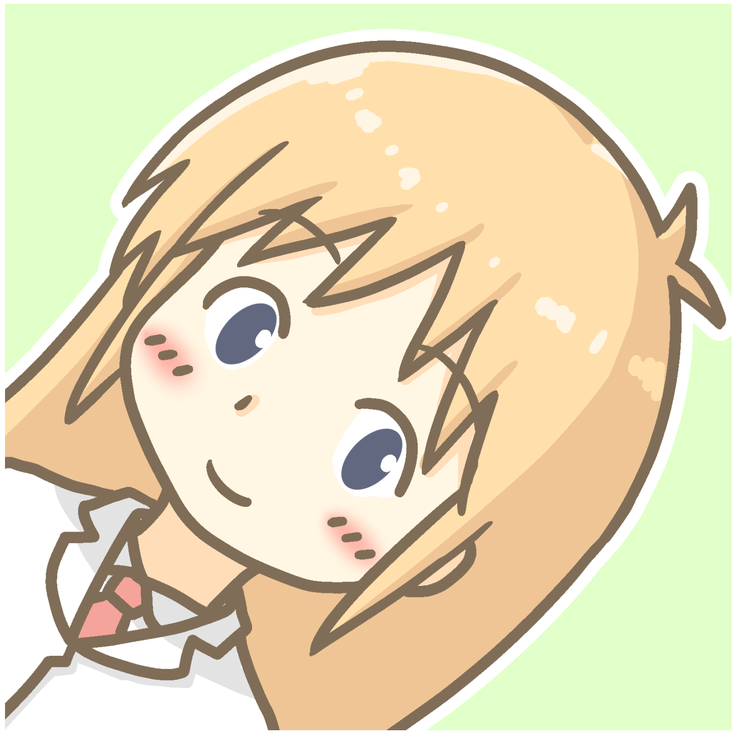
SATソルバー（中級編）

@public_yusuke
筑波大学情報学群情報科学類１年次

自己紹介

@public_yusuke（一木 祐介）



- ペペロンチーノ飽きた😭
- 生活習慣を直したい🤔
- 情報科学特別演習で SAT ソルバーを作っている✌️

免責事項

スライド作ってたら
情報メディア入門Aみたいになった……

眠くなったら教えてください$_{zz}^{z}$

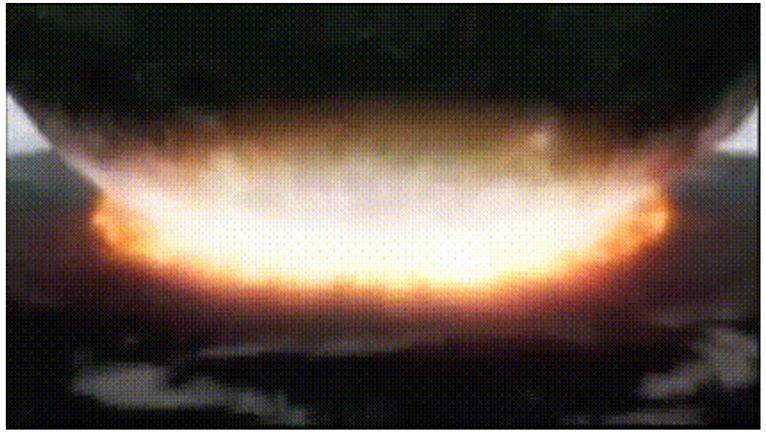
Table of Contents

スライドは 39 枚あります＞＜

すみません……

前回までのあらすじ

DPLL の概要（前回までのあらすじ）



←愚直に $O(2^N)$ で真理値表を
書いてしまった場合の世界線

充足可能性問題 をある程度高速に解きたい

→ 探索の枝刈りをしたい

→ Davis-Putnam-Logemann-Loveland アルゴリズム
（略して DPLL）

DPLL がする枝刈り（前回までのあらすじ）

- **単位伝播（Unit propagation）**
  - 単位節 (l) が存在するとき、リテラル l を真とする変数の割り当てを自動的に行うこと
- **空節（empty clause）の存在**
  - 空節 $()$ が F に存在するとき、少なくともその変数の割り当てでは F を充足することができない

    ⇒ 最後に割り当てた変数の真偽値を変える

    それでも空節が出たら、2番目に最後に割り当てたのを……

用語の確認（前回までのあらすじ）

a, b, c などを**変数**という。

x が変数であるとき、x と $\neg x$ を**リテラル**という。

用語の確認（前回までのあらすじ）

$l_1, \ldots, l_n$ がリテラルであるとき、$(l_1 \vee \cdots \vee l_n)$ を**節**という。

$(\cdots)$ を節としたとき、$(\cdots) \wedge \cdots \wedge (\cdots)$ を **CNF** という。

CNF は、$\bigwedge_i \bigvee_j l_{i,j}$ と表すこともできる。

節の操作（前回までのあらすじ）

節 C にリテラル l が含まれているとする。

- $l = \mathrm{T}$ であるとき
  - 節 C を削除する。

節全体が T だから、これ以降
考える必要がない

$(a \vee l)$

- $l = \mathrm{F}$ であるとき
  - 節 C からリテラル l を削除する。

そのリテラルは、これ以降
考える必要がない

$(a \vee l)$

=> 最終的に全ての節が消えれば SAT
　空節（リテラルを含まない節）が残れば UNSAT

節を操作してみよう（前回までのあらすじ）

$$(a \vee b \vee \neg c) \wedge (\neg b \vee c)$$

$b = \mathsf{F}$

$(a \vee b \vee \neg c)$ は $(a \vee \neg c)$ になり、節 $(\neg b \vee c)$ は削除

$$(a \vee \neg c)$$

$a = \mathsf{T}$

節 $(a \vee \neg c)$ の削除

全ての節が削除された（すなわち、満たされた）ので充足可能！（SAT）

閑話休題

- 「情報数学A」の後続としての科目「論理と形式化」
- DPLL アルゴリズムについて勉強する（と先輩が……）↓

第6回　（担当：水谷）一階述語論理の証明：様々な論理式の証明，演習．
第7回　（担当：海野）命題論理・一階述語論理の計算：証明探索、導出原理、単一化、SAT、SMT．
第8回　（担当：海野）命題論理・一階述語論理の計算：証明探索、導出原理、単一化、SAT、SMT．
第9回　（担当：海野）命題論理・一階述語論理の計算：証明探索、導出原理、単一化、SAT、SMT．
第10回（担当：海野）発展的な内容、授業のまとめ．

