An Abstract Decision Procedure for Satisfiability in the Theory of Recursive Data Types

Clark Barrett
New York University

Joint work with:
Igor Shikanian, New York University
Cesare Tinelli, University of Iowa

Recursive Data Types

Recursive data types

- Built up from basic types using *type constructors*
- *Type selectors* and *type testers* are also defined for convenience

Recursive Data Types

Recursive data types

- Built up from basic types using *type constructors*
- *Type selectors* and *type testers* are also defined for convenience

Example: *list* of *int*

- Constructors: *cons*: $(int, list) \longrightarrow list$, *null*: *list*
- Selectors: *car*: $list \longrightarrow int$, *cdr*: $list \longrightarrow list$
- Testers: *is_cons*, *is_null*

Recursive Data Types

Recursive data types

- Built up from basic types using *type constructors*
- *Type selectors* and *type testers* are also defined for convenience

Example: *list* of *int*

- Constructors: $cons: (int, list) \longrightarrow list$, $null: list$
- Selectors: $car: list \longrightarrow int$, $cdr: list \longrightarrow list$
- Testers: *is_cons*, *is_null*

Convenient Notation

- $list := cons(car : int, cdr : list) | null$
- $nat := succ(pred : nat) | zero$

First Order Theory of Recursive Data Types

Recursive Data Types (*RDTs*) can be modeled very naturally using multi-sorted first order logic:

Syntax

- *Sort* for each data type
- *Function* for each constructor and selector
- *Predicate* for each tester

First Order Theory of Recursive Data Types

Recursive Data Types (*RDTs*) can be modeled very naturally using multi-sorted first order logic:

Syntax

- *Sort* for each data type
- *Function* for each constructor and selector
- *Predicate* for each tester

Examples

- $\forall\, x : \mathit{nat}.\ \mathit{succ}(x) \not\approx \mathit{zero}$
- $\forall\, x : \mathit{list}.\ x \approx \mathit{null} \vee \exists\, y : \mathit{nat}, z : \mathit{list}.\ x \approx \mathit{cons}(y, z)$

First Order Theory of Recursive Data Types

We will consider a generic *RDT* denoted as follows:

- *Constructors*: $C_j : (s_{j,1}, \ldots s_{j,n_j}) \rightarrow \tau, j = 1, \ldots, m$
- *Selectors*: $S_{j,k} : \tau \rightarrow s_{j,k}, j = 1, \ldots, m, k = 1, \ldots, n_j$
- *Testers*: $isC_j : \tau \rightarrow$ bool, $j = 1, \ldots, m$

First Order Theory of Recursive Data Types

We will consider a generic *RDT* denoted as follows:

- *Constructors*: $C_j : (s_{j,1}, \ldots s_{j,n_j}) \rightarrow \tau, j = 1, \ldots, m$
- *Selectors*: $S_{j,k} : \tau \rightarrow s_{j,k}, j = 1, \ldots, m, k = 1, \ldots, n_j$
- *Testers*: $isC_j : \tau \rightarrow$ bool, $j = 1, \ldots, m$

Or, written schematically:

$$\begin{array}{lll} \tau & := & C_1(S_{1,1} : s_{1,1}, \ldots, S_{1,n_1} : s_{1,n_1}) \quad | \\ & & C_2(S_{2,1} : s_{2,1}, \ldots, S_{2,n_2} : s_{2,n_2}) \quad | \\ & & \vdots \\ & & C_m(S_{m,1} : s_{m,1}, \ldots, S_{m,n_m} : s_{m,n_j}) \end{array}$$

First Order Theory of Recursive Data Types

We will consider a generic *RDT* denoted as follows:

- *Constructors*: $C_j : (s_{j,1}, \ldots s_{j,n_j}) \rightarrow \tau, j = 1, \ldots, m$
- *Selectors*: $S_{j,k} : \tau \rightarrow s_{j,k}, j = 1, \ldots, m, k = 1, \ldots, n_j$
- *Testers*: $isC_j : \tau \rightarrow$ bool, $j = 1, \ldots, m$

Or, written schematically:

$$\begin{array}{lll} \tau & := & C_1(S_{1,1} : s_{1,1}, \ldots, S_{1,n_1} : s_{1,n_1}) \quad | \\ & & C_2(S_{2,1} : s_{2,1}, \ldots, S_{2,n_2} : s_{2,n_2}) \quad | \\ & & \vdots \\ & & C_m(S_{m,1} : s_{m,1}, \ldots, S_{m,n_m} : s_{m,n_j}) \end{array}$$

Each $s_{j,k}$ is either τ or some non-*RDT* sort.

First Order Theory of Recursive Data Types

We will consider a generic *RDT* denoted as follows:

- *Constructors*: $C_j : (s_{j,1}, \ldots s_{j,n_j}) \rightarrow \tau, j = 1, \ldots, m$
- *Selectors*: $S_{j,k} : \tau \rightarrow s_{j,k}, j = 1, \ldots, m, k = 1, \ldots, n_j$
- *Testers*: $isC_j : \tau \rightarrow$ bool, $j = 1, \ldots, m$

Or, written schematically:

$$\begin{array}{lll} \tau & := & C_1(S_{1,1} : s_{1,1}, \ldots, S_{1,n_1} : s_{1,n_1}) \quad | \\ & & C_2(S_{2,1} : s_{2,1}, \ldots, S_{2,n_2} : s_{2,n_2}) \quad | \\ & & \vdots \\ & & C_m(S_{m,1} : s_{m,1}, \ldots, S_{m,n_m} : s_{m,n_j}) \end{array}$$

Each $s_{j,k}$ is either τ or some non-*RDT* sort.

We assume an infinite number of constants of each non-*RDT* sort.

Semantics

Let Σ be the signature induced by the *RDT* just described. Let Ω be the signature obtained from Σ by removing the selectors and testers.

Let $\mathcal{T}(\Omega)$ be the set of all ground terms of signature Ω.

Let $\mathcal{R}$ be the model of Σ defined as follows:

- $\mathcal{T}(\Omega)$ is the carrier set for the sort τ
- $isC_j(t)$ is true iff t is an application of C_j.
- $S_{j,k}(C_j(t_1, \ldots, t_{n_j})) \approx t_k$

Semantics

Let Σ be the signature induced by the *RDT* just described. Let Ω be the signature obtained from Σ by removing the selectors and testers.

Let $\mathcal{T}(\Omega)$ be the set of all ground terms of signature Ω.

Let $\mathcal{R}$ be the model of Σ defined as follows:

- $\mathcal{T}(\Omega)$ is the carrier set for the sort τ
- $isC_j(t)$ is true iff t is an application of C_j.
- $S_{j,k}(C_j(t_1, \ldots, t_{n_j})) \approx t_k$

What about $S_{j,k}(C_{j'}(t_1, \ldots, t_{n_{j'}})), j \neq j'$*?*

Semantics

Let Σ be the signature induced by the *RDT* just described. Let Ω be the signature obtained from Σ by removing the selectors and testers.

Let $\mathcal{T}(\Omega)$ be the set of all ground terms of signature Ω.

