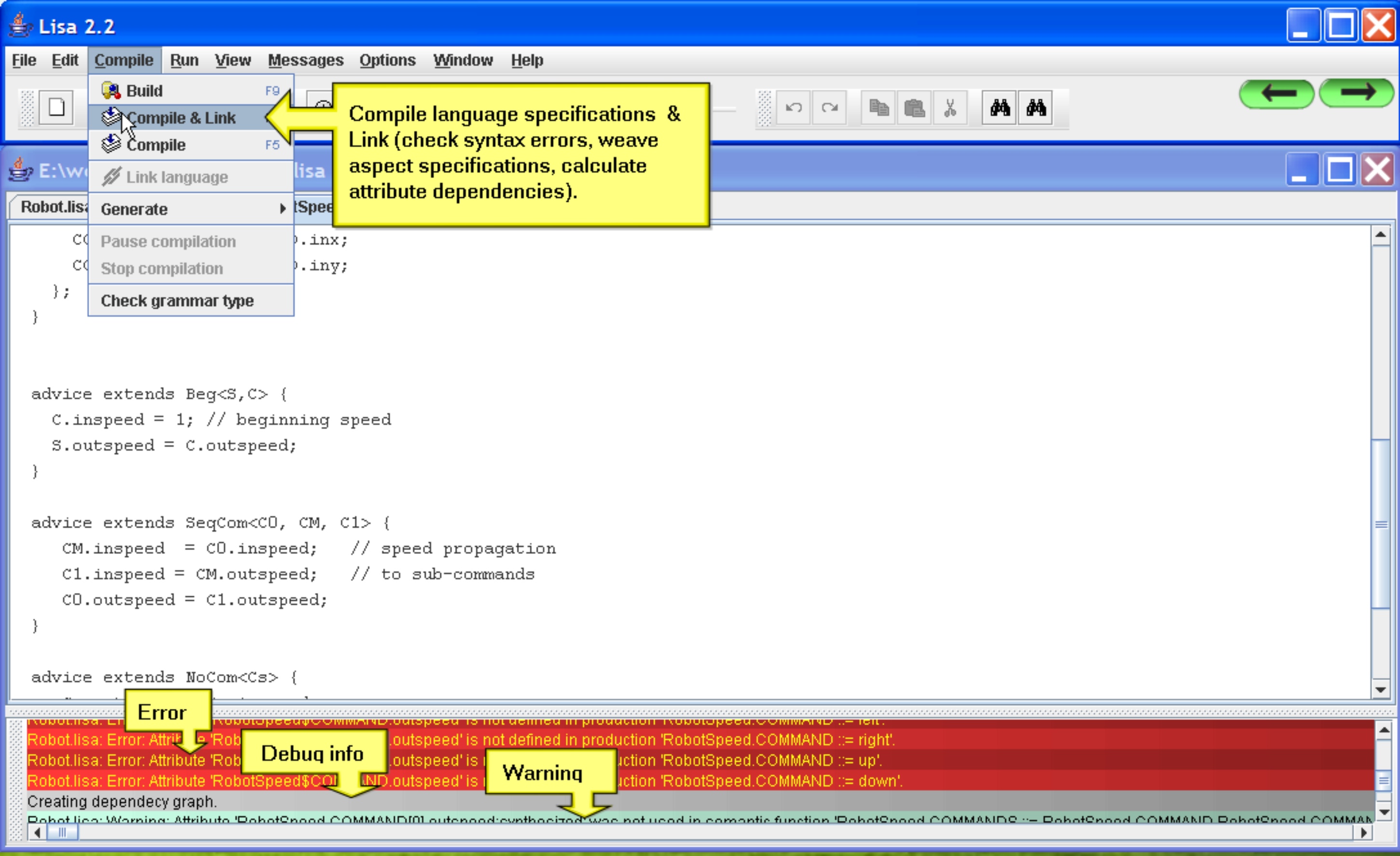
Successfull  
Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Done





Only compile language specifications (check syntax errors, generate data structures).

Error reporting



- Build F9
- Compile & Link**
- Compile F5
- Link language
- Generate
- Pause compilation
- Stop compilation
- Check grammar type

Compile language specifications & Link (check syntax errors, weave aspect specifications, calculate attribute dependencies).

```
CO .inx;  
CO .iny;  
};  
}
```

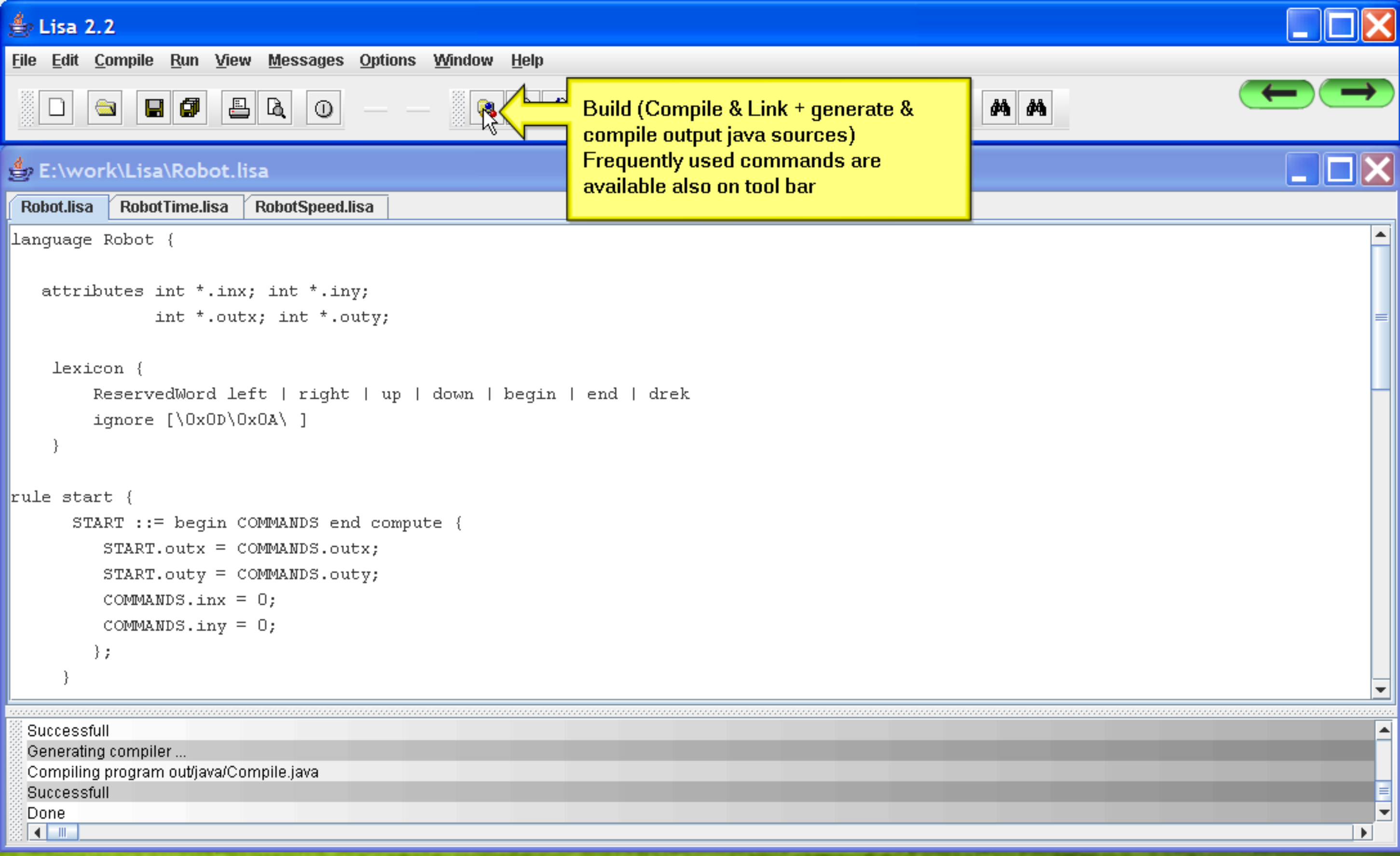
```
advice extends Beg<S,C> {  
  C.inspeed = 1; // beginning speed  
  S.outspeed = C.outspeed;  
}  
  
advice extends SeqCom<CO, CM, C1> {  
  CM.inspeed = CO.inspeed; // speed propagation  
  C1.inspeed = CM.outspeed; // to sub-commands  
  CO.outspeed = C1.outspeed;  
}  
  
advice extends NoCom<Cs> {
```

