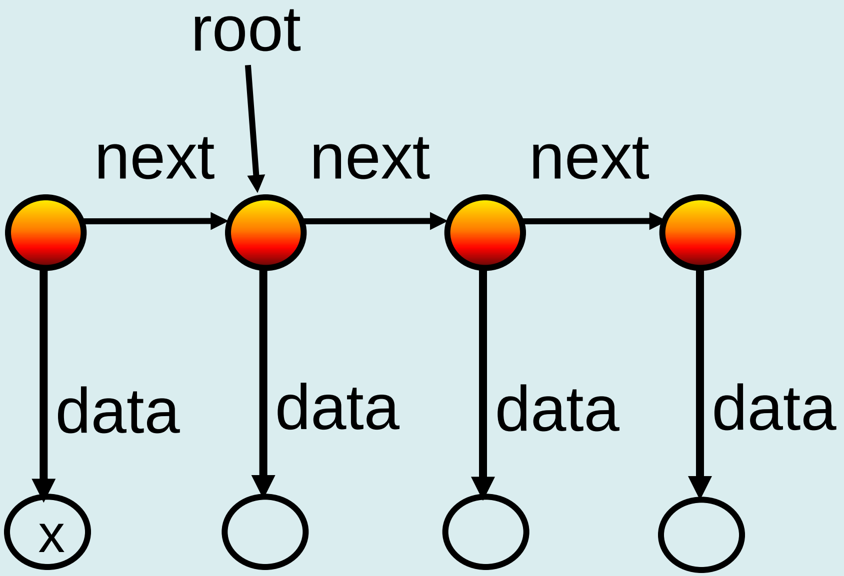
Linked List Implementation

```
class List {
  private List next;
  private Object data;
  private static List root;
  private static int size;
```

invariant : size = |{data(x). next*(root,x)}|

```
  public static void addNew(Object x) {
      List n1 = new List();
      n1.next = root;
      n1.data = x;
      root = n1;
      size = size + 1;
  }
}
```



Verification Condition for addNew

$\neg next0^*(root0,n1) \wedge x \notin \{data0(n) \mid next0^*(root0,n)\} \wedge$
$next=next0[n1:=root0] \wedge data=data0[n1:=x] \rightarrow$
$|\{data(n) \,.\, next^*(n1,n)\}| =$
$|\{data0(n) \,.\, next0^*(root0,n)\}| + 1$

"The number of stored objects has increased by one."

Expressing this VC requires a rich logic

– transitive closure * (in lists and also in trees)

– unconstraint functions (data, data0)

– cardinality operator on sets | ... |

Is there a decidable logic containing all this?

Decomposing the Formula

Consider a (simpler) formula

$$|\{data(x) \,.\, next^*(root,x)\}|=k+1$$

Introduce fresh variables denoting sets:

$A = \{x.\ next^*(root,x)\} \wedge$ — 1) WS2S

$B = \{y.\ \exists\, x.\ data(x,y) \wedge x \in A\} \wedge$ — 2) C^2

$|B|=k+1$ — 3) BAPA

Conjuncts belong to decidable fragments!

Next

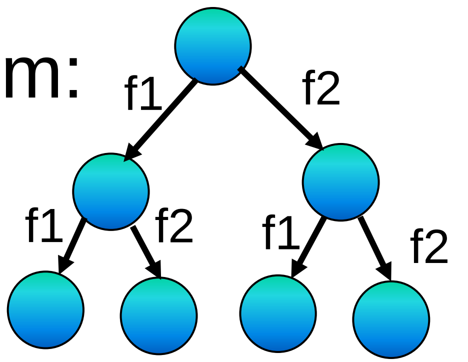
- define these 3 fragments
- sketch a technique to combine them

WS2S: Monadic 2^{nd} Order Logic

Weak Monadic 2^{nd}-order Logic of 2 Successors

In HOL, satisfiability of formulas of the form:

tree[f1,f2] & F(f1,f2,S,T)



where

- tree[f1,f2] means f1,f2 form a tree

F ::= x=f1(y) | x=f2(y) | $x \in S$ | $S \subseteq T$ | $\exists S.F$ | $F_1 \wedge F_2$ | $\neg F$

