

A Polynomial-Time Fragment of Dominance Constraints

Alexander Koller

Kurt Mehlhorn

Joachmin Niehren

Presentation by Torsten Marek

February 9, 2009

Outline

- 1 Constraints
 - Dominance Constraints
 - Constraint Graphs
- 2 Normal Dominance Constraints
 - Solved Forms
- 3 Finding Solved Forms
 - Algorithm for Enumeration
 - Satisfiability
 - Evaluation
- 4 CLLS
- 5 Conclusion

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Informal Representation

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Default Example

Every linguist speaks two languages.

Informal Representation

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Default Example

Every linguist speaks two languages.

Logical Representation of the Possible Readings

$$\forall x.(\text{linguist}(x) \Rightarrow \exists_2 y.(\text{lang}(y) \wedge \text{speak}(x, y)))$$

$$\exists_2 y.(\text{lang}(y) \wedge \forall x.(\text{linguist}(x) \Rightarrow \text{speak}(x, y)))$$

Graph Representation

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

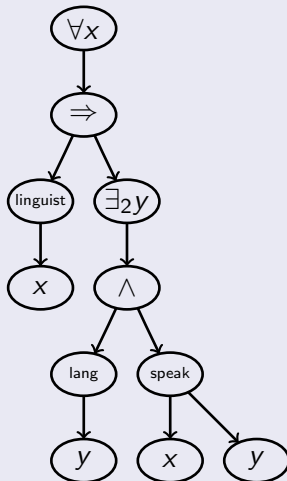
Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Graph Representation



Linking Graphs to Constraints

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Graph Representations

- 1 a constraint graph is a representation of dominance constraint
- 2 can be used as a link between dominance constraints and classic graph algorithms

Linking Graphs to Constraints

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Graph Representations

- 1 a constraint graph is a representation of dominance constraint
- 2 can be used as a link between dominance constraints and classic graph algorithms

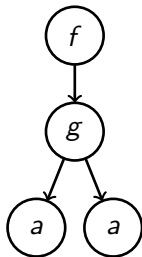
Needed

- 1 definition of dominance constraints
- 2 interpretation of dominance constraints as graphs

Trees & Labels

Trees

- there is a arbitrarily many symbols f, g, h, \dots with an arity $ar(f) \geq 0$.
- symbols with arity 0 (leaves) are written as a, b
- the tree $f(g(a, a))$ corresponds to the graph shown below
- there is a labeling function such that each edge and node can be assigned a variable.



The Syntax of Dominance Constraints

Dominance
Graphs

Constraints

Dominance
Constraints

Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Definition

A dominance constraint is a conjunction of dominance constraint primitives and evaluated on tree structures.

The Syntax of Dominance Constraints

Dominance Graphs

Constraints

Dominance Constraints

Constraint Graphs

Normal

Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for Enumeration

Satisfiability Evaluation

CLLS

Conclusion

Definition

A dominance constraint is a conjunction of dominance constraint primitives and evaluated on tree structures.

Formal Syntax Definition

$$\phi ::= \phi \wedge \phi' \mid X \triangleleft^* Y \mid X \neq Y \mid X : f(X_1, \dots, X_n)$$

Primitives of Dominance Constraints

Dominance
Graphs

Constraints

Dominance
Constraints

Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Dominance

$X \triangleleft^* Y$: there is a path from X to Y

Primitives of Dominance Constraints

Dominance
Graphs

Dominance

$X \triangleleft^* Y$: there is a path from X to Y

Inequality

$X \neq Y$: X and Y are never the same node.

Constraints

Dominance
Constraints

Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Primitives of Dominance Constraints

Dominance
Graphs

Dominance

$X \triangleleft^* Y$: there is a path from X to Y

Constraints

Dominance
Constraints

Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Inequality

$X \neq Y$: X and Y are never the same node.

Labeling

$X:f(X_1, \dots, X_n)$: X is a n -ary node with children X_1 to X_n . If the arity of f is 0, it is a constant.

