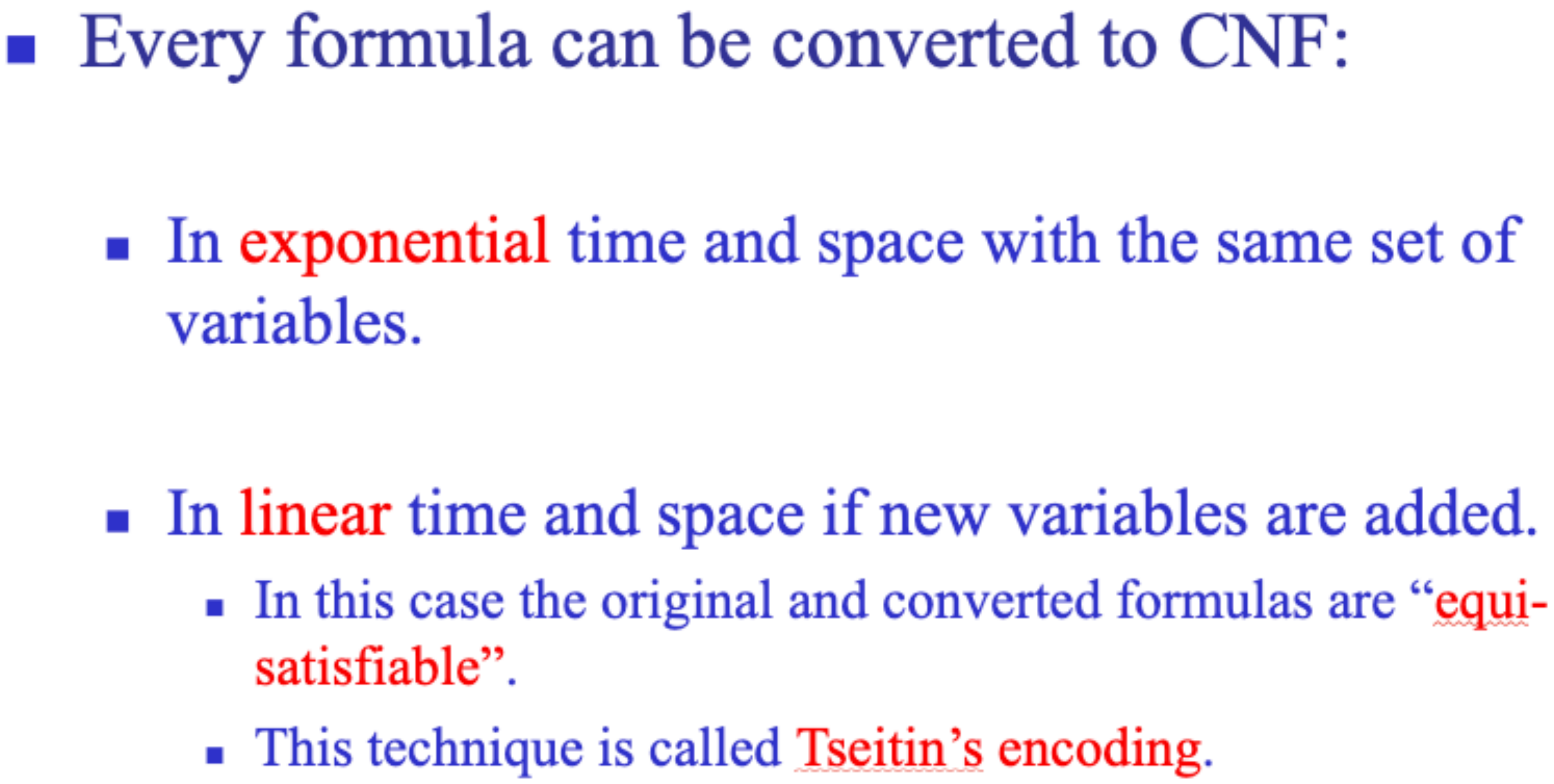
CS6410 Software Verification

#3. Tseitin Transform, Modeling systems, BDD

Ashish Mishra

Converting to CNF



- Every formula can be converted to CNF:
  - In exponential time and space with the same set of variables.
  - In linear time and space if new variables are added.
    - In this case the original and converted formulas are "equi-satisfiable".
    - This technique is called Tseitin's encoding.

source: decision-procedures.org

Converting to CNF: the exponential way

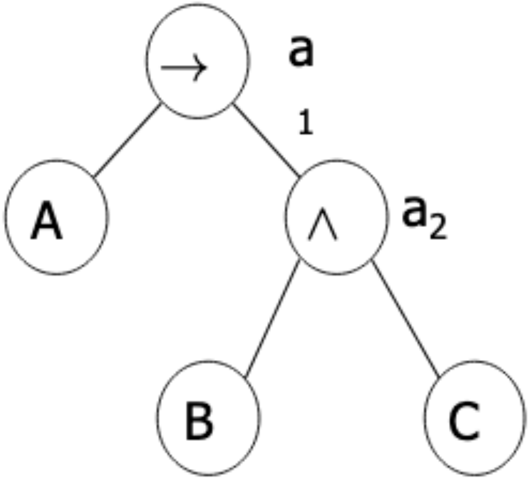
```
CNF($\phi$) {
case
    $\phi$ is a literal: return $\phi$
    $\phi$ is $\psi_1 \wedge \psi_2$: return CNF($\psi_1$) $\wedge$ CNF($\psi_2$)
    $\phi$ is $\psi_1 \vee \psi_2$: return Dist(CNF($\psi_1$),CNF($\psi_2$))
}


Dist($\psi_1$,$\psi_2$) {
case
    $\psi_1$ is $\phi_{11} \wedge \phi_{12}$: return Dist($\phi_{11}$,$\psi_2$) $\wedge$ Dist($\psi_{12}$,$\psi_2$)
    $\psi_2$ is $\phi_{21} \wedge \phi_{22}$: return Dist($\psi_1$,$\phi_{21}$) $\wedge$ Dist($\psi_1$,$\phi_{22}$)
    else: return $\psi_1 \vee \psi_2$
```

