Tutorial 2

大纲

- 传递闭包
- 作业
- Lab
- SMT中的量词处理方法（不考）

作业

下列正确的选项是

- $\{a, b\} \subseteq \{a, \{b\}, \{a, b\}\}$
- 其它选项都不能选
- $\{a, b\} \notin \{a, b, \{b\}\}$
- $\{a, b\} \nsubseteq \{a, b, \{b\}\}$

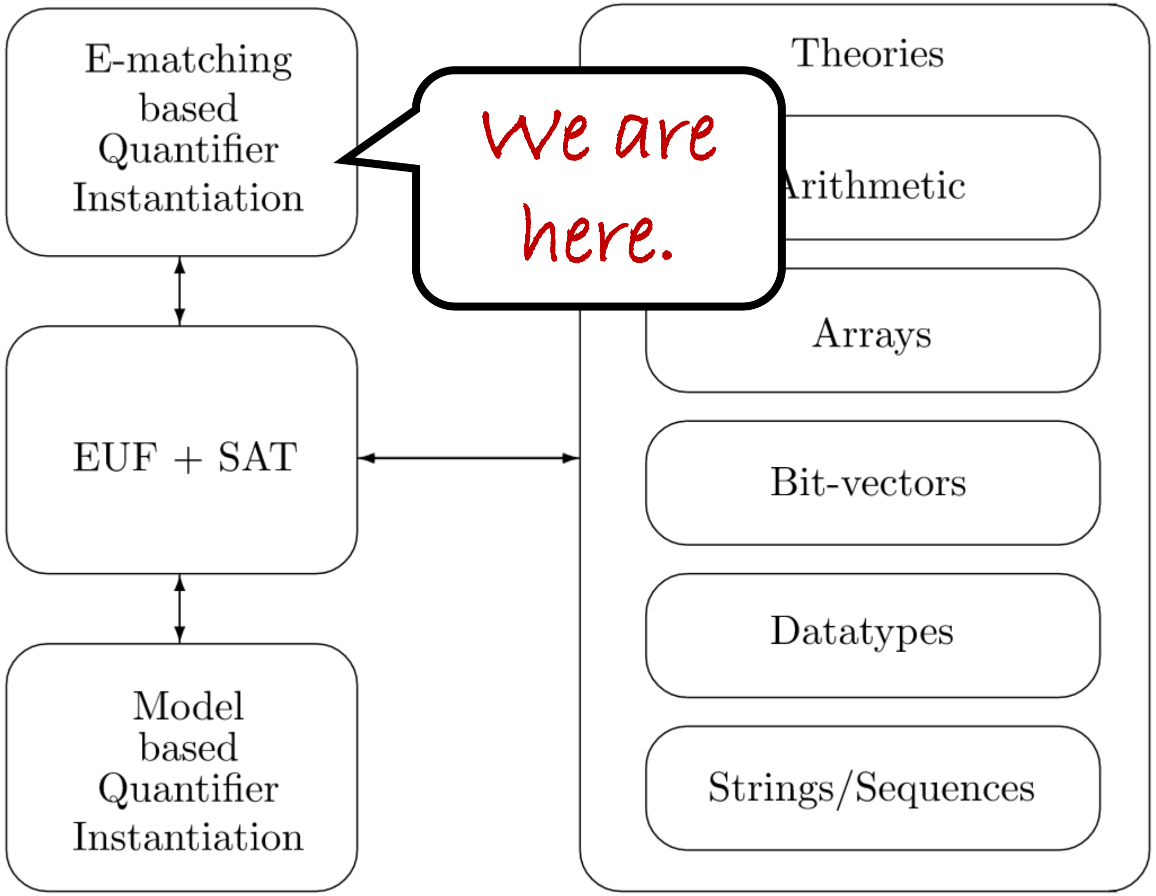
作业

以下叙述错误的是：

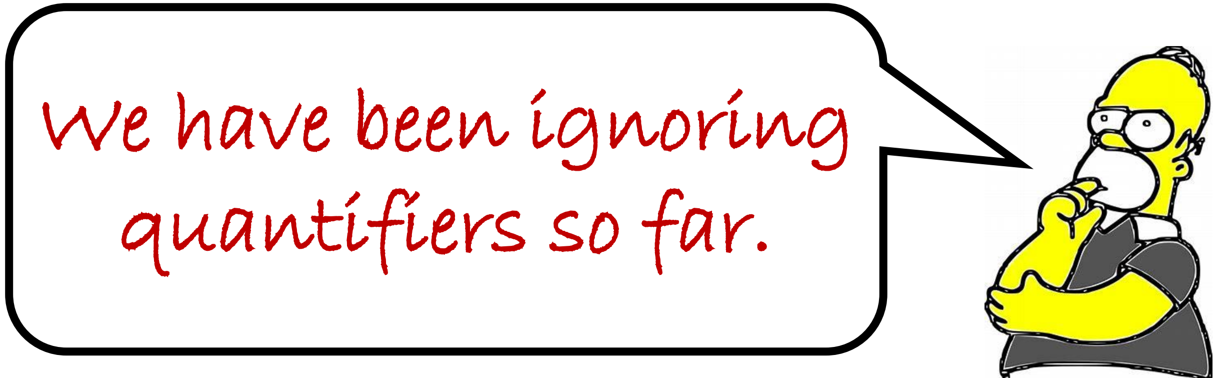
- 偏序关系是有自反性、反对称性、传递性的关系
- 对于非空有限集合，极小元（极大元）不一定唯一
- 对于非空有限集合，最小元（最大元）不一定存在
- **拟序关系有非自反性和传递性，但不一定有反对称性**

