# *Chapter 2: Simplification, Optimization and Implication*

*Where we learn more fun things we can do with constraints.*

1

## *Simplification, Optimization and Implication*

- Constraint Simplification
- Projection
- Constraint Simplifiers
- Optimization
- Implication and Equivalence

2

# Constraint Simplification

- Two equivalent constraints represent the same information, **but**
- One may be simpler than the other

$$
\begin{aligned}
X \ge 1 \wedge X \ge 3 \wedge 2 = Y + X \\
\leftrightarrow X \ge 3 \wedge 2 = Y + X \\
\leftrightarrow 3 \le X \wedge X = 2 - Y \\
\leftrightarrow X = 2 - Y \wedge 3 \le X \\
\leftrightarrow X = 2 - Y \wedge 3 \le 2 - Y \\
\leftrightarrow X = 2 - Y \wedge Y \le -1
\end{aligned}
$$

Removing redundant constraints, rewriting a primitive constraint, changing order, substituting using an equation all preserve equivalence

3

# Redundant Constraints

- One constraint *C1* **implies** another *C2* if the solutions of *C1* are a subset of those of *C2*
- *C2* is said to be **redundant** wrt *C1*
- It is written $C1 \rightarrow C2$

$$
\begin{aligned}
X \ge 3 \rightarrow X \ge 1 \\
Y \le X + 2 \wedge Y \ge 4 \rightarrow X \ge 1 \\
cons(X, X) = cons(Z, nil) \rightarrow Z = nil
\end{aligned}
$$

4

## Redundant Constraints

- We can remove a primitive constraint which is redundant with respect to the rest of the constraint.

$$\underline{X \geq 1} \wedge X \geq 3 \leftrightarrow X \geq 3$$
$$Y \leq X + 2 \wedge \underline{X \geq 1} \wedge Y \geq 4 \leftrightarrow Y \leq X + 2 \wedge Y \geq 4$$
$$cons(X, X) = cons(Z, nil) \wedge \underline{Z = nil} \leftrightarrow cons(X, X) = cons(Z, nil)$$

Definitely produces a simpler constraint

5

## Solved Form Solvers

- Since a solved form solver creates equivalent constraints it can be a simplifier

For example using the term constraint solver

$$cons(X, X) = cons(Z, nil) \wedge Y = succ(X) \wedge succ(Z) = Y \wedge Z = nil$$
$$\leftrightarrow X = nil \wedge Z = nil \wedge Y = succ(nil)$$

Or using the Gauss-Jordan solver

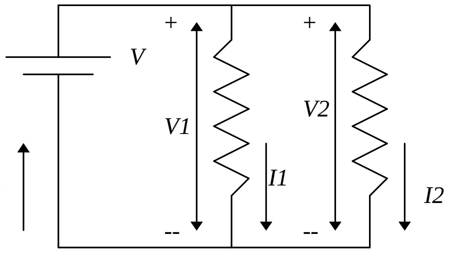
$$X = 2 + Y \wedge 2Y + X - T = Z \wedge X + Y = 4 \wedge Z + T = 5$$
$$\leftrightarrow X = 3 \wedge Y = 1 \wedge Z = 5 - T$$

6

## Projection

It becomes even more important to simplify when we are only interested in some variables in the constraint

$$
\begin{aligned}
V1 &= I1 \times R1 \\
V2 &= I2 \times R2 \\
V - V1 &= 0 \\
V - V2 &= 0 \\
V1 - V2 &= 0 \\
I - I1 - I2 &= 0 \\
-I + I1 + I2 &= 0 \\
R1 &= 5 \\
R2 &= 10
\end{aligned}
$$