Converting to CNF: the exponential way

- Consider the formula
  $\phi = (x_1 \wedge y_1) \vee (x_2 \wedge y_2)$
- CNF(ϕ)=
  $(x_1 \vee x_2) \wedge$
  $(x_1 \vee y_2) \wedge$
  $(y_1 \vee x_2) \wedge$
  $(y_1 \vee y_2)$
- Now consider: $\phi_n = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$
- Q: How many clauses CNF(ϕ) returns ?
- A: 2^n

Converting to CNF: Tseitin's encoding

- Consider the formula $\phi = (A \rightarrow (B \wedge C))$
- The parse tree:



- Associate a new auxiliary variable with each gate.
- Add constraints that define these new variables.
- Finally, enforce the root node.

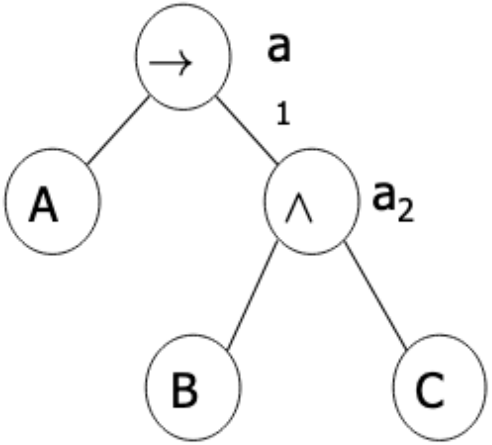
Converting to CNF: Tseitin's encoding

- Need to satisfy:

  $(a_1 \leftrightarrow (A \rightarrow a_2)) \wedge$

  $(a_2 \leftrightarrow (B \wedge C)) \wedge$

  (a_1)



- Each such constraint has a CNF representation with 3 or 4 clauses.

source: decision-procedures.org

Converting to CNF: Tseitin's encoding

- Need to satisfy:

  $(a_1 \leftrightarrow (A \rightarrow a_2)) \wedge$

  $(a_2 \leftrightarrow (B \wedge C)) \wedge$

  (a_1)

- First: $(a_1 \vee A) \wedge (a_1 \vee \neg a_2) \wedge (\neg a_1 \vee \neg A \vee a_2)$
- Second: $(\neg a_2 \vee B) \wedge (\neg a_2 \vee C) \wedge (a_2 \vee \neg B \vee \neg C)$

source: decision-procedures.org

Converting to CNF: Tseitin's encoding

- Let's go back to
  $\phi_n = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$
- With Tseitin's encoding we need:
  - 2n auxiliary variables $a_1,\ldots,a_{2n}$.
  - Each adds 3 constraints.
  - Top clause: $(a_1 \vee \cdots \vee a_n)$
- Hence, we have
  - $6n + 1$ clauses, instead of 2^n.
  - 4n variables rather than 2n.

Before how to solve these formulas

- Q: Suppose we can solve the satisfiability problem... how can this help us?

- A: There are numerous problems in the industry that are solved via the satisfiability problem of propositional logic
  - Logistics...
  - Planning...
  - Electronic Design Automation industry...
  - Cryptography...
  - ... (every NP-P problem...)

source: decision-procedures.org

Modeling with PL

Example: placement of wedding guests

- Three chairs in a row: 1,2,3
- We need to place Aunt, Sister and Father
- Constraints:
  - Aunt doesn’t want to sit near Father
  - Aunt doesn’t want to sit in the left chair
  - Sister doesn’t want to sit to the right of Father

- Q: Can we satisfy these constraints?

Example

- Denote: Aunt = 1, Sister = 2, Father = 3
- Introduce a propositional variable for each pair (person, place).
- $x_{ij} =$ person i is sited at place j, for 1 <= i, j <= 3
- Constraints:
  - Aunt doesn't want to sit near father:

    $(x_{11} \rightarrow \neg x_{32}) \wedge (x_{12} \rightarrow \neg x_{31} \wedge \neg x_{33}) \wedge (x_{13} \rightarrow \neg x_{32})$
  - Aunt doesn't want to sit in the left chair

    $\neg x_{11}$
  - Sister doesn't want to sit to the right (immediate) of the Father

    $(x_{31} \rightarrow \neg x_{22}) \wedge (x_{32} \rightarrow \neg x_{23})$

Example

- More constraints
  - Each person is placed

$$(x_{11} \vee x_{12} \vee x_{13}) \wedge (x_{21} \vee x_{22} \vee x_{23}) \wedge (x_{31} \vee x_{32} \vee x_{33})$$

  - No person is placed in more than one place

$$(x_{11} \rightarrow \neg x_{12} \wedge \neg x_{13}) \wedge \ (x_{12} \rightarrow \neg x_{11} \wedge \neg x_{13}) \wedge \ (x_{13} \rightarrow \neg x_{11} \wedge \neg x_{12}) \wedge$$
$$(x_{21} \rightarrow \neg x_{22} \wedge \neg x_{23}) \wedge \ (x_{22} \rightarrow \neg x_{21} \wedge \neg x_{23}) \wedge \ (x_{23} \rightarrow \neg x_{21} \wedge \neg x_{22}) \wedge$$
$$(x_{31} \rightarrow \neg x_{32} \wedge \neg x_{33}) \wedge \ (x_{32} \rightarrow \neg x_{31} \wedge \neg x_{33}) \wedge \ (x_{33} \rightarrow \neg x_{31} \wedge \neg x_{32})$$

Example 3: assignment of frequencies

- n radio stations
- For each assign one of k transmission frequencies, k < n.
- E -- set of pairs of stations, that are too close to have the same frequency.

- Q: which graph problem does this remind you of ?

