**Fling—A Fluent API Generator**

Ori Roth  
Advisor: Prof. Yossi Gil  
Department of Computer Science, The Technion—Israel Institute of Technology

### The Fluent API Problem

Turning this...

```
SELECT id FROM my_table
WHERE age > 20
```

...into this:

```
select("id").from("my_table")
.where(column("id").gt(20))
```

The problem is parameterized by:
- DSL grammar class
- Host language requirements
- Guaranteed practicality

### Our Answer

We introduce a solution to the fluent API problem which supports:
- Any deterministic context-free grammar (DPDA equivalent: LL, LALR, LR, etc.)
- Type systems endorsing unbounded, unspecialized parametric polymorphism (Java, C++, C#, etc.)
- Linear compilation time and linear number of generated types

### From Grammar to Fluent API to AST

Run-time DPDA stack is encoded in a tree data structure, which can be encoded yet again as a type in a programming language supporting parametric polymorphism.

The root and references to its children are enough to compute the next automaton configuration.

### Compilation Performance

A parametric type can be represented by either a tree or a directed acyclic graph:
- Compilers recording types using DAGs compile fluent API method calls in linear time.
- Using trees instead might cause an exponential compilation time: We used this knowledge to detect a bug in the javac compiler.