Logic and Computability

Topic 1: Theories in Predicate Logic – Lazy Encoding

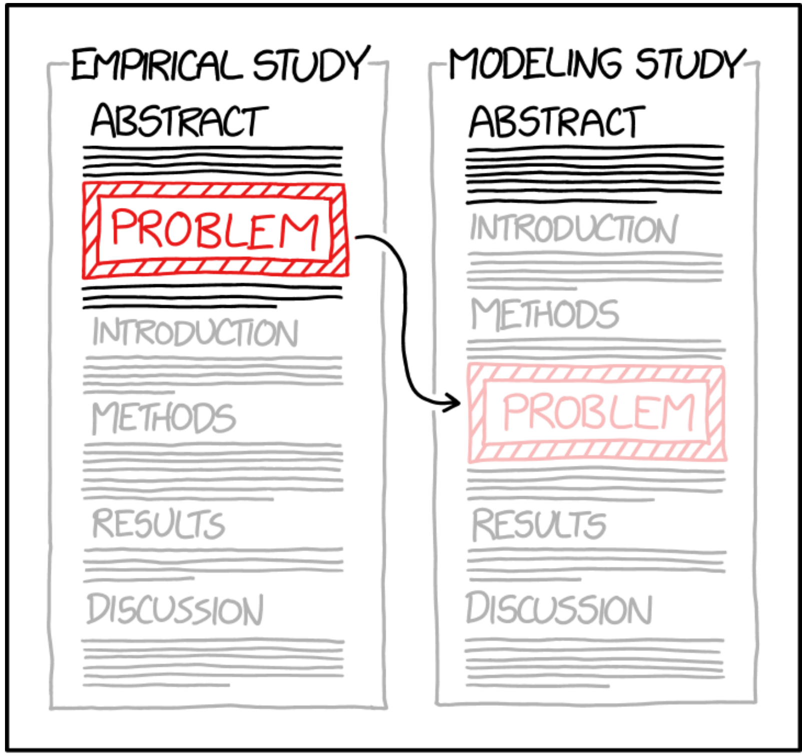
Topic 2: Symbolic Encoding

Bettina Könighofer

bettina.koenighofer@iaik.tugraz.at

Stefan Pranger

stefan.pranger@iaik.tugraz.at



A MATHEMATICAL MODEL IS A POWERFUL TOOL FOR TAKING HARD PROBLEMS AND MOVING THEM TO THE METHODS SECTION.

https://xkcd.com/2323/

TU Graz

SCIENCE PASSION TECHNOLOGY

Plan for Today

- **Part 1 – Lazy Encoding / DPLL(T)**
  - Recap: Theories in Predicate Logic
  - Recap: Lazy Encoding and Congruence Closure
  - Simplified Version of DPLL(T)
    - Discuss via example

- **Part 2 – Symbolic Encoding**
  - Transition systems
  - Symbolic representation of sets of states
  - Symbolic representation of the transition relation
  - Symbolic encodings of arbitrary sets
  - Set operations on symbolically encoded sets

Learning Outcomes

After this lecture…

1. students can explain the simplified version of DPLL(T), especially the interaction of SAT solver and theory solver.
2. students can apply the simplified version of DPPL(T) to decide the satisfiability of formulas in $\mathcal{T}_{UFE}$.

Recap - Definition of a Theory

Definition of a First-Order Theory $\mathcal{T}$:

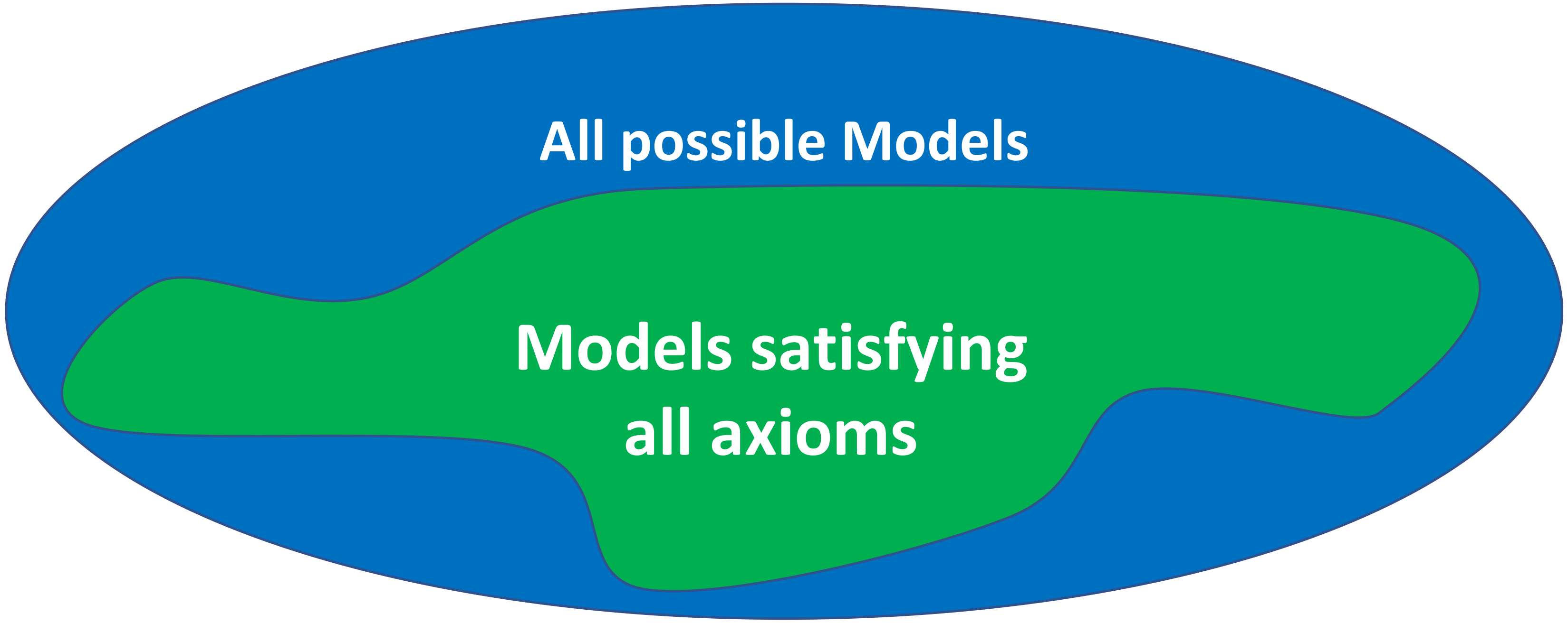
- Signature Σ
  - Defines the set of **constants, predicate and function symbols**
- Set of Axioms $\mathcal{A}$
  - Gives **meaning** to the predicate and function symbols

Example: Theory of Lineare Integer Arithmetic $\mathcal{T}_{\mathrm{LIA}}$:

- $\Sigma_{\mathrm{LIA}} := \mathbb{Z} \cup \{+, -\} \cup \{=, \neq <, \leq, >, \geq\}$
- $\mathcal{A}_{LIA}$: defines the usual meaning to all symbols
  - E.g., The function **+** is interpreted as the addition function, e.g.
    - …
    - 0+0 → 0
    - 0+1 → 1….

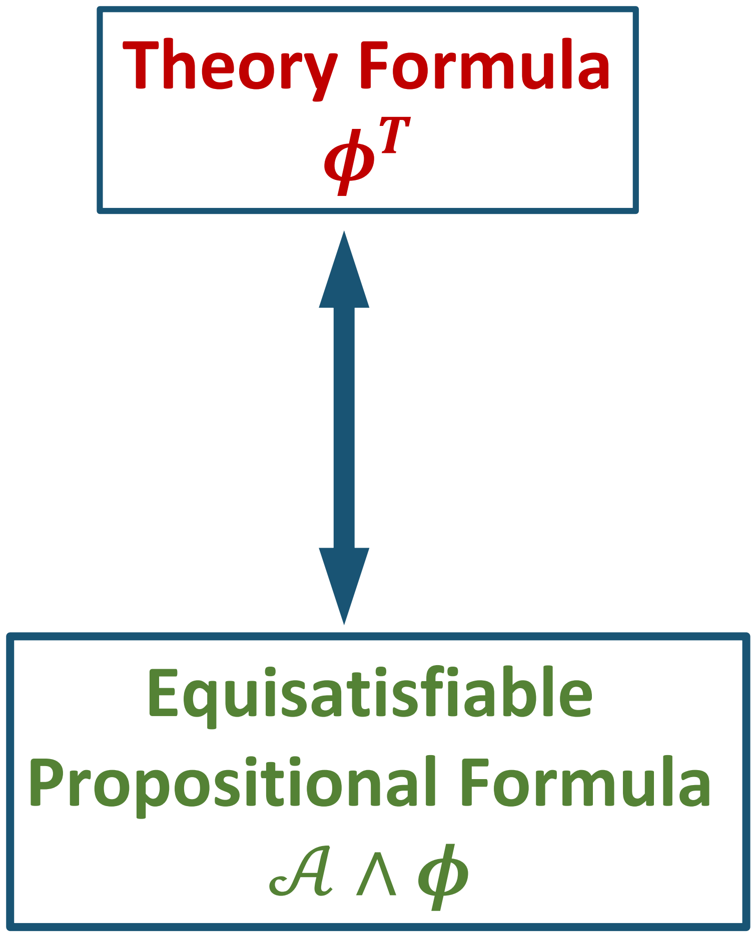
Recap: $\mathcal{T}$-Satisfiability, $\mathcal{T}$-validity, $\mathcal{T}$-Equivalence