履修条件

CDCL まではやらない

CDCL とは

CDCL の概要

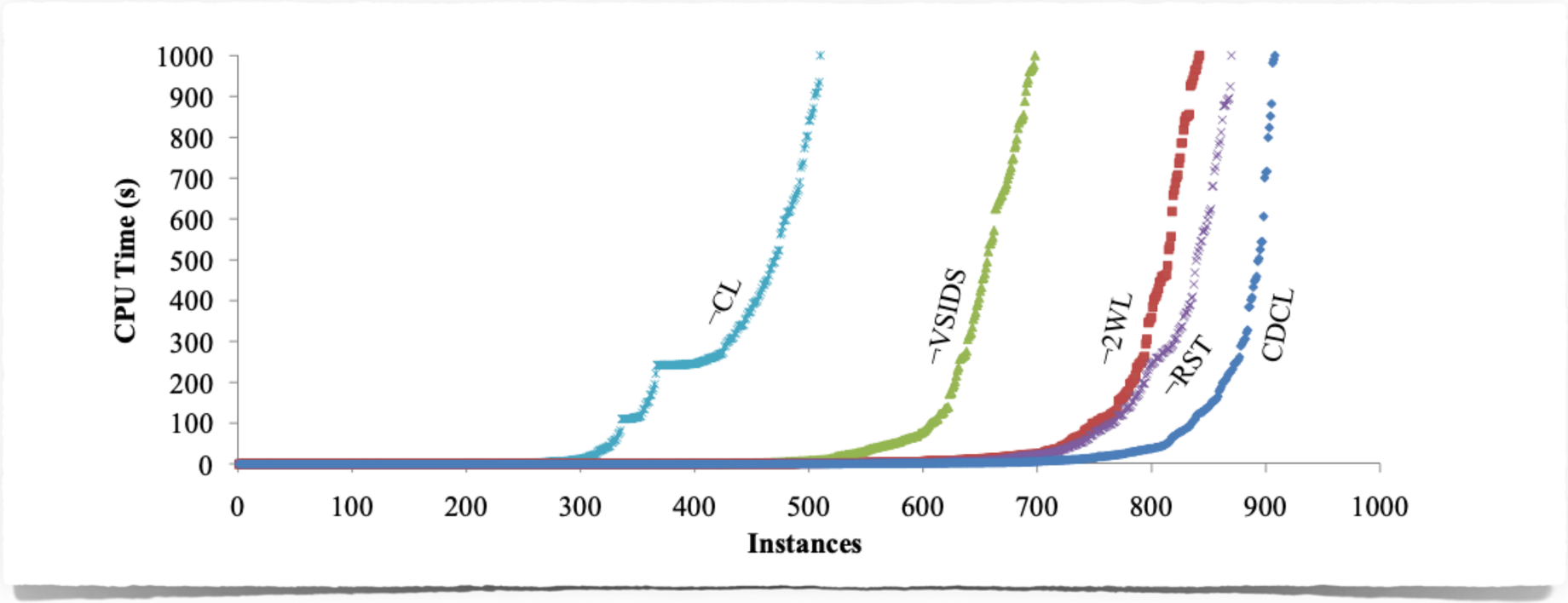
conflict-driven clause learning (CDCL)

「conflict（矛盾） からの節学習」アルゴリズム

- **DPLL** 1962 年
- **CDCL** 1996 年

やはり時代は CDCL！

CDCL の速さ



-CL が DPLLで、プレーンな CDCL が -VSIDS

実装上の工夫を加えると、さらに高速に！

出典: Empirical Study of the Anatomy of Modern Sat Solvers (2011) Katebi, Hadi and Sakallah, Karem A. and Marques-Silva, João P.

CDCL がする枝刈り

- **単位伝播（Unit propagation）**
  - 単位節 (l) が存在するとき、リテラル l を真とする変数の割り当てを自動的に行うこと
- **学習節（learnt clause)**
  - 矛盾が生じたとき、その原因となったリテラルの真偽の割り当て $(l_1 \wedge \cdots \wedge l_n)$ の否定 $(\neg l_1 \vee \cdots \vee \neg l_n)$ を学習する。
    - 少なくとも学習節を満たす必要があることがわかる

CDCL の動作

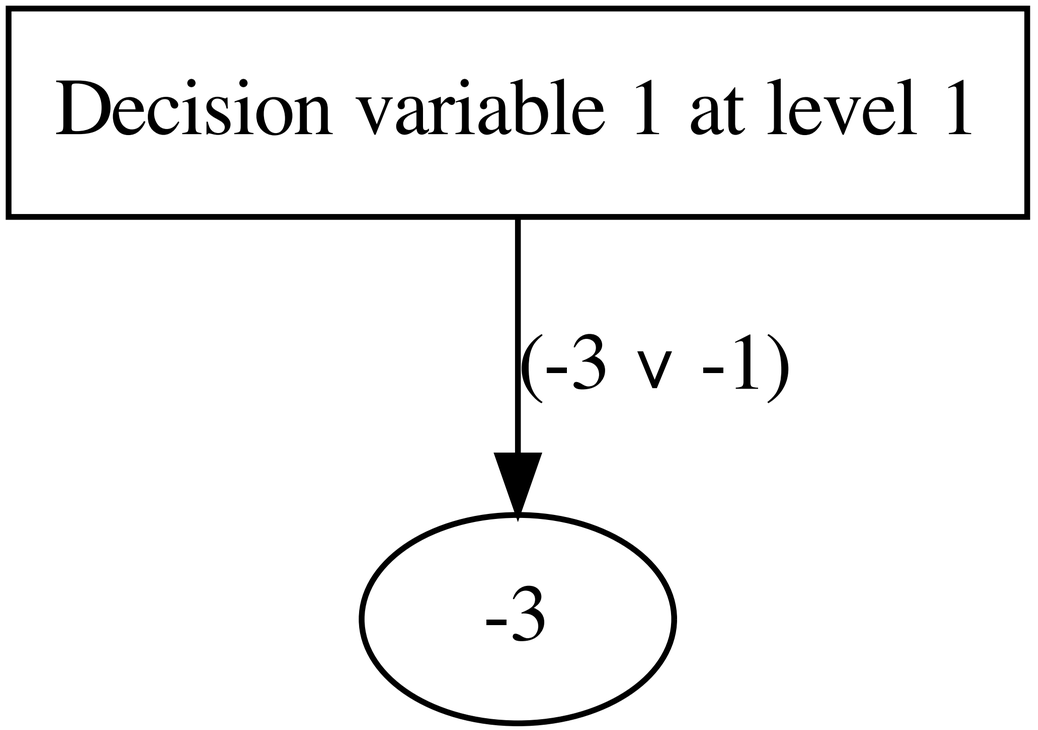
Algorithm 2.2: DPLL-ClauseLearning-Iterative

```
Input  : A CNF formula
Output : UNSAT, or SAT along with a satisfying assignment
begin
    while TRUE do
        DecideNextBranch                          ←——— 変数選択&真偽値割り当て
        while TRUE do
            status ← Deduce                       ←——— 単位伝播
            if status = CONFLICT then             ←——— もし矛盾が生じたら
                blevel ← AnalyzeConflict          ←——— 矛盾の原因を解析し学習
                if blevel = 0 then return UNSAT   ←——— 単位伝播のみで矛盾→×
                Backtrack(blevel)                 ←——— 矛盾する前にバックトラック
            else if status = SAT then             ←——— 矛盾なく割り当てが完了したら
                Output current assignment stack
                return SAT                        ←——— SAT です！
            else break
end
```