Let $\mathcal{R}$ be the model of Σ defined as follows:

- $\mathcal{T}(\Omega)$ is the carrier set for the sort τ
- $isC_j(t)$ is true iff t is an application of C_j.
- $S_{j,k}(C_j(t_1, \ldots, t_{n_j})) \approx t_k$

What about $S_{j,k}(C_{j'}(t_1, \ldots, t_{n_{j'}})), j \neq j'$*?*

We have to define it to be something, so we just designate some ground term (usually as simple as possible).

Semantics

Let Σ be the signature induced by the *RDT* just described. Let Ω be the signature obtained from Σ by removing the selectors and testers.

Let $\mathcal{T}(\Omega)$ be the set of all ground terms of signature Ω.

Let $\mathcal{R}$ be the model of Σ defined as follows:

- $\mathcal{T}(\Omega)$ is the carrier set for the sort τ
- $isC_j(t)$ is true iff t is an application of C_j.
- $S_{j,k}(C_j(t_1, \ldots, t_{n_j})) \approx t_k$

What about $S_{j,k}(C_{j'}(t_1, \ldots, t_{n_{j'}})), j \neq j'$*?*

We have to define it to be something, so we just designate some ground term (usually as simple as possible).

We assume that users will appropriately guard applications of selectors. This can be checked if desired.

Decision Procedure

We are interested in determining the satisfiability of sets of Σ-literals over $\mathcal{R}$.

Related work

- Oppen's procedure: limited to a single constructor
- Zhang et al.: extend Oppen's procedure using nondeterministic *type completion* guess
- Superposition approach: single constructor, efficiency untested

We are interested in a procedure that is both expressive and efficient

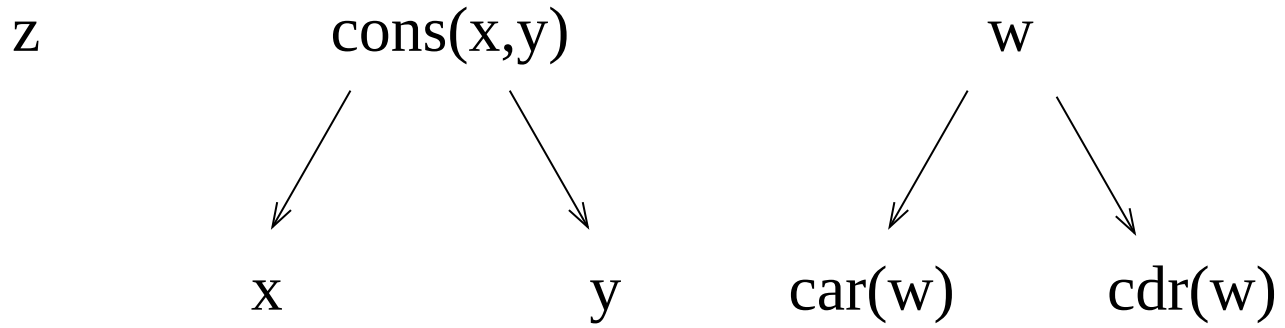
- *Expressive:* Multi-sorted, arbitrary constructors, selectors, testers, mutual recursion.
- *Efficient:* Strategies for delaying expensive operations

Term Graphs

An important concept in understanding the procedure is an associated *graph* that relates terms according to their meaning in $\mathcal{R}$.

Example

- $\mathit{list} := \mathit{cons}(\mathit{car} : \mathit{int}, \mathit{cdr} : \mathit{list}) | \mathit{null}$
- $\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

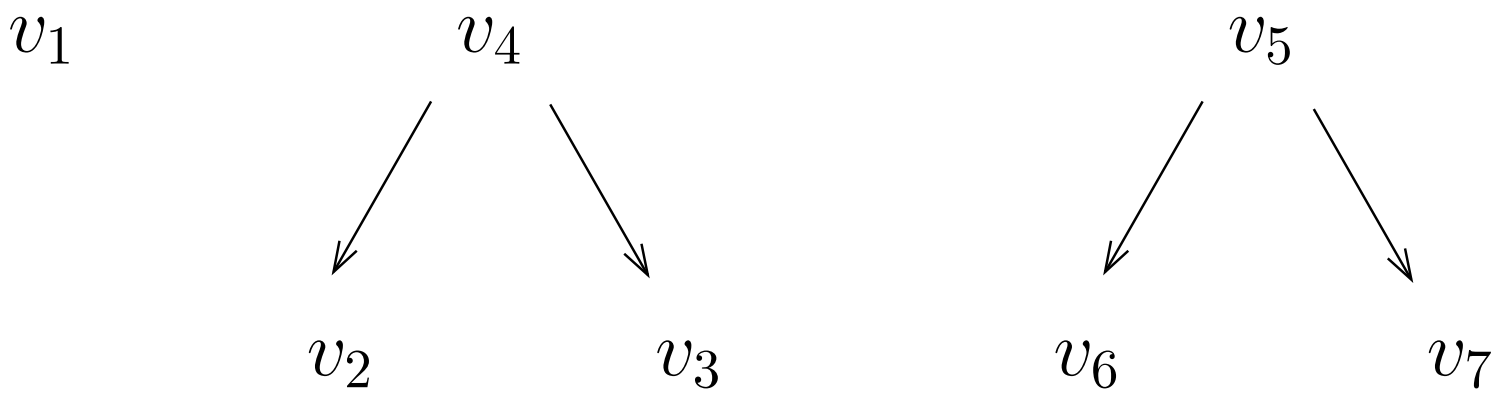


Abstraction Rules

- *Flatten* terms by introducing new abstraction variables.
- *Label* each abstract variable with possible constructors.

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

$z \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \approx v_1$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_2 \approx v_6$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_3 \approx v_7$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}, \mathit{null}\}$	$\mathit{is\_cons}(v_5)$
$\mathit{car}(v_5) \rightarrow v_6$	$v_6 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_5) \rightarrow v_7$	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	



Literal level rules

Orient $$\frac{u \approx v,\ E}{u \to v,\ E} \quad \text{if } u \succ v$$

Inconsistent $$\frac{v \not\approx v,\ E}{\bot}$$

Remove 1 $$\frac{isC_j\, v,\ E}{v \mapsto \{C_j\},\ E}$$

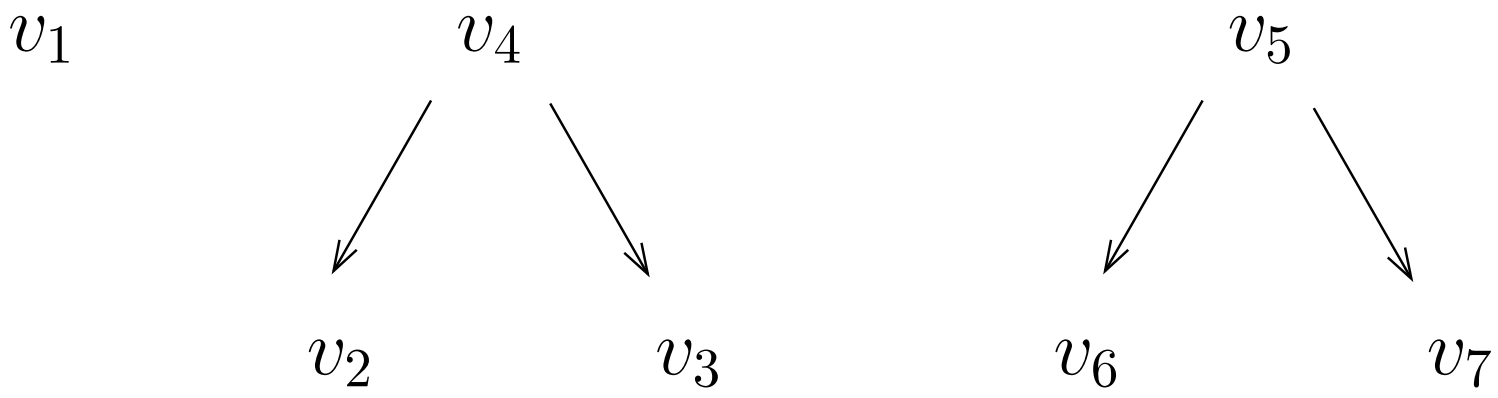
Remove 2 $$\frac{\neg isC_j\, v,\ E}{v \mapsto \mathit{lbls}(sort(v)) \setminus \{C_j\},\ E}$$