- Only models satisfying axioms are relevant
- ➔ "Satisfiability ***modulo*** (='with respect to') theories"



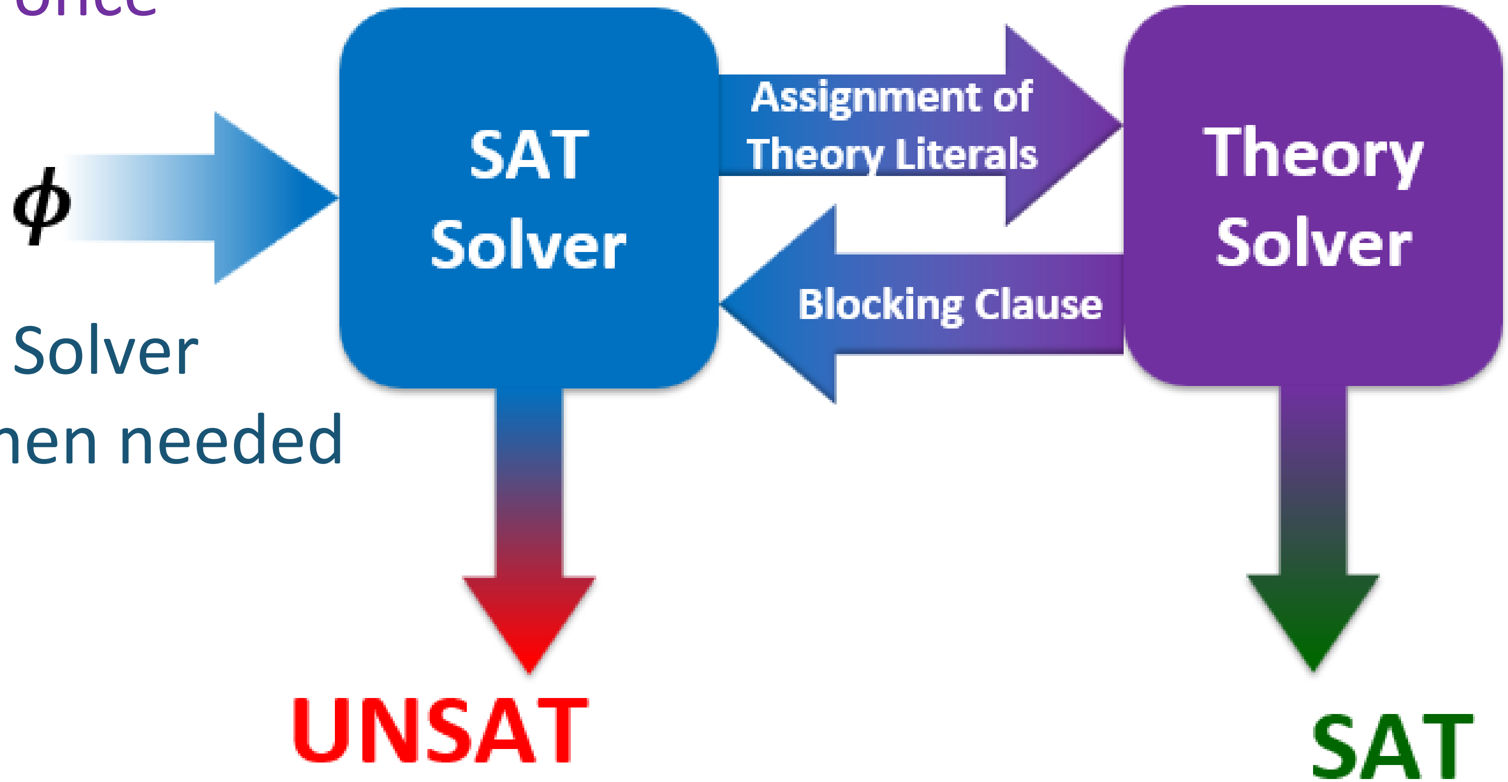
Recap - Implementations of SMT Solvers

- **Eager Encoding**
  - Equisatisfiable propositional formula
    - Adds all constraints that could be needed at once
  - SAT Solver