例

変数 6 個、節は 12 個のケース

リテラル 1 が真であると仮定する →

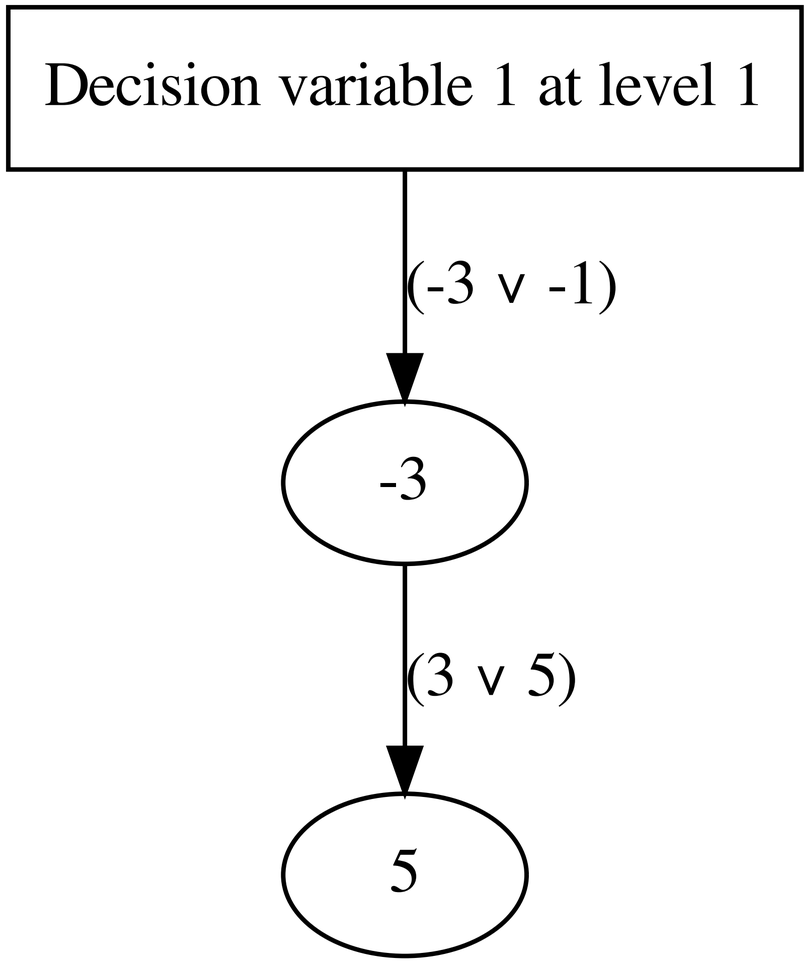


節 $(-3 \vee -1)$ が (-3) になったので、

リテラル -3 が真になる

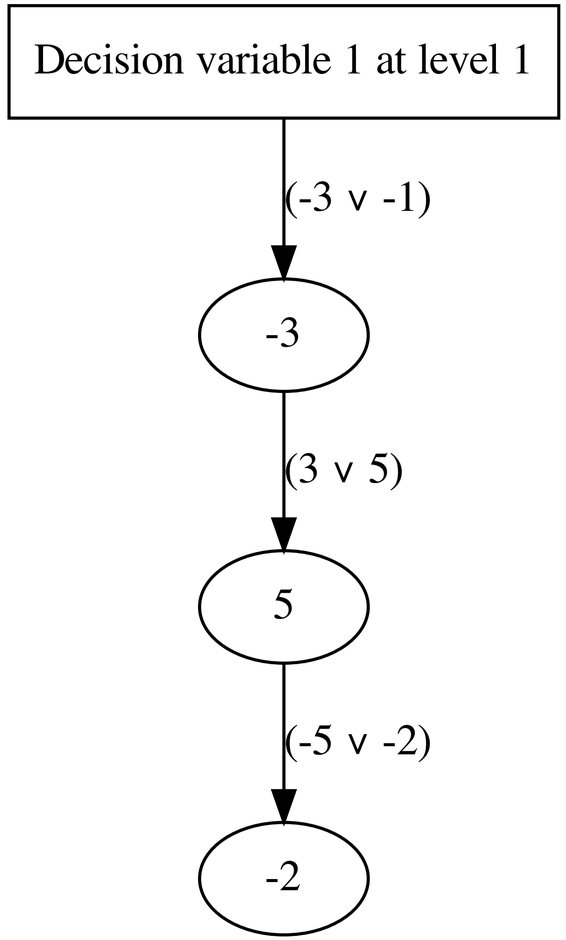
節 (3 ∨ 5) が (5) になったので、

リテラル 5 が真になる



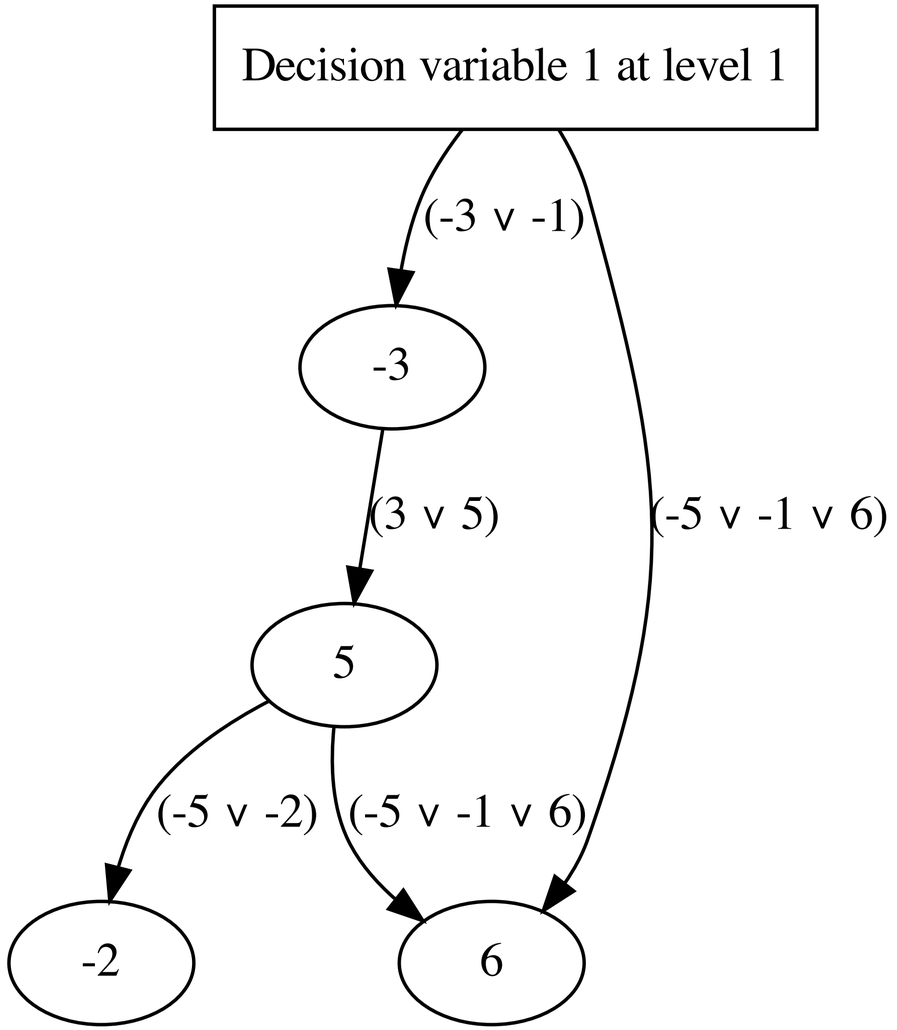
節 $(-5 \vee -2)$ が (-2) になったので、

リテラル -2 が真になる