Simplified wrt to *V* and *I* $V = \frac{10}{3} I$

7

## Projection

- The **projection** of a constraint *C* onto variables *V* is a constraint *C1* such that
  - *C1* only involves variables *V*
  - Every solution of *C* is a solution of *C1*
  - A solution of *C1* can be extended to give a solution of *C*

| | |
|---|---|
| $X \geq Y \wedge Y \geq Z \wedge Z \geq 0$ | $X \geq 0$ |
| $\{X \mapsto 0, Y \mapsto 0, Z \mapsto 0\}$ | $\{X \mapsto 0\}$ |
| $\{X \mapsto 4, Y \mapsto 3, Z \mapsto 1\}$ | $\{X \mapsto 4\}$ |

8

# *Fouriers Algorithm*

- Eliminates variable $y$ from linear ineqs $C$
- Write each ineq with $y$ on one side of ineq

$$t_1 \le y \qquad y \le t_2$$

- For each pair $t_1 \le y \quad y \le t_2$
  - produce a new ineq $t_1 \le y \otimes y \le t_2 \quad t_1 \le t_2$
- The result is all new ineqs and those ineqs in $C$ which do not involve $y$

9

# *Fouriers Algorithm Example*

Projecting out $Y$

$X - 1 \le Y$

$-1 - X \le Y$

$Y \le 1 - X$

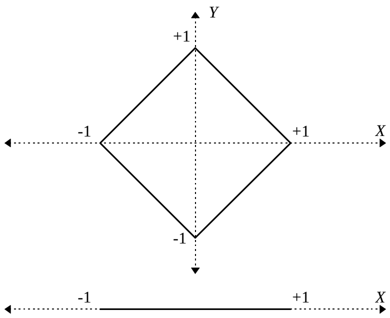
$Y \le 1 + X$

$X - 1 \le Y \otimes Y \le 1 - X \quad X \le 1$

$X - 1 \le Y \otimes Y \le 1 + X \quad 0 \le 2$

$-1 - X \le Y \otimes Y \le 1 - X \quad 0 \le 2$

$-1 - X \le Y \otimes Y \le 1 + X \quad -1 \le X$




Result only involving $X$

$$X \le 1 \wedge -1 \le X$$

10

# Projecting Term Constraints

- We can also project term constraints

$$cons(Y,Y) = cons(X,Z) \wedge succ(Z) = succ(T)$$

- projected onto *{X,Z}* is $X = Z$
- But what is *X = cons(Y,Z)* projected onto *X*?
- Answer: there is no such constraint!

11

# Constraint Simplifiers

- constraints *C1* and *C2* are **equivalent wrt variables** *V* if
  - taking any solution of one and restricting it to the variables *V*, this restricted solution can be extended to be a solution of the other
- Example *X=succ(Y)* and *X=succ(Z)* wrt *{X}*

| $X = succ(Y)$ | $\{X\}$ | $X = succ(Z)$ |
|---|---|---|
| $\{X \mapsto succ(a), Y \mapsto a\}$ | $\{X \mapsto succ(a)\}$ | $\{X \mapsto succ(a), Z \mapsto a\}$ |
| $\{X \mapsto succ(nil), Y \mapsto nil\}$ | $\{X \mapsto succ(nil)\}$ | $\{X \mapsto succ(nil), Z \mapsto nil\}$ |

12

## Simplifier Definition

- A **constraint simplifier** is a function *simpl* which takes a constraint *C* and set of variables *V* and returns a constraint *C1* that is equivalent to *C* wrt *V*
- We can make a simplifier for real inequalities from Fouriers algorithm

13

## Tree Constraint Simplifier

- apply the term solver to *C* obtaining *C1*
- if *C1* is *false* then return *false*
- foreach equation *x=t* in *C1*
  - if *x* is in *V* then
    - if *t* is a variable not in *V*
      - substitute *x* for *t* throughout *C1* and result
    - else add *x=t* to result
- return result

14

# *Tree Simplification Example*

Tree constraint to be simplified wrt *{Y,T}*

$$h(f(X,Y),,g(T)) = h(f(g(T),X), f(X,X), g(U))$$

Equivalent constraint from tree solver

$$Z = f(g(U), g(U)) \wedge X = g(U) \wedge Y = g(U) \wedge T = U$$

Discard the first two equations, keep the third and use the last to substitute for *U* by *T*

$$Y = g(T)$$

15

# *Simplifier Properties*

- Desirable properties of a simplifier are:
  - **projecting**: $vars(simpl(C,V)) \subseteq V$
  - **weakly projecting**: for all constraints *C2* that are equivalant to *C1* wrt *V*

    $$|vars(simpl(C1,V)) - V| \leq |vars(C2) - V|$$

    - a weakly projecting solver never uses more variables than is required
- Both properties allow a simplifier to be used as a solver (How?)

