Verification of Data Structures
Using Jahob

Feride Çetin
Kremena Diatchka

Outline

- Jahob
- Motivation
- Cyclic list example
- In progess: leaf linked tree
- Conclusions

Verifying data structures

- Developer: provides specifications
  - Loop invariants
  - Procedure contracts (pre- & post-conditions)
  - Class invariants

- Jahob:
  - Specs --> logical constraints --> proved by decision procedures

Motivations

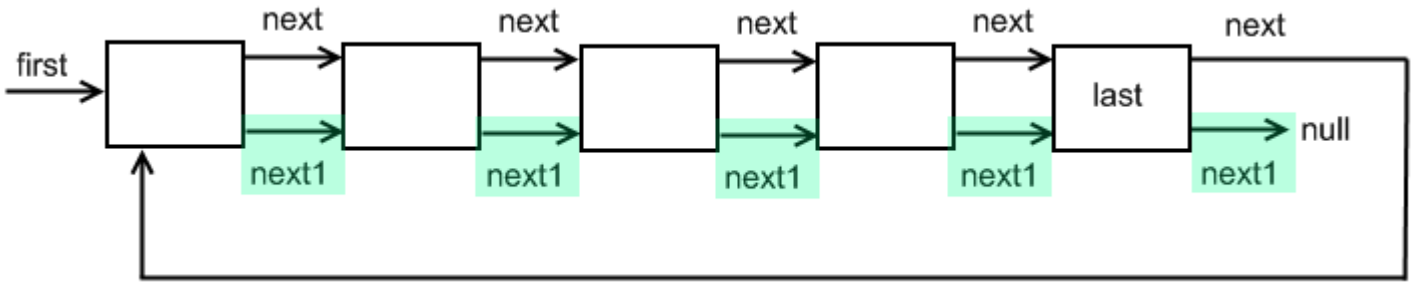
1. We can solve more problems automatically
2. Fast computers --> can use advanced techniques
3. Computers everywhere -> bugs everywhere, bugging everyone
4. Bugs are $$$

Modular verification of data structures

Status

- Verified
  - List with header node
  - Queue
  - Cyclic list
- Tried
  - Instantiable queue
- In progress
  - Leaf-linked tree

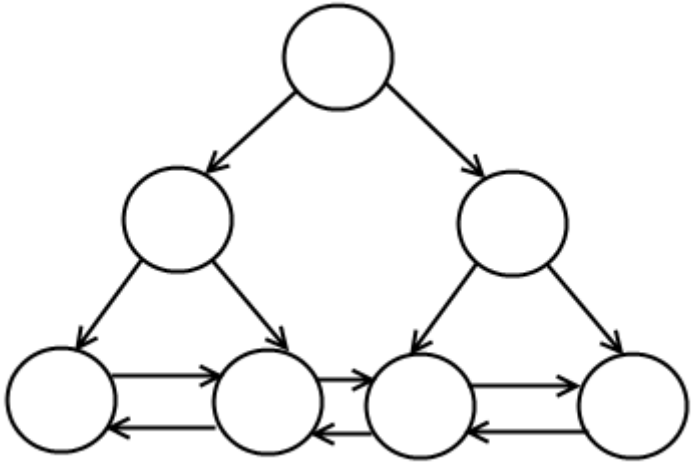
Cyclic list example



Demo (kind of)

Leaf-linked tree

- Motivation: *log(n)* operations on linked list
- Node:
  - Node left, right, next, prev, parent
  - int v



Leaf-linked tree structure

- Binary search tree
- No sort property verification for this project
- Left subtree : values <= parent.v
- Right subtree : values > parent.v
- Each parent node holds the largest left subtree value
- private static Node root;

Insertion method

- Each value is inserted as a leaf
- Two insertion stages:
  - Insertion in the binary search tree
  - Update of the leaf-linked list

Methods

boolean isEmpty() -> verified

boolean isLeaf (Node n) -> verified

void add (int v) -> in progress

void leafUpdate (Node n) -> in progress

Specification variables

- Nodes: all nodes reachable from root using left or right fields
- Content: the values of nodes in Nodes
- Internal nodes: all those nodes for which at least one of the left or right fields is not null

Class invariants

- Tree invariant on left and right fields
- Root Not Pointed: if root is not null then no node exists whose left, right, next, prev fields point to root
- Field constraint on all Node fields of a node: they should point to a node in Nodes (in this tree)
- Field constraint on parent field: if x has a parent, then there exists a node whose left or right field points to x

Conclusions

“The whole problem with the world is that fools and fanatics are always so certain of themselves, but wiser people so full of doubts.”