Literal level rules

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

After Literal level rules

$z \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \rightarrow v_1$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_6 \rightarrow v_2$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_7 \rightarrow v_3$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_5) \rightarrow v_6$	$v_6 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_5) \rightarrow v_7$	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	



Selector Rules

Instantiate 1

$$\frac{S_{j,1}u \to u_1,\ \ldots,\ S_{j,n_j}u \to u_{n_j},\ u \mapsto \{C_j\},\ E}{C_j u_1 \cdots u_{n_j} \to u,\ u \mapsto \{C_j\},\ E}$$

Instantiate 2 Similar, but for finite constructors only.

Collapse 1

$$\frac{C_j u_1 \cdots u_{n_j} \to u,\ S_{j,k}u \to v,\ E}{C_j u_1 \cdots u_{n_j} \to u,\ u_k \approx v,\ E}$$

Collapse 2

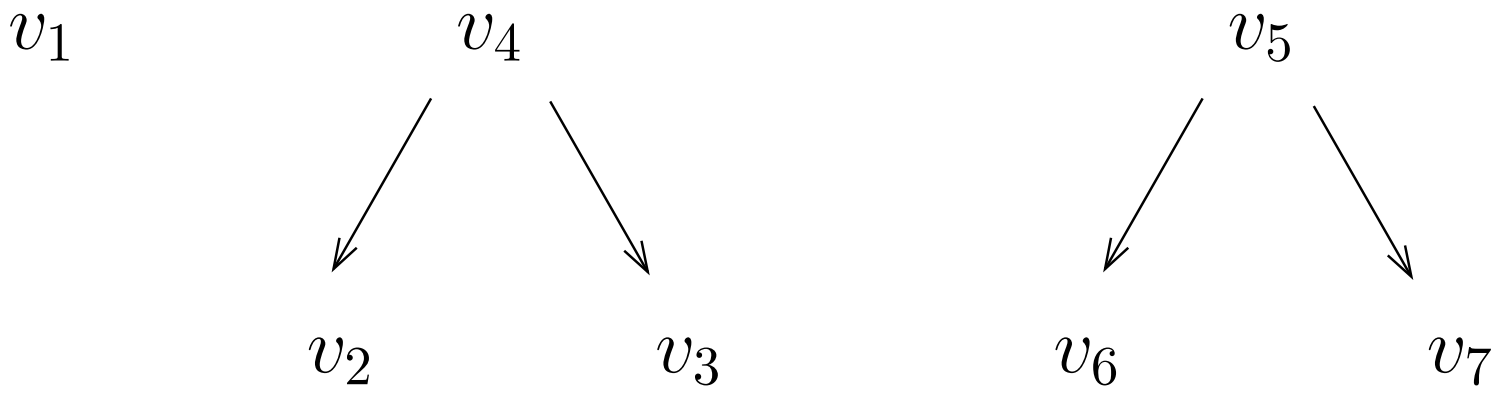
$$\frac{S_{j,k}u \to v,\ u \mapsto L,\ E}{t_{j,k} \approx v,\ u \mapsto L,\ E} \quad \text{if } C_j \notin L$$

Selector Rules

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

After Instantiate 1

$z \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \rightarrow v_1$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_6 \rightarrow v_2$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_7 \rightarrow v_3$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}\}$	
$\mathit{cons}(v_6, v_7) \rightarrow v_5$	$v_6 \mapsto \{\mathit{cons}, \mathit{null}\}$	
	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	

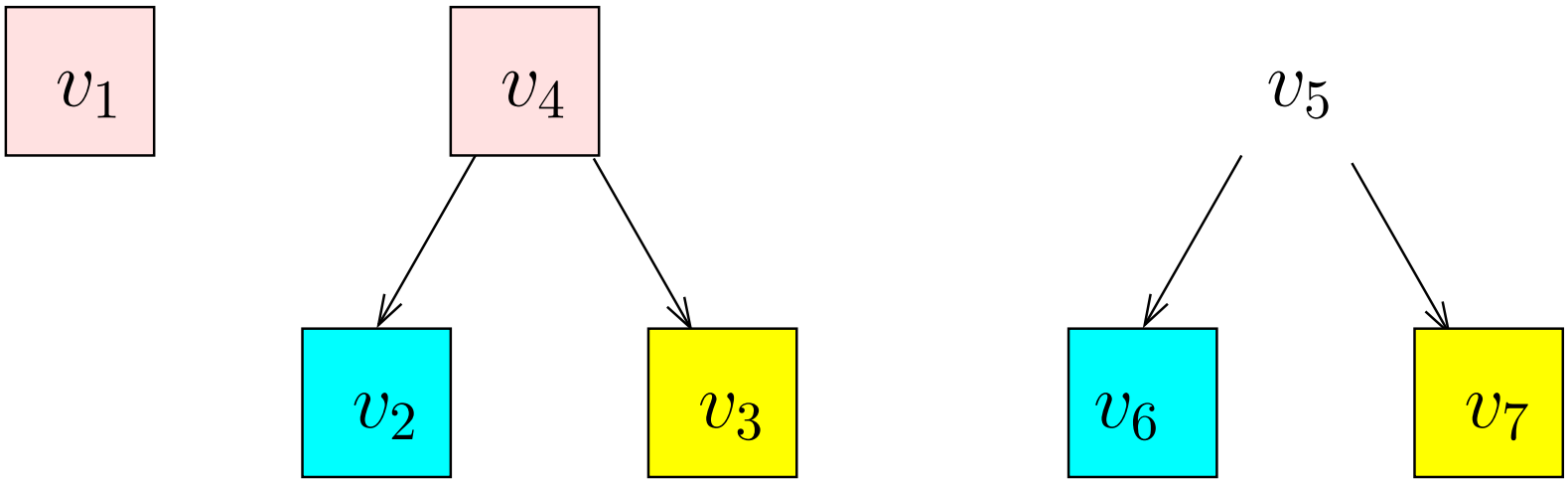


Selector Rules

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

After Instantiate 1

$z \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \rightarrow v_1$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_6 \rightarrow v_2$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_7 \rightarrow v_3$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}\}$	
$\mathit{cons}(v_6, v_7) \rightarrow v_5$	$v_6 \mapsto \{\mathit{cons}, \mathit{null}\}$	
	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	