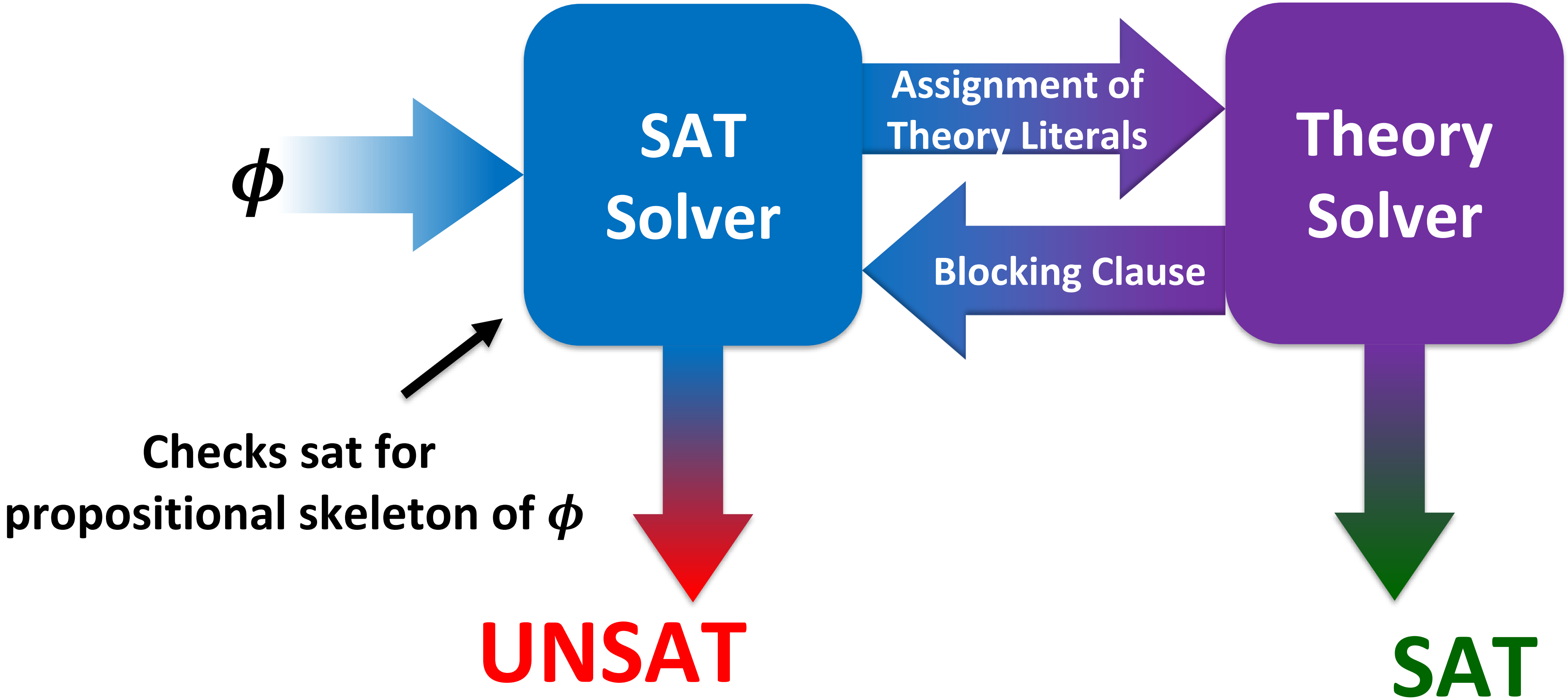


Recap - Implementations of SMT Solvers

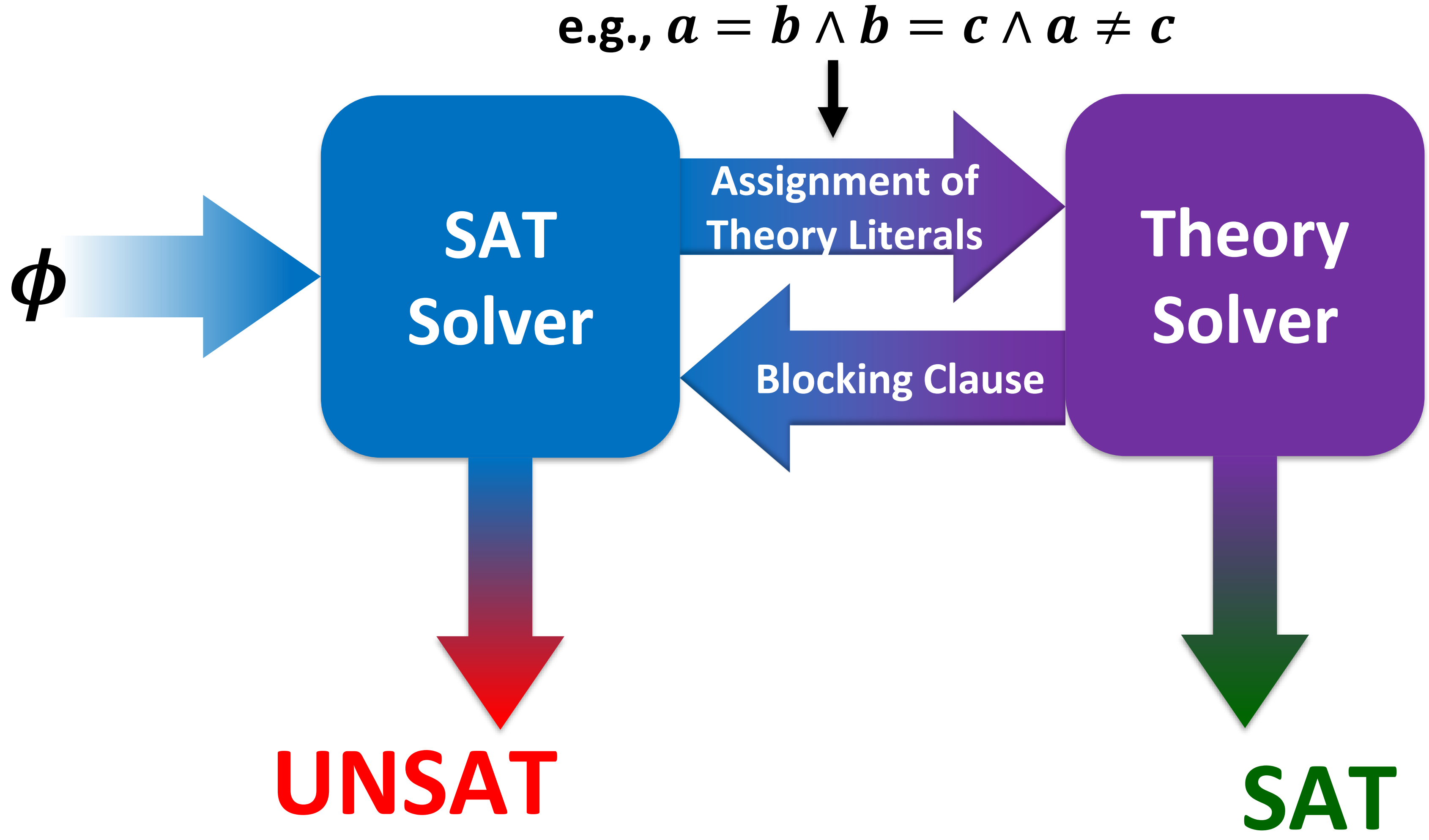
- **Eager Encoding**
  - Equisatisfiable propositional formula
    - Adds all constraints that could be needed at once
  - SAT Solver
- **Lazy Encoding**
  - SAT Solver and Theory Solver
  - Add constrains only when needed