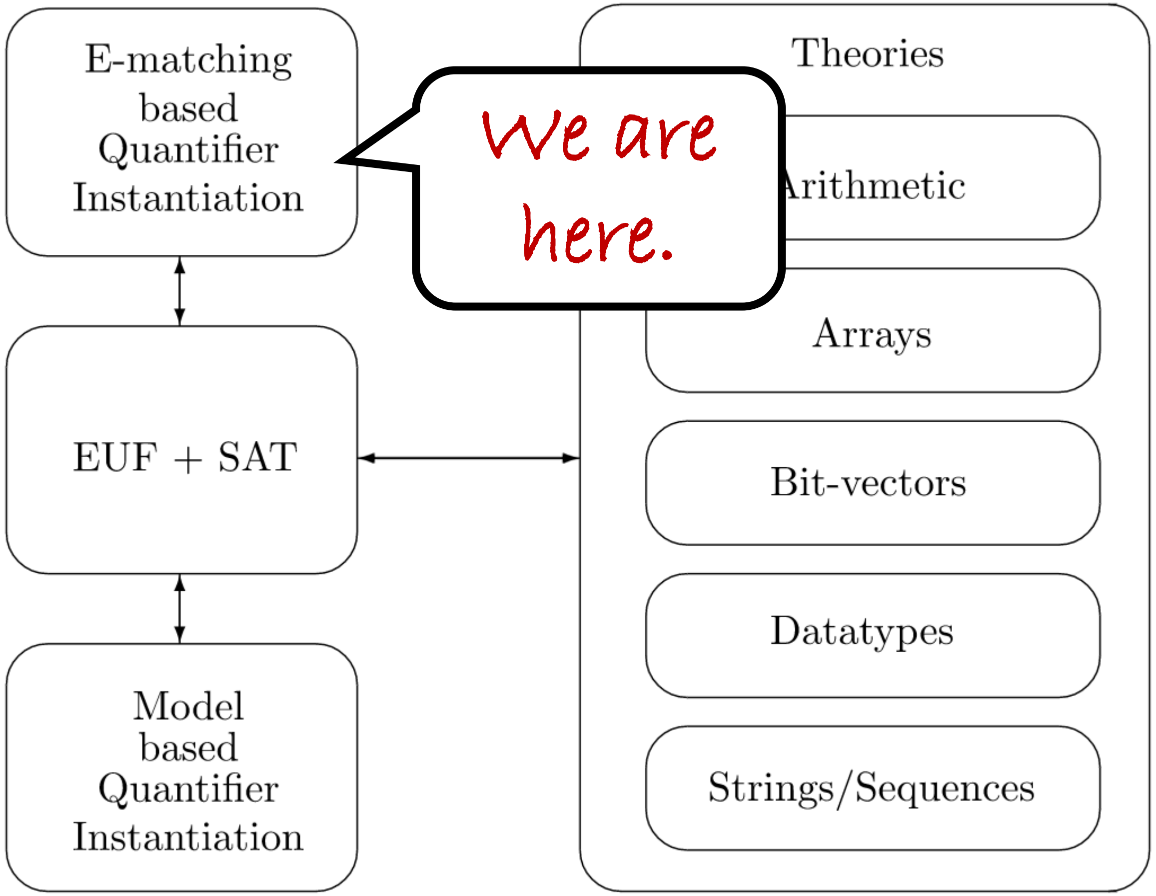
Lab

- Lab1如果有问题，期末会统一安排时间处理
- Lab2常见问题已经整理到谓词逻辑讨论区
- Lab3据大家反馈，时长为2小时左右

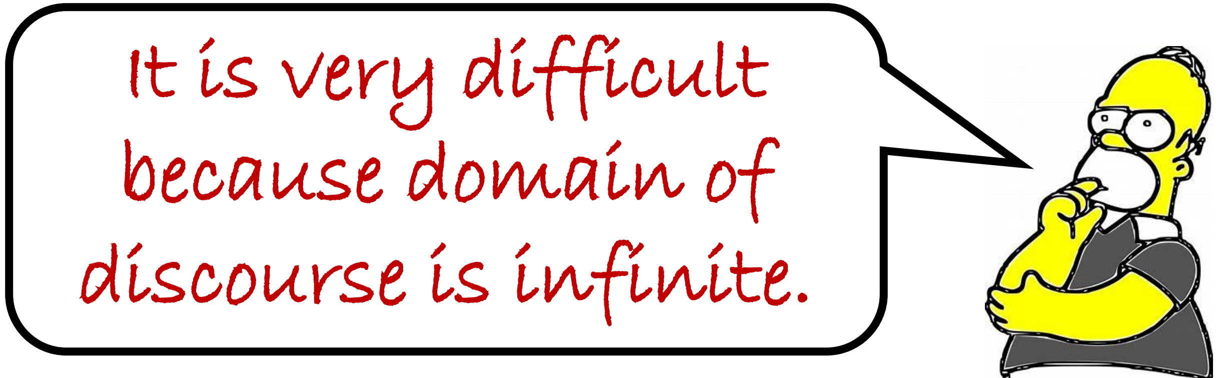


Architecture of Z3's SMT Core solver



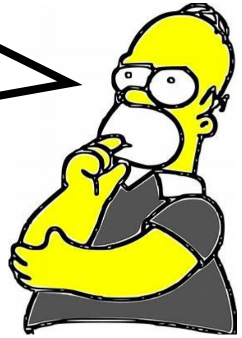


Architecture of Z3's SMT Core solver



Quantifier

To make things easier, we can eliminate all existential quantifiers.



Quantifier

$$\neg(\forall x)(P(x)) = (\exists x)(\neg P(x))$$
$$\neg(\exists x)(P(x)) = (\forall x)(\neg P(x))$$

1) Move “not” inwards.
2) Use skolemization to eliminate existential quantifiers.
3) Now we obtain a formula with only positive universally-quantified literals.

There is no “not” before quantifiers.

A Quick Recap

$$(\forall x)(\exists y)(\exists z)S(a,b,x,y,z)$$

$$\Downarrow$$

y depends on x

$$(\forall x)(\exists z)S(a,b,x,f(x),z)$$

$$\Downarrow$$

z depends on x

$$(\forall x)S(a,b,x,f(x),g(x))$$

We don't need to move all the quantifiers to the front of all formulas. We directly do skolemization on quantifiers.

Quantifier

Now there are only universal quantifiers.

Intuitive approach: replace x with a

$$g(f(a)) = 0 \wedge (\forall x)(g(f(x)) = 1)$$

Quantifier

Now there are only universal quantifiers.

Intuitive approach: replace x with a

$$g(f(a)) = 0 \wedge \boxed{(\forall x)(g(f(x)) = 1)}$$

$$g(f(a)) = 1$$

Quantifier

Now there are only universal quantifiers.

Intuitive approach: replace x with a

$$g(f(a)) = 0 \wedge \boxed{(\forall x)(g(f(x)) = 1)}$$

$$g(f(a)) = 1$$

