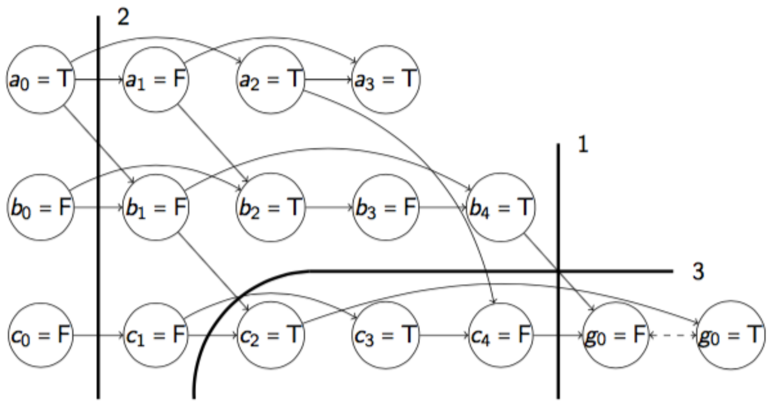
2017-10-11

- **Tseitin transformation**: For every propositional logic formula φ, there is a propositional logic formula ψ in CNF s.t. φ and ψ are equisatisfiable.
  - For every non-atomic subformula α, define C_α as follows:
  - Let $\psi = x_\phi \wedge \bigwedge\{C_\alpha | \alpha \text{ is a non-atomic subformula of } \phi\}$. Then ϕ and ψ are equisatisfiable.

α	C_α	**Remark**
$\neg\beta$	$(x_\alpha \vee x_\beta) \wedge (\neg x_\alpha \vee \neg x_\beta)$	$x_\alpha \Leftrightarrow \neg x_\beta$
$\beta_0 \vee \beta_1$	$(x_\alpha \vee \neg x_{\beta_0}) \wedge (x_\alpha \vee \neg x_{\beta_1}) \wedge (\neg x_\alpha \vee x_{\beta_0} \vee x_{\beta_1})$	$x_\alpha \Leftrightarrow x_{\beta_0} \vee x_{\beta_1}$
$\beta_0 \wedge \beta_1$	$(\neg x_\alpha \vee x_{\beta_0}) \wedge (\neg x_\alpha \vee x_{\beta_1}) \wedge (x_\alpha \vee \neg x_{\beta_0} \vee \neg x_{\beta_1})$	$x_\alpha \Leftrightarrow x_{\beta_0} \wedge x_{\beta_1}$

- SAT algorithms: *backtracking-based* v.s. *stochastic local search* algorithms.
- **Davis-Putnam-Logemann-Loveland (DPLL) algorithm** (*backtracking-based*):
  - **Resolution**: $\frac{\phi_1 \vee \psi \quad \phi_2 \vee \neg\psi}{\phi_1 \vee \phi_2}$. $\phi_1 \vee \phi_2$ is called a **conflict-driven learned clause**.
  - **Non-chronological backtracking**: When a learned clause is generated, backtrack to the next-to-the-last variable in the clause to prevent the conflict from reoccurring.
  - **Unique implication point (UIP)**: Given a cut in an **implication graph**, an internal node causing a conflict is called a *UIP* if it is one and the only one node in the same level as the conflict.
  - Emperically, the first UIP is the best.
- Examples of UIP. Cut 1: no UIP. Cut 2: c_0 is the UIP. Cut 3: c_1 is the UIP.



- MiniSet: http://minisat.se/Main.html (http://minisat.se/Main.html)
- Informally, a **predicate** is a function from objects to truth values.
- Example: every student is younger than some instructor.
  - $S(x)$ means x is a student; $I(y)$ means y is an instructor; $Y(x, y)$ means x is younger than y.
  - $\forall x(S(x) \Rightarrow (\exists y(I(y) \wedge Y(x, y))))$
- Symbols in predicate logic: **predicate symbols** P, **function symbols** F, **constant symbols** C.
- $C \subseteq F$: **0-arity** (or **nullary**) function is in fact a constant.
- **Terms**: $t ::= x \mid c \mid f(t, \ldots, t)$, where x is a variable, $c \in F$ a nullary function symbol, and $f \in F$.
- **Formulae**: $\varphi ::= P(t_1, \ldots, t_n) \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \Rightarrow \varphi) \mid (\forall x\varphi) \mid (\exists x\varphi)$, where x is a variable, $t_1, \ldots, t_n$ are

terms over F, and $P \in P$ is a predicate symbol of arity n.

- Free and bound variables:
  - **Free** variables: An occurrence of x in φ is *free* in φ if it is a leaf node without ancestor nodes $\forall x$ or $\exists x$ in the parse tree of φ.
  - **Bound** variables: Otherwise, the occurrence of x is *bound*.
- **Substitution** rules:
  - Variables can be replaced by terms (but not formulae).
  - Given a variable x, a term t and a formula φ. Define $\varphi[t/x]$ to be the formula obtained by replacing each *free* variable x in φ with t.
  - Let t be a term, x a variable, and φ a formula. t is *free for* x in φ if no free x in φ occurs in the scope of $\forall y$ or $\exists y$ for any variable y occurring in t.
  - Bound variables can always be renamed for substitution.