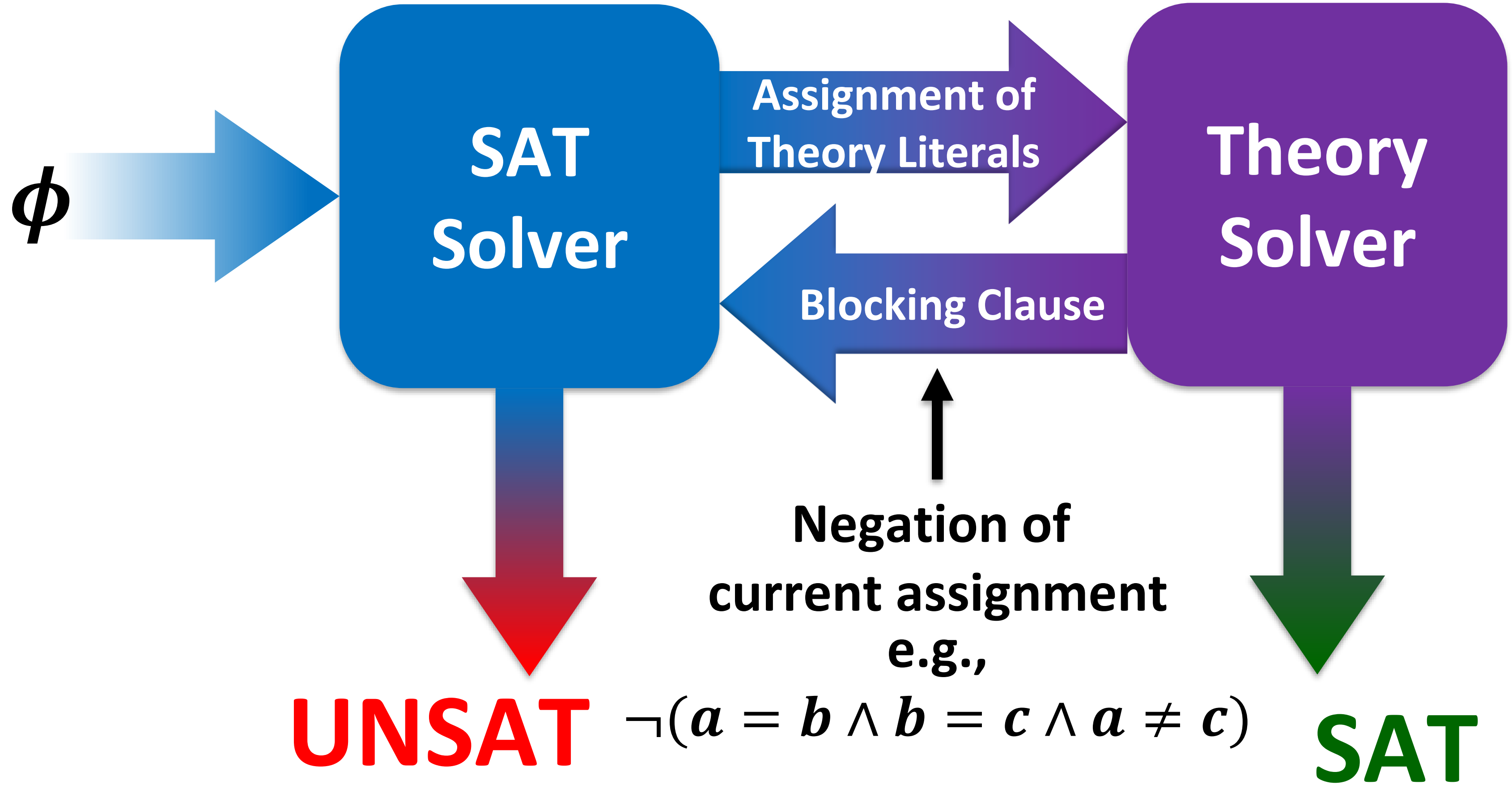
Recap - Lazy Encoding



Recap - Lazy Encoding



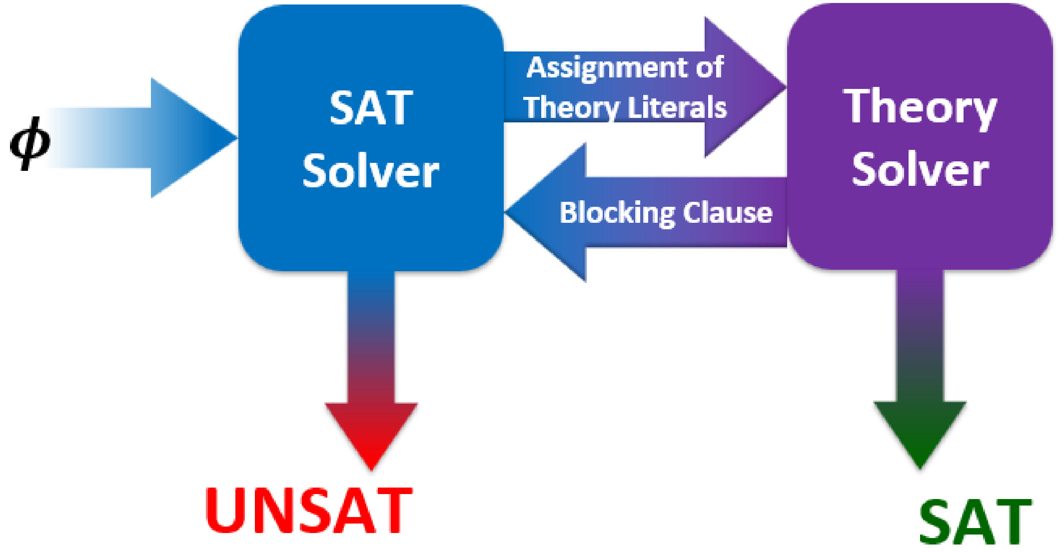
Recap - Lazy Encoding



Recap – Theory Solver for $\mathcal{T}_{\mathrm{UF}E}$

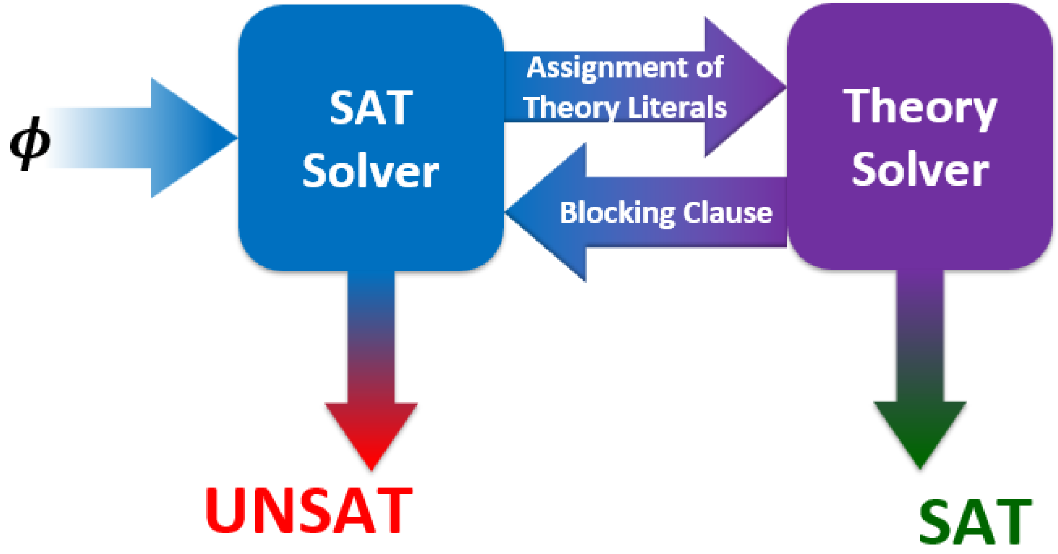
Congruence Closure Algorithm

- Takes conjunctions of theory literals as input
  - Equalities (e.g., $\mathrm{f}(\mathrm{g}(\mathrm{a})) = \mathrm{g}(\mathrm{b})$)
  - Disequalities (e.g., $\mathrm{a} \neq f(b)$)
- Checks whether assignment to literals is consistent with theory
  - e.g., $a = b, b = c, c \neq a$ is $\mathcal{T}_{UFE}$ unsat



Plan for Today

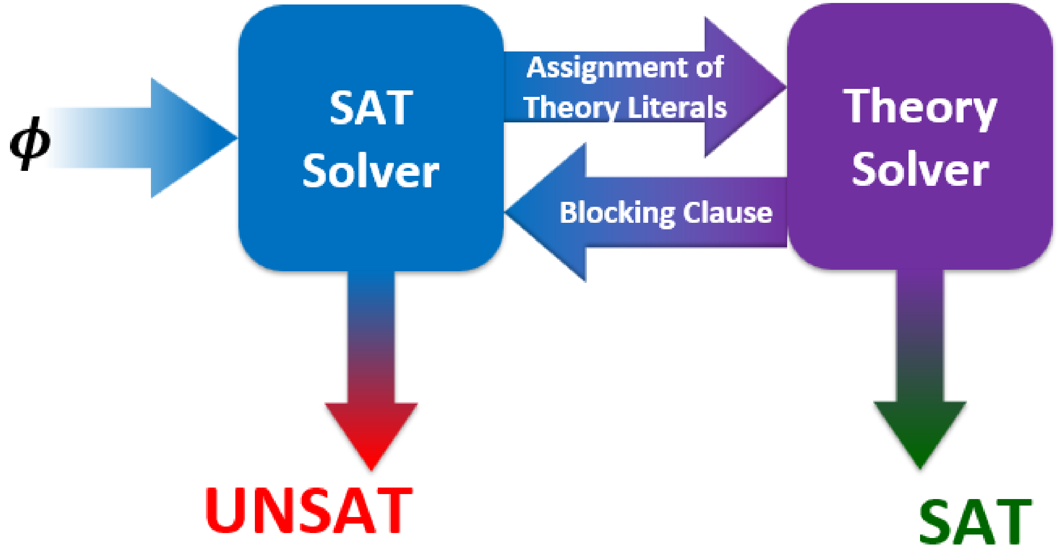
- We did not do an example for lazy encoding yet
  - → Plan for today: Examples ☺



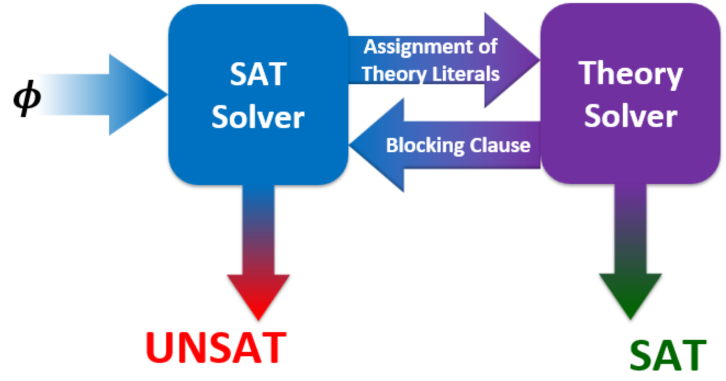
Plan for Today

- We did not do an example for lazy encoding yet
  - → Plan for today: Examples ☺

- **Deciding Satisfiability of Formulas in $\mathcal{T}_{UFE}$ using (a simplified version of) DPLL(T)**
  - Execute **DPLL with theory literals**
  - Use **Congrence Closure** to check assignment of theory literals



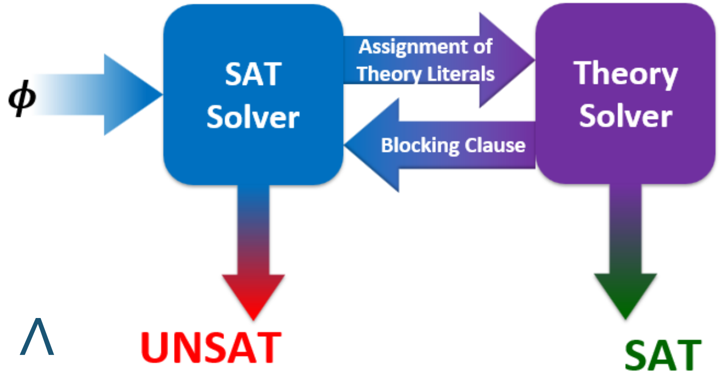
Example



Use the simple version of DPLL(T) to find satisfying assignment for φ within $\mathcal{T}_{UFE}$ (if one exists).

$$\varphi = ((f(g(a)) = b) \vee (f(b) = a)) \wedge ((f(g(a)) \neq b) \vee (f(b) = c)) \wedge$$
$$((f(g(a)) = b) \vee (f(a) \neq b)) \wedge ((f(b) \neq a) \vee (f(b) = c)) \wedge$$
$$((f(b) = c) \vee (f(a) = b)) \wedge ((f(b) \neq c) \vee (f(c) \neq a)) \wedge ((f(a) \neq b) \vee (f(c) \neq a))$$