Dominance Constraints

Dominance Graphs

Constraints

Dominance Constraints

Constraint Graphs

Normal

Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for Enumeration

Satisfiability Evaluation

CLLS

Conclusion

“Every linguist...” formalized

$$\begin{aligned}\phi = & X':f(X) \wedge X:g(X_1, X_2) \wedge X_1:h(x) \\ & \wedge Y':i(Y) \wedge Y:j(Y_1, Y_2) \wedge Y_1:k(y) \\ & \wedge Z:l(x, y) \wedge X_2 \triangleleft^* Z \wedge Y_2 \triangleleft^* Z\end{aligned}$$

Drawing Dominance Constraints as Graphs

Dominance Graphs

Constraints

Dominance
Constraints

Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration

Satisfiability
Evaluation

CLLS

Conclusion

Interpretation of Primitives

- labeling constraints become *tree* edges (solid)
 - if $X:f(\dots, Y, \dots) \in \phi$, then no other node can occur on the path from X to Y
- dominance constraints become *dominance* edges (dashed)
 - if $X \triangleleft^* Y \in \phi$, then Y must be reachable from X
- labels on nodes are ordered from left to right
- inequality constraints are not drawn

Sample Dominance Graph

Dominance Graphs

Constraints

Dominance Constraints

Constraint Graphs

Normal

Dominance Constraints

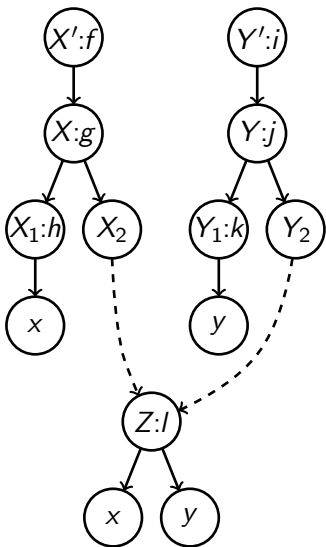
Solved Forms

Finding Solved Forms

Algorithm for Enumeration Satisfiability Evaluation

CLLS

Conclusion



Original Formula

$$\begin{aligned}\phi = & X':f(X) \wedge X:g(X_1, X_2) \\ & \wedge X_1:h(x) \wedge Y':j(Y) \\ & \wedge Y:j(Y_1, Y_2) \wedge Y_1:k(y) \\ & \wedge Z:l(x, y) \wedge X_2 \triangleleft^* Z \\ & \wedge Y_2 \triangleleft^* Z\end{aligned}$$

Major Questions

Dominance
Graphs

Constraints

Dominance
Constraints

Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Natural Problems on Dominance Constraints

- satisfiability, i.e. is there a graph that satisfies a given dominance constraint
- if a constraint is satisfiable, enumerate a solutions

Major Questions

Dominance
Graphs

Constraints

Dominance
Constraints

Constraint
Graphs

Normal

Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration

Satisfiability
Evaluation

CLLS

Conclusion

Natural Problems on Dominance Constraints

- satisfiability, i.e. is there a graph that satisfies a given dominance constraint
- if a constraint is satisfiable, enumerate a solutions

NP-hardness

Testing if a dominance constraint is satisfiable is NP-hard because variables may be bound to the same node if no inequality constraint exists.

Major Questions

Dominance
Graphs

Natural Problems on Dominance Constraints

- satisfiability, i.e. is there a graph that satisfies a given dominance constraint
- if a constraint is satisfiable, enumerate a solutions

Constraints

Dominance
Constraints

Constraint
Graphs

Normal

Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

NP-hardness

Testing if a dominance constraint is satisfiable is NP-hard because variables may be bound to the same node if no inequality constraint exists.

Solution

Limit dominance constraint to a reasonable subset to be able to solve both problems in polynomial time.

Overview

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- 1 Constraints
 - Dominance Constraints
 - Constraint Graphs
- 2 Normal Dominance Constraints
 - Solved Forms
- 3 Finding Solved Forms
 - Algorithm for Enumeration
 - Satisfiability
 - Evaluation
- 4 CLLS
- 5 Conclusion

Restrictions of Normal Dominance Constraints I

Restrictions

- dominance edges only go from holes to roots: if $X \triangleleft^* Y$, then X is unlabeled and Y is labeled but never appears as a child in a labeling literal
- $X \neq Y$ if both X, Y are labeled in ϕ .