Upward (i.e., congruence) closure rules

Simplify 1
$$\frac{u \bowtie t,\ u \to v,\ E}{v \bowtie t,\ u \to v,\ E}$$

Simplify 2
$$\frac{f\mathbf{u}u\mathbf{v} \to w,\ u \to v,\ E}{f\mathbf{u}v\mathbf{v} \to w,\ u \to v,\ E}$$

Compose
$$\frac{t \to v,\ v \to w,\ E}{t \to w,\ v \to w,\ E}$$

Superpose
$$\frac{t \to u,\ t \to v,\ E}{u \to v,\ t \to v,\ E} \quad \text{if } u \succ v$$

Upward (i.e., congruence) closure rules

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

After Simplify rules:

$z \longrightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \longrightarrow v_1$
$x \longrightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_6 \longrightarrow v_2$
$y \longrightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_7 \longrightarrow v_3$
$\mathit{cons}(v_2, v_3) \longrightarrow v_1$	$v_1 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \longrightarrow v_5$	$v_5 \mapsto \{\mathit{cons}\}$	
$\mathit{cons}(v_2, v_3) \longrightarrow v_5$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	
	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	

Upward (i.e., congruence) closure rules

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

After Simplify rules:

$z \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \rightarrow v_1$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_6 \rightarrow v_2$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_7 \rightarrow v_3$
$\mathit{cons}(v_2, v_3) \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}\}$	
$\mathit{cons}(v_2, v_3) \rightarrow v_5$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	
	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	

After Superpose: $v_5 \rightarrow v_1$

Upward (i.e., congruence) closure rules

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

After Simplify rules:

$z \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \rightarrow v_1$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_6 \rightarrow v_2$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_7 \rightarrow v_3$
$\mathit{cons}(v_2, v_3) \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}\}$	
$\mathit{cons}(v_2, v_3) \rightarrow v_5$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	
	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	

After Superpose: $v_5 \rightarrow v_1$

After Simplify: $v_1 \not\approx v_1$

Upward (i.e., congruence) closure rules

$\mathit{cons}(x, y) \approx z, x \approx \mathit{car}(w), y \approx \mathit{cdr}(w), w \not\approx z, \mathit{is\_cons}(w)$

After Simplify rules:

$z \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_4 \rightarrow v_1$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_6 \rightarrow v_2$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_7 \rightarrow v_3$
$\mathit{cons}(v_2, v_3) \rightarrow v_1$	$v_1 \mapsto \{\mathit{cons}\}$	$v_5 \not\approx v_1$
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}\}$	
$\mathit{cons}(v_2, v_3) \rightarrow v_5$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	
	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	

After Superpose: $v_5 \rightarrow v_1$

After Simplify: $v_1 \not\approx v_1$

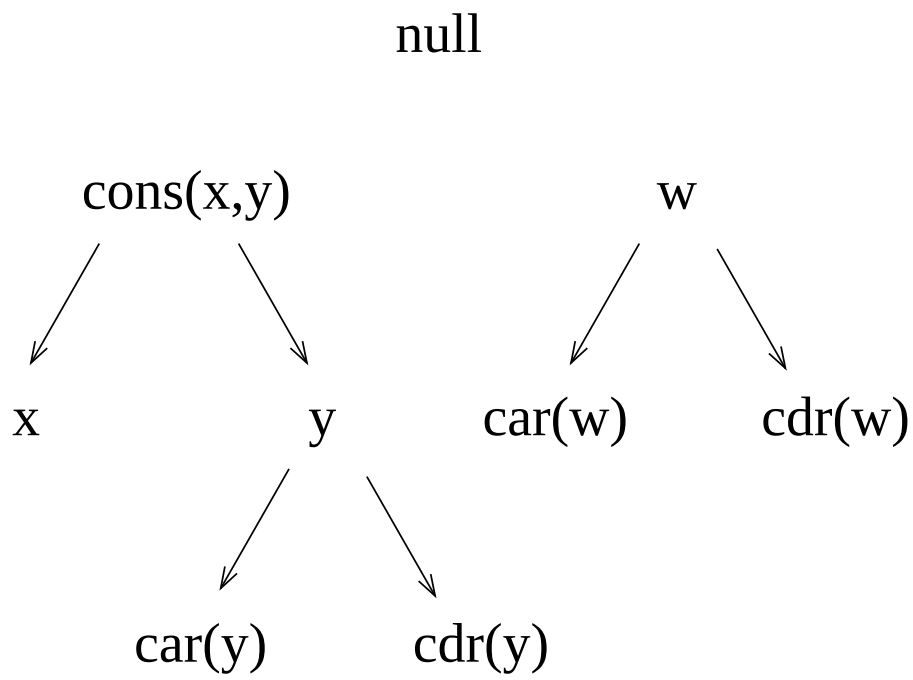
After Inconsistent: $\bot$

Another Example

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}$

After Abstraction and Orient:

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_7$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$w \rightarrow v_5$	$v_5 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{car}(v_5) \rightarrow v_6$	$v_6 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_5) \rightarrow v_7$	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_9$	$v_9 \mapsto \{\mathit{cons}, \mathit{null}\}$	

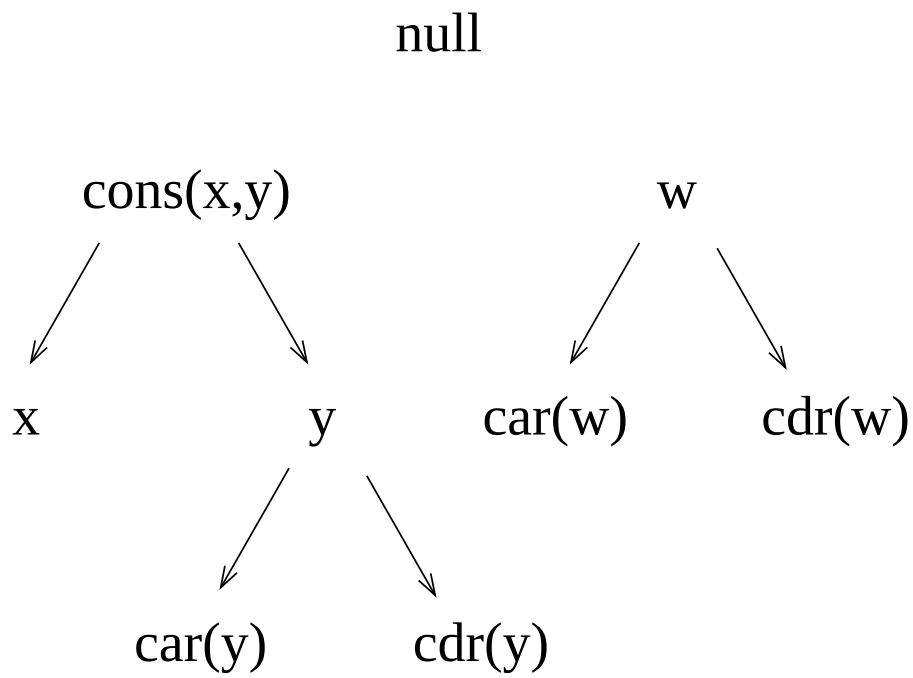


Another Example

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}$

After Simplifying:

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_7$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$w \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{car}(v_4) \rightarrow v_6$	$v_6 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_4) \rightarrow v_7$	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_7$	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	



Labeling rules

Refine
$$\frac{v \mapsto L_1,\ v \mapsto L_2,\ E}{v \mapsto L_1 \cap L_2,\ E}$$

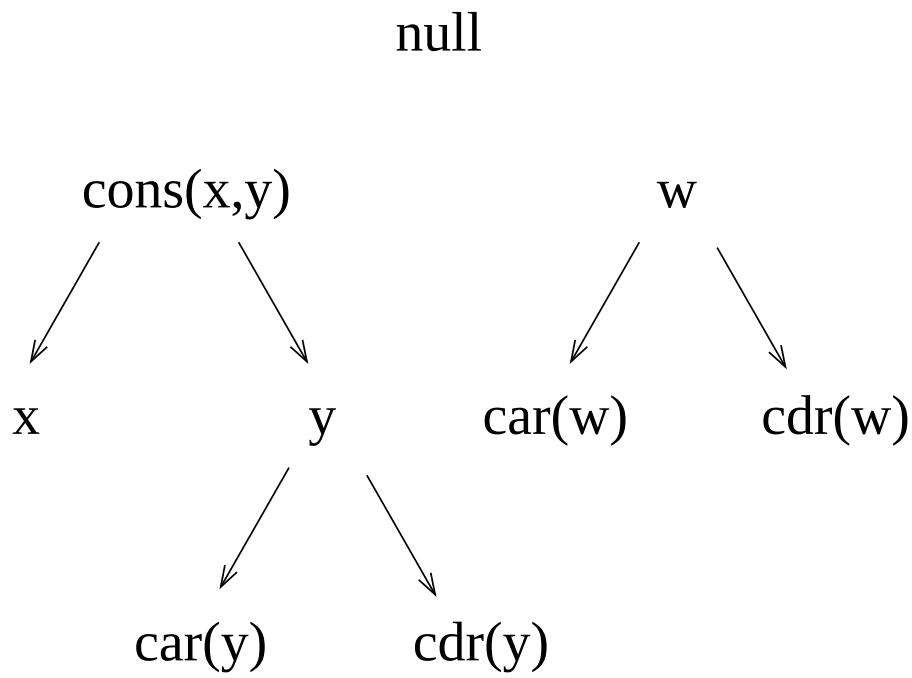
Empty
$$\frac{v \mapsto \emptyset,\ E}{\bot}$$

Another Example

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}$

After Refine and Instantiate:

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_7$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$w \rightarrow v_4$		
$\mathit{cons}(v_6, v_7) \rightarrow v_4$		
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_7$	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	

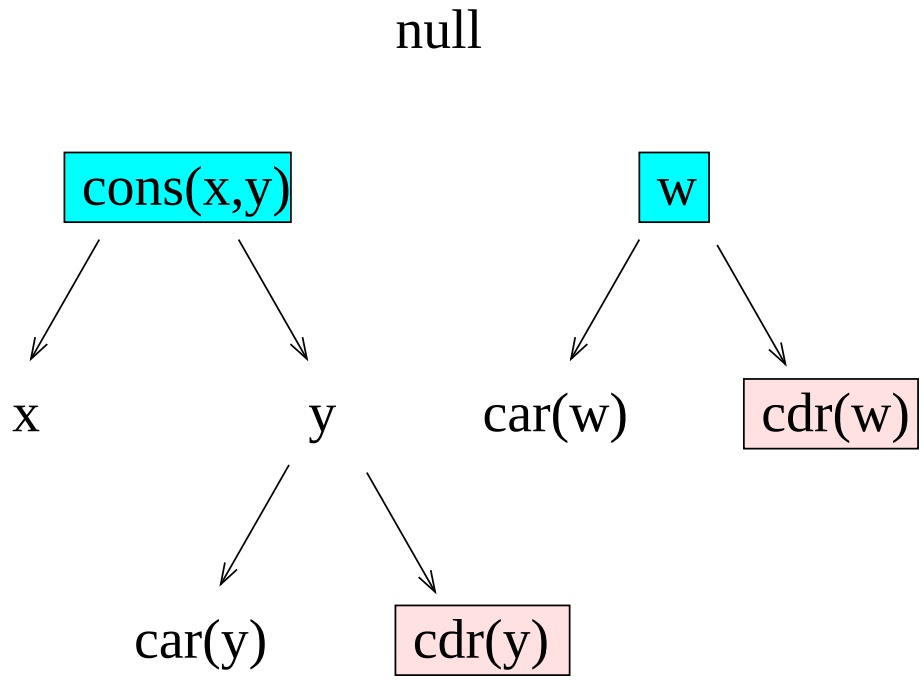


Another Example

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}$

After Refine and Instantiate:

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_7$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$w \rightarrow v_4$		
$\mathit{cons}(v_6, v_7) \rightarrow v_4$		
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_7$	$v_7 \mapsto \{\mathit{cons}, \mathit{null}\}$	



Downward (i.e., unification) closure rules

Decompose $$\frac{C_j\mathbf{u} \to v,\ C_j\mathbf{v} \to v,\ E}{C_j\mathbf{u} \to v,\ \mathbf{u} \to \mathbf{v},\ E}$$

Clash $$\frac{c \to v,\ d \to v,\ E}{\bot} \quad \text{if } c \neq d$$

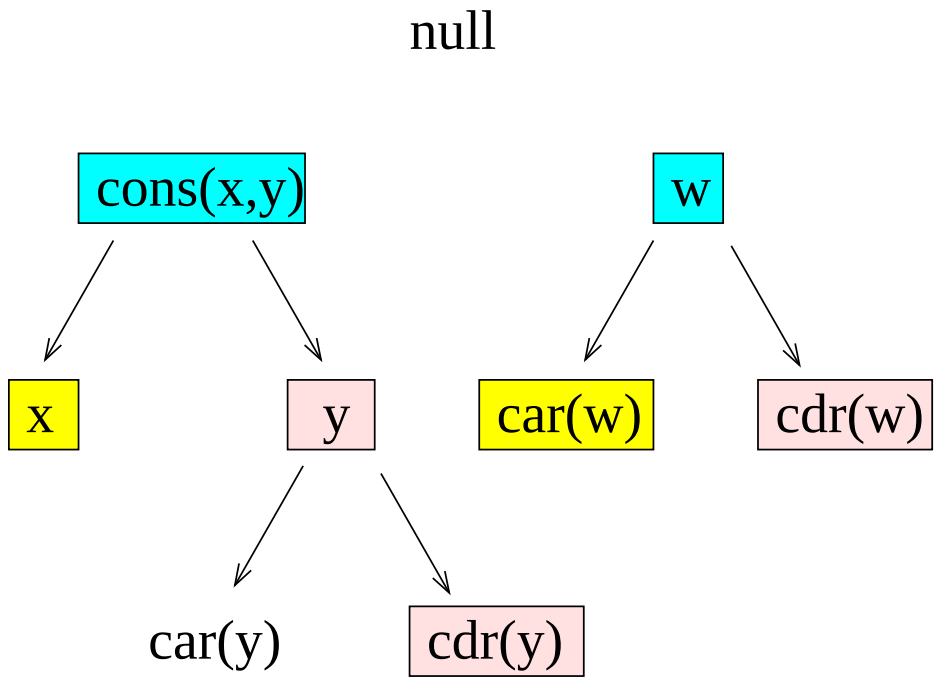
Cycle $$\frac{C_{j_n}\mathbf{u_n}u\mathbf{v_n} \to u_{n-1}, \ldots, C_{j_2}\mathbf{u_2}u_2\mathbf{v_2} \to u_1, C_{j_1}\mathbf{u_1}u_1\mathbf{v_1} \to u, E}{\bot}$$