Example



$$\varphi = ((f(g(a)) = b) \lor (f(b) = a)) \land ((f(g(a)) \neq b) \lor (f(b) = c)) \land$$
$$((f(g(a)) = b) \lor (f(a) \neq b)) \land ((f(b) \neq a) \lor (f(b) = c)) \land$$
$$((f(b) = c) \lor (f(a) = b)) \land ((f(b) \neq c) \lor (f(c) \neq a)) \land ((f(a) \neq b) \lor (f(c) \neq a))$$

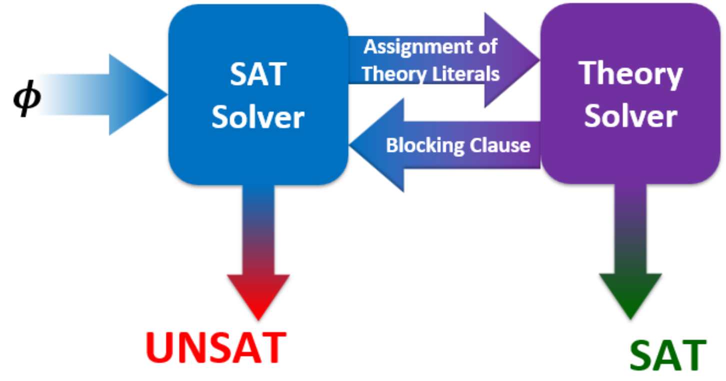
- Step 1: Assign propositional variables to theory literals

$$e_0 \Leftrightarrow (f(g(a)) = b) \qquad e_3 \Leftrightarrow (f(a) = b)$$
$$e_1 \Leftrightarrow (f(b) = a) \qquad e_4 \Leftrightarrow (f(c) = a)$$
$$e_2 \Leftrightarrow (f(b) = c)$$

- Step 2: Compute propositional skeleton $\hat{\varphi}$

$$\hat{\varphi} = (e_0 \lor e_1) \land (\neg e_0 \lor e_2) \land (e_0 \lor \neg e_3) \land (\neg e_1 \lor e_2) \land (e_2 \lor e_3) \land (\neg e_2 \lor e_4) \land (\neg e_3 \lor \neg e_4)$$

Example



$$\hat{\varphi} = (e_0 \vee e_1) \wedge (\neg e_0 \vee e_2) \wedge (e_0 \vee \neg e_3) \wedge (\neg e_1 \vee e_2) \wedge (e_2 \vee e_3) \wedge (\neg e_2 \vee e_4) \wedge (\neg e_3 \vee \neg e_4)$$

- Step 3: Use SAT Solver to find satisfying Model for $\hat{\varphi}$ (if one exists)

$\hat{\varphi} = (e_0 \lor e_1) \land (\neg e_0 \lor e_2) \land (e_0 \lor \neg e_3) \land (\neg e_1 \lor e_2) \land (e_2 \lor e_3) \land (\neg e_2 \lor e_4) \land (\neg e_3 \lor \neg e_4)$

*Decision heuristic: alphabetical order starting with the **negative** phase*

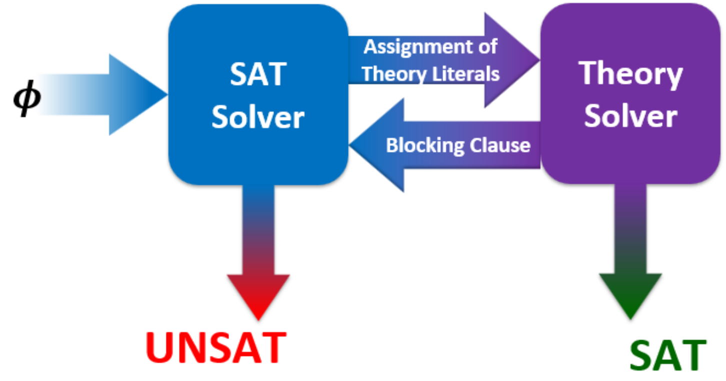
Step	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Dec. Level							
Assignment							
1: $\{e_0, e_1\}$							
2: $\{\neg e_0, e_2\}$							
3: $\{e_0, \neg e_3\}$							
4: $\{\neg e_1, e_2\}$							
5: $\{e_2, e_3\}$							
6: $\{\neg e_2, e_4\}$							
7: $\{\neg e_3, \neg e_4\}$							
LC 1							
LC 2							
BCP							
Pure Literal							
Decision							

$\hat{\varphi} = (e_0 \vee e_1) \wedge (\neg e_0 \vee e_2) \wedge (e_0 \vee \neg e_3) \wedge (\neg e_1 \vee e_2) \wedge (e_2 \vee e_3) \wedge (\neg e_2 \vee e_4) \wedge (\neg e_3 \vee \neg e_4)$

*Decision heuristic: alphabetical order starting with the **negative** phase*

Step	1	2	3	4	5	6
Decision Level	0	1	1	1	1	1
Assignment	-	$\neg e_0$	$\neg e_0, e_1$	$\neg e_0, e_1, e_2$	$\neg e_0, e_1, e_2, \neg e_3$	$\neg e_0, e_1, e_2, \neg e_3, e_4$
Cl. 1: e_0, e_1	e_0, e_1	e_1	✓	✓	✓	✓
Cl. 2: $\neg e_0, e_2$	$\neg e_0, e_2$	✓	✓	✓	✓	✓
Cl. 3: $e_0, \neg e_3$	$e_0, \neg e_3$	$\neg e_3$	$\neg e_3$	$\neg e_3$	✓	✓
Cl. 4: $\neg e_1, e_2$	$\neg e_1, e_2$	$\neg e_1, e_2$	e_2	✓	✓	✓
Cl. 5: e_2, e_3	e_2, e_3	e_2, e_3	e_2, e_3	✓	✓	✓
Cl. 6: $\neg e_2, e_4$	$\neg e_2, e_4$	$\neg e_2, e_4$	$\neg e_2, e_4$	e_4	e_4	✓
Cl. 7: $\neg e_3, \neg e_4$	$\neg e_3, \neg e_4$	$\neg e_3, \neg e_4$	$\neg e_3, \neg e_4$	$\neg e_3, \neg e_4$	✓	✓
BCP	-	e_1	e_2	$\neg e_3$	e_4	-
PL	-	-	-	-	-	-
Decision	$\neg e_0$	-	-	-	-	SAT

Example



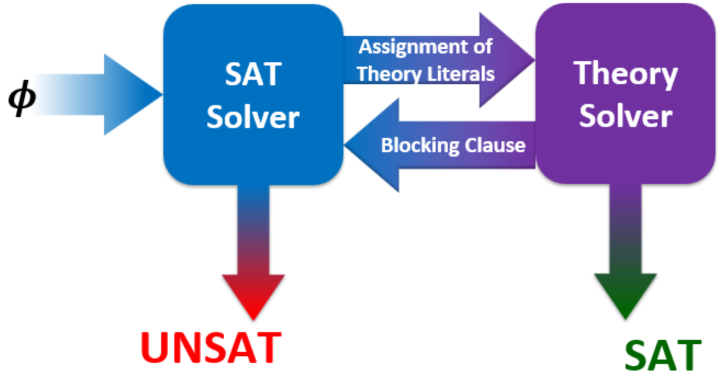
- DPLL returned satisfying assignment from SAT Solver
  - $M_{prop} = \{e_0 = F, e_1 = T, e_2 = T, e_3 = F,\ e_4 = T\}$
  - $M_{prop} \vDash \hat{\varphi}$

- Step 4: Check if assignment of theory literals is consistent with theory
  - Translate back to theory literals using

$$e_0 \Leftrightarrow (f(g(a)) = b) \qquad e_3 \Leftrightarrow (f(a) = b)$$
$$e_1 \Leftrightarrow (f(b) = a) \qquad e_4 \Leftrightarrow (f(c) = a)$$
$$e_2 \Leftrightarrow (f(b) = c)$$

  - $M_{\mathcal{T}_{UFE}} := \{(f(g(a)) \neq b), (f(b) = a), (f(b) = c), (f(a) \neq b), (f(c) = a)\}$

Example



- Execute Congruence Closure Algorithm
  - $M_{\mathcal{T}_{UFE}} := \{(f(g(a)) \neq b), (f(b) = a), (f(b) = c), (f(a) \neq b), (f(c) = a)\}$