**Error**

**Debug info**

**Warning**

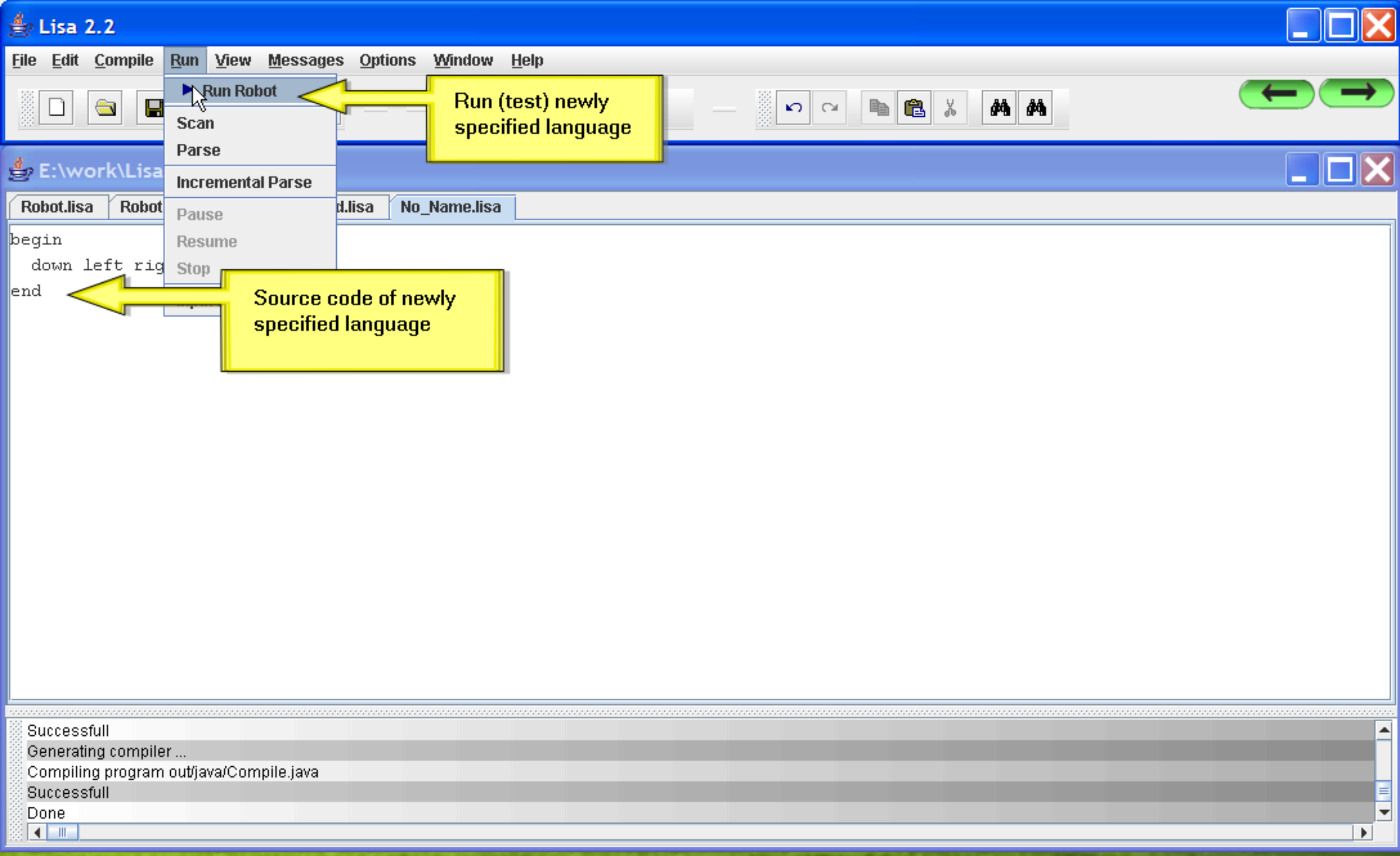
Robot.lisa: Error: Attribute 'RobotSpeed\$COMMAND.outspeed' is not defined in production 'RobotSpeed.COMMAND ::= left'.  
Robot.lisa: Error: Attribute 'RobotSpeed\$COMMAND.outspeed' is not defined in production 'RobotSpeed.COMMAND ::= right'.  
Robot.lisa: Error: Attribute 'RobotSpeed\$COMMAND.outspeed' is not defined in production 'RobotSpeed.COMMAND ::= up'.  
Robot.lisa: Error: Attribute 'RobotSpeed\$COMMAND.outspeed' is not defined in production 'RobotSpeed.COMMAND ::= down'.  
Creating dependency graph.  
Robot.lisa: Warning: Attribute 'RobotSpeed.COMMAND.outspeed: synthesized' was not used in semantic function 'RobotSpeed.COMMANDS ::= RobotSpeed.COMMAND RobotSpeed.COMMAND'



Build (Compile & Link + generate & compile output java sources)  
Frequently used commands are available also on tool bar

```
language Robot {  
  
  attributes int *.inx; int *.iny;  
             int *.outx; int *.outy;  
  
  lexicon {  
    ReservedWord left | right | up | down | begin | end | drek  
    ignore [ \0x0D\0x0A\ ]  
  }  
  
  rule start {  
    START ::= begin COMMANDS end compute {  
      START.outx = COMMANDS.outx;  
      START.outy = COMMANDS.outy;  
      COMMANDS.inx = 0;  
      COMMANDS.iny = 0;  
    };  
  }  
}
```