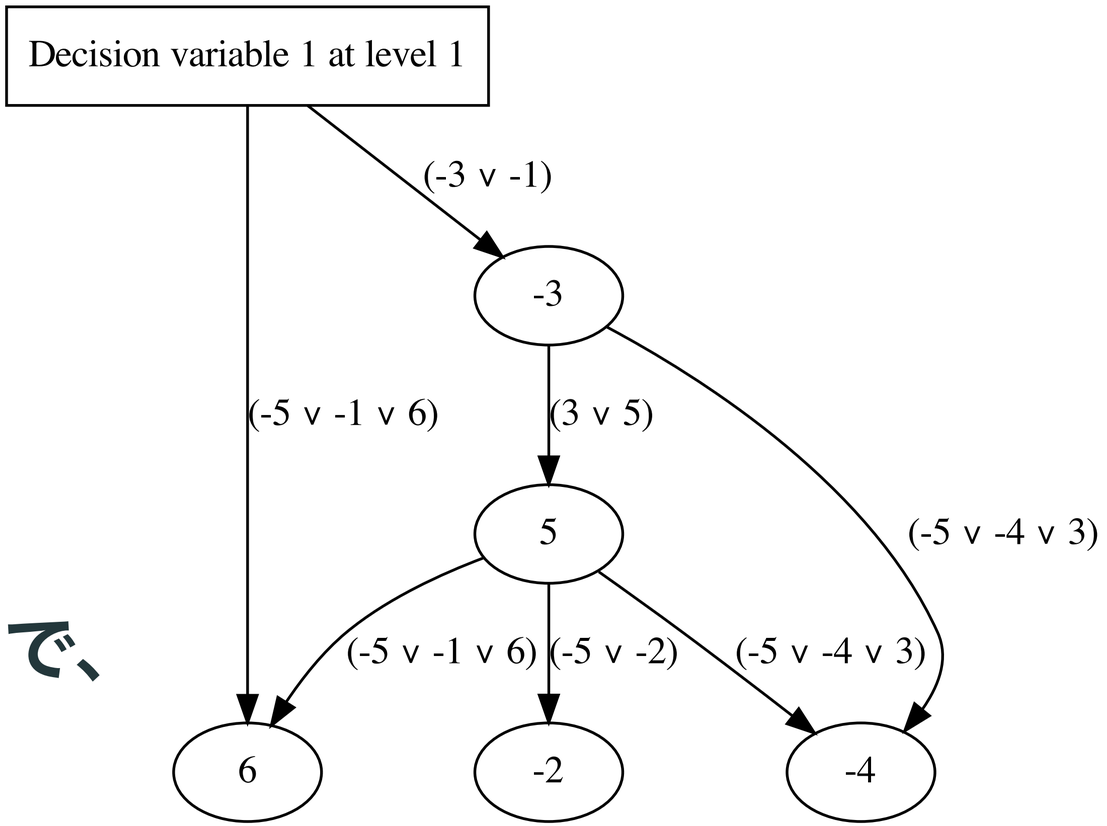
節 $(-5 \vee -1 \vee 6)$ が (6) になったので、

リテラル 6 が真になる



節 $(-5 \vee -4 \vee 3)$ が (-4) になったので、

リテラル -4 が真になる



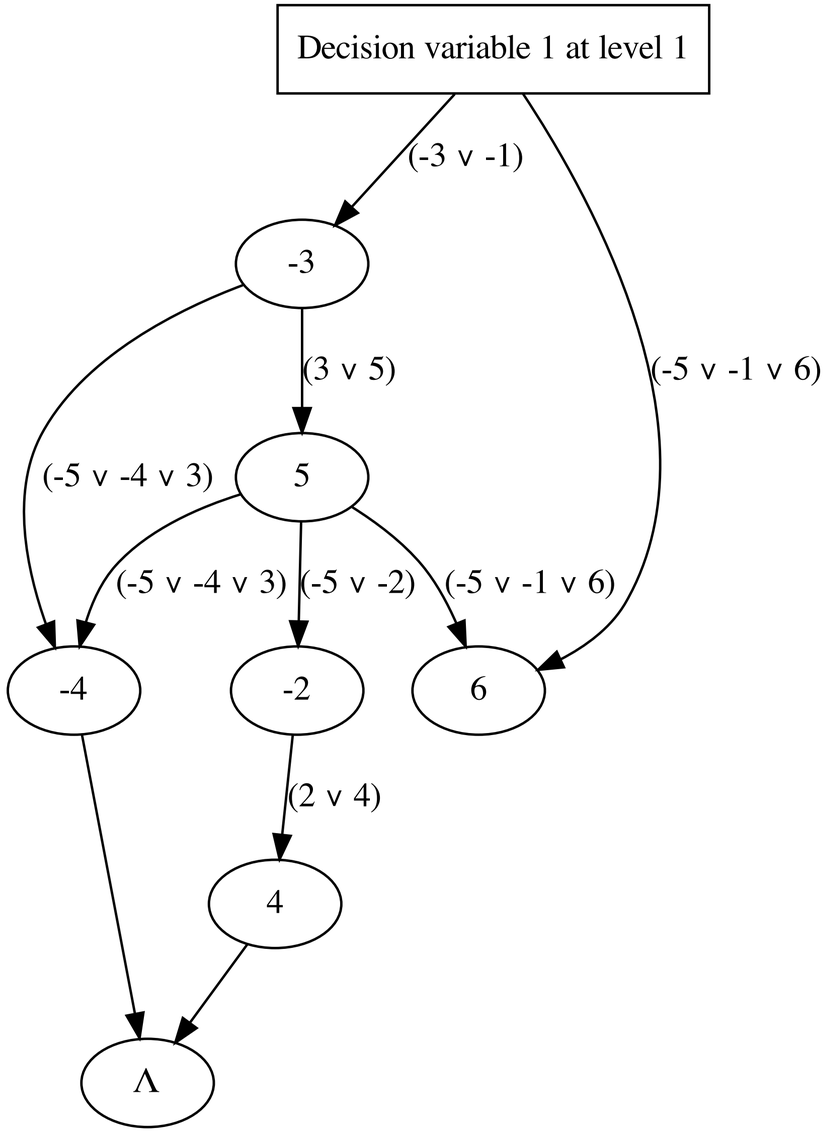
矛盾が発生した →

節 $(2 \vee 4)$ が (4) になったので、

リテラル 4 が真になる

…… リテラル $4, -4$ が真である

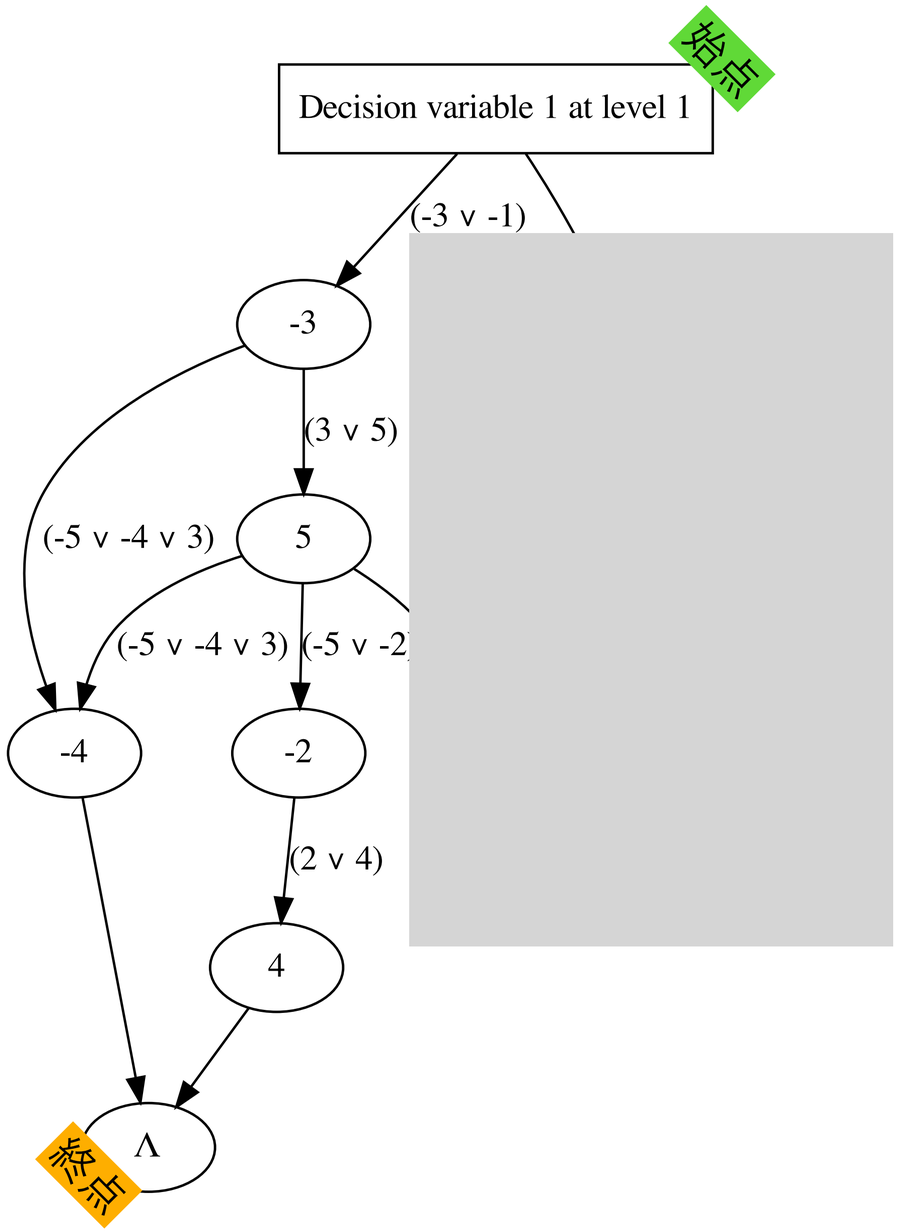