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Restrictions of Normal Dominance Constraints I

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

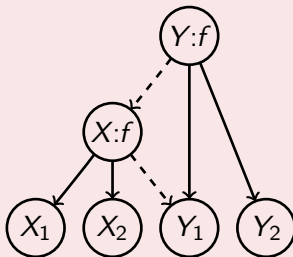
CLLS

Conclusion

Restrictions

- dominance edges only go from holes to roots: if $X \triangleleft^* Y$, then X is unlabeled and Y is labeled but never appears as a child in a labeling literal
- $X \neq Y$ if both X, Y are labeled in ϕ .

Not normal



Normal Dominance Constraints II

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Other Restrictions

- solid fragments are tree-shared or cyclic: every variable in ϕ appears at most once as a parent and at most once as a child in a labeling literal
- no empty fragments: every hole of ϕ occurs in child position

Normal Dominance Constraints II

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Other Restrictions

- solid fragments are tree-shared or cyclic: every variable in ϕ appears at most once as a parent and at most once as a child in a labeling literal
- no empty fragments: every hole of ϕ occurs in child position

Invalid Formula

$$X:f(X_1, X_2) \wedge X:g(X_3)$$

Solutions vs. Solved Forms

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Solutions

If a constraint is satisfiable, it has infinitely many solutions.

Solutions vs. Solved Forms

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Solutions

If a constraint is satisfiable, it has infinitely many solutions.

Constraints are *partial* Graphs

$X:a$: all trees that have a node a

$X:f(X_1, X_2)$: all trees that have a node with two children

Solutions vs. Solved Forms

Dominance
Graphs

Solutions

If a constraint is satisfiable, it has infinitely many solutions.

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Constraints are *partial* Graphs

$X:a$: all trees that have a node a

$X:f(X_1, X_2)$: all trees that have a node with two children

Solved Forms

A solved form only describes the *relevant* parts of a solution.

The Reachability Relation R_ϕ

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Construction

① initialize

$$R_\phi := \{(X, Y) \mid X:f(\dots, Y, \dots) \in \phi \vee X \triangleleft^* Y \in \phi\}$$

② $\forall X, Y, Z (X, Y) \in R_\phi \wedge (Y, Z) \in R_\phi \Leftrightarrow R_\phi := R_\phi \cup \{(X, Z)\}$

③ repeat 2 until no more items can be added

The Reachability Relation R_ϕ

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Construction

① initialize

$$R_\phi := \{(X, Y) \mid X:f(\dots, Y, \dots) \in \phi \vee X \triangleleft^* Y \in \phi\}$$

② $\forall X, Y, Z (X, Y) \in R_\phi \wedge (Y, Z) \in R_\phi \Leftrightarrow R_\phi := R_\phi \cup \{(X, Z)\}$

③ repeat 2 until no more items can be added

Result

If Y can be reached from X , then $(X, Y) \in R_\phi$.

Definition: Solved Forms I

Dominance
Graphs

One Incoming Dominance Edge

If $X \neq Y$, then only one of $X \triangleleft^* Z$, $Y \triangleleft^* Z$ occurs
in ϕ

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Definition: Solved Forms I

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

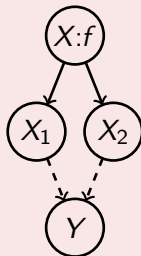
CLLS

Conclusion

One Incoming Dominance Edge

If $X \neq Y$, then only one of $X \triangleleft^* Z$, $Y \triangleleft^* Z$ occurs
in ϕ

Invalid Upbranch



Definition: Solved Forms II

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal

Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

No cycles

$$(X, X) \notin R_\phi$$

Definition: Solved Forms II

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal

Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

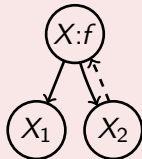
CLLS

Conclusion

No cycles

$$(X, X) \notin R_\phi$$

Graph with Dominance Cycle



Solved Forms: Results

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Forests

A normal dominance constraint is in solved form iff its constraint graph is a forest, i.e. a graph that is a conjunction of trees.

It can be shown that every normal dominance constraint in solved form has a solution.

Solved Forms: Results

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Forests

A normal dominance constraint is in solved form iff its constraint graph is a forest, i.e. a graph that is a conjunction of trees.

It can be shown that every normal dominance constraint in solved form has a solution.

Proof Idea

- 1 dominance constraint in solved form is already a forest

Solved Forms: Results

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Forests

A normal dominance constraint is in solved form iff its constraint graph is a forest, i.e. a graph that is a conjunction of trees.