16

## Optimization

- Often given some problem which is modelled by constraints we dont want just any solution, but a “best” solution
- This is an **optimization problem**
- We need an **objective function** so that we can rank solutions, that is a mapping from solutions to a real value

17

## Optimization Problem

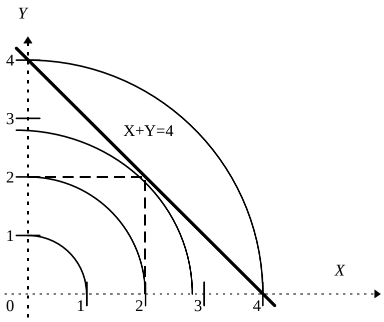
- An **optimization problem** *(C,f)* consists of a constraint *C* and objective function *f*
- A valuation *v1* is **preferred** to valuation *v2* if *f(v1) < f(v2)*
- An **optimal solution** is a solution of *C* such that no other solution of *C* is preferred to it.

18

# Optimization Example

An optimization problem

$(C \equiv X + Y \geq 4, \quad f \equiv X^2 + Y^2)$




Find the closest point to the origin satisfying the *C*.
Some solutions and *f* value

| | |
|---|---|
| $\{X \mapsto 0, Y \mapsto 4\}$ | 16 |
| $\{X \mapsto 3, Y \mapsto 3\}$ | 18 |
| $\{X \mapsto 2, Y \mapsto 2\}$ | 8 |

Optimal solution

$\{X \mapsto 2, Y \mapsto 2\}$

19

# Optimization

- Some optimization problems have no solution.
  - Constraint has no solution
    $(X \geq 2 \wedge X \leq 0, \ X^2)$
  - Problem has no optimum
    $(X \leq 0, \ X)$
  - For any solution there is more preferable one

20

## Simplex Algorithm

- The most widely used optimization algorithm
- Optimizes a linear function wrt to linear constraints
- Related to Gauss-Jordan elimination

21

## Simplex Algorithm

- A optimization problem (*C*, *f*) is in **simplex form**:
  - *C* is the conjunction of *CE* and *CI*
  - *CE* is a conjunction of linear equations
  - *CI* constrains all variables in *C* to be non-negative
  - *f* is a linear expression over variables in *C*

22

## *Simplex Example*

An optimization problem in simplex form

minimize $3X+2Y-Z+1$ subject to

$$\begin{aligned} X \quad + Y \qquad\qquad\qquad & = 3 \wedge \\ -X \quad -3Y \quad +2Z \quad +T \quad & = 1 \wedge \\ X \geq 0 \wedge Y \geq 0 \wedge Z \geq 0 \wedge T \geq 0 & \end{aligned}$$

- An arbitrary problem can be put in simplex form by
  - replacing unconstrained var *X* by new vars $X^{+} - X^{-}$
  - replacing ineq $e \leq r$ by new var *s* and $e + s = r$

23

## *Simplex Solved Form*

- A simplex optimization problem is in **basic feasible solved (bfs) form** if:
  - The equations are in solved form
  - Each constant on the right hand side is non-negative
  - Only parameters occur in the objective
- A **basic feasible solution** is obtained by setting each parameter to 0 and each non-parameter to the constant in its equation

24

## Simplex Example

An equivalent problem to that before in bfs form

$$\text{minimize } 10 - Y - Z \text{ subject to}$$

$$\begin{array}{lllll} X = & 3 & -Y & & \wedge \\ T = & 4 & +2Y & -2Z & \wedge \end{array}$$

$$X \geq 0 \wedge Y \geq 0 \wedge Z \geq 0 \wedge T \geq 0$$

We can read off a solution and its objective value

$$\{X \mapsto 3, T \mapsto 4, Y \mapsto 0, Z \mapsto 0\}$$

$$f = 10$$

25

## Simplex Algorithm

starting from a problem in bfs form

**repeat**

Choose a variable $y$ with negative coefficient in the obj. func.

