

# SMT решавачи

Милан Банковић  
milan@matf.bg.ac.rs

Математички факултет

Аутоматско резоновање, 6. 6. 2017.

# Увод

## Шта до сада знамо?

- Логика првог реда је неодлучива
- Постоје процедуре полуодлучивања (метод резолуције)
- Постоје теорије првог реда које јесу одлучиве (нпр. реална аритметика)
- На језику таквих теорија могуће је изразити многе проблеме

## Циљ:

- Имплементирати процедуре одлучивања за овакве теорије
- Развити ефикасне решаваче
- Примене у разним областима

# Вишесортна логика првог реда са једнакошћу

## Сигнатура $\Sigma = (\mathcal{S}, \mathcal{F}, r)$

- Скуп сорти  $\mathcal{S}$  (међу којима је и Bool)
- Скуп функцијских симбола  $\mathcal{F}$
- Функција ранга:  $r : \mathcal{F} \rightarrow \mathcal{S}^* \times \mathcal{S}$
- $r(f) = [s_1, \dots, s_n] \rightarrow s$  ( $s_1, \dots, s_n$  су сорте аргумената, а  $s$  је повратна сорта)
- $r(a) = [] \rightarrow s$  ( $a$  је симбол константе сорте  $s$ )
- За сваку сорту  $s \in \mathcal{S}$ , скуп променљивих  $V^s$

## Изрази

- Сваки израз е има своју сорту  $s$
- Симбол константе  $a$  ранга  $[] \rightarrow s$  је израз сорте  $s$
- Променљива  $x \in V^s$  је израз сорте  $s$
- Ако је  $t_i$  израз сорте  $s_i$  ( $i = 1, \dots, n$ ) и  $r(f) = [s_1, \dots, s_n] \rightarrow s$ , тада је  $f(t_1, \dots, t_n)$  израз сорте  $s$
- Изрази сорте Bool зовемо формулама, а остале изразе термовима
- Исказни симболи:  $r(\perp) = r(\top) = [] \rightarrow \text{Bool}$ ,  $r(\neg) = [\text{Bool}] \rightarrow \text{Bool}$ ,  $r(\wedge) = [\text{Bool}, \text{Bool}] \rightarrow \text{Bool}$ , ...
- Симбол једнакости:  $r(=) = [s, s] \rightarrow \text{Bool}$  за сваку сорту  $s \in \mathcal{S}$
- Формуле се могу квантификовати

# Семантика

## $\Sigma$ -структура $\mathcal{L} = (\mathcal{D}, \_{}^{\mathcal{L}})$

- $\mathcal{D} = \{D^s \mid s \in \mathcal{S}\}$
- $D^{\text{Bool}} = \{0, 1\}$
- $a^{\mathcal{L}} \in D^s$  ( $r(a) = [ ] \longrightarrow s$ )
- $f^{\mathcal{L}} : D^{s_1} \times \dots \times D^{s_n} \longrightarrow D^s$  ( $r(f) = [s_1, \dots, s_n] \longrightarrow s$ )
- $\perp^{\mathcal{L}} = 0$ ,  $\top^{\mathcal{L}} = 1$ ,  $\wedge^{\mathcal{L}}(x, y) = 1$  ако  $x = 1$  и  $y = 1$ , ...
- $=_s^{\mathcal{L}}(x, y) = 1$  ако су  $x$  и  $y$  исти елемент скупа  $D^s$

## Интерпретација

- Валуација  $v : V^s \longrightarrow D^s$  (за свако  $s \in \mathcal{S}$ )
- $x \in V^s$ :  $I_v^{\mathcal{L}}(x) = v(x)$
- $r(a) = [ ] \longrightarrow s$ :  $I_v^{\mathcal{L}}(a) = a^{\mathcal{L}}$
- $r(f) = [s_1, \dots, s_n] \longrightarrow s$ :  $I_v^{\mathcal{L}}(f(t_1, \dots, t_n)) = f^{\mathcal{L}}(I_v^{\mathcal{L}}(t_1), \dots, I_v^{\mathcal{L}}(t_n))$
- $I_v^{\mathcal{L}}((\forall x)F) = 1$  ако  $I_{v'}^{\mathcal{L}}(F) = 1$  за свако  $v'$  ( $v'(y) = v(y)$  за  $y \neq x$ )
- $I_v^{\mathcal{L}}((\exists x)F) = 1$  ако  $I_{v'}^{\mathcal{L}}(F) = 1$  за неко  $v'$  ( $v'(y) = v(y)$  за  $y \neq x$ )
- Интерпретација затворених формула не зависи од  $v$  ( $I_v^{\mathcal{L}}(F) = I^{\mathcal{L}}(F)$ )

