Inter-Hierarchy Comparison in HCLP

Roman Barták

Department of Theoretical Computer Science
Charles University, Prague

e-mail: bartak@kti.mff.cuni.cz
URL: http://kti.ms.mff.cuni.cz/~bartak

Motivation

- over-constrained systems (no solution)
- expressive power of constraints
  - preferences
  - disjunctions
- modelling real problems
  - optimization techniques
- stability of the solution

© Roman Barták, 1998

Constraint Hierarchies

- approach to solving over-constrained systems
- hard and soft constraints
  - hierarchy of labeled (preferential) constraints
- solution
  - stronger constrains are strictly preferred to weaker constraints
  - comparators (local, regional, global)
- relaxing of constraints (combining function)

© Roman Barták, 1998

Constraint Hierarchies - Example

- hierarchy:

```
required A+B=C
  strong A=3
    weak B=3
    weak C=5
```

- solution: (depends on the comparator)
  - local comparator:

    A=3,B=3,C=6

    A=3,B=2,C=5
  - global (metric) comparator:

    A=3,B=2.5,C=5.5 (worst-case metric)

© Roman Barták, 1998

Inter-Hierarchy Comparison

- Example:

```
f(X):-strong X>3, g(X).
g(1).
g(5).
        required X=1, strong X>3
        required X=5, strong X>3
```

- comparison of solutions arising from different hierarchies
- rules out unintuitive solutions
- global comparators (combining function)

© Roman Barták, 1998

Inter-Hierarchy Comp. - Example

```
• can_meet(Person1,Person2,Day,Time,Length):-
     free(Person1,Day,StartTime1,EndTime1),
     free(Person2,Day,StartTime2,EndTime2),
     required StartTime1≤Time,
     required StartTime2≤Time,
     required Time+Length≤EndTime1,
     required Time+Length≤EndTime2.

  free(judy,saturday,11,12).
  free(judy,friday,13,14).
  free(chris,saturday,10,13).
  free(chris,friday,T,16):-
     required T≥12, prefer T≥14.

• Day=saturday (Length=1)
    11≤Time, Time+1≤12, 10≤Time, Time+1≤13
  Day=friday (Length=1)
    13≤Time, 12≤Time, Time+1≤14, Time+1≤16, prefer 14≤Time
```

© Roman Barták, 1998

HCLP

- Hierarchical Constraint Logic Programming
- natural framework for
  - constraint hierarchies (labeled constraints)
  - inter-hierarchy comparison (via alternative rules)
- declarative style of programming
- constraints prune the search space
  - improved efficiency
- practical applications

© Roman Barták, 1998

Benefits

- representation of over-constrained systems
  - via weakening some constraints
- natural expressivity
  - preferential constraints
  - disjunctions (unlimited)
- optimization techniques
  - weak constraints (`Optimum=CritFunction`)
  - combinations of criteria functions
- stability of the solution
  - via very weak constraints (`Var=PrevValue`)

© Roman Barták, 1998

Initial Situation

- constraint hierarchies are used sparingly
  - GUIs, geometric layout, document formatting
- hierarchical solvers for local comparators
- inter-hierarchy comparison
  - high potential
  - scarce usage (if at all)
- efficient algorithm needed

© Roman Barták, 1998

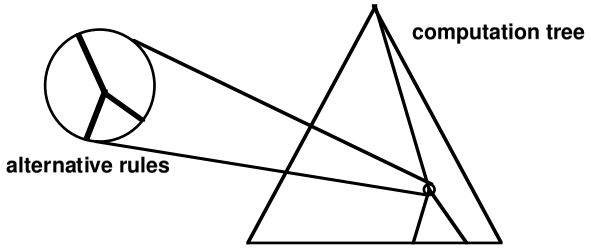
Background

- problematic features
  - disorderly property ($S_H \supsetneq S_{H\cup\{c\}}$)
  - non-monotonicity ($S_\Delta \not\subseteq S_{\Delta\cup\Gamma}$)
- error vector EV
  - vector of combined errors at each level
  - `strong a+b=5, prefer a=3, weak b=3`
    σ={a/3,b/2} -> $EV_{H,\sigma}$=[0,0,1]
  - $EV_H = \min\{EV_{H,\sigma}\}$
  - monotony $EV_H \leq EV_{H\cup\{c\}}$
  - persistency $EV_H < EV_J \Rightarrow EV_H < EV_{J\cup\{c\}}$

© Roman Barták, 1998

Behind the Algorithm

- inter-hierarchy comparison within HCLP

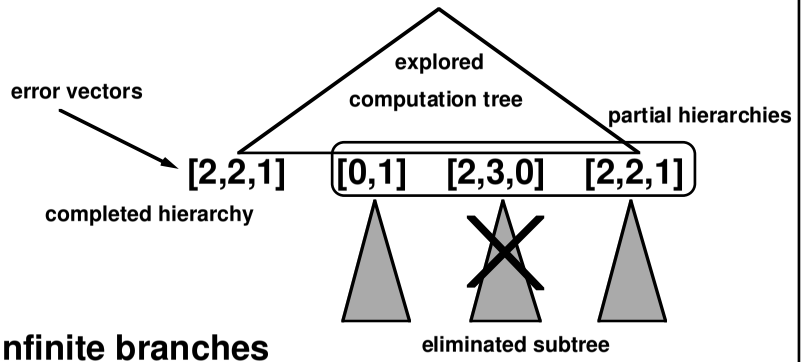


- error vectors
  - early elimination of useless branches

© Roman Barták, 1998

The Algorithm

- breadth-first search
- switching between reduction and elimination



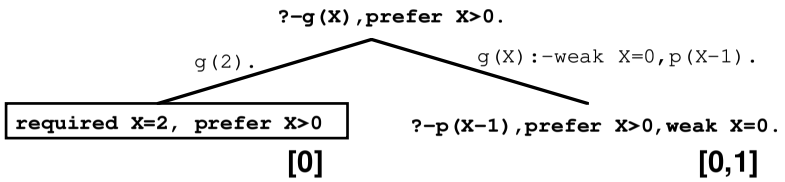
- infinite branches

© Roman Barták, 1998

Example

- HCLP program (over integers)

```
g(2).
g(X):-weak X=0, p(X-1).
p(1).
p(X):-p(X-1).
```



- solution X=2

© Roman Barták, 1998

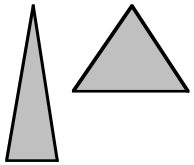
Results

- application of inter-hierarchy comparison
  - modelling (preferences, disjunction)
- error vectors
  - monotony & persistency
- algorithm for inter-hierarchy comparison
  - within HCLP framework
  - exploits underlying hierarchy solver
  - early elimination of useless branches

© Roman Barták, 1998

Further Steps

- implementation
- empirical studies
  - deep vs wide computation tree



  - balance between reduction and elimination
  - knowledge driven search
- integration with CSP

© Roman Barták, 1998