$\{f(b), a\}, \{f(b), c\}, \{f(c), a\}, \{f(g(a))\}, \{b\}, \{f(a)\}$

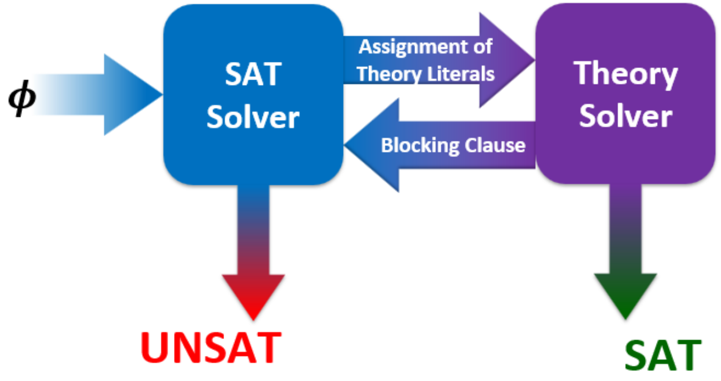
$\{a, c, f(b)\}, \{f(c), a\}, \{f(g(a))\}, \{b\}, \{f(a)\}$

$\{a, c, f(b), f(c)\}, \{f(g(a))\}, \{b\}, \{f(a)\}$

$\{a, c, f(a)\ f(b), f(c)\}, \{f(g(a))\}, \{b\}$

- $\mathcal{T}_{\text{UFE}}$-Satisfiable since $f(g(a))$ and b as well as $f(a)$ and b are in different equivalence classes.
- **→ $M_{\mathcal{T}_{\text{UFE}}}$ is a satisfying assignment for φ. Algorithm terminates with SAT.**

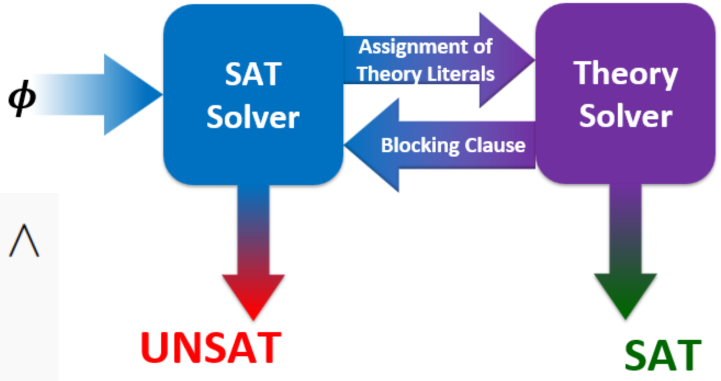
Example 2



Use the simple version of DPLL(T) to find satisfying assignment for φ within $\mathcal{T}_{UFE}$ (if one exists).

$$\begin{aligned}\varphi = &((f(a) = b) \lor (f(a) = c) \lor \neg(b = c)) \land ((b = c) \lor (a = b) \lor (f(a) = b)) \land \\ &(\neg(f(a) = b) \lor (a = b)) \land ((b = c) \lor \neg(a = b) \lor \neg(f(a) = b)) \land \\ &(\neg(f(a) = c) \lor (b = c)) \land (\neg(f(a) = c) \lor (b = c) \lor \neg(a = b)) \land \\ &((f(a) = b) \lor (f(a) = c))\end{aligned}$$

Example 2

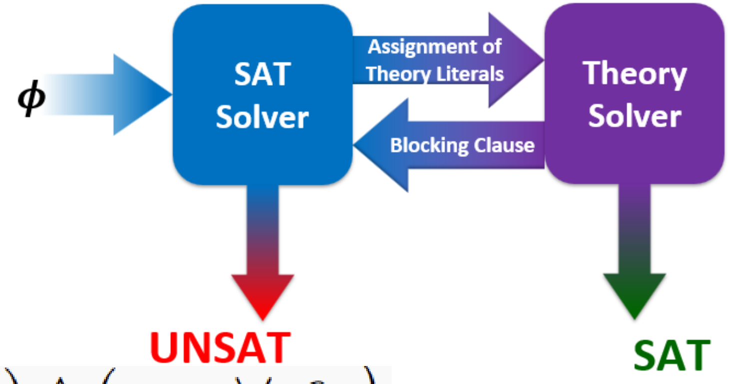


$$\varphi = ((f(a) = b) \vee (f(a) = c) \vee \neg(b = c)) \wedge ((b = c) \vee (a = b) \vee (f(a) = b)) \wedge$$
$$(\neg(f(a) = b) \vee (a = b)) \wedge ((b = c) \vee \neg(a = b) \vee \neg(f(a) = b)) \wedge$$
$$(\neg(f(a) = c) \vee (b = c)) \wedge (\neg(f(a) = c) \vee (b = c) \vee \neg(a = b)) \wedge$$
$$((f(a) = b) \vee (f(a) = c))$$

- Step 1: Assign propositional variables to theory literals
  - $e_0 \Leftrightarrow (f(a) = b)$
  - $e_1 \Leftrightarrow (f(a) = c)$
  - $e_2 \Leftrightarrow (b = c)$
  - $e_3 \Leftrightarrow (a = b)$
- Step 2: Compute propositional skeleton $\hat{\varphi}$

$$\hat{\varphi} = (e_0 \vee e_1 \vee \neg e_2) \wedge (e_2 \vee e_3 \vee e_0) \wedge (\neg e_0 \vee e_3) \wedge (e_2 \vee \neg e_3 \vee \neg e_0) \wedge (\neg e_1 \vee e_2)$$
$$\wedge (\neg e_1 \vee e_2 \vee \neg e_3) \wedge (e_0 \vee e_1)$$

Example 2



$$\hat{\varphi} = (e_0 \vee e_1 \vee \neg e_2) \wedge (e_2 \vee e_3 \vee e_0) \wedge (\neg e_0 \vee e_3) \wedge (e_2 \vee \neg e_3 \vee \neg e_0) \wedge (\neg e_1 \vee e_2) \wedge (\neg e_1 \vee e_2 \vee \neg e_3) \wedge (e_0 \vee e_1)$$

- Step 3: Use SAT Solver to find satisfying Model for $\hat{\varphi}$ (if one exists)

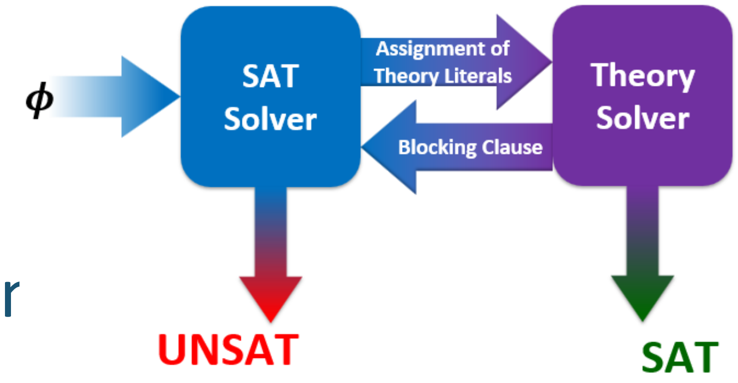
$$\hat{\varphi} = (e_0 \lor e_1 \lor \neg e_2) \land (\neg e_1 \lor e_2 \lor e_3) \land (e_2 \lor e_3 \lor e_0) \land$$
$$(\neg e_0 \lor e_3) \land (e_0 \lor e_1 \lor \neg e_3) \land (e_2 \lor \neg e_3 \lor \neg e_0)$$