Another Example

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}$

After Decompose, Orient, Simplify:

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		



Split Rules

Split 1

$$\frac{S_{j,k}(u) \to v,\ u \mapsto \{C_j\} \cup L,\ E}{S_{j,k}(u) \to v,\ u \mapsto \{C_j\},\ E \qquad\qquad S_{j,k}(u) \to v,\ u \mapsto L,\ E}$$

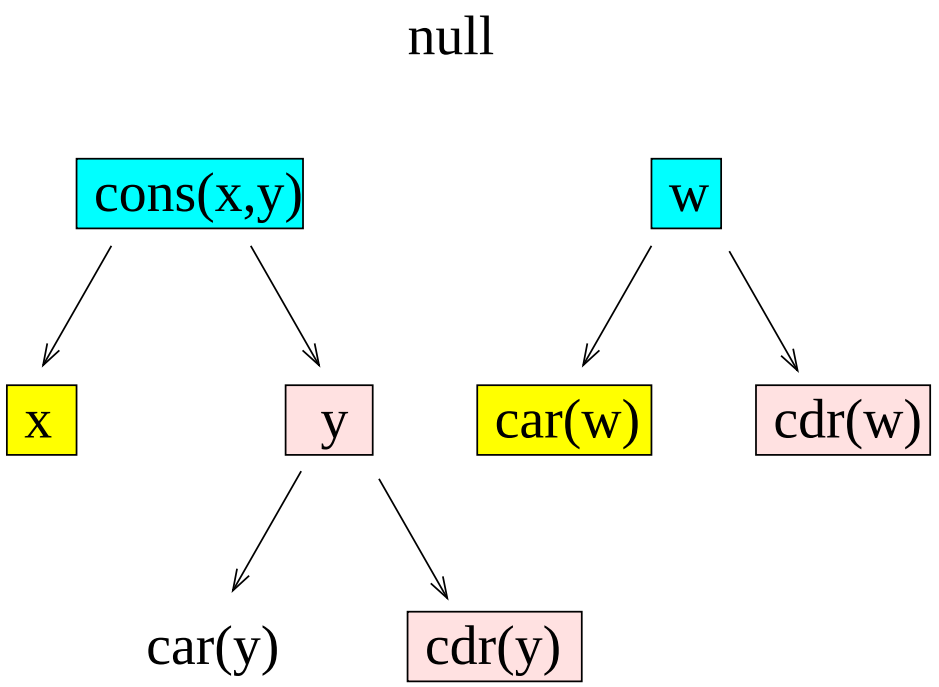
Split 2 Special case for finite constructors only.

Another Example

After Split 1: (first case)

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{cons}\}$

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		



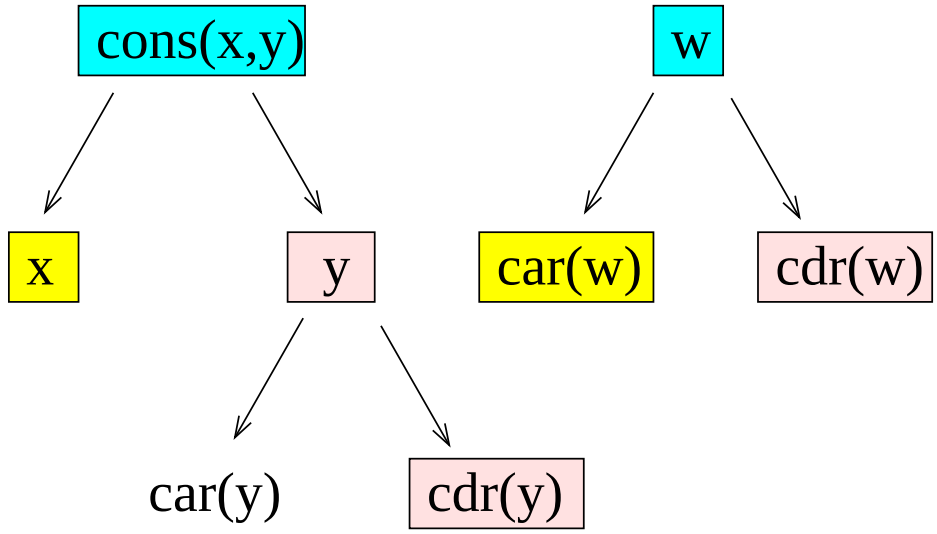
Another Example

After Split 1: (first case)

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{cons}\}$

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		

After Instantiate 1: $\mathit{cons}(v_8, v_3) \rightarrow v_3$



Another Example

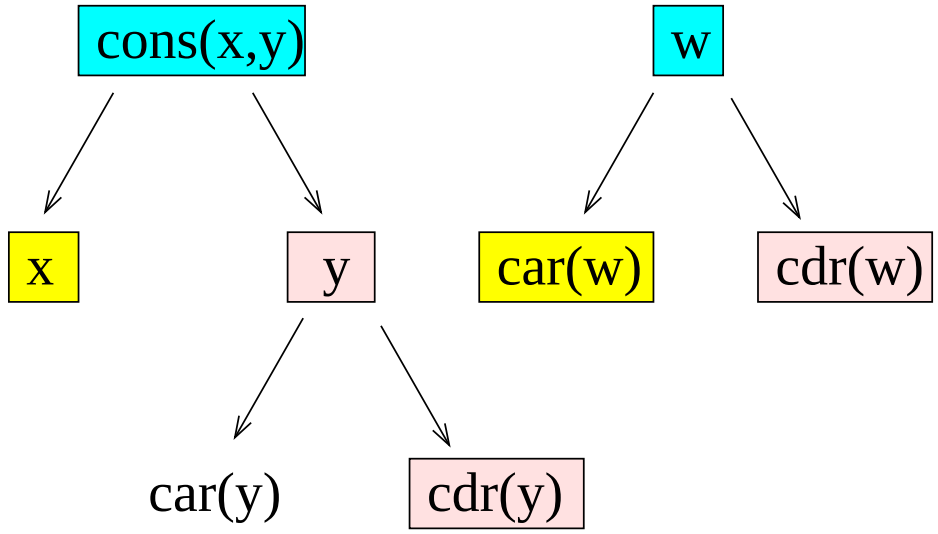
After Split 1: (first case)

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{cons}\}$

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{cons}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		

