September 22, 2011

Algorithm for 3-SAT

I assume that the input formula

$$\Psi \equiv (q_{11} \vee q_{12} \vee q_{13}) \wedge (q_{21} \vee q_{22} \vee q_{23}) \wedge \cdots \wedge (q_{t1} \vee q_{t2} \vee q_{t3})$$

is so that there is no pair of conjugated literals in any clause. That is, for all $q_{i1} \vee q_{i2} \vee q_{i3}$, there is no l so that l and $\neg l$ is a literal of $q_{i1} \vee q_{i2} \vee q_{i3}$. A linear verification replaces $(r \vee l \vee \neg l)$ by $\top$ ($\top \equiv (r \vee l \vee \neg l) \equiv (r \vee \top)$). Denote the literals of Ψ by $Literal(\Psi)$.

Denote by $\Psi \setminus (q_{r1} \vee q_{r2} \vee q_{r3})$ the formula

$$\begin{array}{l} (q_{11} \vee q_{12} \vee q_{13}) \wedge \cdots \wedge (q_{r-11} \vee q_{r-12} \vee q_{r-13}) \wedge \\ (q_{r+11} \vee q_{r+12} \vee q_{r+13}) \wedge \cdots \wedge (q_{t1} \vee q_{t2} \vee q_{t3}) \end{array}$$

(remove the clause $q_{r1} \vee q_{r2} \vee q_{r3}$ from Ψ) and by $C_i \setminus q_{ij} = (q_{i1} \vee q_{i2} \vee q_{i3}) \setminus q_{ij}$ the formula $q_{ik} \vee q_{il}$, $k, l \neq i$.

Denote by X' the complementary of a set X.

Assume Procedures

- **Stract**(ψ): write the literals of a formula ψ.
- $\star$: Given strings s, $s \star t$ is the concatenation of s with t. Strings are, in this text, formed by literals of Ψ;
- $Length(\Psi) = \max\{|\textbf{Stract}(C_i)| C_i \text{ is a clause of } \Psi\}$. It is the maximum of the number of literals in the clauses of Ψ.

Algorithm 1 Factorize by p, `Fac`(Ψ, p)

Input: $\Psi \equiv (q_{11} \vee q_{12} \vee q_{13}) \wedge \cdots \wedge (q_{t1} \vee q_{t2} \vee q_{t3}) \equiv C_1 \wedge \cdots \wedge C_t$, $p \in Literals(\Psi)$

```
  S_p ← ∅
  for 1 ≤ i ≤ t do
    Read C_i
    A ← Stract(C_i)
    if p ∈ A then
      S_p ← S_p ∧ (C_i \ p)
    end if
  end for
```

Output: S_p

Algorithm 2 Partition

Input: $\Psi \equiv (q_{11} \vee q_{12} \vee q_{13}) \wedge \cdots \wedge (q_{t1} \vee q_{t2} \vee q_{t3}) \equiv C_1 \wedge \cdots \wedge C_t$

```
Label ← ∅, S_G ← Ψ, S_⊤ ← ∅, PTTΨ ← ∅, Str ← ∅, FS ← ∅
while Lenght(S_G) = 3 do
  Str ← Stract(S_G)
  litAll ← ∅
  while Str ≠ ∅ do
    if there is a conjugated pair p, ¬p in Str \ (litAll)' then
      C_p ← ∅
      C_¬p ← ∅
      if p ∈ FS then
        Necfalse ← Necfalse ∪ {p}
      end if
      if ¬p ∈ FS then
        Necfalse ← Necfalse ∪ {¬p}
      end if
      for all Clause C of Ψ do
        if p ∈ C then
          C_p ← C_p ∧ C
          S_G ← S_G \ C
        else
          if ¬p ∈ C then
            C_¬p ← C_¬p ∧ C
            S_G ← S_G \ C
          end if
        end if
        Factorize C_p by p
        Factorize C_¬p by ¬p
        PTTΨ ← PTTΨ ∧ (p ∨ S_P)
        PTTΨ ← PTTΨ ∧ (¬p ∨ S_¬p)
        litAll ← litAll \ ({p, ¬p} ∪ Stract(S_p) ∪ Stract(S_¬p))
        Label ← Label ∪ {p, ¬p}
      end for
    else {there are no pair of conjugated literals in Str \ (litAll)'}
      FS ← FS ∪ (Stract(S_P) ∪ Stract(S_¬p))
      S_G ← S_G \ PTTΨ
    end if
  end while
end while
S_⊤ ← S_G
PTTΨ ← PTTΨ ∧ S_⊤
Output: Labels, Partitioned Formula Ψ, Necfalse
```

