



BDS

BIOCOMPUTING AND
DEVELOPMENTAL SYSTEMS

Context Sensitive Constraints in Grammatical Evolution

R. Muhammad Atif Azad

Extending libGE with Attribute Grammars

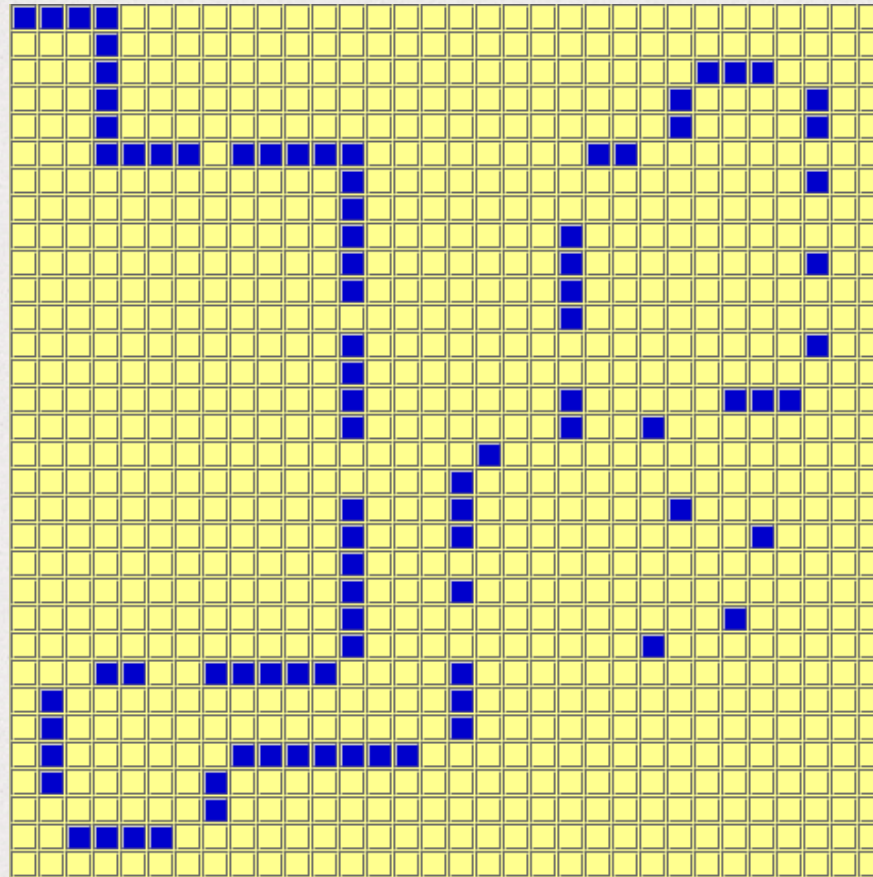


UNIVERSITY of LIMERICK
OILLSCOIL LUIMNIGH





An example problem: Efficient Navigation



Context Free Grammars

- Typically represented in Backus Naur Form (BNF).
- BNF uses a tuple $\langle \mathbf{T}, \mathbf{N}, \mathbf{P}, \mathbf{S} \rangle$:
 - \mathbf{T} = Set of Terminals;
 - \mathbf{N} = Set of Non-Terminals;
 - \mathbf{P} = Set of Production rules;
 - \mathbf{S} = Start Symbol; $\mathbf{S} \in \mathbf{N}$.

