CIS 771: Software Specifications

Lecture: Alloy Whirlwind Tour (part A)

Objectives of the Quick Tour lectures

- Understand primary features of the Alloy modeling language
  - modeling structures
  - specifying constraints
- Understand the basic capabilities of the Alloy Constraint Analyzer (ACA) automated tool
- Be able to write Alloy specifications that model simple systems
- Be able to run ACA to analyze simple systems

...this short lecture gives a quick tour of the quick tour
Address Book Application

Presentation will be based on a simple application

- Name
- Address
- Lists

Text-based Modeling Language

```
module tour/addressBook1

sig Name, Addr {}

sig Book {
    addr: Name -> lone Addr
}
```

Formal semantics based on sets and relations...

Signatures hold sets of abstract elements.

Fields form relations between signature elements and elements of field type.

CIS 771 --- Alloy Whirlwind Tour (part A)
User Interface

![Image of Alloy Whirlwind Tour (part A) interface]

An instance has been found. Click **Instance found** to launch visualization.

Results

Result of execution from previous slide...

- Alloy automatically looks for system configurations that satisfy the specifications.
- If we expected instances, but none can be found we know we have made a mistake by over-constraining our specification.
- If instances are found that we didn’t expect, this may mean that we need to add more constraints to our specification to rule out undesirable configurations.

Settings: Go to Theme, then Book, then turn on Project over this sig.
Generating Instances

Add constraints to direct the analysis...

```alloy
pred show1 (b: Book) {
    #b.addr > 1
}
```

```
run show1 for 3 but 1 Book
```

Example Operations

...adding a new book
Modeling \textit{Add Address}

\begin{verbatim}
pred add2(b, b': Book, n:Name, a:Addr) {
  // pre-condition
  // n is not currently in the book
  no n.(b.addr)
  // post-condition
  // invoking b' addr map on n yield's a
  n.(b'.addr) = a
  // frame-condition
  // for all other names, the addr map should
  // yield the same value
  all nl: (Name - n) | nl.(b.addr) = nl.(b'.addr)
}
\end{verbatim}

Results

\begin{verbatim}
r
\end{verbatim}

\textit{run add2 for 2 Book, exactly 4 Name, exactly 4 Addr}

\textit{using project over Book sig option in visualization}
Modeling System Executions

Consider the address book example...

- Consider an address book example...
- Use built-in functions from ordering library to impose an ordering on signature elements.
- Initially, the address book is empty.
- `first()` yields state `b`.
- `b'' = next(b')` yields state `b''`.
- `add` or `del` yields next state.
- Relate states/elements in the sequence using operations (output of one operation forms the input of the next).
- `last()` yields state.
- Initially, the address book is empty.
- Add or delete yields next state.

Results

- Initially, the address book is empty.
- Add `Group`, `Addr`.
- Add `Alias`, `Group`.
- Add `Alias`, `Group`.
- Delete `Group`, `Addr`.
- Assertion violation -- Group doesn't map to anything.
Acknowledgements

- The material in this lecture is based on Chapter 2 from...