Successfull  
Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Done



Lisa 2.2

File Edit Compile Run View Messages Options Window Help

Run Robot

Run (test) newly specified language

Scan

Parse

Incremental Parse

Pause

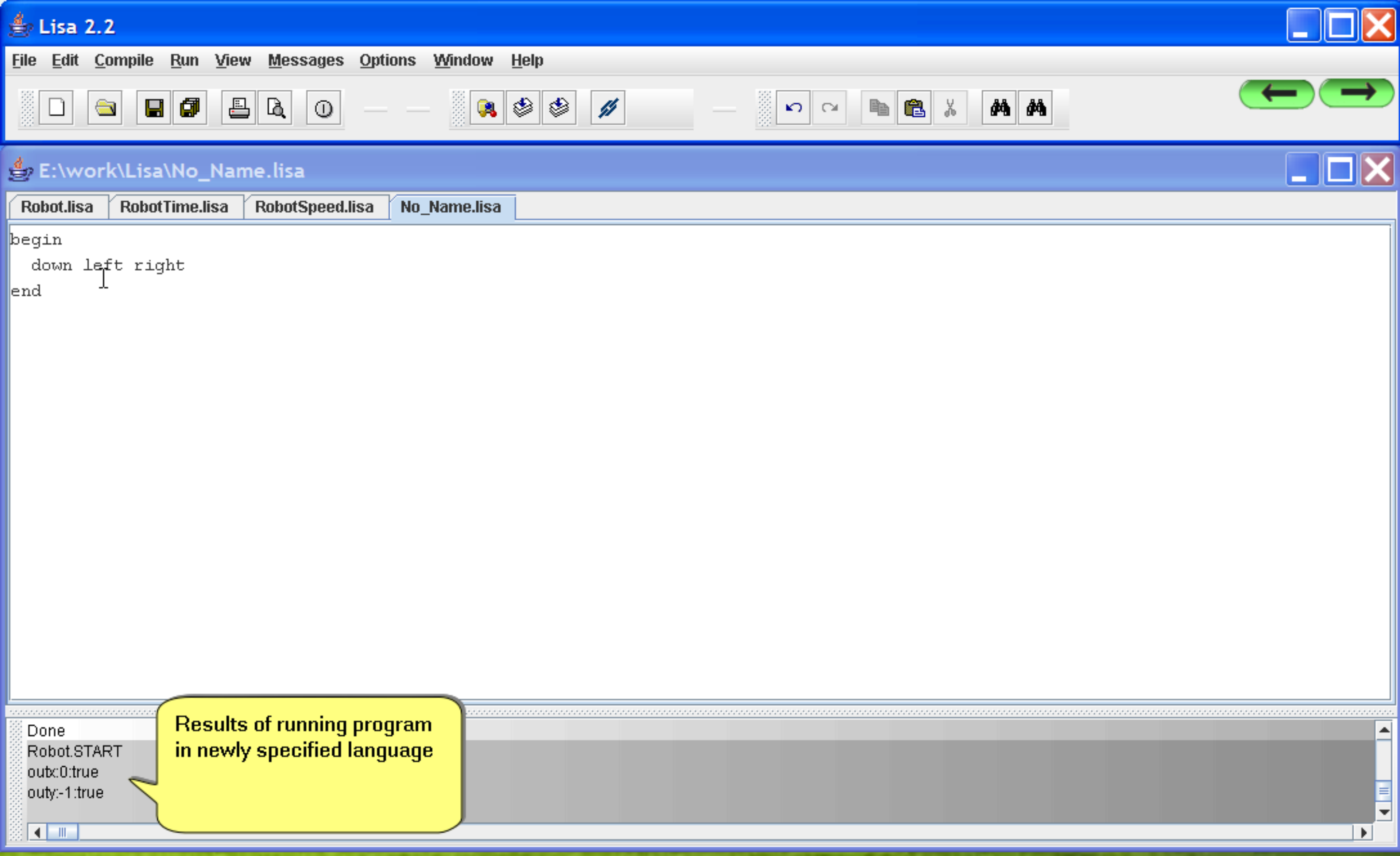
Resume

Stop

Source code of newly specified language

```
begin
  down left rig
end
```

Successfull  
Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Done



Lisa 2.2

File Edit Compile Run View Messages Options Window Help



E:\work\Lisa\No\_Name.lisa

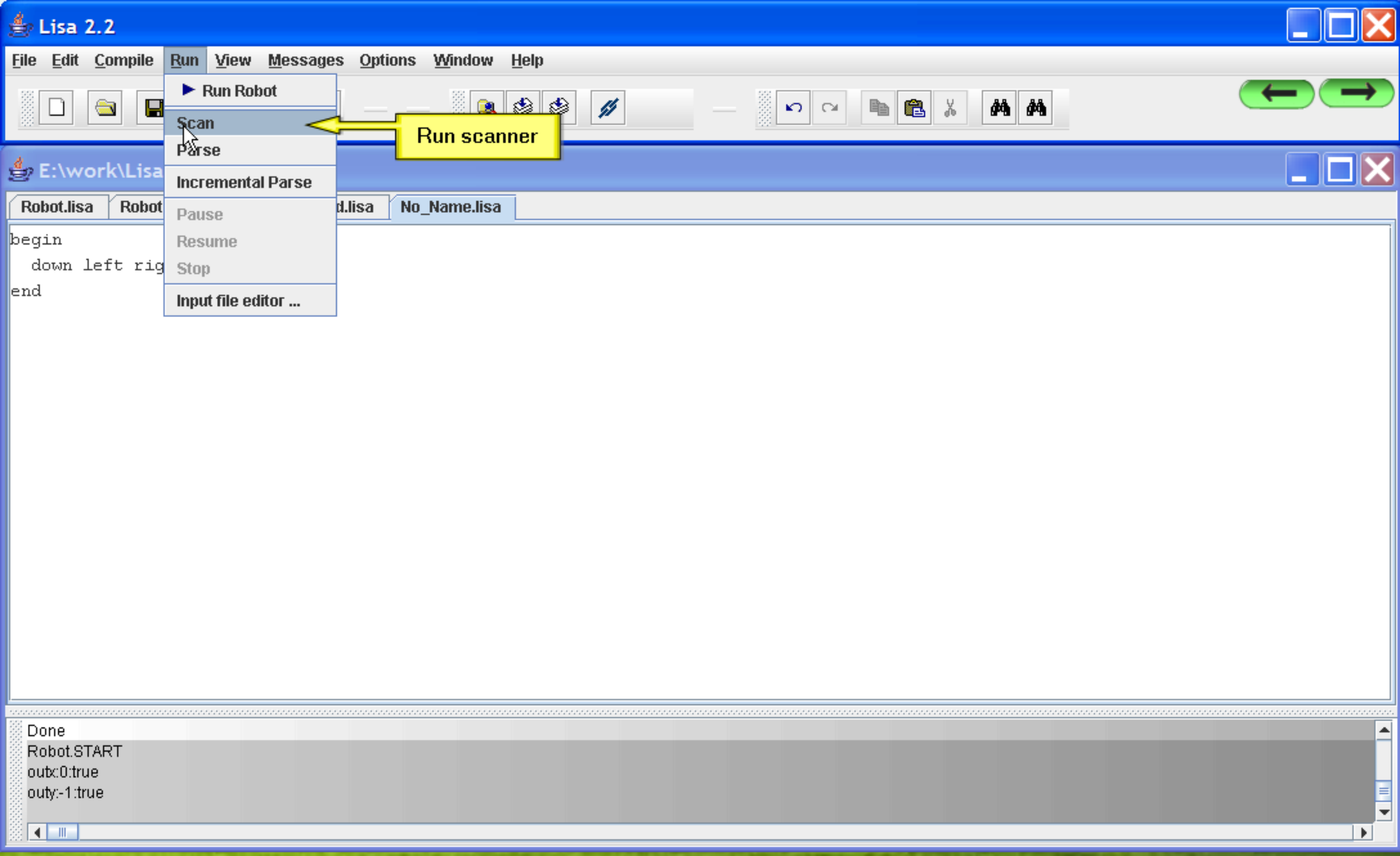
Robot.lisa RobotTime.lisa RobotSpeed.lisa No\_Name.lisa

```
begin
  down left right
end
```

```
Done
Robot.START
out:0:true
out:-1:true
```

Results of running program in newly specified language





- ▶ Run Robot
- Scan
- Parse
- Incremental Parse
- Pause
- Resume
- Stop
- Input file editor ...

Run scanner

```
begin
  down left rig
end
```

```
Done
Robot.START
out:0:true
outy:-1:true
```

**Lisa 2.2** [Window Controls]

File Edit Compile Run View Messages Options Window Help

[Icons: File, Folder, Disk, Print, Magnifying Glass, Clock, Undo, Redo, Copy, Paste, Erase, Undo, Redo, Copy, Paste, Erase]

[Navigation: Left Arrow, Right Arrow]

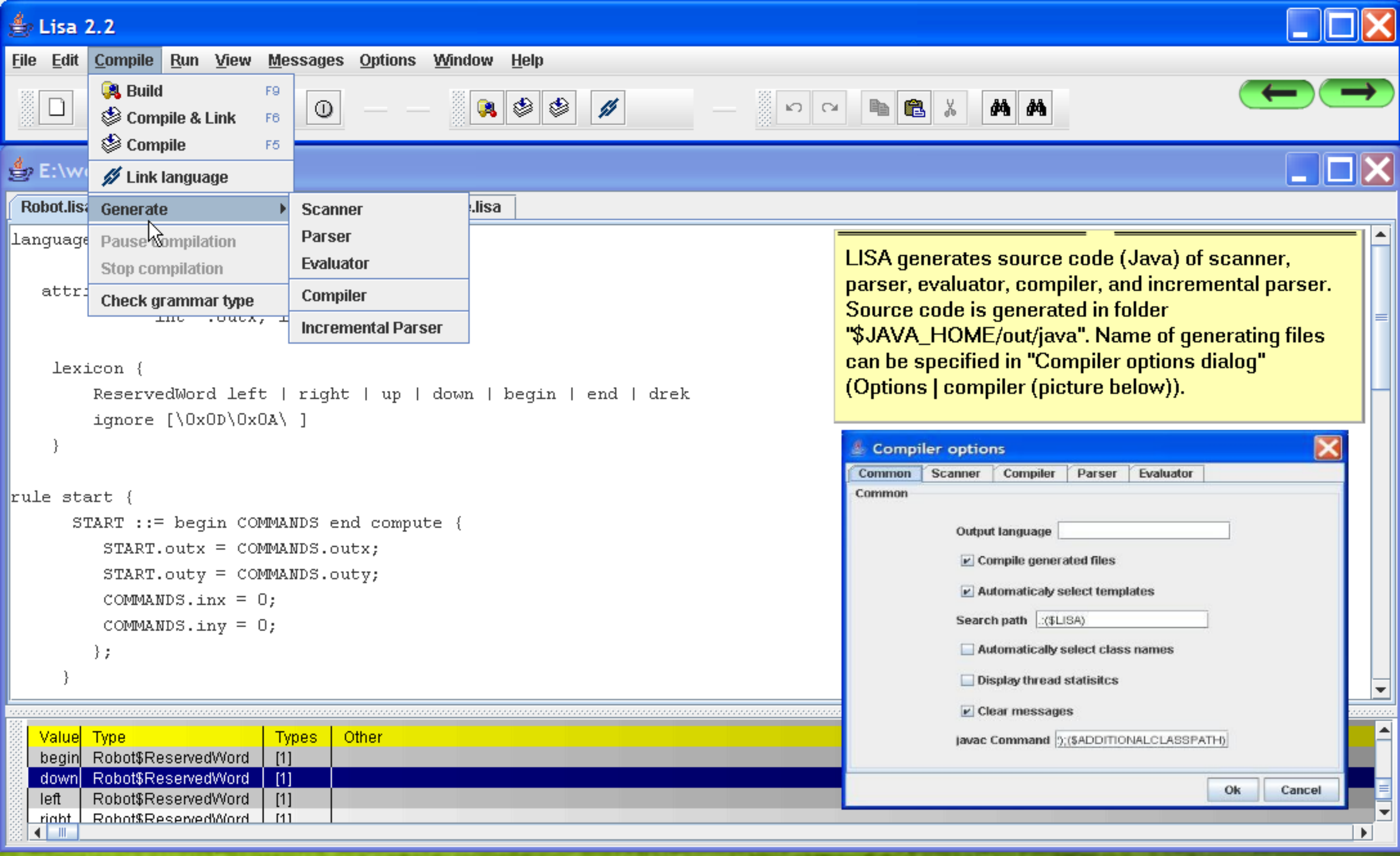
**E:\work\Lisa\No\_Name.lisa** [Window Controls]

Robot.lisa RobotTime.lisa RobotSpeed.lisa **No\_Name.lisa**

```
begin
  down left right
end
```

**Results of scanning**

Value	Type	Types	Other
begin	Robot\$ReservedWord	[1]	
down	Robot\$ReservedWord	[1]	
left	Robot\$ReservedWord	[1]	
right	Robot\$ReservedWord	[1]	



LISA generates source code (Java) of scanner, parser, evaluator, compiler, and incremental parser. Source code is generated in folder "\$JAVA\_HOME/out/java". Name of generating files can be specified in "Compiler options dialog" (Options | compiler (picture below)).