It can be shown that every normal dominance constraint in solved form has a solution.

Proof Idea

- 1 dominance constraint in solved form is already a forest
- 2 by successively replacing dominance edges by tree edges, we get a solution for a solved form

Example

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

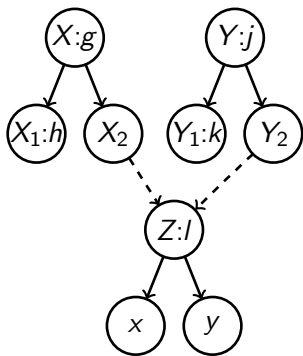
Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

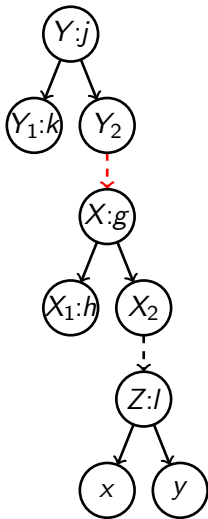
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion



Normal constraint, not in solved form.



One possible solved form of the graph.



Overview

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- 1 Constraints
 - Dominance Constraints
 - Constraint Graphs
- 2 Normal Dominance Constraints
 - Solved Forms
- 3 Finding Solved Forms
 - Algorithm for Enumeration
 - Satisfiability
 - Evaluation
- 4 CLLS
- 5 Conclusion

Enumeration of Solved Forms

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Result from previous section

From the previous section, we know that in order to find solutions for a dominance constraint, we only need its solved forms.

Enumeration of Solved Forms

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Result from previous section

From the previous section, we know that in order to find solutions for a dominance constraint, we only need its solved forms.

Needed

We need an algorithm to

- 1 find out if a given normal dominance constraint is satisfiable
- 2 create all its solved forms

Simplification Rules

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

1: Transitivity Elimination

All redundant *dominance* edges which are implied by transitivity can be removed.

Simplification Rules

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

1: Transitivity Elimination

All redundant *dominance* edges which are implied by transitivity can be removed.

2: Choice

If $X \triangleleft^* Y$ and $X' \triangleleft^* Y$ and R, R' are the roots of the leaves X, X' , the create two a new graph for each of

- $X \triangleleft^* R'$
- $X' \triangleleft^* R$

Enumeration Algorithm

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Outline

- 1 apply simplification rule
- 2 if the graph is unsatisfiable, stop processing the current graph
- 3 if the graph is in solved form
 - 1 report it and stop processing
 - 2 otherwise, apply the choice rule and recursively run the algorithm with both new graphs

Sample Iteration

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

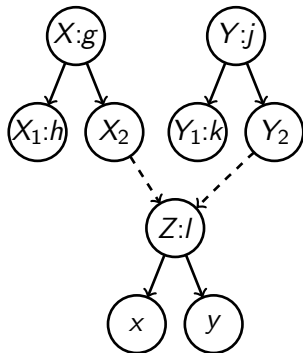
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Steps

1 apply rule 2



Sample Iteration

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal

Dominance
Constraints
Solved Forms

Finding Solved Forms

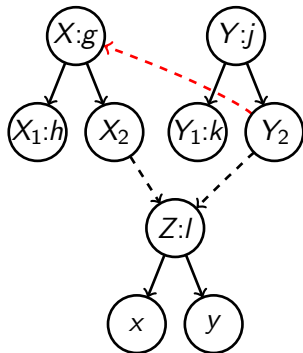
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Steps

- 1 apply rule 2
- 2 create graph with new dominance edge



Sample Iteration

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal

Dominance
Constraints
Solved Forms

Finding Solved Forms

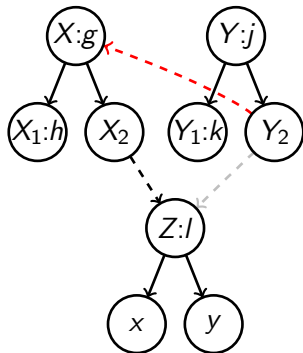
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Steps

- 1 apply rule 2
- 2 create graph with new dominance edge
- 3 apply rule 1



Sample Iteration

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal

Dominance
Constraints
Solved Forms

Finding Solved Forms

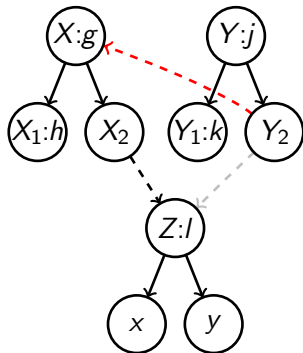
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Steps

- 1 apply rule 2
- 2 create graph with new dominance edge
- 3 apply rule 1
- 4 solved form!