- quantification is over finite sets of positions in tree
- transitive closure encoded using set quantification

Decision procedure

- recognize WS2S formula within HOL
- run the MONA tool (tree automata, BDDs)

C^2 : Two-Variable Logic w/ Counting

Two-Variable Logic with Counting

$F ::= P(v_1,...,v_n) \mid F_1 \wedge F_2 \mid \neg F \mid \exists^{\mathbf{count}} v_i.F$

where

P : is a predicate symbol

v_i : is one of the ***two*** variable names x,y

count : is =k, ≤k, or ≥k for nonnegative *constants* k

We can write $(\exists^{\leq k} v_i.F)$ as $|\{v_i.F\}| \leq k$

We can define $\exists, \forall$ and axiomatize total functions:

$\forall x \exists^{=1} y.R(x,y)$

Decidable sat. and fin-sat. (1997), NEXPTIME

even for binary-encoded k: Pratt-Hartman '05

BAPA:
Boolean Algebra with Presburger Arithmetic

$$S ::= V \mid S_1 \cup S_2 \mid S_1 \cap S_2 \mid S_1 \setminus S_2$$

$$T ::= k \mid C \mid T_1 + T_2 \mid T_1 - T_2 \mid C \cdot T \mid |S|$$

$$A ::= S_1 = S_2 \mid S_1 \subseteq S_2 \mid T_1 = T_2 \mid T_1 < T_2$$

$$F ::= A \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \neg F \mid \exists S.F \mid \exists k.F$$

Essence of decidability: Feferman, Vaught 1959

Our results

- first implementation for BAPA (CADE'05)
- first, exact, complexity for full BAPA (JAR'06)
- first, exact, complexity for QFBAPA (CADE'07)
- generalize to multisets (VMCAI'08,CAV'08,CSL'08)

New: role of BAPA in combination of logics

Back to Decomposing the Formula

Consider a (simpler) formula

$$|\{data(x)\ .\ next^*(root,x)\}|=k+1$$

Introduce fresh variables denoting sets:

$A = \{x.\ next^*(root,x)\} \wedge$ 1) WS2S

$B = \{y.\ \exists\ x.\ data(x,y) \wedge x \in A\} \wedge$ 2) C^2

$|B|=k+1$ 3) BAPA

Conjuncts belong to decidable fragments

Next

- define these 3 fragments – we have seen this
- **sketch a technique to combine them**

Combining Decidable Logics

Satisfiability problem expressed in HOL:

(all free symbols existentially quantified)

$\exists$ next,data,k. $\exists$ root,A,B.

A = {x. next*(root,x)} $\wedge$

B = {y. $\exists$ x. data(x,y) $\wedge$ x $\in$ A} $\wedge$

|B|=k+1

1) WS2S

2) C^2

3) BAPA

We assume formulas share only:

- **set variables** (sets of uninterpreted elems)
- individual variables, as a special case - {x}

Combining Decidable Logics

Satisfiability problem expressed in HOL,

after moving fragment-specific quantifiers

$\exists$ root,A,B.