## Семантика – наставак

### Дефиниције и ознаке

- $\mathcal{L} \models F$ : затворена формула  $F$  је тачна у  $\Sigma$ -структури  $\mathcal{L}$  ( $I^{\mathcal{L}}(F) = 1$ )
- Задовољива формула: постоји  $\Sigma$ -структура  $\mathcal{L}$  таква да је  $\mathcal{L} \models F$
- $\models F$ : ваљана формула (тачна у свим  $\Sigma$ -структурама)
- $\Delta \models F$ : логичка последица (тачна кад год су тачне све формуле из  $\Delta$ )
- $F_1 \equiv F_2$ : логичка еквиваленција ( $F_1 \models F_2$  и  $F_2 \models F_1$ )
- $F \models \perp$ : незадовољива (негација је ваљана)

# Дедукција

## Дедукциони систем

- Правила извођења облика:  $\frac{P_1, \dots, P_n}{Q}$
- Доказ: извођење применом правила
- $\Delta \vdash F$ :  $F$  је доказива из  $\Delta$
- Сагласност: ако  $\Delta \vdash F$ , онда  $\Delta \models F$
- Потпуност: ако  $\Delta \models F$ , онда  $\Delta \vdash F$
- Хилбертов систем, природна дедукција, рачун секвената...

# Теорија првог реда

## Синтаксно-дедуктивна дефиниција

- Теорија  $\mathcal{T}$  над сигнатуром  $\Sigma$  задата скупом аксиома  $Ax(\mathcal{T})$  је скуп свих формула  $F$  таквих да је  $Ax(\mathcal{T}) \vdash F$
- Формуле  $F \in \mathcal{T}$  зовемо теоремама теорије  $\mathcal{T}$
- Модел теорије: структура  $\mathcal{L}$  у којој су све аксиоме из  $Ax(\mathcal{T})$  тачне
- Важи  $Ax(\mathcal{T}) \vdash F$  ако  $Ax(\mathcal{T}) \models F$  ако  $\mathcal{L} \models F$  за сваки модел  $\mathcal{L}$  теорије  $\mathcal{T}$

## Семантичка дефиниција

- Теорија  $\mathcal{T}$  над сигнатуром  $\Sigma$  задата је скупом  $\Sigma$ -структура које називамо моделима теорије
- Формула  $F$  је ваљана у теорији  $\mathcal{T}$  ( $\mathcal{T}$ -ваљана) ако је тачна у свим њеним моделима ( $\models_{\mathcal{T}} F$ )
- Формула је задовољива у теорији  $\mathcal{T}$  ( $\mathcal{T}$ -задовољива) ако је тачна у бар једном моделу теорије  $\mathcal{T}$
- Логичка последица у теорији ( $\Delta \models_{\mathcal{T}} F$ ):  $F$  је тачна у свим моделима теорије  $\mathcal{T}$  у којима су тачне све формуле из  $\Delta$

# SMT проблем и SMT решавачи

## SMT проблем

- SMT проблем (енгл. Satisfiability Modulo Theory) за теорију  $\mathcal{T}$  је проблем испитивања  $\mathcal{T}$ -задовољивости дате формуле  $F$
- Одлучивост SMT проблема зависи од избора теорије  $\mathcal{T}$
- За поједине теорије SMT проблем је одлучив само за неке фрагменте (формуле одређеног облика)

## SMT решавачи

- Софтверски алати који имплементирају процедуре одлучивања за (одлучиве) SMT проблеме зову се SMT решавачи
- Релативно нова технологија (почетак 21. века)
- SAT технологија + процедуре одлучивања (лењи приступ)
- Примене: верификација софтвера и хардвера, проблеми задовољавања ограничења



# Квантификатори и SMT

## Квантфикатори

- Егзистенцијални квантификатори: сколемизација
- Универзалне квантификаторе није увек могуће уклонити
- Неке теорије допуштају елиминацију квантификатора
- Инстанцирање квантификатора
- SMT проблем за базне формуле – најчешћи случај

# ЕУФ теорија

## ЕУФ теорија

- Equality with Uninterpreted Functions
- Нема других предикатских симбола осим симбола једнакости (сви атоми су облика  $u = v$ )
- Сигнатура може садржати произвољан број сорти и функцијских симбола који се могу потпуно слободно интерпретирати (неинтерпретирани симболи и сорте)
- Модели теорије су сви (нормални) модели
- SMT проблем за ову теорију је неодлучив
- SMT проблем за базни фрагмент теорије је одлучив
- Задовољивост конјункције базних литерала ове теорије је одлучив у полиномијалном времену (Нелсон-Опен)

# Реална аритметика

## Реална аритметика

- Сигнатура: сорта Real, симболи  $0, 1, +, \cdot, -, /, \leq$
- Модел: структура реалних бројева  $\mathbb{R}$
- Теорија је одлучива (елиминацијом квантификатора)
- Базни линеарни фрагмент (QF\_LRA)
- Проблем испитивања задовољивости конјункције линеарних базних литерала је одлучив у полиномијалном времену
- Неке од процедура одлучивања (експоненцијалне сложености) су Фурије-Моцкинова процедура и Симплекс процедура