Example 3 (cont'd)

- $x_{i,j}$ – station i is assigned frequency j, for $1 \leq i \leq n, 1 \leq j \leq k$
  - Every station is assigned at least one frequency:

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{k} x_{ij}$$

  - Every station is assigned not more than one frequency:

$$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{k-1} (x_{ij} \rightarrow \bigwedge_{j<t\leq k} \neg x_{it})$$

  - Close stations are not assigned the same frequency.

    For each (i,j) \in E,

$$\bigwedge_{t=1}^{k} (x_{it} \rightarrow \neg x_{jt})$$

Example 2 (Lewis Carroll)

- (1) All the dated letters in this room are written on blue paper;

  (2) None of them are in black ink, except those that are written in the third person;

  (3) I have not filed any of them that I can read;

  (4) None of them, that are written on one sheet, are undated;

  (5) All of them, that are not crossed, are in black ink;

  (6) All of them, written by Brown, begin with "Dear Sir";

  (7) All of them, written on blue paper, are filed;

  (8) None of them, written on more than one sheet, are crossed;

  (9) None of them, that begins with "Dear Sir", are written in the third person.

  Therefore, I cannot read any of Brown’s letters.
- Is this statement valid ?

Example 2 (cont’d)

- p = “the letter is dated”
- q = “the letter is written on blue paper”

  (1) All the dated letters in this room are written on blue paper;

  $p \rightarrow q$

- r = “the letter is written in black ink”
- s = “the letter is written in the third person”

  (2) None of them are in black ink, except those that are written in the third person;

  $\neg s \rightarrow \neg r$

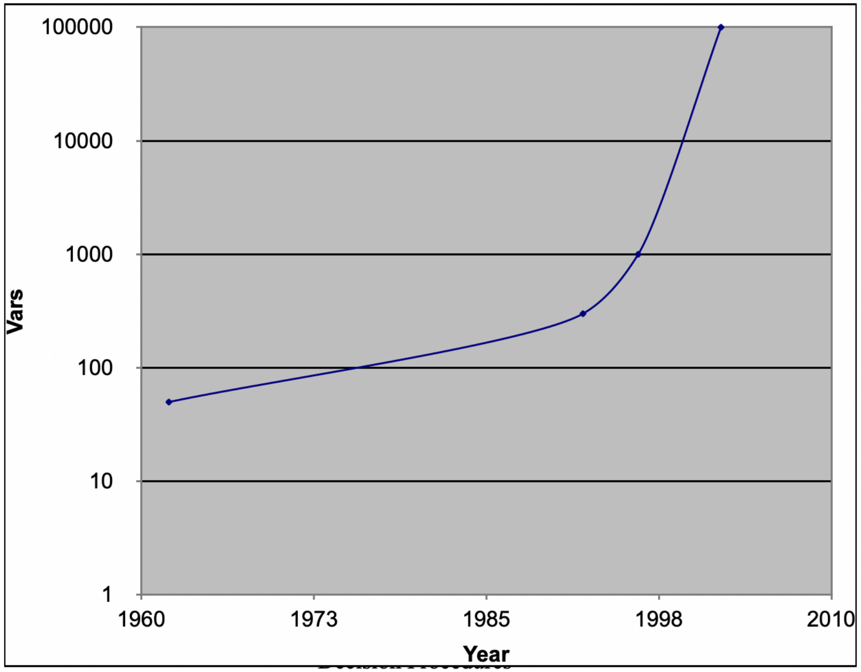
- ...