After Instantiate 1: $\mathit{cons}(v_8, v_3) \rightarrow v_3$

After Cycle: $\perp$

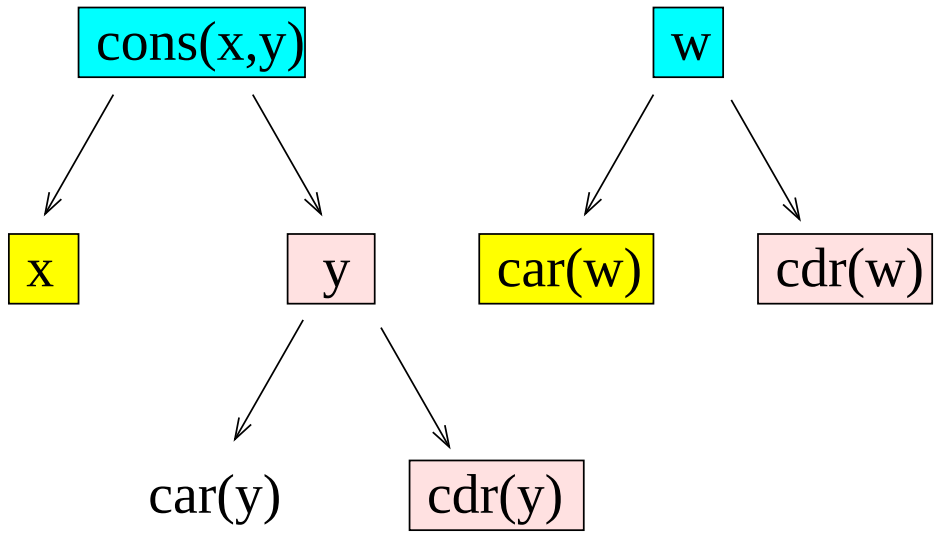


Another Example

After Split 1: (second case)

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{null}\}$

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		



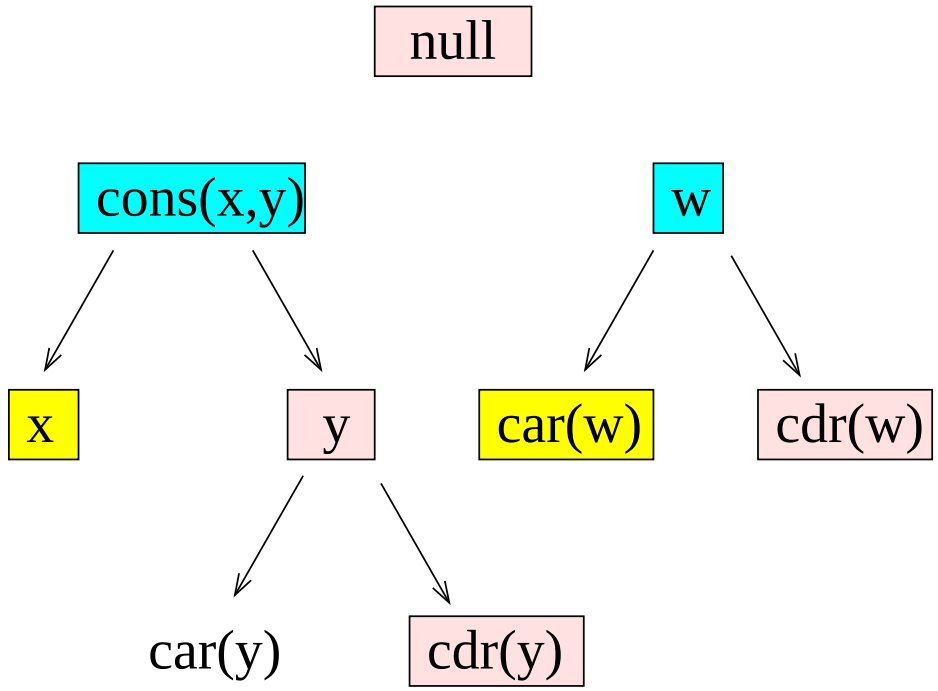
Another Example

After Split 1: (second case)

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{null}\}$

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		

After Instantiate 1: $\mathit{null} \rightarrow v_3$



Another Example

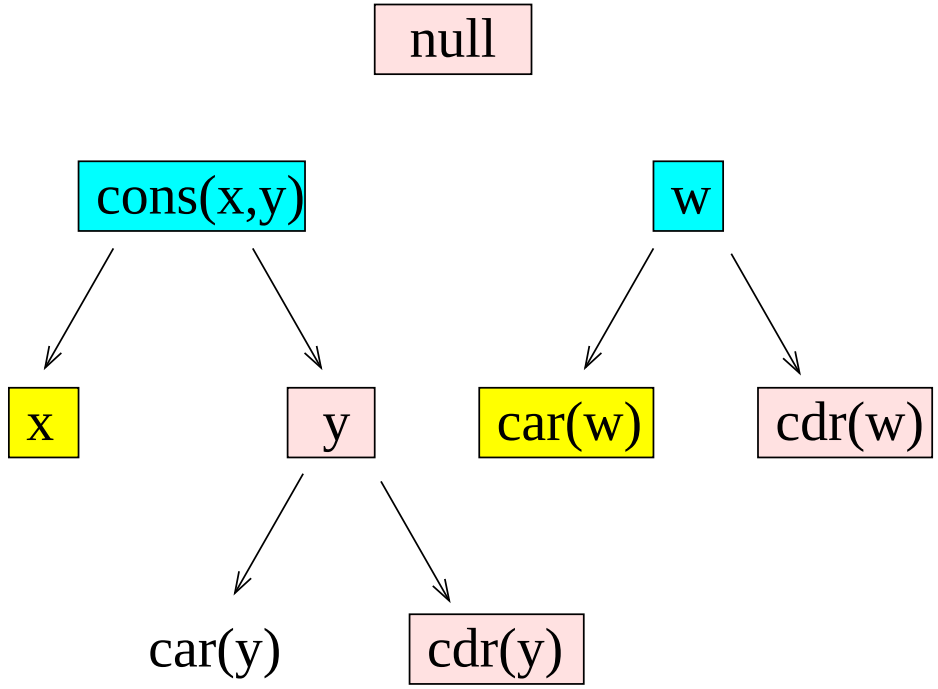
After Split 1: (second case)

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{null}\}$

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		

After Instantiate 1: $\mathit{null} \rightarrow v_3$

After Superpose: $v_3 \rightarrow v_1$



Another Example

After Split 1: (second case)

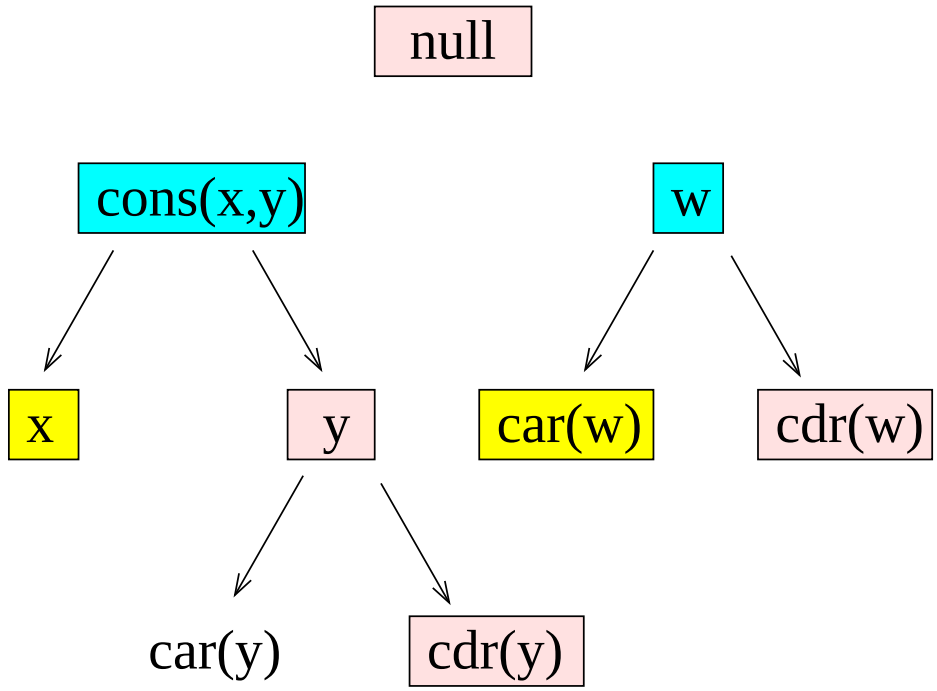
$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{null}\}$