$$\boxed{\text{Theory Solvers}}$$

UNSAT

Quantifier

Now there are only universal quantifiers.

Intuitive approach: replace x with a.

$$g(f(a)) = 0 \land \boxed{(\forall x)(g(f(x)) = 1)}$$

$$g(f(a)) = 1$$

Theory Solvers

How to combine instantiation with SMT?

The remaining problem is how to generate instances.

Trigger Matching

DEFINITION

A trigger is a term t, satisfying the following criteria:

1) t must contain all of the variables quantified by the quantifier
2) t must contain at least one non-constant function symbol

$$(\forall x.\{f(x)\})(f(x) > 0)$$

trigger

$$(\forall x.\{g(f(x))\})(g(f(x)) = 1)$$

trigger

Trigger Matching

DEFINITION

A ground term is a term containing no quantified variables.

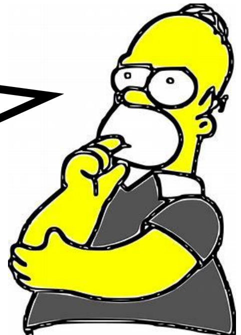
$$(\forall x.\{f(x)\})(f(x) > 0)$$

Not ground term

$$f(3)$$

ground term

We can compare the form of triggers and ground terms to generate instances



Trigger Matching

DEFINITION

A ground term is a term containing no quantified variables.

$$(\forall x.\{f(x)\})(f(x) > 0)$$

Not ground term

$$f(3)$$

ground term

$$\boxed{g(f(a))} = 0 \wedge (\forall x.\{\boxed{g(f(x))}\})(g(f(x)) = 1)$$

$$\Downarrow$$

$$g(f(a)) = 1$$

Term g(f(a))
matches the trigger,
causing the instantiation.

Trigger Matching

Without a matching ground term, no information will be deduced from the quantifier body.

$$(\forall x.p(x))(p(x) = 1) \wedge (\forall x.p(x))(p(x) = 0)$$

With trigger matching, solver returns unknown rather than sat.