*Decision heuristic: alphabetical order starting with the **negative** phase*

Step	1	2	3	4
Decision Level	0	1	1	1
Assignment	-	$\neg e_0$	$\neg e_0, e_1$	$\neg e_0, e_1, e_2$
Cl. 1: $e_0, e_1, \neg e_2$	$e_0, e_1, \neg e_2$	$e_1, \neg e_2$	✓	✓
Cl. 2: e_2, e_3, e_0	e_2, e_3, e_0	e_2, e_3	e_2, e_3	✓
Cl. 3: $\neg e_0, e_3$	$\neg e_0, e_3$	✓	✓	✓
Cl. 4: $e_2, \neg e_3, \neg e_0$	$e_2, \neg e_3, \neg e_0$	✓	✓	✓
Cl. 5: $\neg e_1, e_2$	$\neg e_1, e_2$	$\neg e_1, e_2$	e_2	✓
Cl. 6: $\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$e_2, \neg e_3$	✓
Cl. 7: e_0, e_1	e_0, e_1	e_1	✓	✓
BCP	-	e_1	e_2	-
PL	-	-	-	-
Decision	$\neg e_0$	-	-	-

$$\mathcal{M}_{\mathcal{T}_{EUF}} := \{(f(a) \neq b), (f(a) = c), (b = c)\}$$

Example 2

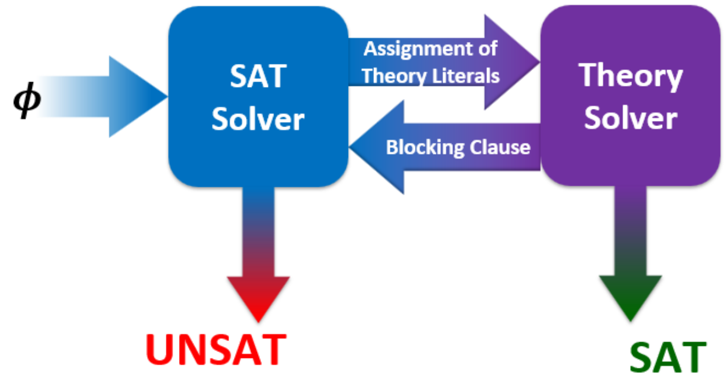


- Step 4: Check if assignment of theory literals is consister
  - Translate back to theory literals using

    - $e_0 \Leftrightarrow (f(a) = b)$
    - $e_1 \Leftrightarrow (f(a) = c)$
    - $e_2 \Leftrightarrow (b = c)$
    - $e_3 \Leftrightarrow (a = b)$

  - $\mathcal{M}_{\mathcal{T}_{EUF}} := \{(f(a) \neq b), (f(a) = c), (b = c)\}$

Example 2

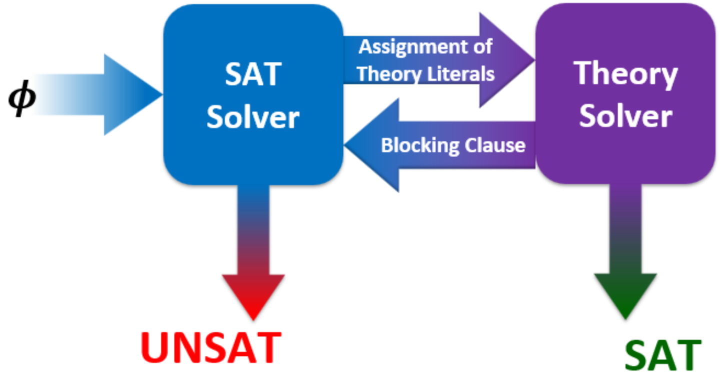


- Execute Congruence Closure Algorithm
  - $\mathcal{M}_{\mathcal{T}_{EUF}} := \{(f(a) \neq b), (f(a) = c), (b = c)\}$

$\{f(a), c\}, \{b, c\}$
$\{b, c, f(a)\}$

$\mathcal{M}_{\mathcal{T}_{EUF}}$ is not consistent with the theory, because of: $(f(a) \neq b)$
$\Rightarrow$ We need to add a blocking clause from $\mathcal{M}_{\mathcal{T}_{EUF}}$:
$BC_8 := e_0 \vee \neg e_1 \vee \neg e_2$

Example 2

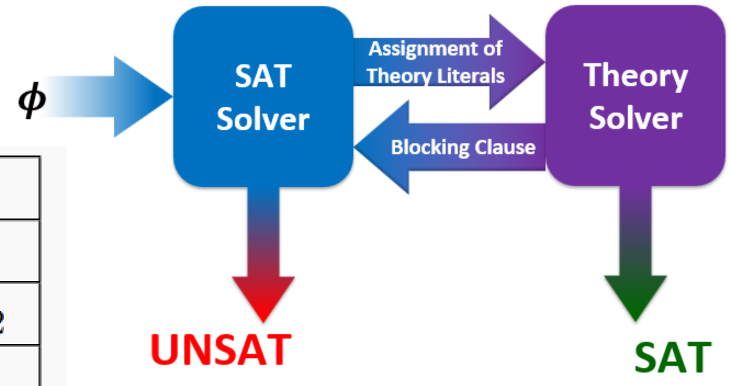


- Execute Congruence Closure Algorithm
  - $\mathcal{M}_{\mathcal{T}_{EUF}} := \{(f(a) \neq b), (f(a) = c), (b = c)\}$

$\{f(a), c\}, \{b, c\}$
$\{b, c, f(a)\}$

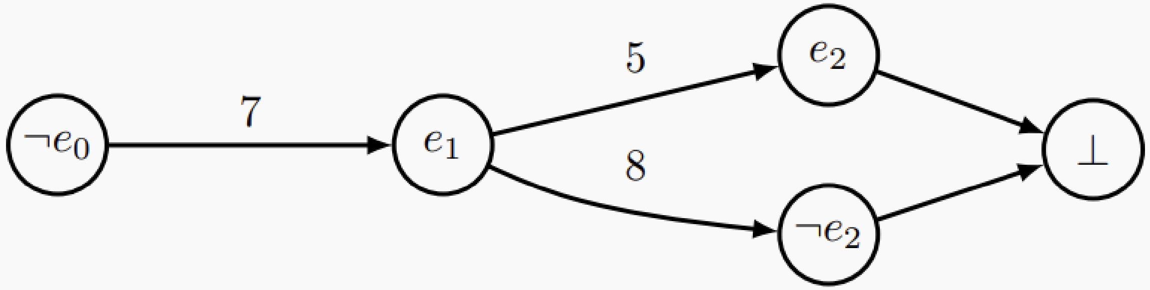
$\mathcal{M}_{\mathcal{T}_{EUF}}$ is not consistent with the theory, because of: $(f(a) \neq b)$
$\Rightarrow$ We need to add a blocking clause from $\mathcal{M}_{\mathcal{T}_{EUF}}$:
$BC_8 := e_0 \vee \neg e_1 \vee \neg e_2$

Example 2



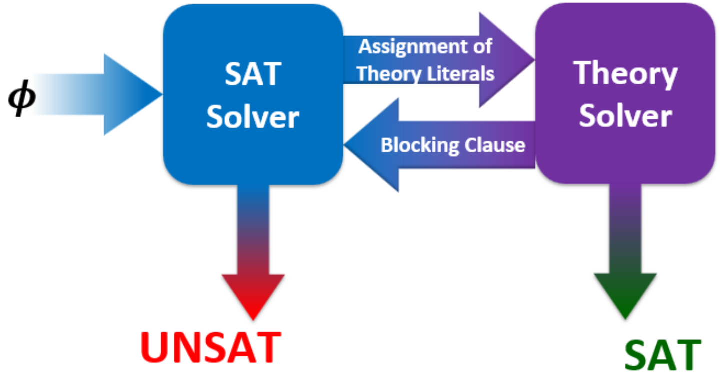
Step	5	6	7	8
Decision Level	0	1	1	1
Assignment	-	$\neg e_0$	$\neg e_0, e_1$	$\neg e_0, e_1, \neg e_2$
Cl. 1: $e_0, e_1, \neg e_2$	$e_0, e_1, \neg e_2$	$e_1, \neg e_2$	✓	✓
Cl. 2: e_2, e_3, e_0	e_2, e_3, e_0	e_2, e_3	e_2, e_3	e_3
Cl. 3: $\neg e_0, e_3$	$\neg e_0, e_3$	✓	✓	✓
Cl. 4: $e_2, \neg e_3, \neg e_0$	$e_2, \neg e_3, \neg e_0$	✓	✓	✓
Cl. 5: $\neg e_1, e_2$	$\neg e_1, e_2$	$\neg e_1, e_2$	e_2	{} ✗
Cl. 6: $\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$e_2, \neg e_3$	$\neg e_3$
Cl. 7: e_0, e_1	e_0, e_1	e_1	✓	✓
Blocking Cl. 8: $e_0, \neg e_1, \neg e_2$	$e_0, \neg e_1, \neg e_2$	$\neg e_1, \neg e_2$	$\neg e_2$	✓
BCP	-	e_1	$\neg e_2$	-
PL	-	-	-	-
Decision	$\neg e_0$	-	-	-