Find the equation $x = b + cy + ...$ where $c<0$ and $-b/c$ is minimal

Rewrite this equation with $y$ the subject $y = -b/c + 1/c\ x + ...$

Substitute $-b/c + 1/c\ x + ...$ for $y$ in all other eqns and obj. func.

**until** no such variable $y$ exists or no such equation exists

**if** no such $y$ exists optimum is found

**else** there is no optimum solution

26

## *Simplex Example*

minimize $10 - Y - Z$ subject to

$$X = 3 - Y \wedge$$
$$T = 4 + 2Y - 2Z \wedge$$
$$X \geq 0 \wedge Y \geq 0 \wedge Z \geq 0 \wedge T \geq 0$$

Choose variable $Y$, the first eqn is only one with neg. coeff $Y = 3 - X$

minimize $7 + X - Z$ subject to

$$Y = 3 - X \wedge$$
$$T = 10 - 2X - 2Z \wedge$$
$$X \geq 0 \wedge Y \geq 0 \wedge Z \geq 0 \wedge T \geq 0$$

Choose variable $Z$, the 2nd eqn is only one with neg. coeff $Z = 5 - X - 0.5T$

minimize $2 + 2X + 0.5T$ subject to

$$Y = 3 - X \wedge$$
$$Z = 5 - X - 0.5T \wedge$$
$$X \geq 0 \wedge Y \geq 0 \wedge Z \geq 0 \wedge T \geq 0$$

No variable can be chosen, optimal value 2 is found

27

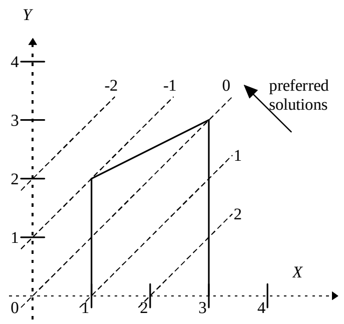
## *Another example*

minimize $X - Y$ subject to

$$Y \geq 0 \wedge$$
$$X \geq 1 \wedge$$
$$X \leq 3 \wedge$$
$$2Y \leq X + 3$$

An equivalent simplex form is:

$$X - S_2 = 1 \wedge$$
$$X + S_3 = 3 \wedge$$
$$-X + 2Y - S_1 = 3 \wedge$$
$$X \geq 0 \wedge Y \geq 0 \wedge S_1 \geq 0 \wedge S_2 \geq 0 \wedge S_3 \geq 0$$




An optimization problem showing contours of the objective function

28

# *Another example*

Basic feasible solution form: circle

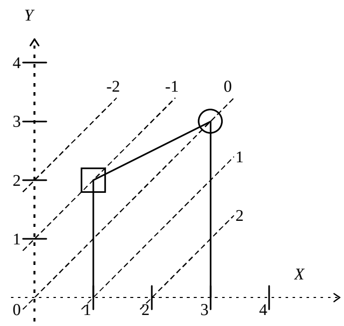
minimize $0 + 0.5S_1 - 0.5S_3$ subject to

$$
\begin{aligned}
Y &= 3 - 0.5S_1 - 0.5S_3 \wedge \\
S_2 &= 2 - S_3 \wedge \\
X &= 3 - S_3 \wedge
\end{aligned}
$$