Algorithm 3 Cylindrical Digraph

Input: Ψ, $Labels = \{p_1, \neg p_1, \dots, p_s, \neg p_s, p_{s+1} = \bot\}$

```
  Nectrue ← ∅
  V_0 ← ∅
  V_1 ← ∅
  E_0 ← ∅
  E_1 ← ∅
  for all a, b ∈ LiteralΨ do
    label(a, b) ← ∅
  end for
  for all l ∈ Labels do
    for all Clauses C of S_l do
      if a ∨ b is a disjunction of C then
        if (a ≠ ⊥) AND (b ≠ ⊥) then
          V_0 ← V_0 ∪ {a_0, b_0}
          V_1 ← V_0 ∪ {a_1, b_1}
          E_0 ← E_0 ∪ {a_0 ⇒ b_1, b_0 ⇒ a_1}
          E_1 ← E_1 ∪ {a_1 → ¬a_0, b_1 → ¬b_0}
          label(a, b) ← label(a, b) ⋆ l
        else {a = ⊥ OR b = ⊥}
          variable c
          If a ≠ ⊥, then c = a, else c = b
          Vertice ← Vertice ∪ {c_0, c_1, ⊥}
          Edges ← Edges ∪ {c_0 ⇒ ⊥, ⊥ ⇒ c_1}
          label((c, ⊥)) ← label(c, ⊥) ⋆ l
          Nectrue ← Nectrue ∪ {c_0}
        end if
      end if
    end for
    label(a, b) and label(c, d) are compatible iff there is no literal p so taht
    p and ¬p are sequente of label(a, b) ⋆ label(c, d)
  end for
  Output: label(a, b), Nectrue
```

Algorithm 4 Write Non Empty Intervals Part 1 of 2

Input $Cyl = (V_0, V_1, E_0, E_1, label(a, b))$, $Nectrue$

for all $a_0 \in V_0$ **do**
  $V_1 \leftarrow V_1 \setminus \{\neg a_1\}$
  $V_0 \leftarrow V_0 \setminus \{\neg a_0\}$
  $V_{0a_0} \leftarrow \{a_0\}$
  $V_{1a_0} \leftarrow \{b_1 | a_0 \Rightarrow b_1 \in E_0\}$
  $E_{0a_0} = \{a_0 \overset{l}{\Rightarrow} b_1 \in E_0\}$
  $\underline{label_{a_0}(a, b)} \leftarrow \emptyset$
  $\overline{Set_1} = \overline{Set_0} = \emptyset$
  $Set_0 = \{a_0\}$
  $Set_1 = \{b_1 \in V_1 | (a_0 \overset{l}{\Rightarrow} b_1 \in E_0\}$
  $E_{1a_0} = \emptyset$
  while $Set_0 \neq \emptyset$ **do**
    $\overline{Set_0} \leftarrow Set_0$
    $\overline{Set_1} \leftarrow Set_1$
    $Set_0 \leftarrow \{\neg b_0 \in V_0 | b_1 \in V_{1a_1}\}$
    $Set_1 \leftarrow \{d_1 \in V_1 | \exists c_0 \in Set_0((c_0 \overset{l_i}{\Rightarrow} d_1) \in E_0)\}$
    $E_{1a_0} \leftarrow E_{1a_0} \cup \{b_1 \rightarrow \neg b_0 \in E_1 | b_1 \in \overline{Set_1}\}$
    $E_{0a_0} \leftarrow E_{0a_0} \cup \{c_0 \overset{l}{\Rightarrow} d_1 | \exists c_0 \in Set_0(c_0 \overset{l}{\Rightarrow} d_1)\}$
    $E_0 \leftarrow E_0 \setminus E_{0a_0}$
    $E_1 \leftarrow E_1 \setminus E_{1a_0}$
    $V_{0a_0} \leftarrow V_{0a_0} \cup (Set_0)$
    $V_{1a_0} \leftarrow V_1 \cup (Set_1)$
    $V_0 \leftarrow V_0 \setminus (\overline{Set_0})$
    $V_1 \leftarrow V_1 \setminus (\overline{Set_1})$
  end while
  if $E_{a_0} \neq \emptyset$ **then**
    for all $c_0 \Rightarrow d_1$ and $d_0 \Rightarrow c_1$ in V_{0a_0} **do**
      $label_{a_0}(c, d) = label(c, d)$
    end for
    $[a_0, q] = (V_{0a_0}, V_{1a_0}, E_{a_0}, E_{a_0}, label_{a_0}(c, d))$
  else {interval starting at a_0 are empty}
    $[a_0, q] = \emptyset$
  end if
end for