Conflict in step 8



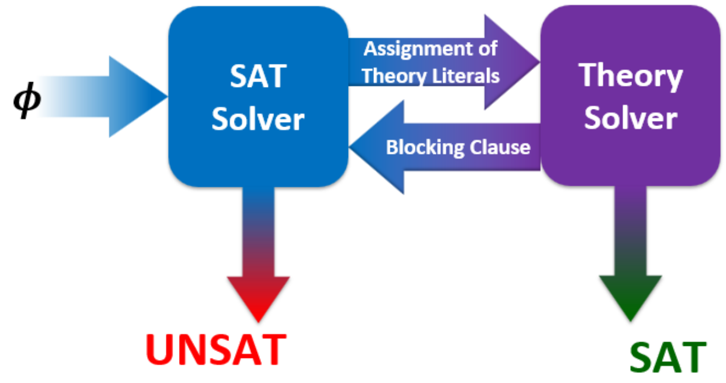
$$\dfrac{\dfrac{5.\ \neg e_1 \vee e_2 \qquad 8.\ e_0 \vee \neg e_1 \vee \neg e_2}{\neg e_1 \vee e_0} \qquad 7.\ e_0 \vee e_1}{e_0}$$

Example 2



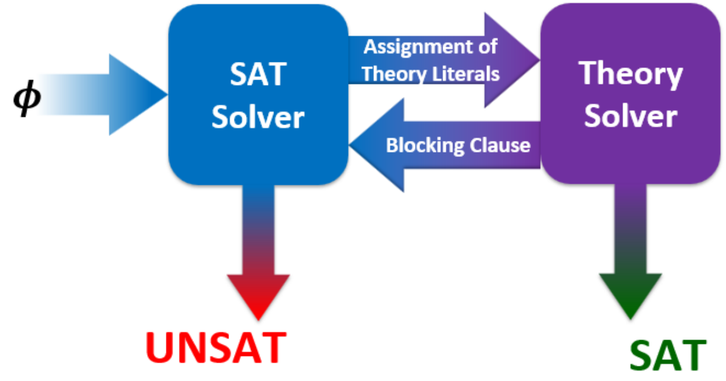
Step	9	10	11	12
Decision Level	0	0	0	0
Assignment	-	e_0	e_0, e_3	e_0, e_3, e_2
Cl. 1: $e_0, e_1, \neg e_2$	$e_0, e_1, \neg e_2$	✓	✓	✓
Cl. 2: e_2, e_3, e_0	e_2, e_3, e_0	✓	✓	✓
Cl. 3: $\neg e_0, e_3$	$\neg e_0, e_3$	e_3	✓	✓
Cl. 4: $e_2, \neg e_3, \neg e_0$	$e_2, \neg e_3, \neg e_0$	$e_2, \neg e_3$	e_2	✓
Cl. 5: $\neg e_1, e_2$	$\neg e_1, e_2$	$\neg e_1, e_2$	$\neg e_1, e_2$	✓
Cl. 6: $\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2$	✓
Cl. 7: e_0, e_1	e_0, e_1	✓	✓	✓
Cl. 8: $e_0, \neg e_1, \neg e_2$	$e_0, \neg e_1, \neg e_2$	✓	✓	✓
Cl. 9: e_0	e_0	✓	✓	✓
BCP	e_0	e_3	e_2	-
PL	-	-	-	-
Decision	-	-	-	SAT

Example 2



Step	9	10	11	12
Decision Level	0	0	0	0
Assignment	-	e_0	e_0, e_3	e_0, e_3, e_2
Cl. 1: $e_0, e_1, \neg e_2$	$e_0, e_1, \neg e_2$	✓	✓	✓
Cl. 2: e_2, e_3, e_0	e_2, e_3, e_0	✓	✓	✓
Cl. 3: $\neg e_0, e_3$	$\neg e_0, e_3$	e_3	✓	✓
Cl. 4: $e_2, \neg e_3, \neg e_0$	$e_2, \neg e_3, \neg e_0$	$e_2, \neg e_3$	e_2	✓
Cl. 5: $\neg e_1, e_2$	$\neg e_1, e_2$	$\neg e_1, e_2$	$\neg e_1, e_2$	✓
Cl. 6: $\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2, \neg e_3$	$\neg e_1, e_2$	✓
Cl. 7: e_0, e_1	e_0, e_1	✓	✓	✓
Cl. 8: $e_0, \neg e_1, \neg e_2$	$e_0, \neg e_1, \neg e_2$	✓	✓	✓
Cl. 9: e_0	e_0	✓	✓	✓
BCP	e_0	e_3	e_2	-
PL	-	-	-	-
Decision	-	-	-	SAT

Example 2



- Execute Congruence Closure Algorithm

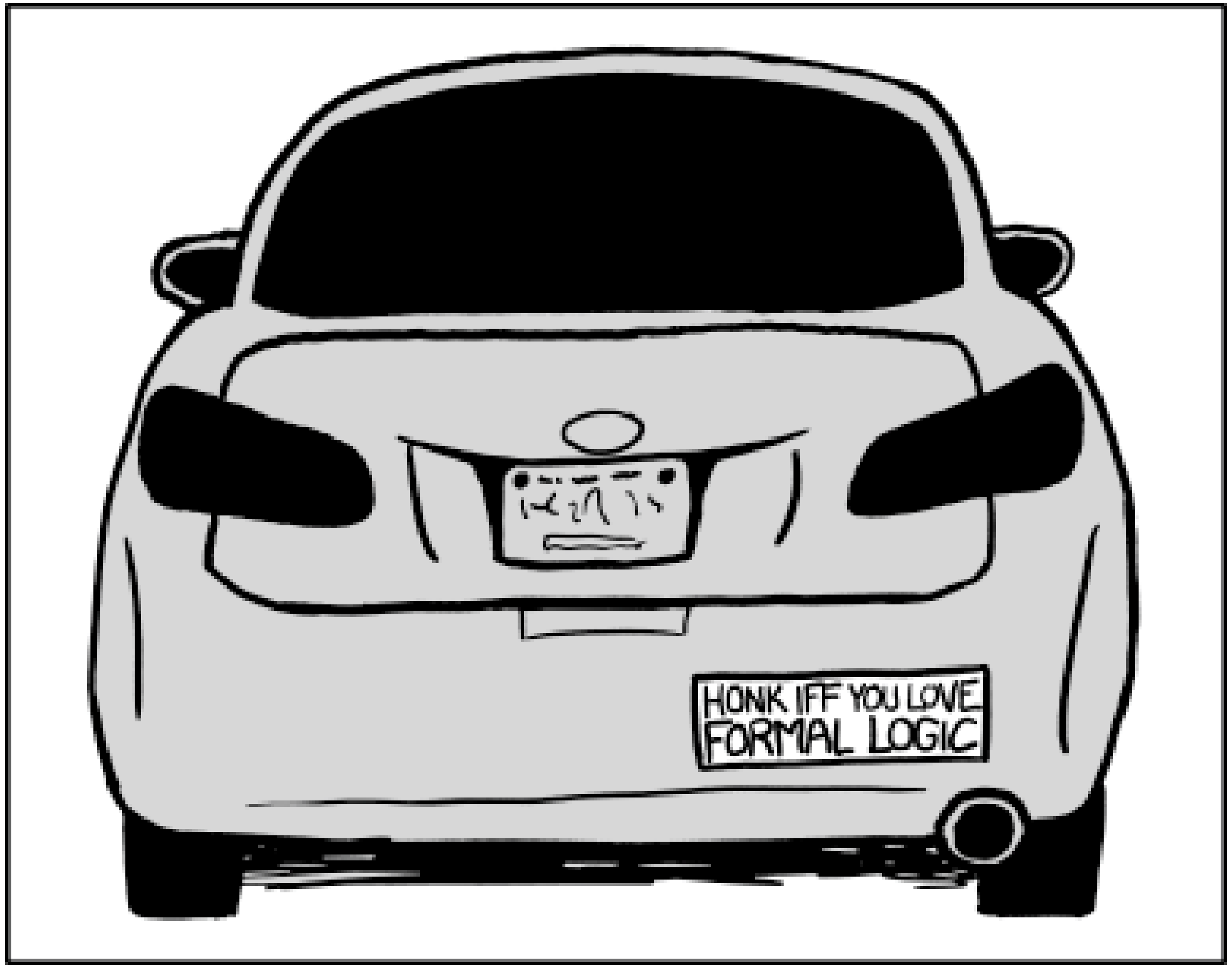
$\mathcal{M}_{\mathcal{T}_{EUF}} := (f(a) = b) \land (b = c) \land (a = b)$

Check if the assignment is consistent with the theory:

$$\{f(a), b\}, \{b, c\}, \{a, b\}$$
$$\{a, b, c, f(a)\}$$

- $\mathcal{T}_{\text{UFE}}$-Satisfiable since there are no disequalities that could be violated.
- → $M_{\mathcal{T}_{\text{UFE}}}$ **is a satisfying assignment for φ. Algorithm terminates with SAT.**

Thank You



https://xkcd.com/1033/