$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		

After Instantiate 1: $\mathit{null} \rightarrow v_3$

After Superpose: $v_3 \rightarrow v_1$

After Simplify: $v_1 \not\approx v_1$



Another Example

After Split 1: (second case)

$\mathit{cons}(x, y) \approx w, \mathit{cdr}(w) \approx \mathit{cdr}(y), y \not\approx \mathit{null}, v_3 \mapsto \{\mathit{null}\}$

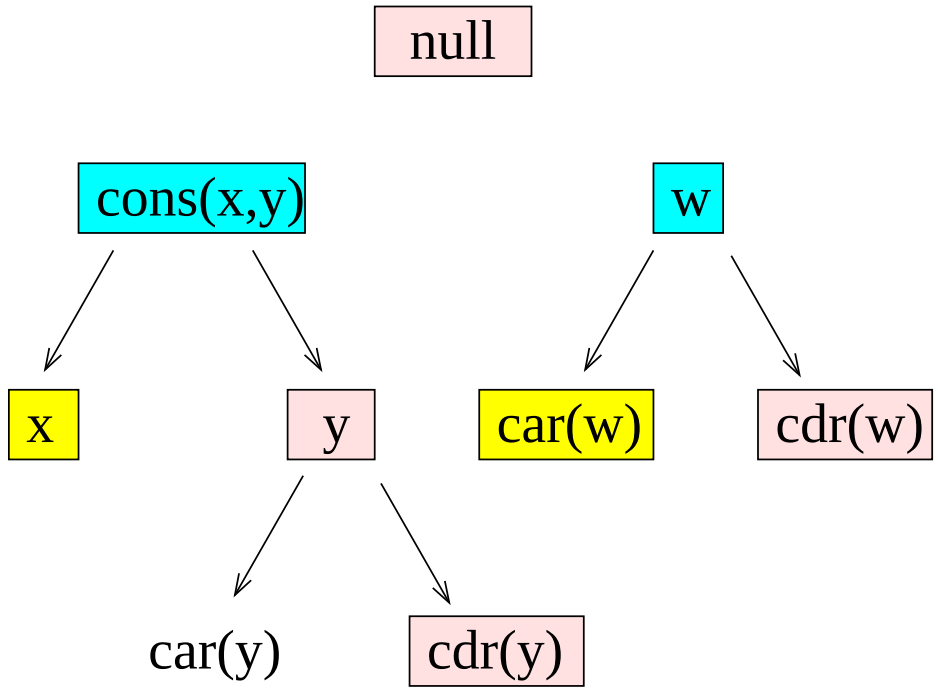
$\mathit{null} \rightarrow v_1$	$v_1 \mapsto \{\mathit{null}\}$	$v_5 \rightarrow v_4$
$x \rightarrow v_2$	$v_2 \mapsto \{\mathit{cons}, \mathit{null}\}$	$v_9 \rightarrow v_3$
$y \rightarrow v_3$	$v_3 \mapsto \{\mathit{null}\}$	$v_3 \not\approx v_1$
$\mathit{cons}(v_2, v_3) \rightarrow v_4$	$v_4 \mapsto \{\mathit{cons}\}$	
$\mathit{car}(v_3) \rightarrow v_8$	$v_8 \mapsto \{\mathit{cons}, \mathit{null}\}$	
$\mathit{cdr}(v_3) \rightarrow v_3$		
$v_6 \rightarrow v_2$		
$v_7 \rightarrow v_3$		

After Instantiate 1: $\mathit{null} \rightarrow v_3$

After Superpose: $v_3 \rightarrow v_1$

After Simplify: $v_1 \not\approx v_1$

After Inconsistent: $\perp$



Decision Procedure

Rules

- *Abstraction Rules:* Abstract 1, Abstract 2, Abstract 3
- *Literal Rules:* Orient, Inconsistent, Remove 1, Remove 2
- *Upward Rules:* Simplify 1, Simplify 2, Compose, Superpose
- *Downard Rules:* Decompose, Clash, Cycle
- *Selector Rules:* Instantiate 1, Instantiate 2, Collapse 1, Collapse 2
- *Labeling Rules:* Refine, Empty
- *Split Rules:* Split 1, Split 2

Correctness

- The rules have been proved *sound*, *complete*, and *terminating*
- Termination does not depend on the order in which rules are applied
- These results are available in an NYU tech report

Strategies

The type completion approach described in previous work is essentially equivalent to first applying the following naive Split rule:

Split $$\frac{u \mapsto \{C_j^i\} \cup L,\ E}{u \mapsto \{C_j^i\},\ E \qquad\qquad u \mapsto L,\ E}$$

Our approach includes two important contributions:

1. We have a *stronger side-condition* for the splitting rule
2. It is not hard to see that *splitting should be delayed* as long as possible. The description of the procedure as independent rules makes it easy to do this.

Experimental Results

We have implemented the procedure in *CVC3*.

We compared the type completion approach with our delayed splitting approach on 8000 randomly generated benchmarks.

Splits	Cases	Sat	Unsat	Type Completion		Delayed Splitting	
				Splits	Time (s)	Splits	Time (s)
0	4416	306	4110	0	24.6	0	24.6
1-5	2520	2216	304	6887	16.8	2414	17.0
6-10	692	571	121	4967	5.8	1597	5.7
11-20	178	112	66	2422	2.3	517	1.6
21-100	145	73	72	6326	4.5	334	1.1
101+	49	11	38	16593	9.8	73	0.3

Table 1: Type Completion vs. Delayed Splitting

Conclusion

Abstract presentation of decision procedure has several advantages

- Relatively easy to prove correct
- Easy to construct a variety of strategies
- Also easy to implement (implement one rule at a time)

Conclusion

Abstract presentation of decision procedure has several advantages

- Relatively easy to prove correct
- Easy to construct a variety of strategies
- Also easy to implement (implement one rule at a time)

Achieved our goals

- Can handle expressive RDTs
- Implementation is efficient

Conclusion

Abstract presentation of decision procedure has several advantages

- Relatively easy to prove correct
- Easy to construct a variety of strategies
- Also easy to implement (implement one rule at a time)

Achieved our goals

- Can handle expressive RDTs
- Implementation is efficient

Future work

- Applications
- Richer data structures