Algorithm 5 Write Non Empty Intervals Part 2 of 2

for all $[a_0, q] = (V_{0a_0}, V_1a_0, E_{0a_0}, E_{1a_0}, label_{a_0}(c, d)) \neq \emptyset$ **do**
  for all $b_1 \in V_{a_01}$ **do**
    $V_{1a_0b_1} \leftarrow \{b_1\}$
    $V_{0a_0b_1} \leftarrow \{c_0 | c_0 \Rightarrow b_1 \in E_{0a_0}\}$
    $E_{0a_0b_1} \leftarrow \{c_0 \stackrel{l}{\Rightarrow} b_1 \in E_{0a_0}\}$
    $\overline{Set_1} \leftarrow \overline{Set_0} = \emptyset$
    $Set_1 \leftarrow \{b_1\}$
    $Set_0 \leftarrow \{c_0 \in V_{0a_0} | (c_0 \stackrel{l}{\Rightarrow} b_1) \in E_{0a_0}\}$
    $E_{1a_0b_1} \leftarrow \emptyset$
    $label_{a_0b_1}(c, d) \leftarrow \emptyset$
    while $Set_1 \neq \emptyset$ **do**
      $\overline{Set_0} \leftarrow Set_0$
      $\overline{Set_1} \leftarrow Set_1$
      $Set_1 \leftarrow \{\neg c_1 \in V_{1a_0} | c_0 \in V_{0a_0b_1}\}$
      $Set_0 \leftarrow \{e_0 \in V_{0a_0} | \exists d_1 \in Set_1(e_0 \Rightarrow d_1 \in E_{0a_0})\}$
      $E_{1a_0b_1} \leftarrow E_{1a_0b_1} \cup \{\neg c_1 \rightarrow c_0 \in E_{1a_0} | c_0 \in \overline{Set_0}\}$
      $E_{0a_0b_1} \leftarrow E_{0a_0b_1} \cup \{c_0 \stackrel{l_i}{\Rightarrow} d_1 | \exists d_1 \in Set_1(c_0 \stackrel{l}{\Rightarrow} d_1))$
      $E_{0a_0} \leftarrow E_{0a_0} \setminus E_{0a_0b_1}$
      $E_{1a_0} \leftarrow E_{1a_0} \setminus E_{1a_0b_1}$
      $V_{0a_0b_1} \leftarrow V_{0a_0b_0} \cup (Set_0)$
      $V_{1a_0b_1} \leftarrow V_{1a_0b_1} \cup (Set_1)$
      $V_{0a_0} \leftarrow V_{0a_0} \setminus (\overline{Set_0})$
      $V_{1a_0} \leftarrow V_{1a_0} \setminus (\overline{Set_1})$
    end while
    if $E_{a_0b_1} \neq \emptyset$ **then**
      for all $c_0 \Rightarrow d_1$ and $d_0 \Rightarrow c_1$ in $V_{0a_0b_1}$ **do**
        $label_{a_0b_1}(c, d) = label_{a_0}(c, d)$
      end for
      $[a_0, b_1] = (V_{0a_0b_1}, V_{1a_0b_1}, E_{0a_0b_1}, E_{1a_0b_1}, label_{a_0b_1}(c, d))$
    else
      $[a_0, b_1] = \emptyset$
    end if
  end for
end for
for all a so that $[a_0, a_1] \neq \emptyset$ **do**
  $Nectrue \leftarrow Nectrue \cup \{a\}$
end for