- Earl Bertrand Russell

Conclusions

"The whole problem with the world is that fools and fanatics are always so certain of themselves, but wiser people so full of doubts."

- Earl Bertrand Russell

Jahob: helping prove fools wrong and wise people correct since 2007

Verification of Data Structures
Using Jahob

Feride Çetin
Kremena Diatchka

Outline

- Jahob
- Motivation
- Cyclic list example
- In progess: leaf linked tree
- Conclusions

2

Motivation
Jahob and decision procedures it uses
    Mona, spass (FOL theorem prover)
What data structures we have done
Cyclic list example – show it verifies
In progess: leaf linked tree
Conclusions

Verifying data structures

- Developer: provides specifications
  - Loop invariants
  - Procedure contracts (pre- & post-conditions)
  - Class invariants
- Jahob:
  - Specs --> logical constraints --> proved by decision procedures

Motivations

1. We can solve more problems automatically
2. Fast computers --> can use advanced techniques
3. Computers everywhere -> bugs everywhere, bugging everyone
4. Bugs are $$$

Modular verification of data structures

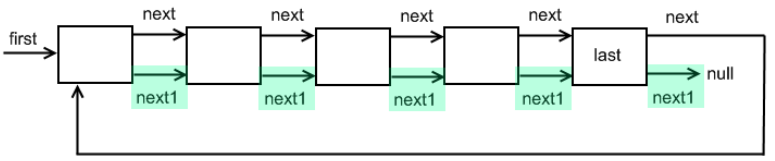
4

Software bugs, or errors, are so prevalent and so detrimental that they cost the U.S. economy an estimated $59.5 billion annually, or about 0.6 percent of the gross domestic product, according to a newly released study commissioned by th

Status

- Verified
  - List with header node
  - Queue
  - Cyclic list
- Tried
  - Instantiable queue
- In progress
  - Leaf-linked tree

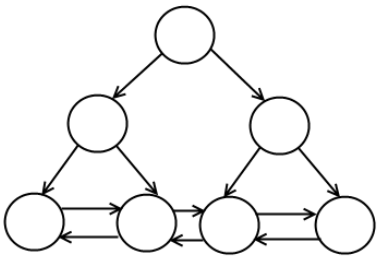
Cyclic list example



Demo (kind of)

Leaf-linked tree

- Motivation: *log(n)* operations on linked list
- Node:
  - Node left, right, next, prev, parent
  - int v



Leaf-linked tree structure

- Binary search tree
- No sort property verification for this project
- Left subtree : values <= parent.v
- Right subtree : values > parent.v
- Each parent node holds the largest left subtree value
- private static Node root;

Insertion method

- Each value is inserted as a leaf
- Two insertion stages:
  - Insertion in the binary search tree
  - Update of the leaf-linked list

Methods

boolean isEmpty() -> verified

boolean isLeaf (Node n) -> verified

void add (int v) -> in progress

void leafUpdate (Node n) -> in progress

Specification variables

- Nodes: all nodes reachable from root using left or right fields
- Content: the values of nodes in Nodes
- Internal nodes: all those nodes for which at least one of the left or right fields is not null

Class invariants

- Tree invariant on left and right fields
- Root Not Pointed: if root is not null then no node exists whose left, right, next, prev fields point to root
- Field constraint on all Node fields of a node: they should point to a node in Nodes (in this tree)
- Field constraint on parent field: if x has a parent, then there exists a node whose left or right field points to x

Conclusions

“The whole problem with the world is that fools and fanatics are always so certain of themselves, but wiser people so full of doubts.”

- Earl Bertrand Russell

13

We are taking little steps toward creating more reliable software, where we can say for certain that parts of it perform what they are meant to do.

Conclusions

"The whole problem with the world is that fools and fanatics are always so certain of themselves, but wiser people so full of doubts."

- Earl Bertrand Russell

Jahob: helping prove fools wrong and wise people correct since 2007

14

We are taking little steps toward creating more reliable software, where we can say for certain that parts of it perform what they are meant to do.