→ これはおかしい。



どうしよう

矛盾を学習

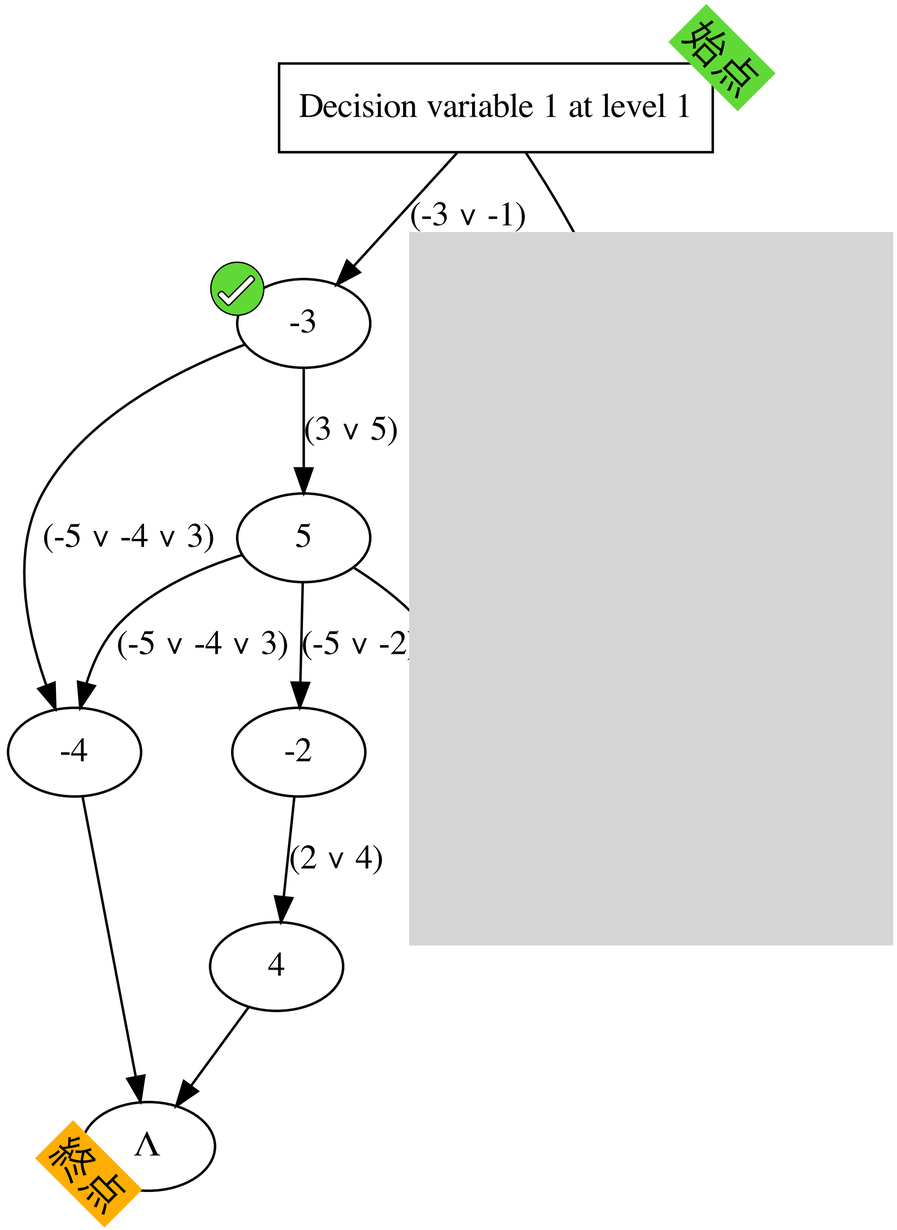
そして、バックトラック！



1UIP (first unique implication point) を探せ！

現在の level での decision variable を始点として、conflict node Λ を終点とする。

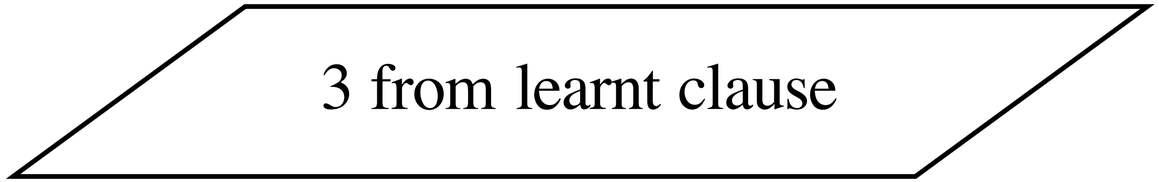
始点から終点までの任意の道に含まれる、Λ 以外で最も Λ に近い位置にある頂点のことを 1UIP と呼ぶ。