**Compiler options**

Common Scanner Compiler Parser Evaluator

Common

Output language

Compile generated files

Automatically select templates

Search path

Automatically select class names

Display thread statistics

Clear messages

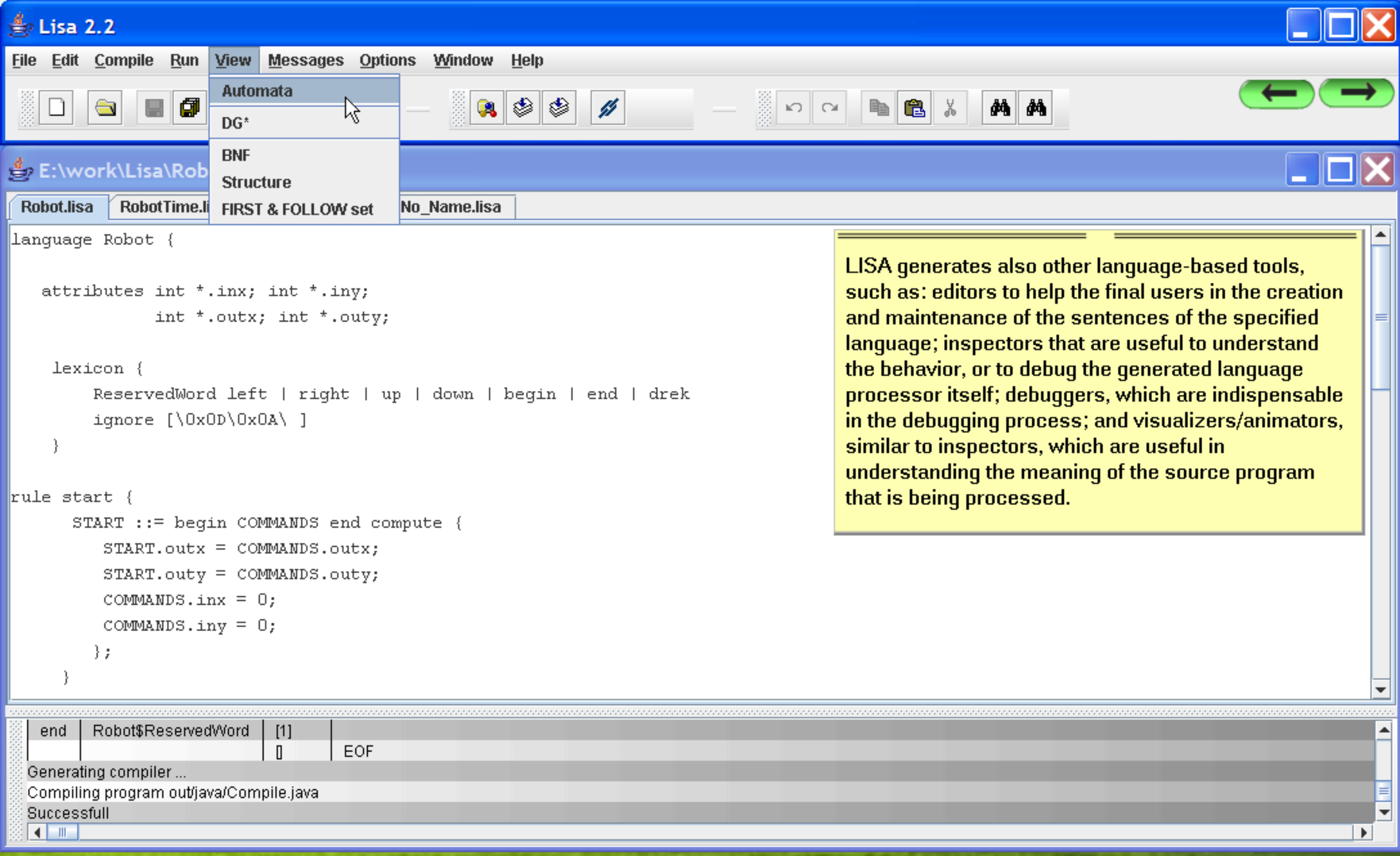
javac Command

Ok Cancel

```
lexicon {
  ReservedWord left | right | up | down | begin | end | drek
  ignore [ \0x0D\0x0A\ ]
}

rule start {
  START ::= begin COMMANDS end compute {
    START.outx = COMMANDS.outx;
    START.outy = COMMANDS.outy;
    COMMANDS.inx = 0;
    COMMANDS.iny = 0;
  };
}
```

Value	Type	Types	Other
begin	Robot\$ReservedWord	[1]	
down	Robot\$ReservedWord	[1]	
left	Robot\$ReservedWord	[1]	
right	Robot\$ReservedWord	[1]	



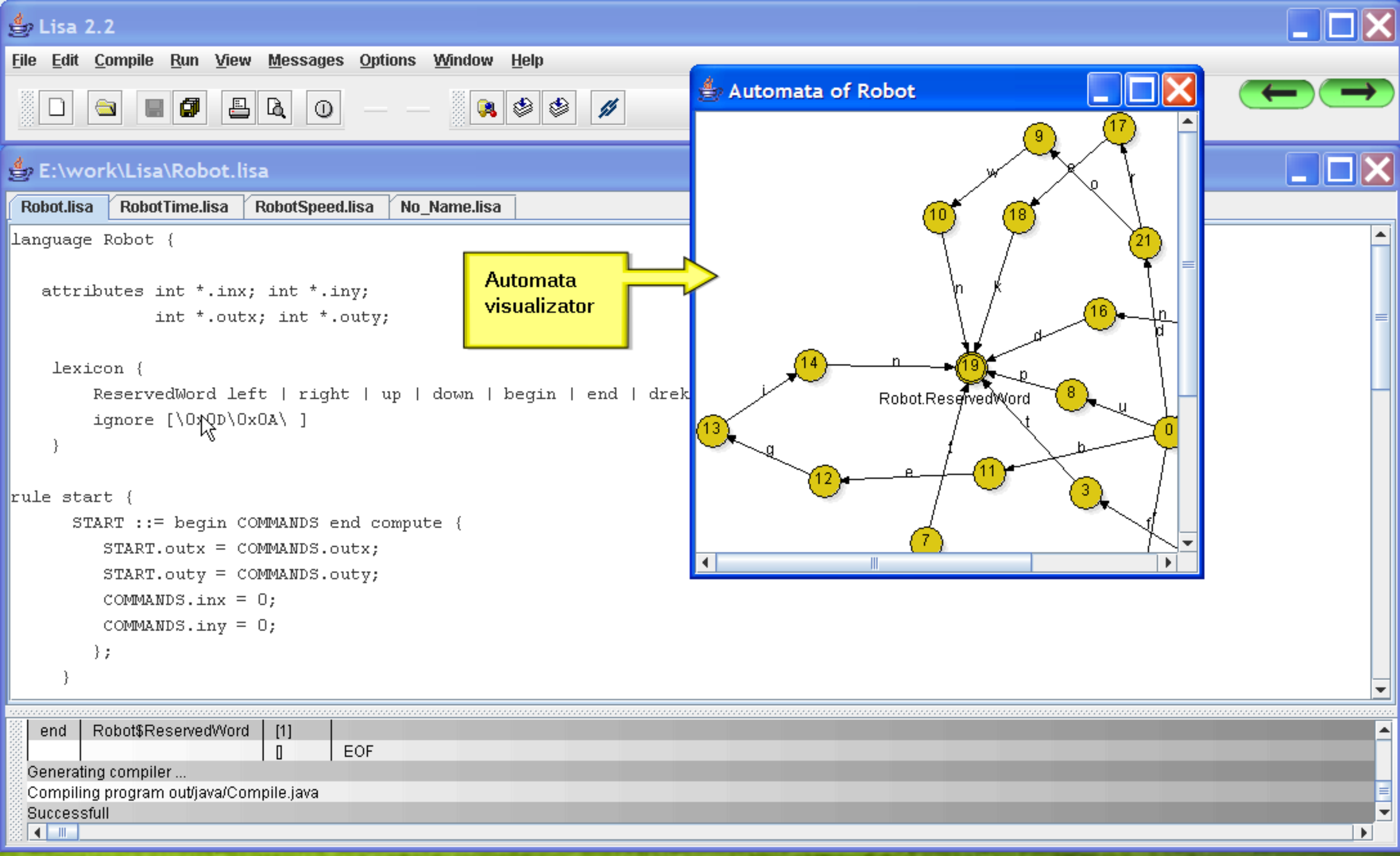
- Automata
- DG\*
- BNF
- Structure
- FIRST & FOLLOW set