Algorithm 6 Closed Digraph Part 1 of 2

$Input\ :\ Closed\ Intervals, Nectrue = \{a_0^1, \ldots, a_0^k\}, Necfalse = \{b_0^1, \ldots .b_0^n\},$

$V_{0Cls} \leftarrow \emptyset$, $V_{1Cls} \leftarrow \emptyset$, $E_{0Cls} \leftarrow \emptyset$, $E_{1Cls} \leftarrow \emptyset$,
$label_{Cls}(a, b) \leftarrow \emptyset$
$disting \leftarrow 0$
for all $1 \leq i < k$ and $i < j \leq k$ **do**
  if $[a_0^i, a_1^i]$ and $[\neg a_0^i, \neg a_1^j]$ and $[a_0^j, a_1^j]$ are non empty **then**
    for all $c_0 \in [a_0^i, a_1^i]$ **do**
      $V_{0Cls} \leftarrow V_{0Cls} \cup \{(c_0, disting)\}$
    end for
    for all $c_1 \in [a_0^i, a_1^i]$ **do**
      $V_{1Cls} \leftarrow V_{1Cls} \cup \{(c_0, disting)\}$
    end for
    $E_{1Cls} \leftarrow \{(a_1^i, disting) \rightarrow (\neg a_0^i, disting + 1)\}$
    $disting \leftarrow disting + 1$
    for all $c_0 \in [\neg a_0^i, \neg a_1^j]$ **do**
      $V_{0Cls} \leftarrow V_{0Cls} \cup \{(c_0, disting)\}$
    end for
    for all $c_1 \in [\neg a_0^i, \neg a_1^j]$ **do**
      $V_{1Cls} \leftarrow V_{1Cls} \cup \{(c_1, disting)\}$
    end for
    $E_{1Cls} \leftarrow \{(\neg a_1^j, disting) \rightarrow (a_0^j, disting + 1)\}$
    $disting \leftarrow disting + 1$
    for all $c_0 \in [a_0^j, a_1^j]$ **do**
      $V_{0Cls} \leftarrow V_{0Cls} \cup \{(c_0, disting)\}$
    end for
    for all $c_1 \in [a_0^j, a_1^j]$ **do**
      $V_{1Cls} \leftarrow V_{1Cls} \cup \{(c_1, disting)\}$
    end for
    $disting \leftarrow disting + 1$
  end if
end for

Algorithm 7 Closed Digraph Part 2 of 2

for all $1 \leq i \leq k$ and $1 \leq j \leq n$ **do**
  if $[a_0^i, a_1^i]$ and $[\neg a_0^i, b_1^j]$ are non empty **then**
    for all $c_0 \in [a_0^i, a_1^i]$ **do**
      $V_{0Cls} \leftarrow V_{0Cls} \cup \{(c_0, disting)\}$
    end for
    for all $c_1 \in [a_0^i, a_1^i]$ **do**
      $V_{1Cls} \leftarrow V_{1Cls} \cup \{(c_1, disting)\}$
    end for
    $E_{1Cls} \leftarrow \{(a_1^i, disting) \rightarrow (\neg a_0^i, disting + 1)\}$
    $disting \leftarrow disting + 1$
    for all $c_0 \in [\neg a_0^i, b_1^j]$ **do**
      $V_{0Cls} \leftarrow V_{0Cls} \cup \{(c_0, disting)\}$
    end for
    for all $c_1 \in [\neg a_0^i, b_1^j]$ **do**
      $V_{1Cls} \leftarrow V_{1Cls} \cup \{(c_1, disting)\}$
    end for
    $V_{0Cls} \leftarrow V_{0Cls} \cup \{(\neg b_0^j, disting + 1)\}$
    $E_{1Cls} \leftarrow \{(b_1^j, disting) \rightarrow (\neg b_0^j, disting + 1)\}$
    $disting \leftarrow disting + 2$
  end if
end for
for all $1 \leq i < n$ and $i < j \leq n$ **do**
  if $[b_0^i, b_1^j]$ in non empty **then**
    for all $c_0 \in [b_0^i, b_1^j]$ **do**
      $V_{0Cls} \leftarrow V_{0Cls} \cup \{(c_0, disting)\}$
    end for
    for all $c_1 \in [b_0^i, b_1^j]$ **do**
      $V_{1Cls} \leftarrow V_{1Cls} \cup \{(c_1, disting)\}$
    end for
    $V_{0Cls} \leftarrow V_{0Cls} \cup \{(\neg b_0^i, disting - 1), (\neg b_0^j, disting + 1)\}$
    $E_{1Cls} \leftarrow \{(b_1^j, disting) \rightarrow (\neg b_0^j, disting + 1)\}$
    $disting \leftarrow disting + 3$
  end if
  for all $(a_0, n) \in V_{0Cls}$ and $(b_1, n) \in V_{1Cls}$ **do**
    if $a_0 \Rightarrow b_1 \in E_0$ (is an edge of the cylindrical digraph) **then**
      $(a_0, n) \Rightarrow (b_1, n) \in E_{0Cls}$
      $label((a_0, n) \Rightarrow (b_1, n)) = label(a_0 \Rightarrow b_1)$
    end if
  end for
  for all $(a_0, n) \in V_0$, $(\neg a_1, n) \in V_1$ **do**
    $(\neg a_1, n) \rightarrow (a_0, n) \in E_1$
  end for
end for