Overview

- Deciding Propositional Logic
  - SAT tools
  - BDDs

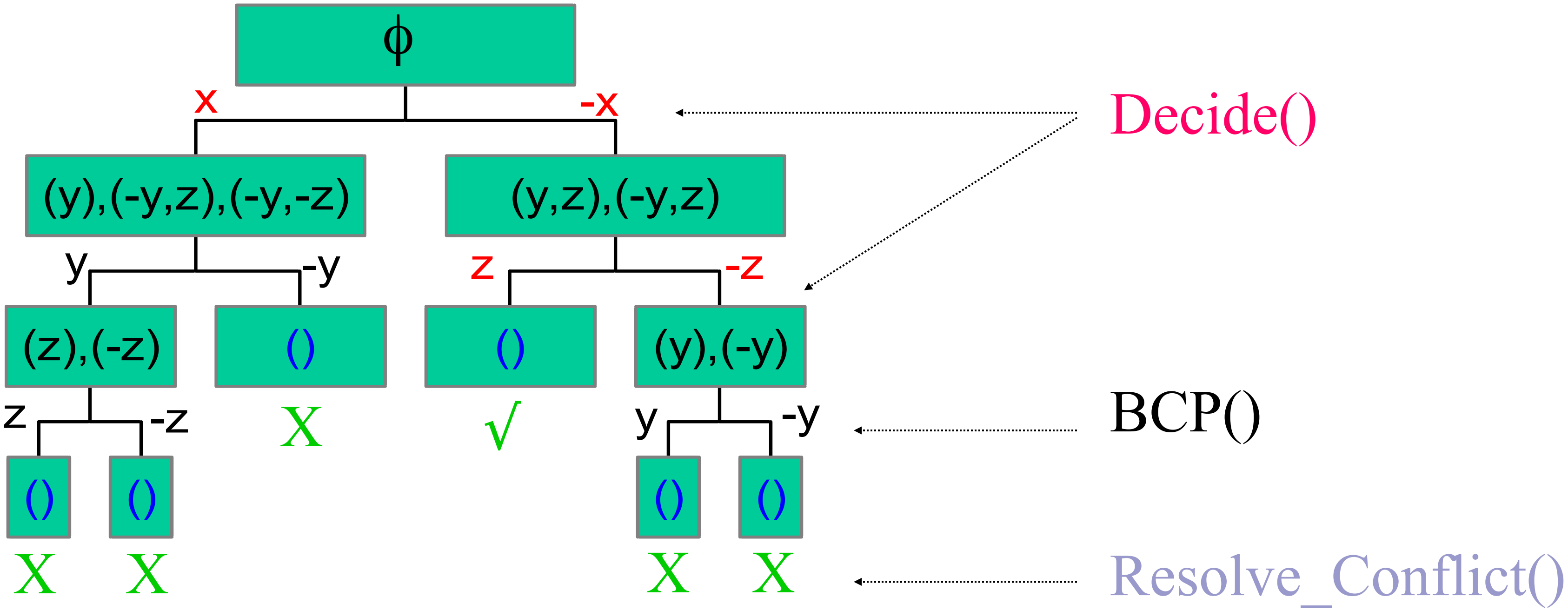
Modern SAT Solvers

- Modern SAT solvers can solve many real-life CNF formulas with hundreds of thousands or even millions of variables in a reasonable amount of time.



A High-level Basic SAT algorithm

- Given ϕ in CNF: $(x \lor y \lor z) \land (-x \lor y) \land (-y \lor z) \land (-x \lor -y \lor -z)$

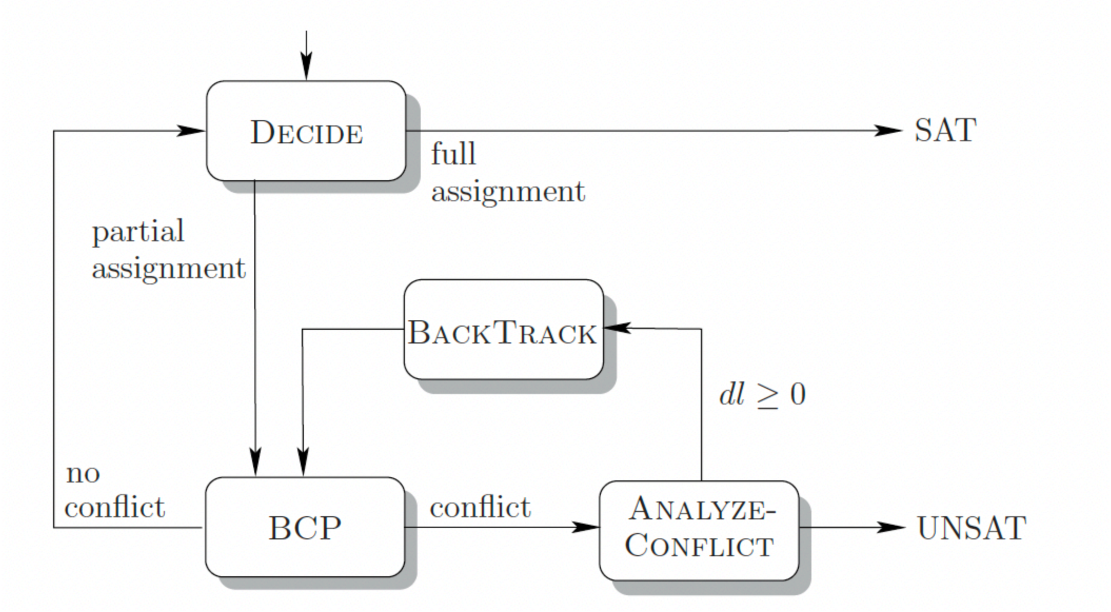


Decide()

BCP()

Resolve_Conflict()

More realistic DPLL



Basic Backtracking Search

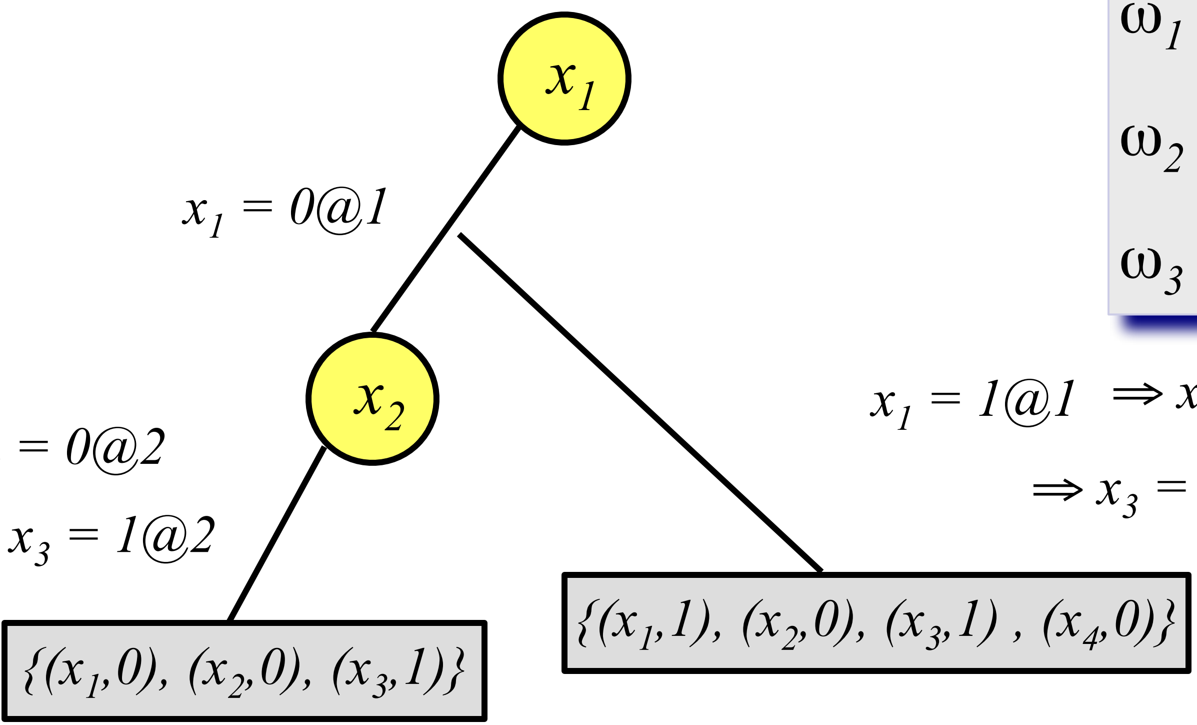
- Organize the search in the form of a decision tree
  - □ Each node corresponds to a decision
  - □ Definition: Decision Level (DL) is the depth of the node in the decision tree.
  - □ Notation: *x*=*v*@*d*
    x \in {0,1} is assigned to *v* at decision level *d*