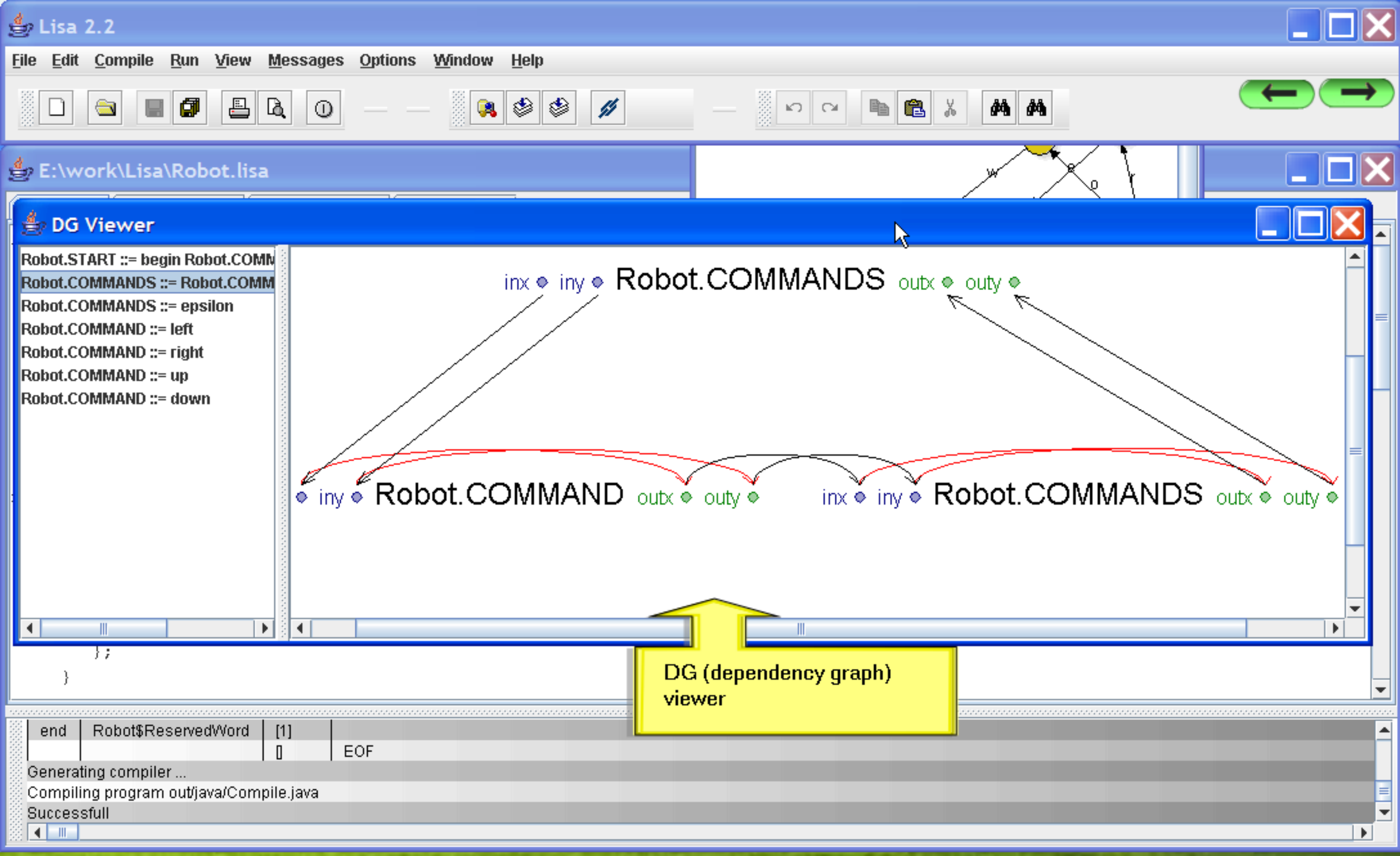
```
language Robot {  
  
  attributes int *.inx; int *.iny;  
             int *.outx; int *.outy;  
  
  lexicon {  
    ReservedWord left | right | up | down | begin | end | drek  
    ignore [ \0x0D\0x0A\ ]  
  }  
  
  rule start {  
    START ::= begin COMMANDS end compute {  
      START.outx = COMMANDS.outx;  
      START.outy = COMMANDS.outy;  
      COMMANDS.inx = 0;  
      COMMANDS.iny = 0;  
    };  
  }  
}
```

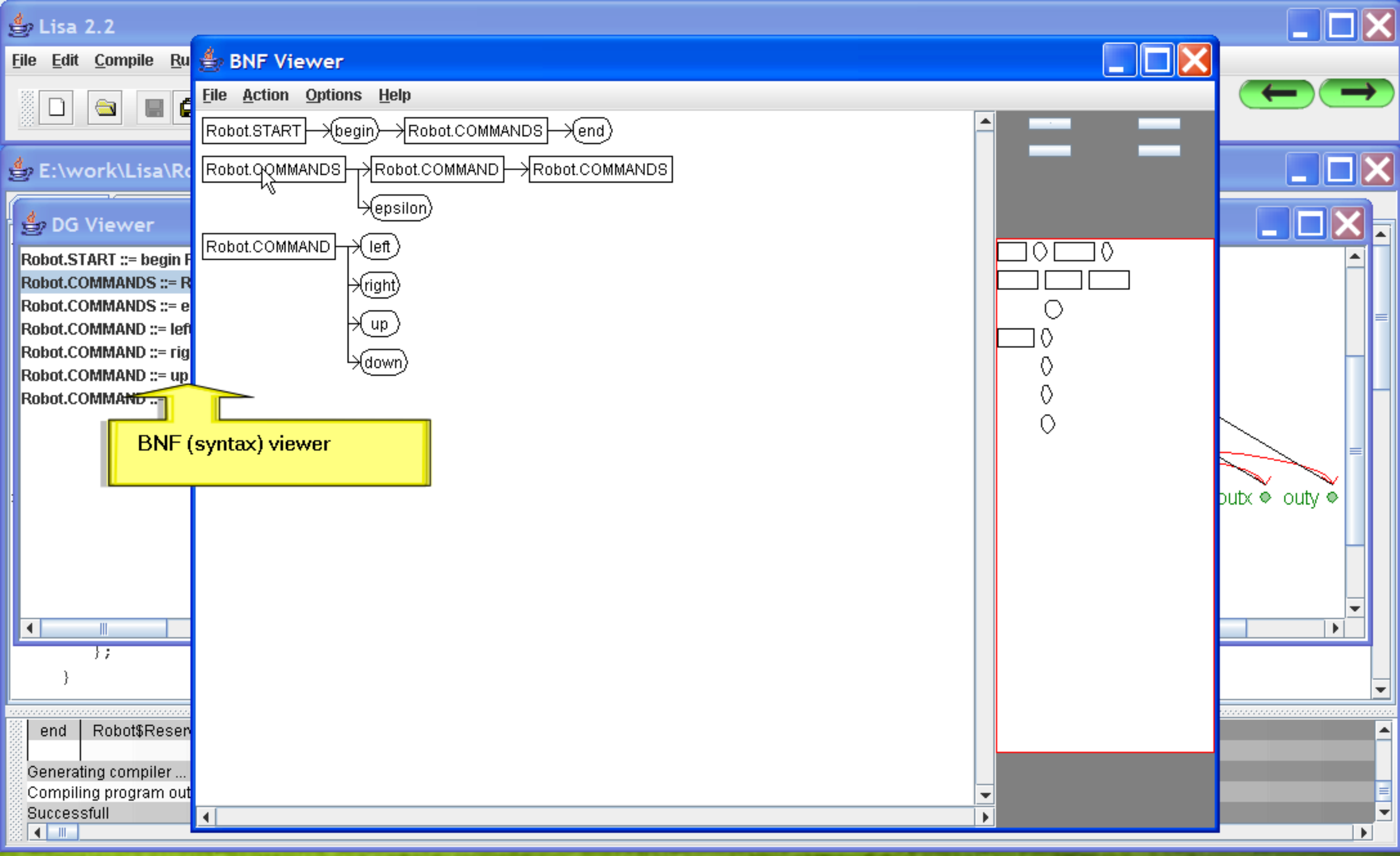
LISA generates also other language-based tools, such as: editors to help the final users in the creation and maintenance of the sentences of the specified language; inspectors that are useful to understand the behavior, or to debug the generated language processor itself; debuggers, which are indispensable in the debugging process; and visualizers/animators, similar to inspectors, which are useful in understanding the meaning of the source program that is being processed.