Algorithm 8 Roots

Input: Closed Digraph, **CSDG** $= (V_{Cls0}, V_{Cls1}, E_{Cls0}, E_{Cls1})$

$\mathbb{R} \leftarrow \emptyset$
for all $(a_0, n) \in V_{0Cls}$ **do**
  if $(\neg a_1, n) \notin V_{1Cls}$ **then**
    $\mathbb{R} \leftarrow \mathbb{R} \cup \{(a_0, n)\}$
  end if
end for

Algorithm 9 Source

Input: **CSDG** $= (V_{Cls0}, V_{Cls1}, E_{Cls0}, E_{Cls1})$
$\mathbb{F} \leftarrow \emptyset$
for all $(a_0, n) \in V_{0Cls}$ **do**
  if $\exists(b_1, n) \in L_1 \exists(b'_1, n) \in L_1\big((a_0, n) \Rightarrow (b_1, n)\ AND (a_0, n) \Rightarrow (b'_1, n)\ AND\ (b_1 \neq b'_1)\big)$ OR $(a_0, n) \in \mathbb{R}$ **then**
    $\mathbb{F} \leftarrow \mathbb{F} \cup \{a_0\}$
  end if
end for

Algorithm 10 Spillway

Input: **CSDG** $= (V_{Cls0}, V_{Cls1}, E_{Cls0}, E_{Cls1})$
$\mathbb{S} \leftarrow \emptyset$
for all $(b_1, n) \in V_{1Cls}$ **do**
  if $\exists(a_0, n) \in V_{0Cls} \exists(a'_0, n) \in V_{0Cls}\big((a_0, n) \Rightarrow (b_1, n)\ AND\ (a'_0, n) \Rightarrow b_1, n)\ AND\ (a_0 \neq a'_0)\big)\ OR\ \nexists(\neg b_0, m) \in L_1((b_1, n) \rightarrow (\neg b_0, m) \in E_1)$ **then**
    $\mathbb{S} \leftarrow \mathbb{S} \cup \{b_1\}$
  end if
end for

Algorithm 11 Antichain

Input partially ordered sets $V \subseteq \mathbb{U}$

% Given a set V, does V contain an antichain?

```
for all v ∈ V do
  for all u ∈ 𝕌 do
    if COMP(u, v) then
      𝕌 ← 𝕌 \ {u}
    end if
  end for
end for
if 𝕌 = ∅ then
  V has an antichain of 𝕌
else {there in an element of 𝕌 non comparable with any element of V}
  V has no antichain of 𝕌
end if
```

Algorithm 12 Alternated Sequence Label

Input: **CSDG** $= (V_0, V_1, E_0, E_1, label(a, b))$, $\mathbb{S}$, $\mathbb{F}$, $\mathbb{R}$. I will write (a_0, m) or (a_1, m) as a_0 and a_1. Edges are clearly defined.

```
for all a_0 ⇒ b_1 ∈ E_0 do
  ASLabel(a_0 ⇒ b_1) ← 0
end for
for all a_0 ∈ 𝔽 do
  for all b_1 ∈ V_1 do
    if a_0 ⇒ b_1 ∈ E_0 then
      ASLabel(a_0 ⇒ b_1) ← ASLabel(a_0 ⇒ b_1) + 1
    end if
  end for
end for
while ∃a_0 ⇒ b_1 ∈ E_0(ASLabel(a_0 ⇒ b_1) = 0) do
  for all c_0 ⇒ d_1 ∈ E_0 so that ASLabel(c_0 ⇒ d_1) = 0) do
    if ASLabel(e_0 ⇒ ¬c_1) ≠ 0 then
      ASLabel(c_0 ⇒ d_1) = ASLabel(e_0 ⇒ ¬c_1)
    end if
  end for
end while
```