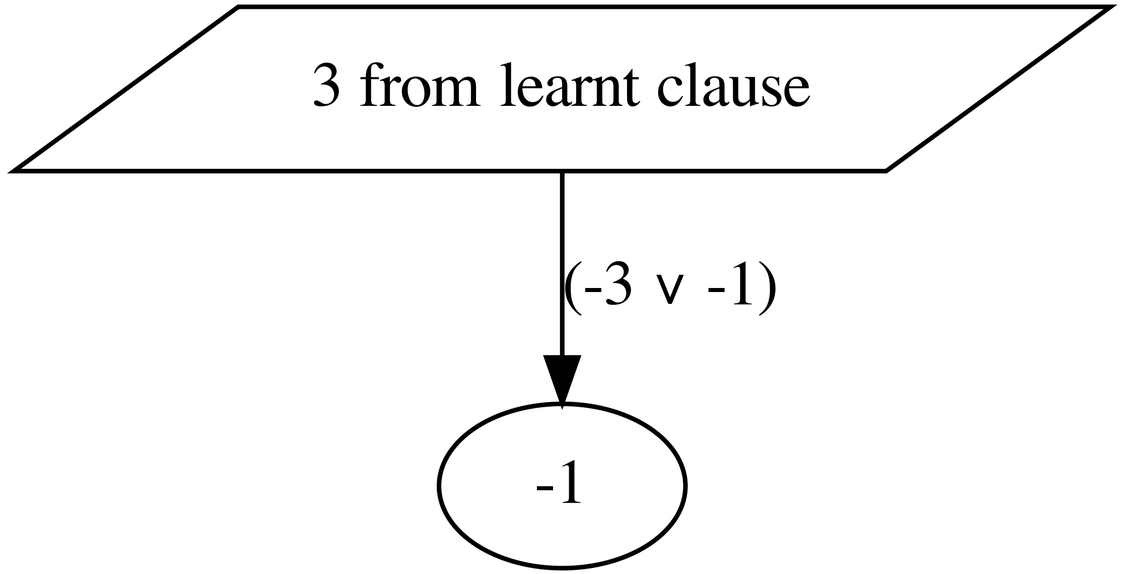
…… 今回の場合は、−3 が
1UIP になります。

(−3) の 否定は (3) ですから……

3 が学習され
単位伝播された→



バックトラックにより、変数の割り当てが無かったところまで戻された

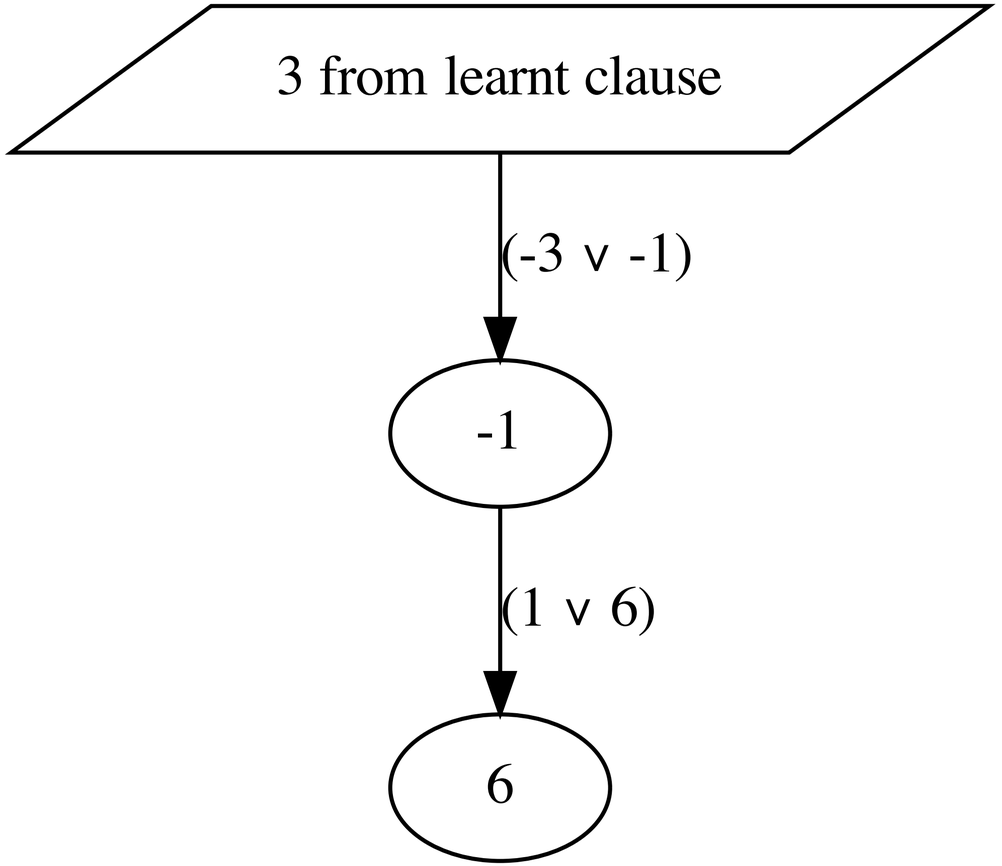


節 $(-3 \vee -1)$ が (-1) になったので、

リテラル -1 が真になる