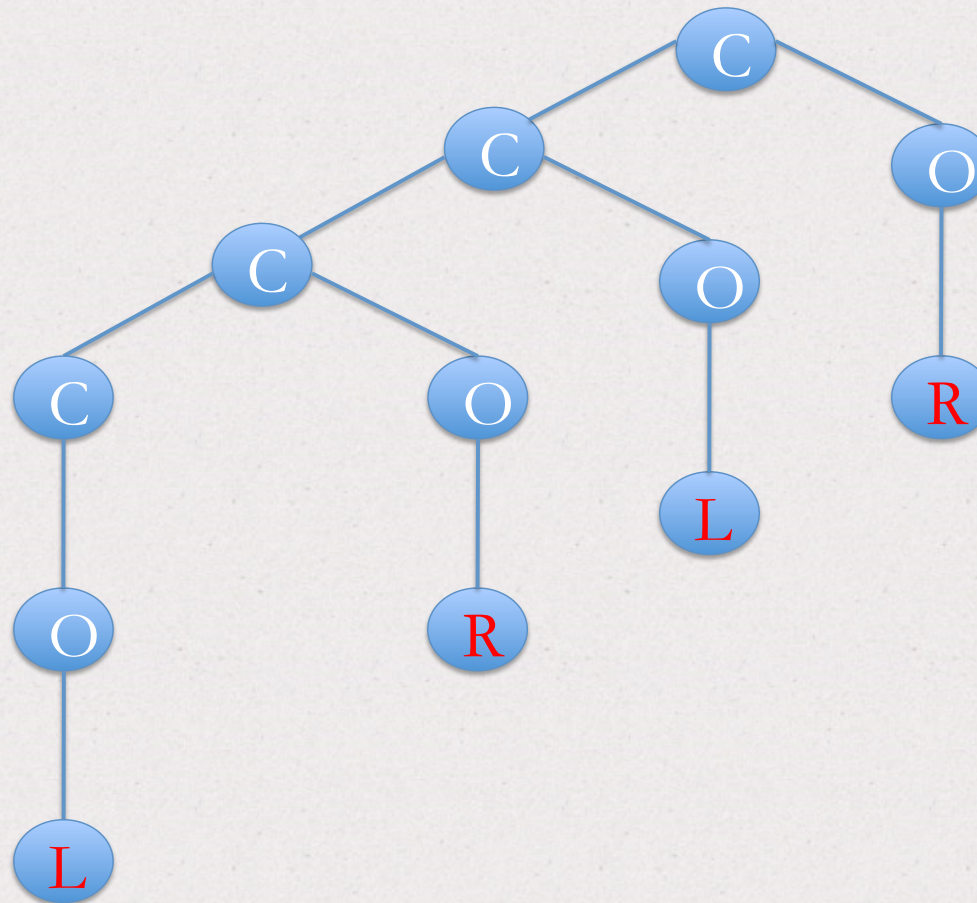
Example: Generate Moves

- $\mathbf{N} = \{ \langle \text{code} \rangle, \langle \text{op} \rangle \}$
- $\mathbf{T} = \{ \mathbf{M}, \mathbf{L}, \mathbf{R} \}$
- $\mathbf{S} = \langle \text{code} \rangle$
- $\mathbf{P} = \{$
 - $\langle \text{code} \rangle ::= \langle \text{op} \rangle \mid \langle \text{code} \rangle \langle \text{op} \rangle$
 - $\langle \text{op} \rangle ::= \mathbf{M} \mid \mathbf{L} \mid \mathbf{R}$ $\}$

Can generate *any* sequence: $\mathbf{LRLR}\dots, \mathbf{MLMR}..$



An Example of a Derivation Tree



Scenario: Constraints

- Useless sequence:
 - **L R L R...**
- Restrict choices based on the history (context):
 - Each op has an *attribute*, e.g. *is_valid*.
 - **L** is only valid if **R** was *not* the previous move.
- **Context Sensitivity:** Present choice of rules affects future choices.

Attribute Grammars

- Allow context sensitivity through a set of attributes.
 - Terminals/Non-Terminals have attributes:
 - $\langle \text{op} \rangle.\textit{last_move}$,
 - $L.\textit{is_valid}$.
- Semantic functions determine the attribute values:
 - $L.\textit{is_valid} = f(\langle \text{op} \rangle.\textit{last_move})$.