Algorithm 13 Label the ordered set U in closed digraph

Input $\mathbb{R}, \mathbb{F}, \mathbf{CSDG} = (L_0, L_1, E_0, E_1), ASLabel(c_0 \Rightarrow d_1)$,

for all $a_0 \Rightarrow b_1$ and $c_0 \Rightarrow d_1$ **do**

$a_0 \Rightarrow b_1 \sim c_0 \Rightarrow d_1$ if $ASLabel(a_0 \Rightarrow b_1) = ASLabel(c_0 \Rightarrow d_1)$

end for

RECURSIVELY,

if $a_0 \Rightarrow b_1 \in E_0\ AND\ a_0 \in \mathbb{R}$ **then**

$u(a_0 \Rightarrow b_1) = ASLabel(c_0 \Rightarrow d_1)$

end if

if $a_0 \Rightarrow b_1 \in E_0\ AND\ (a_0 \in \mathbb{F} \setminus \mathbb{R}\ OR\ \neg a_1 \in \mathbb{S})$ **then**

$u(a_0 \Rightarrow b_1) = (B_1 \bullet \cdots \bullet B_i)A$, where $B_j = u(c_0^j \Rightarrow \neg a_1)$, for all $c_0^j \Rightarrow \neg a_1 \in E_0$, and $ASLabel(a_0 \Rightarrow b_1) = A$

for all $c_0 \Rightarrow d_1$ **do**

if $c_0 \Rightarrow d_1 \sim a_0 \Rightarrow b_1\ AND\ a_0 \in \mathbb{F}\ OR\ \neg a_1 \in \mathbb{S})$ **then**

$u(c_0 \Rightarrow d_1) = u(a_0 \Rightarrow b_1)$

end if

end for

end if

Define the partial order $\leq$ over $\mathbb{U} = \{v | \exists a_0 \Rightarrow b_1 \in E_0 (v = u(a_0 \Rightarrow b_1))\}$ as $v_1 \leq v_2$ as the transitive closure of the relation $v_2 = (B_1 \bullet \cdots \bullet B_i)A$ and $v_1 = B_i$, for some i.

Output: $u(v), v \in E_0$

Algorithm 14 SOLVE

Input **CSDG** $= (V_0, V_1, E_0, E_1, label(a,b)), u(c_0 \Rightarrow d_1)$

Let $\trianglelefteq$ be a linear order over E_0,
for all $(r \trianglelefteq s)$, so that $u(r)$ and $u(s)$ are INCOMPARABLE with respect to $\leq$ **do**
  if $Compatible(label(r), label(s))$ (see Algorithm 3) **then**
    setarray$(s) \leftarrow r$
    Nest$(r) \leftarrow s$
  end if
end for
while no new operation can be performed **do**
  for all $(r \trianglelefteq s)$, so that $INCOMPARABLE(u(r), u(s))$ **do**
    $C \leftarrow \big(\{u(x)|x \in$ **setarray**$(r)\} \cup \{u(y)|y \in Nest(r)\}\big) \cap \big(\{u(z)|z \in$ **setarray**$(s)\} \cup \{u(w)|w \in Nest(s)\}\big)$
    if C has no antichain (w.r.t. $\mathbb{U}$) **then**
      $Nest(r) \leftarrow Nest(r) \setminus \{s\}$
      setarray$(s) \leftarrow$ **setarray**$(s) \setminus \{r\}$
    end if
  end for
end while
if all $Nest$ and **setarray** were erased **then**
  stop the computation with the output **There is no compatible antichain**
else {the computation stops because the set of macronodes was kept unchanged}
  the output is **There are compatible antichains**.
end if
Obtain the output **Solve**, a list with **setarray** of all macronodes.