Sample Iteration

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal

Dominance
Constraints
Solved Forms

Finding Solved Forms

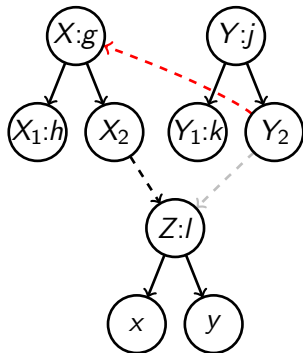
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Steps

- 1 apply rule 2
- 2 create graph with new dominance edge
- 3 apply rule 1
- 4 solved form!
- 5 repeat for $X_2 \triangleleft^* Y:j$



Testing for Satisfiability

Dominance Graphs

Constraints

Dominance
Constraints

Constraint
Graphs

Normal

Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration

Satisfiability
Evaluation

CLLS

Conclusion

Directed Cycles

Check for a directed cycle using dominance and tree edges (cf. solved form).

Testing for Satisfiability

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Directed Cycles

Check for a directed cycle using dominance and tree edges (cf. solved form).

Hypernormal Cycles

A faster check (in linear time) can be done using hypernormal cycles. If a graph has a hypernormal cycle, it is unsatisfiable.

Hypernormal Cycles

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Undirected Graphs

- convert a directed dominance graph to undirected one
- check for certain cycles in the undirected version
- a cycle in an undirected graph visits each node in the graph at most once

Hypernormal Cycles

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Undirected Graphs

- convert a directed dominance graph to undirected one
- check for certain cycles in the undirected version
- a cycle in an undirected graph visits each node in the graph at most once

Hypernormal Cycle

A hypernormal cycle in an undirected dominance graph is a simple cycle such that for all visited nodes, at most one outgoing dominance edge is traversed.

Hypernormal Cycles: Examples

Dominance
Graphs

No Hypernormal Cycle



Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration

Satisfiability
Evaluation

CLLS

Conclusion

Hypernormal Cycles: Examples

Dominance
Graphs

No Hypernormal Cycle



\Rightarrow
choice
rule

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration

Satisfiability
Evaluation

CLLS

Conclusion

Hypernormal Cycles: Examples

Dominance Graphs

Constraints

Dominance Constraints
Constraint Graphs

Normal Dominance Constraints

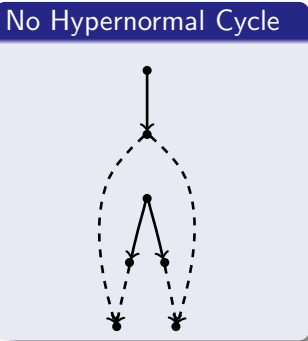
Solved Forms

Finding Solved Forms

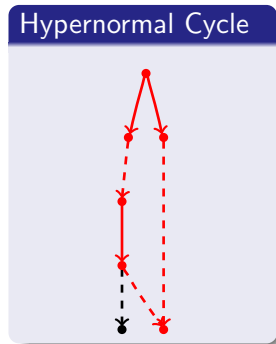
Algorithm for Enumeration
Satisfiability Evaluation

CLLS

Conclusion



\Rightarrow
choice
rule



Hypernormal Cycles: Examples

Dominance
Graphs

No Hypernormal Cycle



\Rightarrow
choice
rule

Hypernormal Cycle



The graph is unsatisfiable!

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Testing for Hypernormal Cycles

Dominance
Graphs

Constraints

Dominance
Constraints

Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration

Satisfiability
Evaluation

CLLS

Conclusion

Testing

Testing for hypernormal cycles can be done using matching graph algorithms.

Testing for Hypernormal Cycles

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Testing

Testing for hypernormal cycles can be done using matching graph algorithms.