$\exists$ next. A = {x. next*(root,x)} $\wedge$

$\exists$ data. B = {y. $\exists$ x. data(x,y) $\wedge$ x $\in$ A} $\wedge$

$\exists$ k. |B|=k+1

F_{WS2S} : {root}$\subseteq$A

F_{BAPA} : 1 $\leq$ |B|

F_{C2} : |B|$\leq$|A|

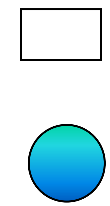
Extend decision procedures into

projection procedures for WS2S,C^2,BAPA

Conjunction of projections satisfiable → so is original formula

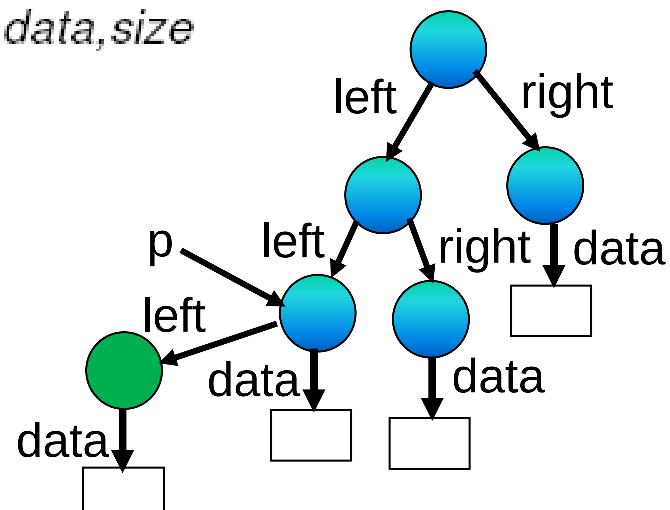
$\exists$ root,A,B. {root}$\subseteq$A $\wedge$|B|$\leq$|A| $\wedge$ 1 $\leq$ |B|

Fragment of Insertion into Tree



```
class Node {Node left,right; Object data;}
class Tree {
    private static Node root;
    private static int size; /*:
    private static specvar nodes :: objset;
    vardefs "nodes=={x. (root,x) ∈ {(x,y). left x = y ∨ right x = y}* }";
    private static specvar content :: objset;
    vardefs "content=={x. ∃ n. n ≠ null ∧ n ∈ nodes ∧ data n = x} " */

    private void insertAt(Node p, Object e) /*:
      requires "tree [ left , right ] ∧ nodes ⊆ Object.alloc ∧ size = card content ∧
                e ∉ content ∧ e ≠ null ∧ p ∈ nodes ∧ p ≠ null ∧ left p = null"
      modifies nodes,content,left,right ,data,size
      ensures "size = card content" */
    {
        Node tmp = new Node();
        tmp.data = e;
        p. left = tmp;
        size = size + 1;
    }
}
```



Verification Condition for Tree Insertion

SHARED SETS: nodes, nodes1, content, content1, $\{e\}$, $\{tmp\}$

WS2S FRAGMENT:

tree[left , right] $\wedge$ left p = null $\wedge$ p $\in$ nodes $\wedge$ left tmp = null $\wedge$ right tmp = null $\wedge$

nodes=$\{$x. (root,x) $\in$ $\{$(x,y). left x = y| right x = y$\}$^$*$$\}$ $\wedge$

nodes1=$\{$x. (root,x) $\in$ $\{$(x,y). (left (p:=tmp)) x = y) | right x = y$\}$

CONSEQUENCE: nodes1=nodes $\cup$ $\{$tmp$\}$

C2 FRAGMENT:

data tmp = null $\wedge$ ($\forall$ y. data y $\neq$ tmp) $\wedge$ tmp $\notin$ alloc $\wedge$ nodes $\subseteq$ alloc $\wedge$

content=$\{$x. $\exists$ n. n $\neq$ null $\wedge$ n $\in$ nodes $\wedge$ data n = x$\}$ $\wedge$

content1=$\{$x. $\exists$ n. n $\neq$ null $\wedge$ n $\in$ nodes1 $\wedge$ (data(tmp:=e)) n = x$\}$

CONSEQUENCE: nodes1 $\neq$ nodes $\cup$ $\{$tmp$\}$ $\vee$ content1 = content $\cup$ $\{$e$\}$

BAPA FRAGMENT: e $\notin$ content $\wedge$ card content1 $\neq$ card content + 1

CONSEQUENCE: e $\notin$ content $\wedge$ card content1 $\neq$ card content + 1

Conjunction of projections unsatisfiable → so is original formula

Decision Procedure for Combination

1. Separate formula into WS2S, C^2, BAPA parts
2. For each part, compute projection onto set vars
3. Check satisfiability of conjunction of projections

Definition: Logic is *effectively cardinality-linear* iff there is an algorithm that computes projections of formulas onto set variables, and these projections are quantifier-free BAPA formulas.

Theorem: WS2S, C^2, BAPA are all cardinality linear.

Proof: WS2S – Parikh image of tree language is in PA
C^2 – proof by Pratt-Hartmann reduces to PA
BAPA - has quantifier elimination