Backtracking Search in Action

$\omega_1 = (x_2 \vee x_3)$

$\omega_2 = (\neg x_1 \vee \neg x_4)$

$\omega_3 = (\neg x_2 \vee x_4)$



$x_1 = 0@1$

$x_2 = 0@2$

$\Rightarrow x_3 = 1@2$

$\{(x_1,0), (x_2,0), (x_3,1)\}$

$x_1 = 1@1 \;\Rightarrow x_4 = 0@1 \Rightarrow x_2 = 0@1$

$\Rightarrow x_3 = 1@1$

$\{(x_1,1), (x_2,0), (x_3,1)\;, (x_4,0)\}$

No backtrack in this example, regardless of the decision!

Backtracking Search in Action

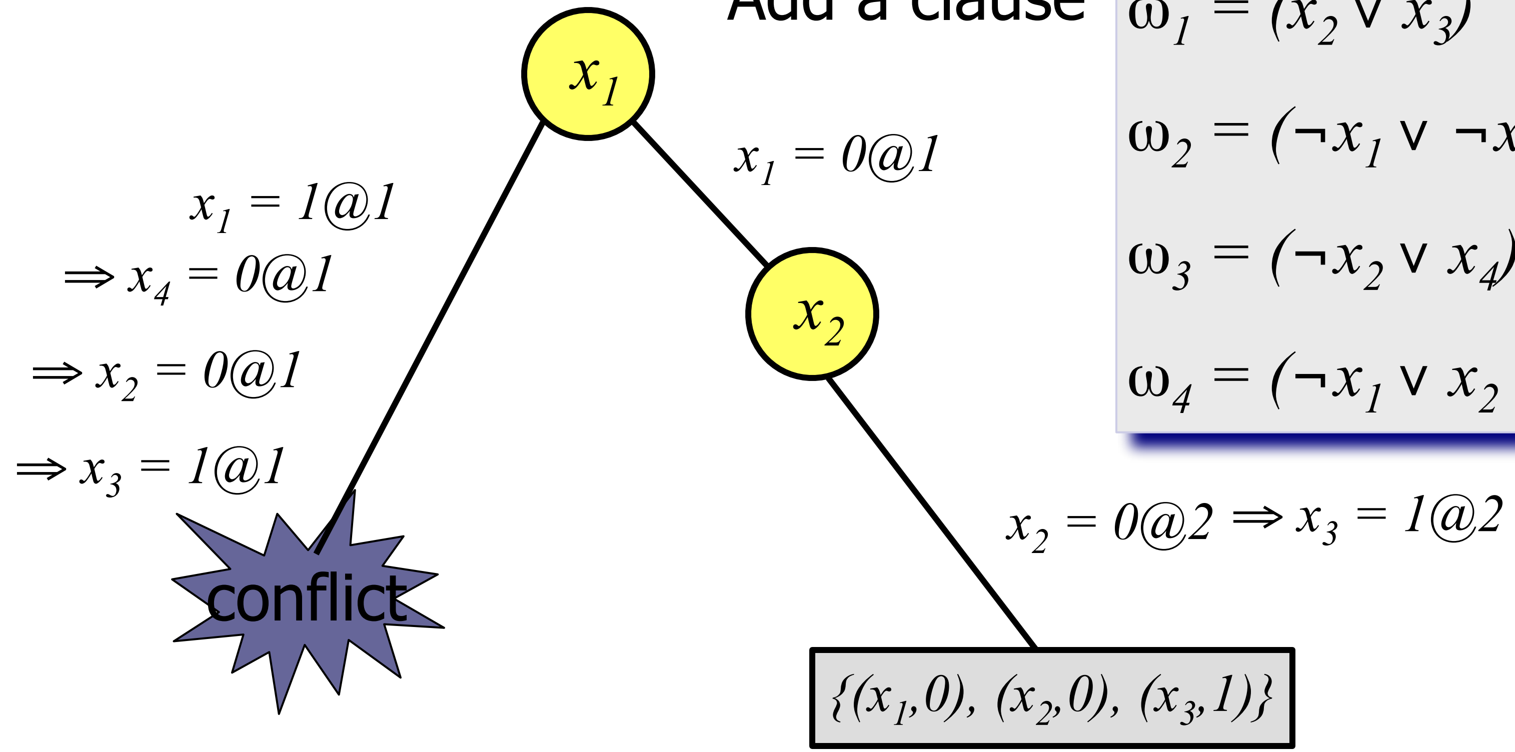
Add a clause

$\omega_1 = (x_2 \vee x_3)$

$\omega_2 = (\neg x_1 \vee \neg x_4)$

$\omega_3 = (\neg x_2 \vee x_4)$

$\omega_4 = (\neg x_1 \vee x_2 \vee \neg x_3)$



Status of a clause

- A clause can be
  - □ Satisfied: at least one literal is satisfied
  - □ Unsatisfied: all literals are assigned but non are satisfied
  - □ Unit: all but one literals are assigned but none are satisfied
  - □ Unresolved: all other cases
- Example: $C = (x_1 \vee x_2 \vee x_3)$

x_1	x_2	x_3	C
1	0		Satisfied
0	0	0	Unsatisfied
0	0		Unit
	0		Unresolved

Decision heuristics - DLIS

DLIS (Dynamic Largest Individual Sum) – choose the assignment that increases the most the number of satisfied clauses

- For a given variable x:
  - □ C_{xp} – # unresolved clauses in which x appears positively
  - □ C_{xn} - # unresolved clauses in which x appears negatively
  - □ Let x be the literal for which C_{xp} is maximal
  - □ Let y be the literal for which C_{yn} is maximal
  - □ If $C_{xp} > C_{yn}$ choose x and assign it TRUE
  - □ Otherwise choose y and assign it FALSE
- Requires l (#literals) queries for each decision.