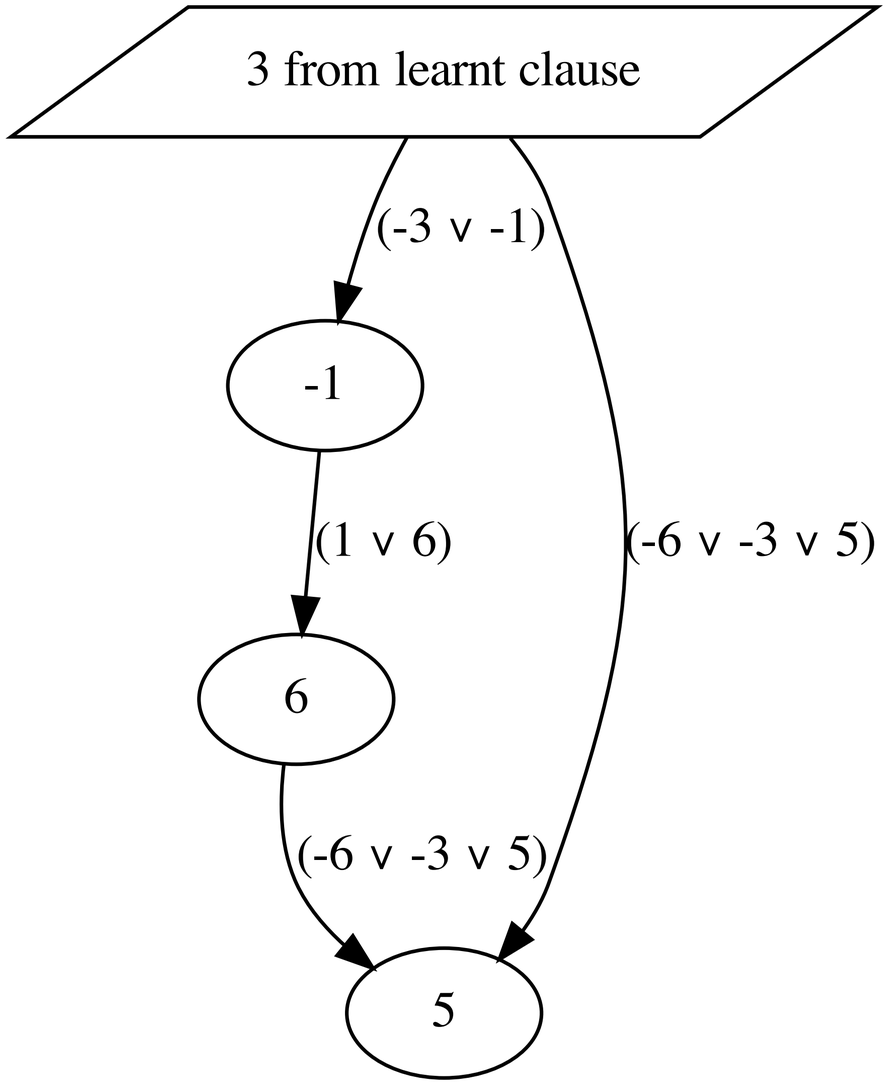
節 (1 ∨ 6) が (6) になったので、

リテラル 6 が真になる



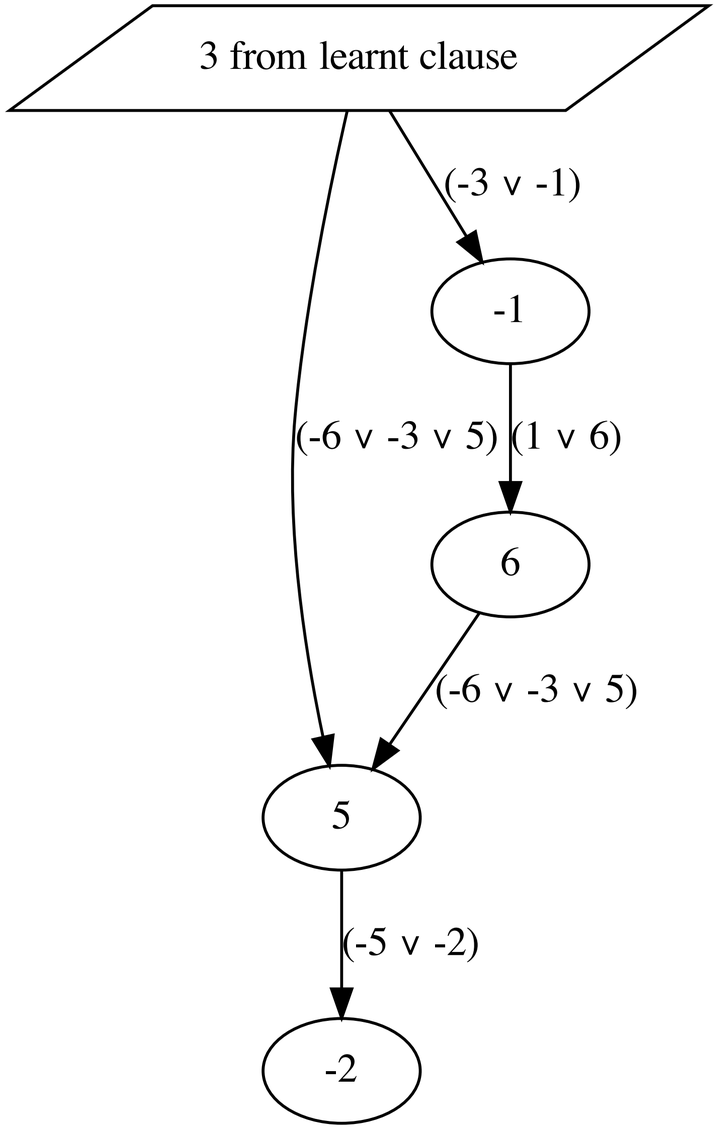
節 $(-6 \vee -3 \vee 5)$ が (5) になったので、

リテラル 5 が真になる



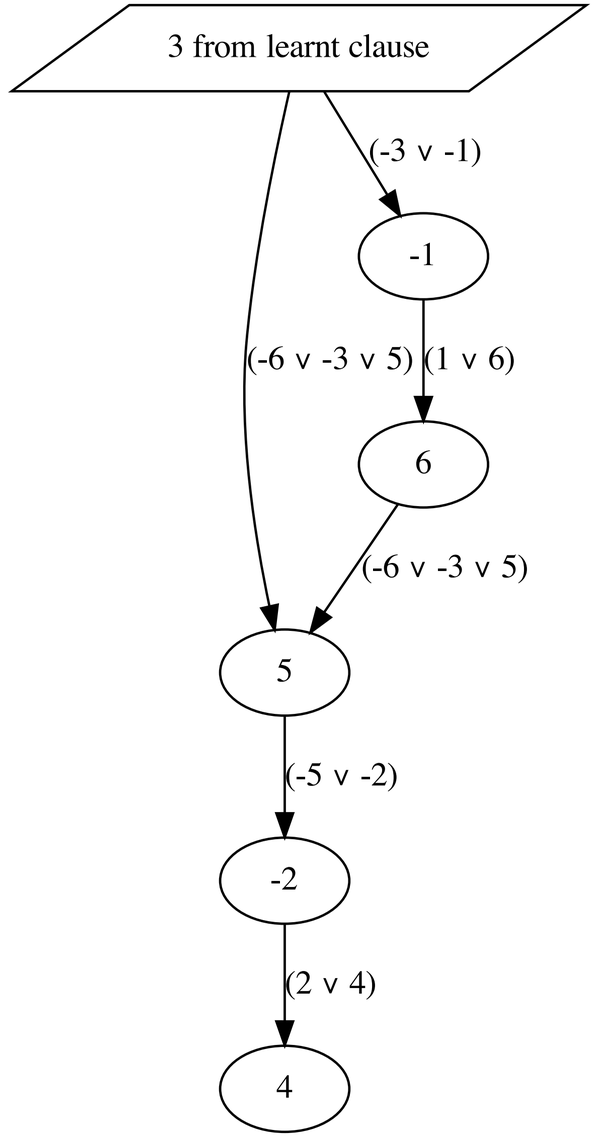
節 $(-5 \vee -2)$ が (-2) になったので、

リテラル -2 が真になる