Testing in Linear Time

If each node has at most one outgoing dominance edge, testing for hypernormal cycles can be done in linear time.

Runtime

Dominance
Graphs

Embedded chains

Jon says that every linguist speaks two languages.
is an embedded chain of length two.

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Runtime

Embedded chains

Jon says that every linguist speaks two languages.
is an embedded chain of length two.

Results

| k | Time (new) | Time (old) |
|---|------------|------------|
| 3 | 20 | 180 |
| 4 | 190 | 670 |
| 5 | 1210 | 5900 |
| 6 | 4130 | 12740 |
| 7 | 16630 | 46340 |
| 8 | 255000 | n/a |

Table: k: length of chain, times in ms CPU time

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Overview

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- 1 Constraints
 - Dominance Constraints
 - Constraint Graphs
- 2 Normal Dominance Constraints
 - Solved Forms
- 3 Finding Solved Forms
 - Algorithm for Enumeration
 - Satisfiability
 - Evaluation
- 4 CLLS
- 5 Conclusion

Constraint Language for Lambda Structures

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Unified Analysis for ...

- scope
 - *Every linguist speaks two languages.*

Constraint Language for Lambda Structures

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms
Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Unified Analysis for ...

- scope
 - *Every linguist speaks two languages.*
- ellipsis
 - *Everybody lies, especially in an election year.*

Constraint Language for Lambda Structures

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal

Dominance
Constraints

Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Unified Analysis for ...

- scope
 - *Every linguist speaks two languages.*
- ellipsis
 - *Everybody lies, especially in an election year.*
- anaphora
 - *Bush said he thinks that the bible is not literally true.*

Constraint Language for Lambda Structures

Dominance
Graphs

Constraints
Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Unified Analysis for ...

- scope
 - *Every linguist speaks two languages.*
- ellipsis
 - *Everybody lies, especially in an election year.*
- anaphora
 - *Bush said he thinks that the bible is not literally true.*

Basic Idea

- write λ terms as tree-like lambda structures
- use dominance constraints in lambda structures

Conclusion

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- notion of dominance constraints ...
 - ... based on logical formulae
 - ... have a graph-theoretic interpretation, not just graph-based representation

Conclusion

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- notion of dominance constraints ...
 - ... based on logical formulae
 - ... have a graph-theoretic interpretation, not just graph-based representation
- checking satisfiability of dominance constraints is NP-hard

Conclusion

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- notion of dominance constraints ...
 - ... based on logical formulae
 - ... have a graph-theoretic interpretation, not just graph-based representation
- checking satisfiability of dominance constraints is NP-hard
- linguistic phenomena only require normal dominance constraints, which are a restricted subset of dominance constraints

Conclusion

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- notion of dominance constraints ...
 - ... based on logical formulae
 - ... have a graph-theoretic interpretation, not just graph-based representation
- checking satisfiability of dominance constraints is NP-hard
- linguistic phenomena only require normal dominance constraints, which are a restricted subset of dominance constraints
- normal dominance constraints can be checked for satisfiability in polynomial time

Conclusion

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal Dominance Constraints

Solved Forms

Finding Solved Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

- notion of dominance constraints ...
 - ... based on logical formulae
 - ... have a graph-theoretic interpretation, not just graph-based representation
- checking satisfiability of dominance constraints is NP-hard
- linguistic phenomena only require normal dominance constraints, which are a restricted subset of dominance constraints
- normal dominance constraints can be checked for satisfiability in polynomial time
- efficient algorithm for enumerating all solved forms of a dominance constraint

References

Dominance Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion



Althaus, E., Duchier, D., Koller, A., Mehlhorn, K., Niehren, J., and Thiel, S. (2003).

An efficient graph algorithm for dominance constraints.
Journal of Algorithms, pages 194–219.



Koller, A., Mehlhorn, K., and Niehren, J. (2000).

A polynomial-time fragment of dominance constraints.
In *Proceedings of the 38th Annual Meeting of the ACL*,
Hong Kong.

Thank You for Your Attention

Dominance
Graphs

Constraints

Dominance
Constraints
Constraint
Graphs

Normal
Dominance
Constraints
Solved Forms

Finding Solved
Forms

Algorithm for
Enumeration
Satisfiability
Evaluation

CLLS

Conclusion

Questions?