Decision heuristics - JW

Jeroslow-Wang method

Compute for every clause ω and every variable l (in each phase):

- J(l) := $\sum_{l \in \omega, \omega \in \varphi} 2^{-|\omega|}$

- Choose a variable *l* that maximizes J(l).
- This gives an exponentially higher weight to literals in shorter clauses.

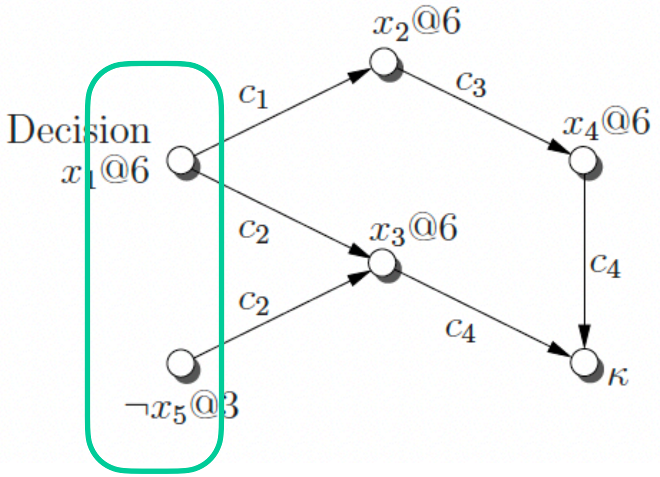
Pause... ||

- We will see other (more advanced) decision Heuristics soon.
- These heuristics are integrated with a mechanism called Learning with Conflict-Clauses, which we will learn next.

Implication Graph

- The process of BCP is best illustrated with an implication graph.
- An implication graph represents the current partial assignment and the reason for each of the implications.
- An implication graph is a labeled directed acyclic graph G(V,E), where:
  - V represents the literals of the current partial assignment.
  - E = {(vi, vj) | vi, vj ∈ V,} denotes the set of directed edges where each edge (vi, vj) is labeled with **Antecedent (vj).**
  - G can also contain a single conflict node labeled with κ and incoming edges {(v, κ)} labeled with c for some conflicting clause c.

Example

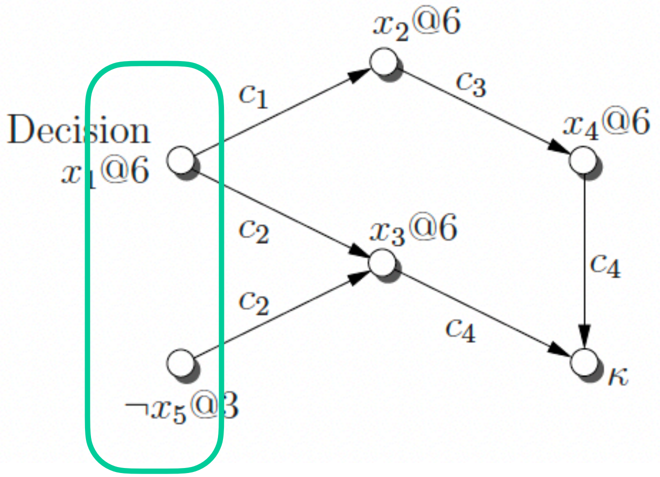


$$\begin{aligned}
c_1 &= (\neg x_1 \vee x_2)\ , \\
c_2 &= (\neg x_1 \vee x_3 \vee x_5)\ , \\
c_3 &= (\neg x_2 \vee x_4)\ , \\
c_4 &= (\neg x_3 \vee \neg x_4)\ , \\
c_5 &= (x_1 \vee x_5 \vee \neg x_2)\ , \\
c_6 &= (x_2 \vee x_3)\ , \\
c_7 &= (x_2 \vee \neg x_3)\ , \\
c_8 &= (x_6 \vee \neg x_5)\ .
\end{aligned}$$

sufficient to create the conflict

In fact, This is a partial implication graph,
A subgraph which
illustrates the BCP at a specific decision level

Example



$$
\begin{aligned}
c_1 &= (\neg x_1 \vee x_2)\ ,\\
c_2 &= (\neg x_1 \vee x_3 \vee x_5)\ ,\\
c_3 &= (\neg x_2 \vee x_4)\ ,\\
c_4 &= (\neg x_3 \vee \neg x_4)\ ,\\
c_5 &= (x_1 \vee x_5 \vee \neg x_2)\ ,\\
c_6 &= (x_2 \vee x_3)\ ,\\
c_7 &= (x_2 \vee \neg x_3)\ ,\\
c_8 &= (x_6 \vee \neg x_5)\ .
\end{aligned}
$$

sufficient to create the conflict

We learn the *conflict clause* *c9* : $(: \sim x_1 \vee x_5)$ Prunes the space

Reflects the fact that this is the solver's way to learn from its past mistakes.