end	Robot\$ReservedWord	[1]	EOF
		0	

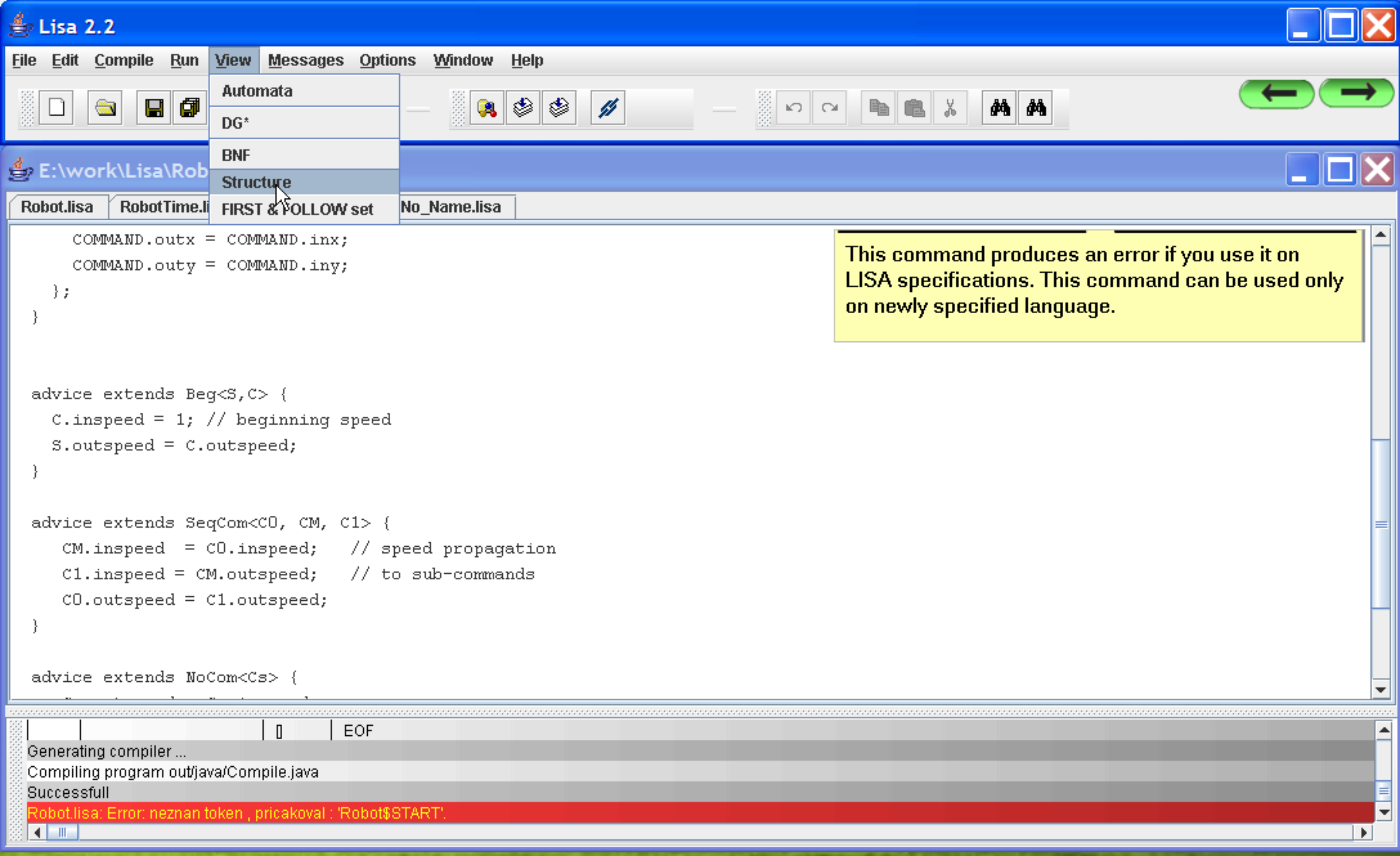
Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull







BNF (syntax) viewer



- Automata
- DG\*
- BNF
- Structure
- FIRST & FOLLOW set

```
COMMAND.outx = COMMAND.inx;
COMMAND.outy = COMMAND.iny;
};
}

advice extends Beg<S,C> {
  C.inspeed = 1; // beginning speed
  S.outspeed = C.outspeed;
}

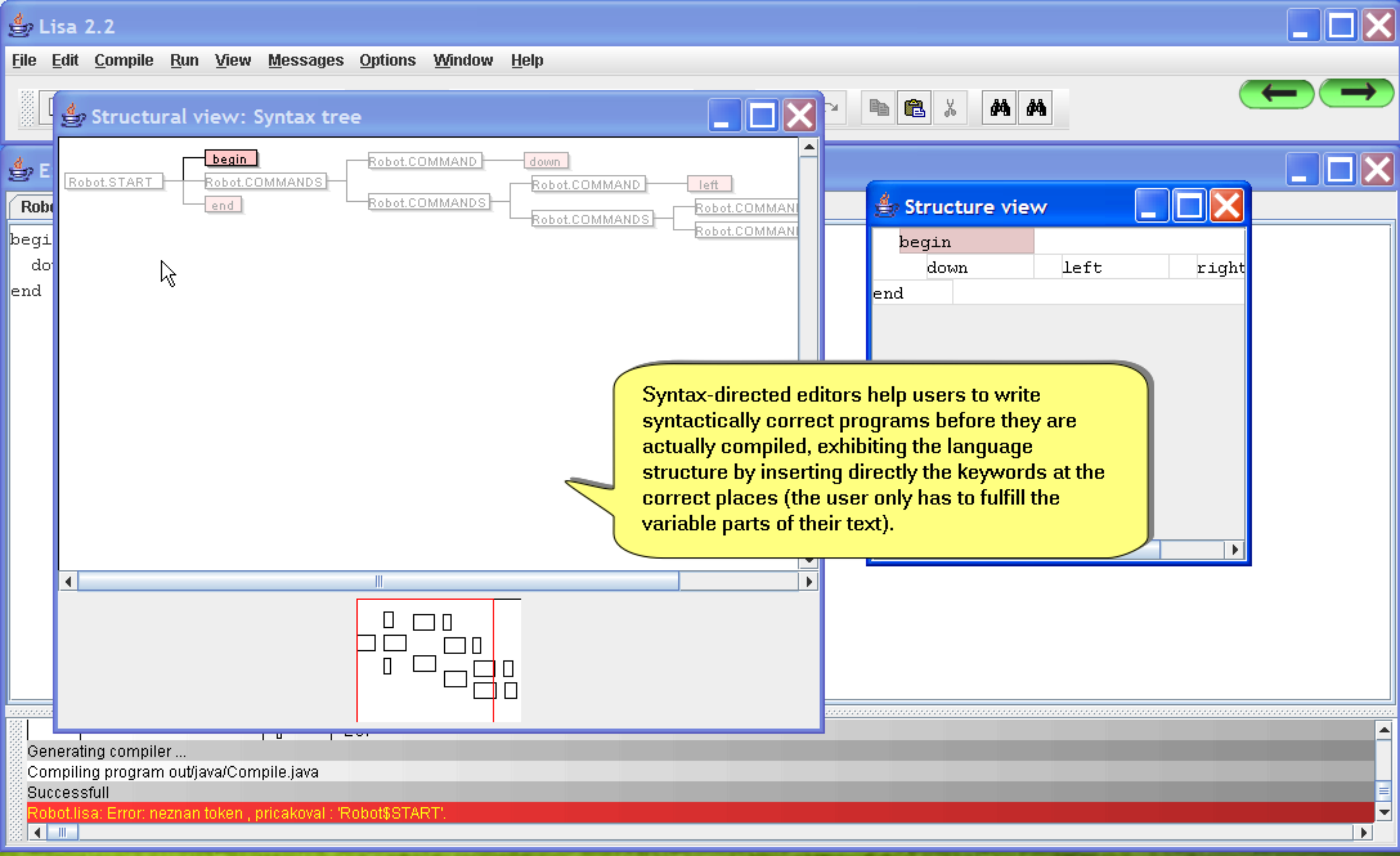
advice extends SeqCom<C0, CM, C1> {
  CM.inspeed = C0.inspeed; // speed propagation
  C1.inspeed = CM.outspeed; // to sub-commands
  C0.outspeed = C1.outspeed;
}

advice extends NoCom<Cs> {
```

This command produces an error if you use it on LISA specifications. This command can be used only on newly specified language.