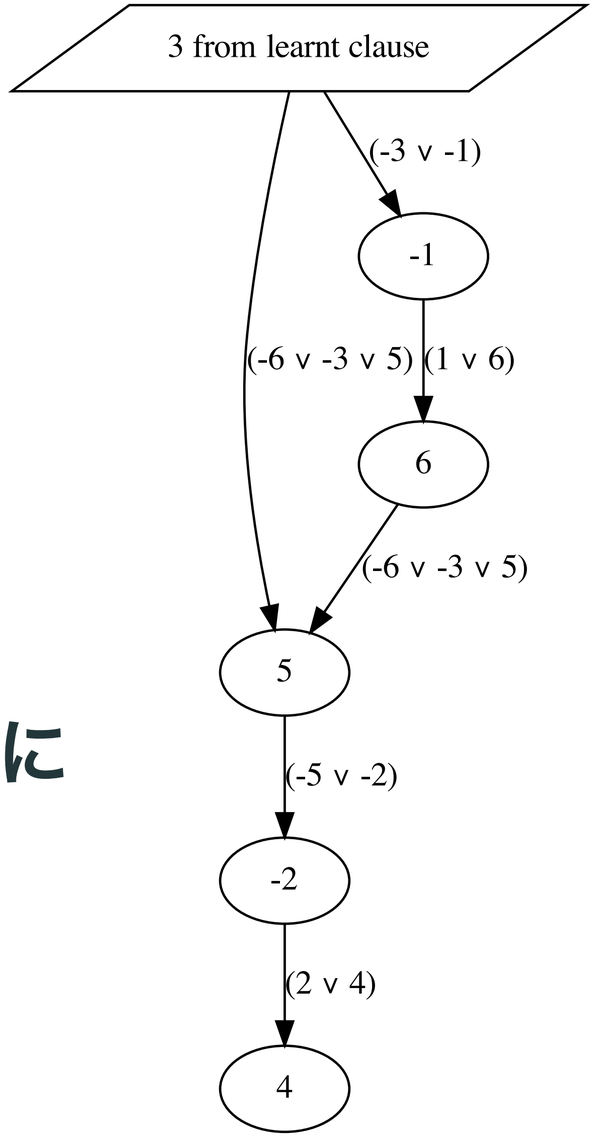


節 (2 ∨ 4) が (4) になったので、

リテラル 4 が真になる



変数 6 個それぞれを、矛盾が生じないように
定めることができたので、SAT！🎉
（すべて単位伝播で確定できちゃった例）



CDCL の D言語での実装

CDCL の D 言語での実装

https://github.com/private-yusuke/sat-d

SATソルバー（中級編）

@public_yusuke
筑波大学情報学群情報科学類１年次