Implication graphs and learning: option #1

Current truth assignment: $\{x_9=0@1\ ,x_{10}=0@3,\ x_{11}=0@3,\ x_{12}=1@2,\ x_{13}=1@2\}$

Current decision assignment: $\{x_1=1@6\}$

$\omega_1 = (\neg x_1 \vee x_2)$

$\omega_2 = (\neg x_1 \vee x_3 \vee x_9)$

$\omega_3 = (\neg x_2 \vee \neg x_3 \vee x_4)$

$\omega_4 = (\neg x_4 \vee x_5 \vee x_{10})$

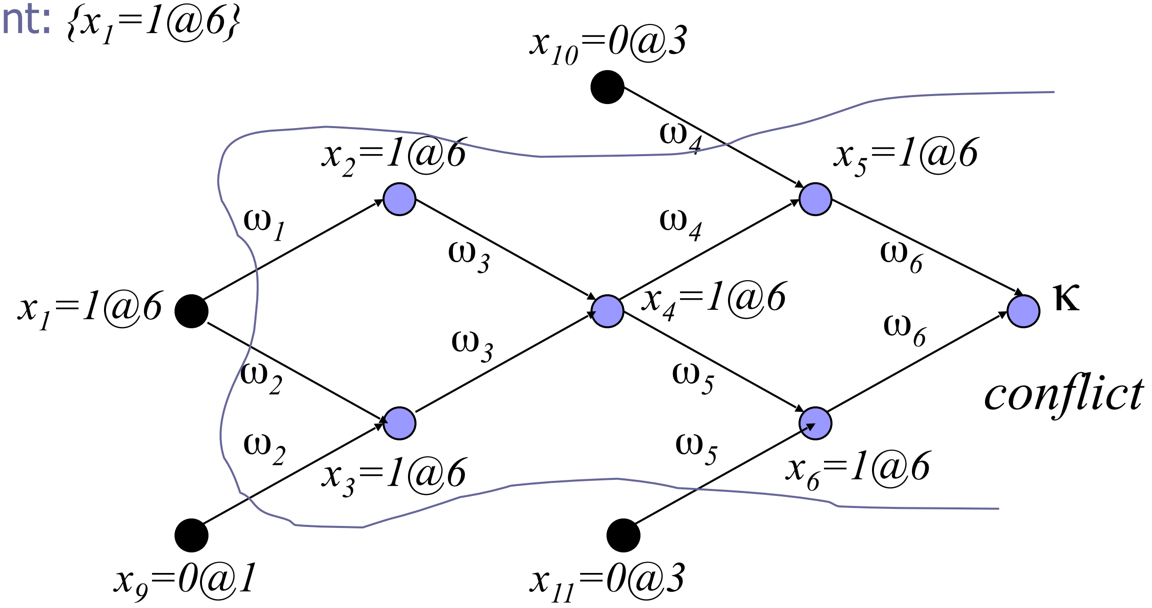
$\omega_5 = (\neg x_4 \vee x_6 \vee x_{11})$

$\omega_6 = (\neg x_5 \vee \neg\ x_6)$

$\omega_7 = (x_1 \vee x_7 \vee \neg x_{12})$

$\omega_8 = (x_1 \vee x_8)$

$\omega_9 = (\neg x_7 \vee \neg x_8 \vee \neg\ x_{13})$



We learn the *conflict clause* ω_{10} : $(\colon x_1 \vee x_9 \vee x_{11} \vee x_{10})$