Choose S3, replace using 2nd eq

minimize $-1 + 0.5S_1 + 0.5S_2$ subject to

$$
\begin{aligned}
Y &= 3 - 0.5S_1 + 0.5S_2 \wedge \\
S_3 &= 2 - S_2 \wedge \\
X &= 1 + S_2 \wedge
\end{aligned}
$$




Optimal solution: box

29

# *The Missing Part*

- How do we get the initial basic feasible solution?
- Solve a different simplex problem
  - Add artificial variables to make equations in basic feasible solved form
  - Minimize the sumof the artificial variables
  - If the sum is zero we can construct a bfs for the original problem

30

# The Missing Part Example

Original simplex form equations

$$\begin{array}{lllllll} X & & & -S_2 & & = 1 \wedge \\ X & & & & +S_3 & = 3 \wedge \\ -X & +2Y & -S_1 & & & = 3 \end{array}$$

With artificial vars in bfs form:

$$\begin{array}{lllllll} A_1 = & 1 & -X & & +S_2 & & \wedge \\ A_2 = & 3 & -X & & & -S_3 & \wedge \\ A_3 = & 3 & +X & -2Y & -S_1 & & \end{array}$$

Objective function: minimize

$A_1 + A_2 + A_3$

$= 7 - X - 2Y - S_1 + S_2 - S_3$

31

# Missing Part Example II

Problem after minization of objective function

minimize $A_1 + A_2 + A_3$ subject to

$$\begin{array}{llllllll} Y & = & 3 & -0.5S_1 & -0.5S_3 & & -0.5A_2 & -0.5A_3 & \wedge \\ S_2 & = & 2 & & -S_3 & +A_1 & -A_2 & & \wedge \\ X & = & 3 & & -S_3 & & -A_2 & & \end{array}$$

Removing the artificial variables, the original problem

$$\begin{array}{lllll} Y = & 3 & -0.5S_1 & -0.5S_3 & \wedge \\ S_2 = & 2 & & -S_3 & \wedge \\ X = & 3 & & -S_3 & \wedge \end{array}$$

32

# Implication and Equivalence

- Other important operations involving constraints are:
- **implication**: test if *C1* implies *C2*
  - *impl(C1, C2)* answers *true*, *false* or *unknown*
- **equivalence**: test if *C1* and *C2* are equivalent
  - *equiv(C1, C2)* answers *true*, *false* or *unknown*

33

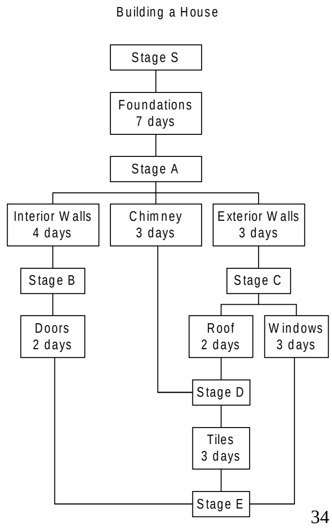
# Implication Example

For the house constraints *CH*, will stage B have to be reached after stage C?

$$CH \rightarrow T_B \geq T_C$$

For this question the answer if *false*, but if we require the house to be finished in 15 days the answer is *true*

$$CH \wedge T_E = 15 \rightarrow T_B \geq T_C$$




34

## *Implication and Equivalence*

- We can use *impl* to define *equiv* and vice versa
  $impl(C1, C2) = equiv(C1, C1 \wedge C2)$
  $equiv(C1, C2) = impl(C1, C2) \wedge impl(C2, C1)$
- We can use a solver to do simple *impl* tests
  $impl(C1, C2) = \neg solv(C1 \wedge \neg C2)$
- e.g. $impl(CH, T_B \geq T_C) = \neg solv(CH \wedge T_B < T_C)$

35

## *Simplication, Optimization and Implication Summary*

- Equivalent constraints can be written in many forms, hence we desire simplification
- Particularly if we are only interested in the interaction of some of the variables
- Many problems desire a optimal solution, there are algms (simplex) to find them
- We may also be interested in asking questions involving implication

36