BDS

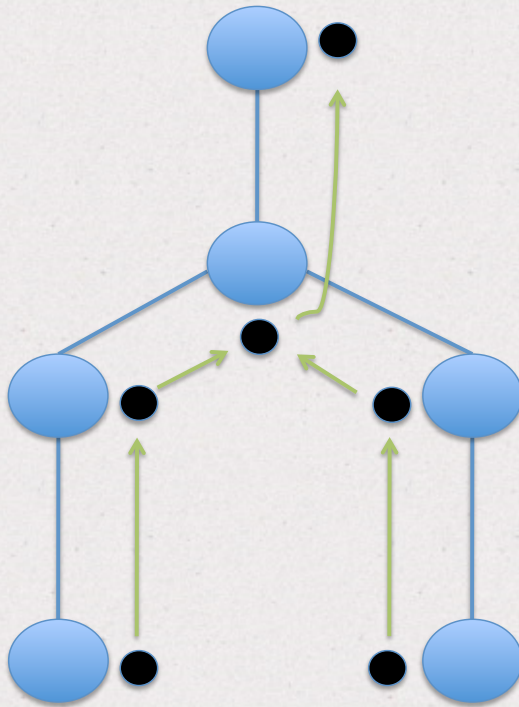
BIOCOMPUTING AND
DEVELOPMENTAL SYSTEMS



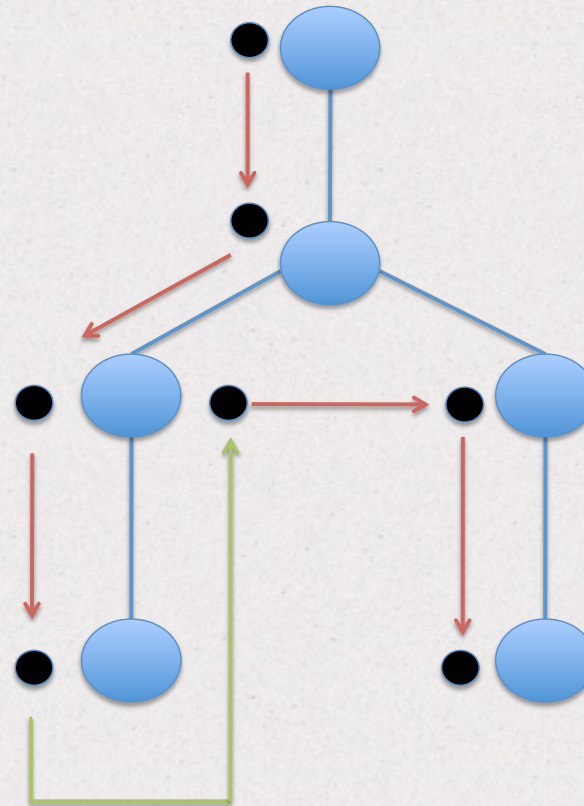
UNIVERSITY of LIMERICK
OGLSCOIL LUIMNIGH



Types of Attributes



Synthesised Attributes



Inherited Attributes

Courtesy: Karim and Ryan, *Memetic Computing*, 2012

Previous Work

- Outside BDS:
 - Patterson (2002): reflexive attribute grammars
 - Ortega et al. (2007): Christiansen grammars.
 - Cleary (2005): attribute grammars with GE.
- Inside BDS:
 - Karim and Ryan
 - 4 conference publications: 2011 – 2012
 - 1 journal publication: *Memetic Computing*, 2012.

libGE Overview

- An open source library for GE.
- Implemented in C++ for linux platforms.
- Currently supports a standard Context Free Grammar (CFG) specified in BNF notation
 - Does not yet support Attribute Grammars
- Current version: 0.26.
- Available from
 - <http://www.grammaticalevolution.org/>



BDS

BIOCOMPUTING AND
DEVELOPMENTAL SYSTEMS

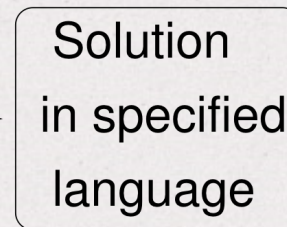
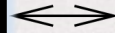
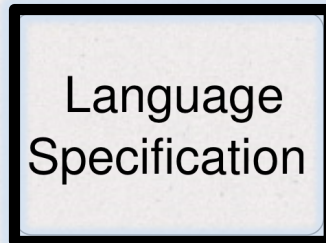
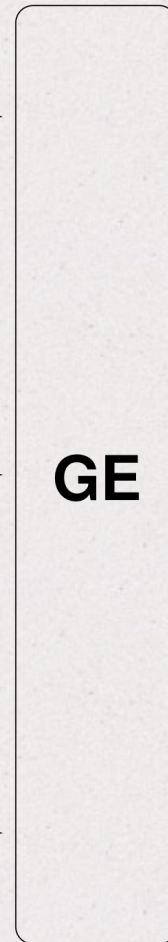
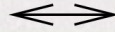
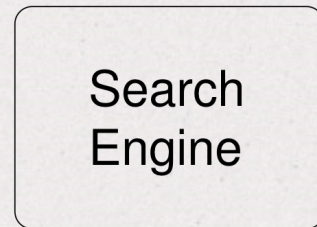


UNIVERSITY of LIMERICK
OILESCOIL LUIMNIGH



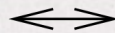
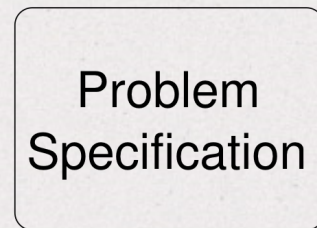
libGE – Design Philosophy

- MIT GAlib
- EO (Evolving Objects)
- illiGAL GA



- S-lang
- Lua
- TinyCC
- GCC

- Cart Centering
- Intertwined Spirals
- Santa Fe Ant Trail



libGE Extensions for Attribute Grammars

- Allow symbols (N , T) to have attributes.
- Add facility for incorporating and firing semantic rules appropriately while mapping.
- Implement a lookup manager that ascribes a suitable class object to each symbol.
- Incorporate mechanisms for propagating attribute values up and down.



