Explanations for constraint solvers
Definition, Computation, Use

Narendra Jussien
Ecole des Mines de Nantes
Context

- Constraint programming
  - *declarative, efficient, ...*
- What if there is no solution?
  - *standard/commercial systems*
    - no solution
  - *not acceptable in decision support systems*
- What can be done
  - *Why is there no solution?*
  - *What do we have to do to get one?*
- Key concept: *explanation*
Outline

- Definition of *explanation*

- Integration with filtering algorithms

- Usages

- Conclusion
Definitions

Let \( <X,C> \) be a constraint problem
the domains are expressed as constraints

\( E \subset C \) is an **explanation for failure**
iff \( <X,E> \) has no solution

What sort of explanations

- **Trivial explanation** : \( C \) (not efficient)
- **Ideal explanation** : minimal for inclusion (very hard)
- **Compromise between time, space and efficiency**
Explanations and filtering algorithms

- Domain reduction
  = prevention against contradictions

- Filtering algorithm
  = explanation provider

- Removing \( a \) from \( x \) because of \( c_1 \) and \( c_2 \)
  \[ \implies x=a \land c_1 \land c_2 \] is an explanation
Examples

\[ x \succ y \rightarrow y \succ z \rightarrow y \succ x \]

\[ x \succ y \land y \succ z \land y \succ x \]

as a whole

\[ x: 1 \quad 2 \quad 3 \]

\[ y: 1 \quad 2 \quad 3 \]

\[ z: 1 \quad 2 \quad 3 \]

The accuracy cannot be guaranteed
Different computations

■ For discrete CSPs
  ● using the AC4 algorithm: explanation computed thanks to the maintained lists of supports
  ● using the AC6 algorithm: explanations less precise but still useful
    – yet AC6 is more efficient than AC4 for constraint propagation

■ For numeric CSPs
  ● the explanation calculation is delegated to the projection functions
Usage of explanations

- Over-constrained problems
- Search guidance
- User dialog
Solving over-constrained problems

- The original context

- Explanations are used to
  - provide explanations in case of contradiction
  - determine a set of constraints to remove in order to overcome the contradiction (using a comparator)
  - easily remove a given constraint (the past effects are known)

=> Dynamic problems can be solved the same way
Search guidance

- **Tree-based search**
  - *explanations provide hints on decisions responsible for the current fail*

- **Intelligent backtracking**
  - implementation done for solving scheduling problems (Jussien & Guéret 98, 99)

- **Dynamic backtracking (Ginsberg 93)**
  - integration of constraint propagation within that algorithm
  - applied to numeric CSPs (Jussien & Lhomme 98)
User dialog

- Explanations can explain
  - contradiction, domain reductions, ...

- Difference between the user perception of its problem and the constraint representation
  - explanations are expressed in terms of internal constraints
  - one solution: using S-boxes but seems to need a dedicated solving mechanism
User dialog

- Translate the explanation into the user language
  
  - *Problem seen as a hierarchy of concepts*
  
  - *User represented as a projection of this hierarchy*
  
  - *Explanation projected on the user representation*

- Current project for the CRITT Pays de la Loire
Conclusion

- Explanations
  - *simple*
  - *powerful*
  - *time and space complexities can be adjusted*

- Links to the user
  - *machinery provided*
  - *but communications need to be improved*