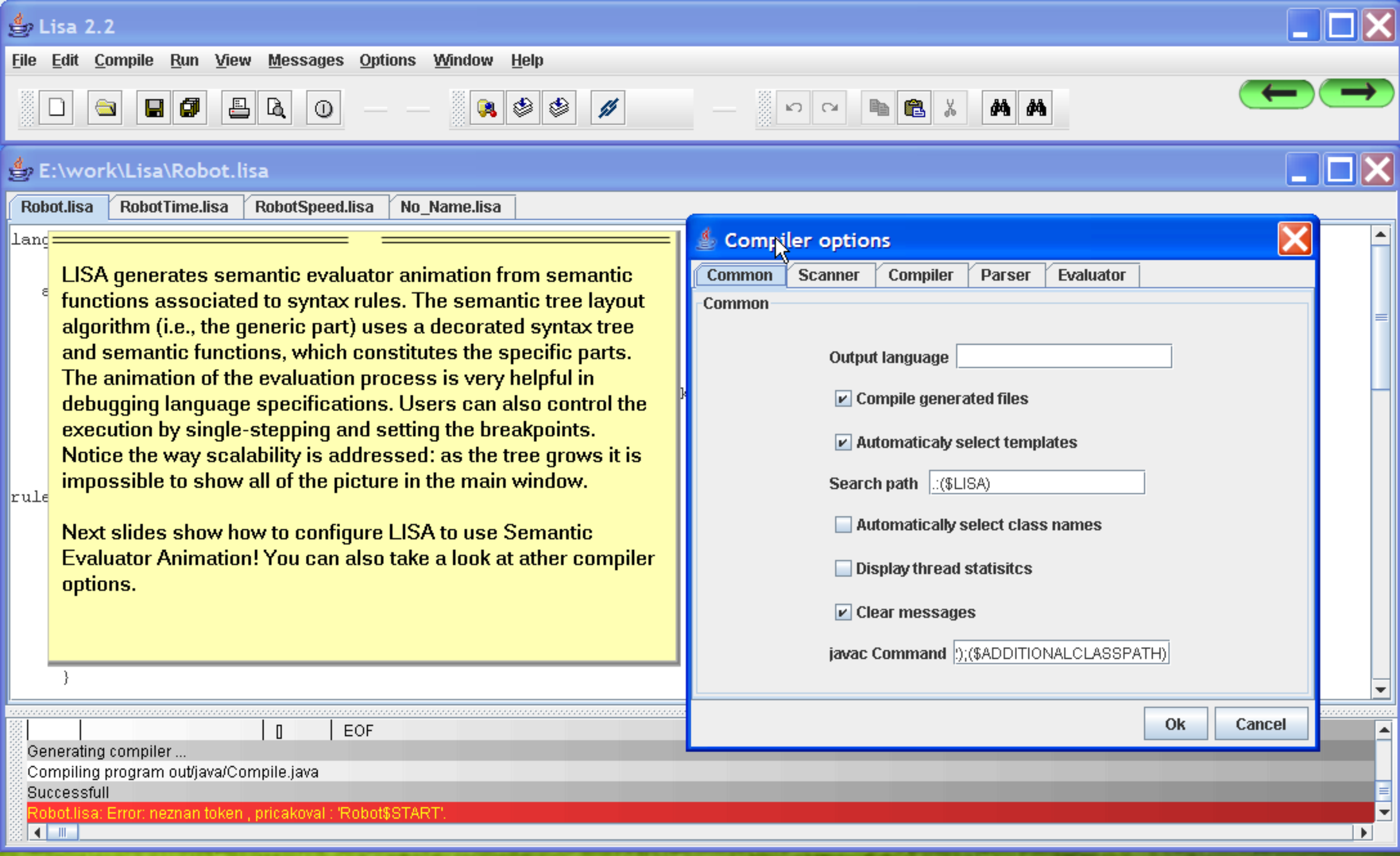
Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'





Syntax-directed editors help users to write syntactically correct programs before they are actually compiled, exhibiting the language structure by inserting directly the keywords at the correct places (the user only has to fulfill the variable parts of their text).

Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'



LISA generates semantic evaluator animation from semantic functions associated to syntax rules. The semantic tree layout algorithm (i.e., the generic part) uses a decorated syntax tree and semantic functions, which constitutes the specific parts. The animation of the evaluation process is very helpful in debugging language specifications. Users can also control the execution by single-stepping and setting the breakpoints. Notice the way scalability is addressed: as the tree grows it is impossible to show all of the picture in the main window.

Next slides show how to configure LISA to use Semantic Evaluator Animation! You can also take a look at other compiler options.

### Compiler options

Common Scanner Compiler Parser Evaluator

Common

Output language

Compile generated files

Automatically select templates

Search path

Automatically select class names

Display thread statistics

Clear messages

javac Command

Ok Cancel

EOF

Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull

Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'

Lisa 2.2

File Edit Compile Run View Messages Options Window Help

```
language Robot {  
  
  attributes int *.inx; int *.iny;  
             int *.outx; int *.outy;  
  
  lexicon {  
    ReservedWord left | right | up | down | begin | end | drek  
    ignore [ \0x0D\0x0A\ ]  
  }  
  
  rule start {  
    START ::= begin COMMANDS end compute {  
      START.outx = COMMANDS.outx;  
      START.outy = COMMANDS.outy;  
      COMMANDS.inx = 0;  
      COMMANDS.iny = 0;  
    };  
  }  
}
```

### Compiler options

Common Scanner Compiler Parser Evaluator

Scanner

Regular def. Lang.rdf

Generate CScanner.java

Template :S)/Scanner/Scanner.template

Compress Tables

Allow Multiple Scanners

Ok Cancel

Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'

Lisa 2.2

File Edit Compile Run View Messages Options Window Help

E:\work\Lisa\Robot.lisa

Robot.lisa RobotTime.lisa RobotSpeed.lisa No\_Name.lisa

```
language Robot {  
  
  attributes int *.inx; int *.iny;  
             int *.outx; int *.outy;  
  
  lexicon {  
    ReservedWord left | right | up | down | begin | end | drek  
    ignore [ \0x0D\0x0A\ ]  
  }  
  
  rule start {  
    START ::= begin COMMANDS end compute {  
      START.outx = COMMANDS.outx;  
      START.outy = COMMANDS.outy;  
      COMMANDS.inx = 0;  
      COMMANDS.iny = 0;  
    };  
  }  
}
```

### Compiler options

Common Scanner **Compiler** Parser Evaluator

Compiler

Generate

Template

Ok Cancel

Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'

Lisa 2.2

File Edit Compile Run View Messages Options Window Help

```
language Robot {  
  
  attributes int *.inx; int *.iny;  
             int *.outx; int *.outy;  
  
  lexicon {  
    ReservedWord left | right | up | down | begin | end | drek  
    ignore [ \0x0D\0x0A\ ]  
  }  
  
  rule start {  
    START ::= begin COMMANDS end compute {  
      START.outx = COMMANDS.outx;  
      START.outy = COMMANDS.outy;  
      COMMANDS.inx = 0;  
      COMMANDS.iny = 0;  
    };  
  }  
}
```

### Compiler options

Common Scanner Compiler **Parser** Evaluator

Parser

- BNF Lang.bnf
- Generate CParser.java
- Template .TES)/Parser/LR/LR1.template
- Parser LR(1)
- Incremental CIncrementalParser.java
- Parse abstract syntax

Ok Cancel

Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'

Lisa 2.2

File Edit Compile Run View Messages Options Window Help

```
language Robot {  
  
  attributes int *.inx; int *.iny;  
             int *.outx; int *.outy;  
  
  lexicon {  
    ReservedWord left | right | up | down | begin | end | drek  
    ignore [ \0x0D\0x0A\ ]  
  }  
  
  rule start {  
    START ::= begin COMMANDS end compute {  
      START.outx = COMMANDS.outx;  
      START.outy = COMMANDS.outy;  
      COMMANDS.inx = 0;  
      COMMANDS.iny = 0;  
    };  
  }  
}
```