BDS

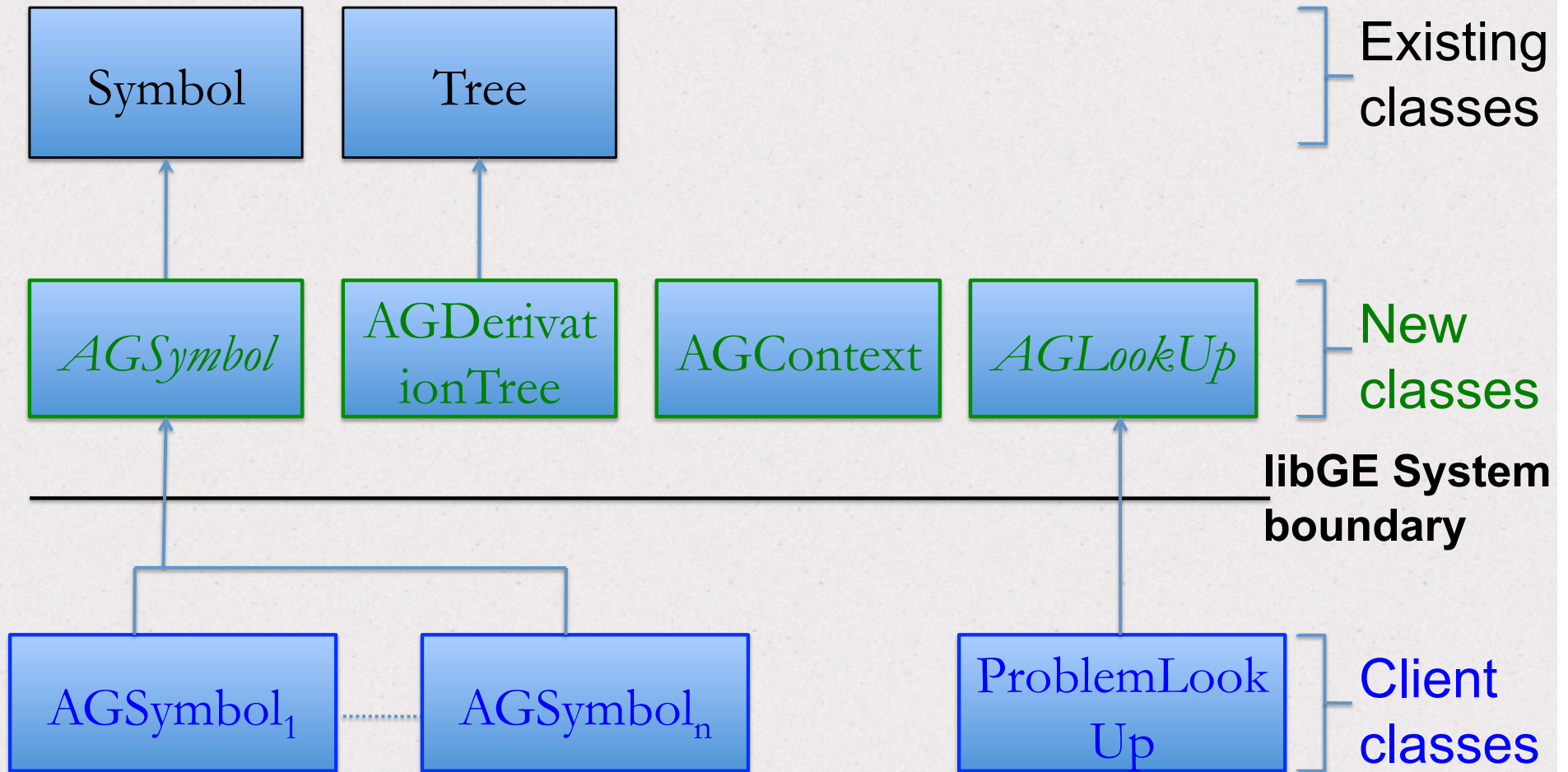
BIOCOMPUTING AND
DEVELOPMENTAL SYSTEMS



UNIVERSITY of LIMERICK
OILESCOIL LUIMNIGH



libGE Extension



Progress

- Released internally to BDS.
 - Implemented a sample problem.
- Undergoing internal testing.
- Expected Release at Christmas - 2014