Implication graph, flipped assignment option #1

$\omega_1 = (\neg x_1 \vee x_2)$

$\omega_2 = (\neg x_1 \vee x_3 \vee x_9)$

$\omega_3 = (\neg x_2 \vee \neg x_3 \vee x_4)$

$\omega_4 = (\neg x_4 \vee x_5 \vee x_{10})$

$\omega_5 = (\neg x_4 \vee x_6 \vee x_{11})$

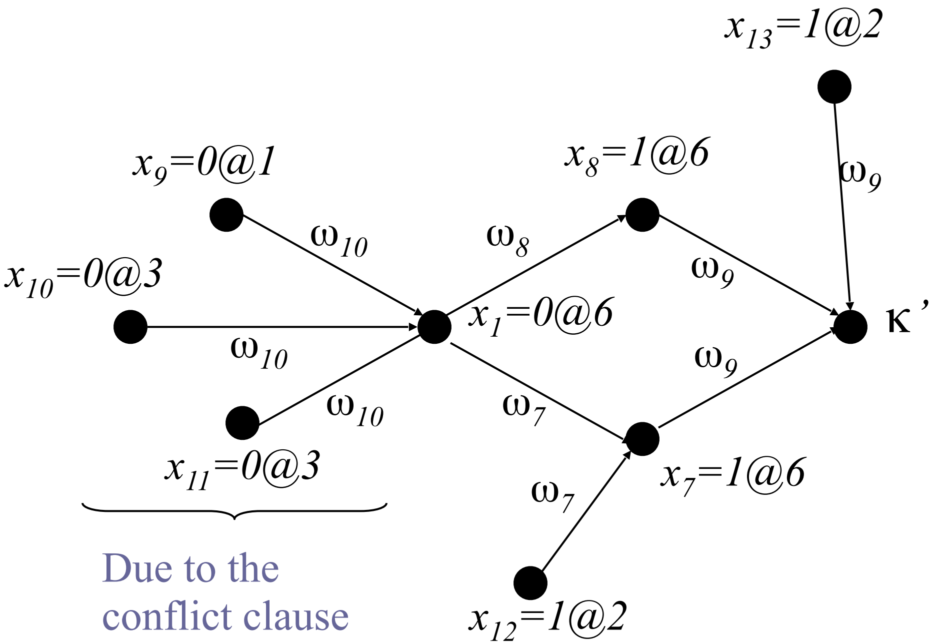
$\omega_6 = (\neg x_5 \vee x_6)$

$\omega_7 = (x_1 \vee x_7 \vee \neg x_{12})$

$\omega_8 = (x_1 \vee x_8)$

$\omega_9 = (\neg x_7 \vee \neg x_8 \vee \neg x_{13})$

$\omega_{10} : (:x_1 \vee x_9 \vee x_{11} \vee x_{10})$

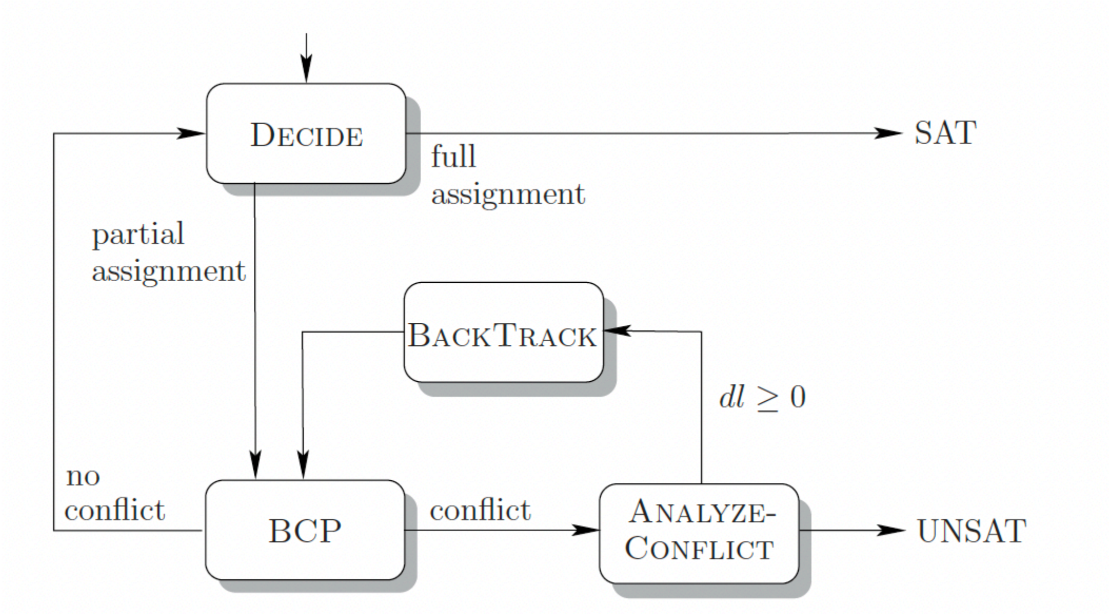


No decision here

Another conflict clause:ω_{11}: $(:x_{13} \vee :x_{12} \vee x_{11} \vee x_{10} \vee x_9)$

where should we backtrack to now ?

More realistic DPLL



Non-chronological backtracking

Which assignments caused the conflicts ?

$x_9 = 0@1$

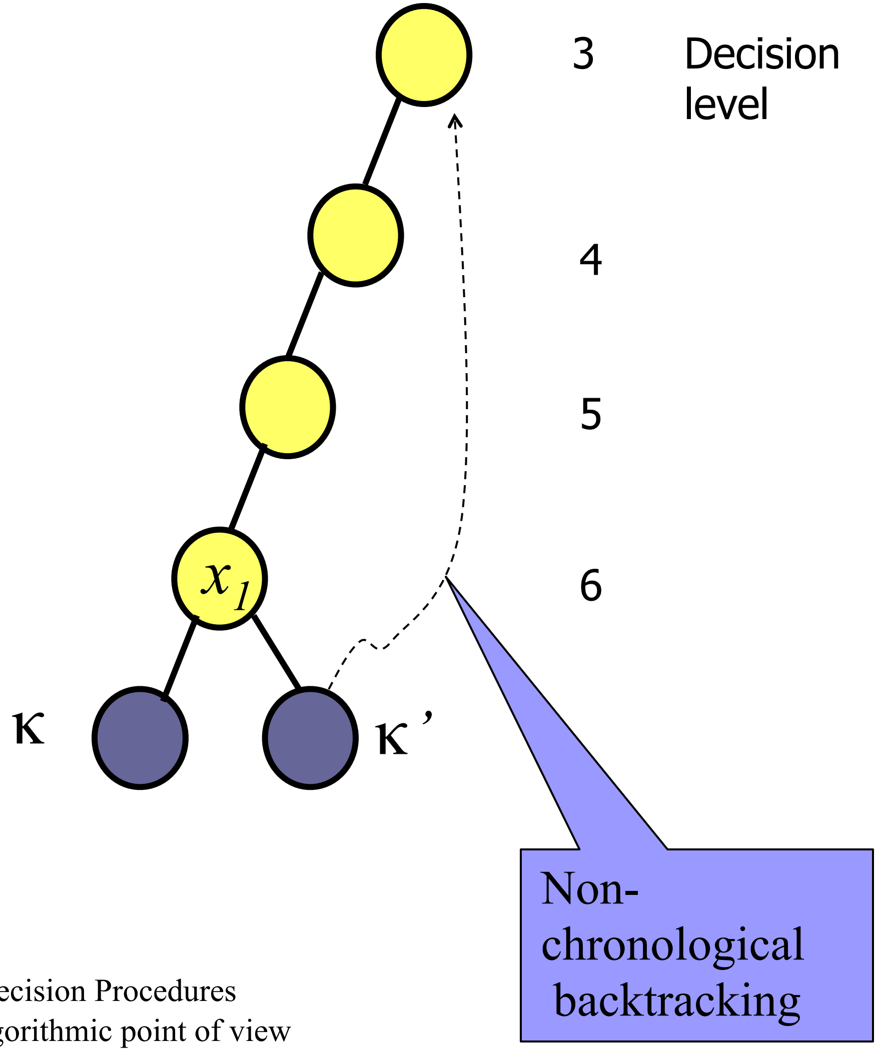
$x_{10} = 0@3$

$x_{11} = 0@3$

$x_{12} = 1@2$

$x_{13} = 1@2$

These assignments Are sufficient for Causing a conflict.



Non-chronological backtracking

Backtrack to DL = 3

Back to the logistics

- Assignment 1
  - Practice: Work out exercise at the end of Chapter 1 in the CoC book. Due Tuesday.
  - Reading: M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. Communications of the ACM, 5(7):394–397, July 1962
- Next Class:
  - Backtracking, Decide heuristics
- Class rescheduling.