### Compiler options

Common Scanner Compiler Parser Evaluator

Evaluator

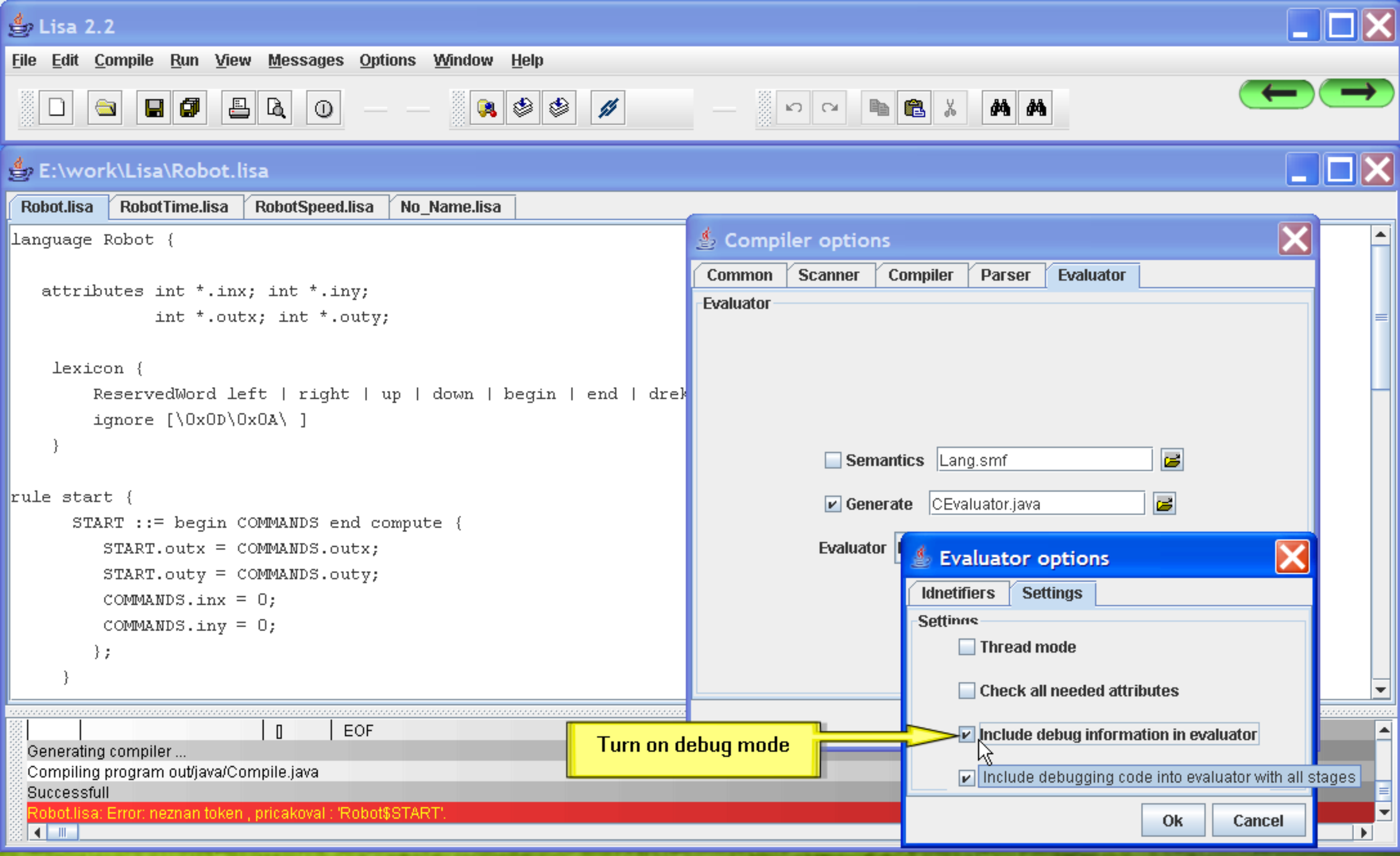
Semantics Lang.smf

Generate CEvaluator.java

Evaluator Lenic Tree Walk Evaluator

Ok Cancel

Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'



```
language Robot {  
  
  attributes int *.inx; int *.iny;  
             int *.outx; int *.outy;  
  
  lexicon {  
    ReservedWord left | right | up | down | begin | end | drek  
    ignore [ \0x0D\0x0A\ ]  
  }  
  
  rule start {  
    START ::= begin COMMANDS end compute {  
      START.outx = COMMANDS.outx;  
      START.outy = COMMANDS.outy;  
      COMMANDS.inx = 0;  
      COMMANDS.iny = 0;  
    };  
  }  
}
```

### Compiler options

Common Scanner Compiler Parser Evaluator

Evaluator

Semantics Lang.smf

Generate CEvaluator.java

Evaluator

### Evaluator options

Identifiers Settings

Settings

Thread mode

Check all needed attributes

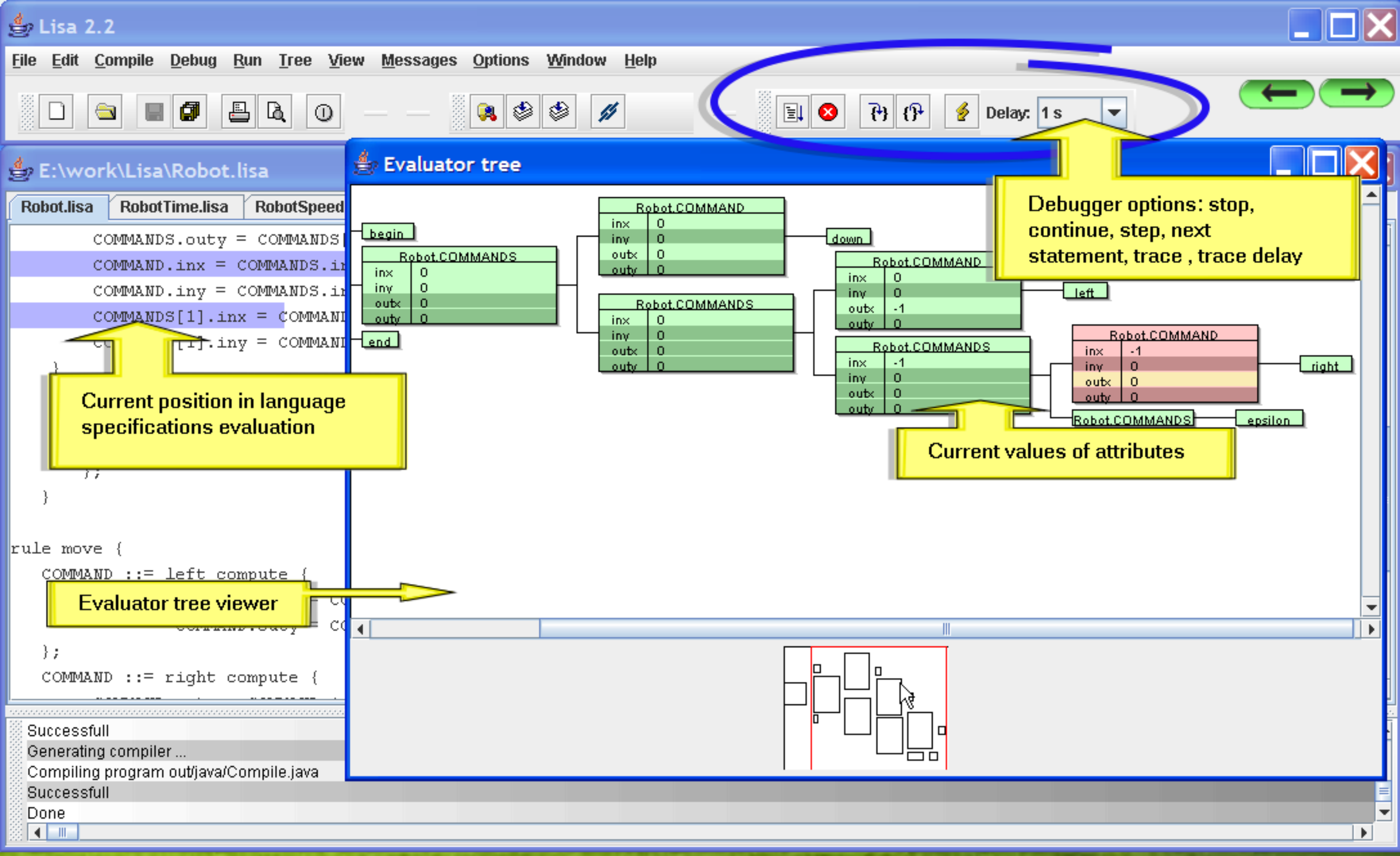
Include debug information in evaluator

Include debugging code into evaluator with all stages

Ok Cancel

Turn on debug mode

Generating compiler ...  
Compiling program out/java/Compile.java  
Successfull  
Robot.lisa: Error: neznan token , pricakoval : 'Robot\$START'





```
Select C:\WINDOWS\system32\cmd.exe
E:\work\Lisa>type test.robot
begin
  down down left down left down right
end
E:\work\Lisa>java -cp .\out\classes\;Lisa.jar Compile test.robot
Lisa Version 2.2
Generated compiler/interpreter
Switching to default
Parsing
File parsed in 0.0 s.
Evaluating
Robot.START
outx:-1:true
outy:-4:true
Program evaluated in 0.0 s.
E:\work\Lisa>
```

You can use generated sources as standalone applications or as integrated parts of your applications

Close window to return to LISA's home page