# Целобројна аритметика

## Целобројна аритметика

- Сигнатура: сорта `Int`, симболи  $0, 1, +, \cdot, -, \leq$
- Модел: структура целих бројева  $\mathbb{Z}$
- Теорија је неодлучива
- Њен линеарни фрагмент (Презбургерова аритметика) је одлучив
- Базни линеарни фрагмент (QF\_LIA)
- Проблем испитивања задовољности конјункције линеарних базних литерала је одлучив и NP-комплетан

# Теорија низова

## Теорија низова

- Сигнатура: сорте Index, Value и Array, симболи
  - $select : [Array, Index] \rightarrow Value$
  - $store : [Array, Index, Value] \rightarrow Array$
- Аксиоме:
  - $(\forall x)(\forall y)(\forall z)(select(store(x, y, z), y) = z)$
  - $(\forall x)(\forall y_1)(\forall y_2)(\forall z)(y_1 \neq y_2 \Rightarrow select(store(x, y_1, z), y_2) = select(x, y_2))$
  - $(\forall x_1)(\forall x_2)((\forall y)(select(x_1, y) = select(x_2, y)) \Rightarrow x_1 = x_2)$
- Теорија је неодлучива у општем случају
- Базни фрагмент теорије (QF\_AX) је одлучив
- Проблем испитивања задовољивости конјункције базних литерала је NP-комплетан

# Теорија битвектора

## Теорија битвектора

- Сигнатура: сорте  $\text{BitVec}_n$  ( $n \in \mathbb{N}$ ), симболи:
  - $bvnot_n, bvneg_n : [\text{BitVec}_n] \rightarrow \text{BitVec}_n$
  - $bvadd_n, bvshl_n, \dots : [\text{BitVec}_n, \text{BitVec}_n] \rightarrow \text{BitVec}_n$
  - $bvult_n, bvslt_n, \dots : [\text{BitVec}_n, \text{BitVec}_n] \rightarrow \text{Bool}$
- Модел: хардверска аритметика
- Теорија је одлучива
- Базни фрагмент  $\text{QF\_BV}$
- Проблем испитивања задовољивости конјункције базних литерала је NP-комплетан

# Лењи приступ

## Лењи приступ

- Исказна апстракција: атоми првог реда се замењују исказним словима
- SAT решавач утврђује задовољивост добијене исказне формуле
- Задовољавајућа исказна валуација одређује конјункцију базних литерала
- Посебна процедура одлучивања проверава задовољивост добијене конјункције базних литерала у теорији
- Резултат: ефикасност претраге SAT решавача + процедуре одлучивања прилагођене теорији
- „Лења ДНФ трансформација”

# DPLL( $\mathcal{T}$ )

## DPLL( $\mathcal{T}$ ) (Ниевенхуис, Оливерас, Тинели (2006))

- Најчешће коришћена архитектура заснована на лењом приступу
- DPLL заснован SAT решавач +  $\mathcal{T}$ -решавач
- SAT решавач: инкрементално конструише задовољавајуће исказне валуације
- $\mathcal{T}$ -решавач: испитује задовољивост одговарајуће конјункције литерала првог реда у теорији  $\mathcal{T}$  у току конструкције задовољавајуће валуације
- Могућност резоновања *унапред* у теорији (теоријске пропагације)



# DPLL( $\mathcal{T}$ )

## DPLL( $X$ ) – систем заснован на правилима

- Имплементира класичан CDCL заснован SAT решавач проширен додатним правилима за резонување у теорији
- Стање ( $F, M, C$ ):  $F$  скуп клауза,  $M$  је стек литерала (парцијална валуација),  $C$  је конфликтни скуп (или *no\_cflt* ако нема конфликта)
- Правило: дефинише начин промене стања, као и услове под којима се може применити
- Гранање, јединична пропагација, анализа конфликта и нехронолошко враћање уназад
- Додатно: теоријске пропагације и конфликти

# Функционалност $\mathcal{T}$ -решавача

## Обавезна функционалност

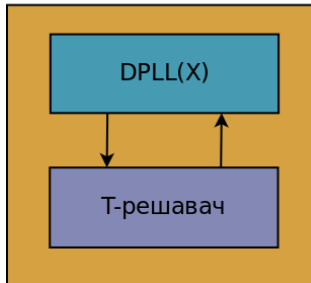
- да може да утврди да ли је конјункција литерала на стеку задовољива у теорији
- да може да конструише објашњење конфликта ( $R \subset M$  такав да је  $R \models_{\mathcal{T}} \perp$ )

## Пожељна функционалност

- да може да врши теоријске пропагације и генерише њихова објашњења ( $R \subset M$  такав да је  $R \models_{\mathcal{T}} I$ , где је  $I$  пропагирани литерал)
- инкременталност (могућност ефикасне провере задовољивости у случају додавања нових литерала у конјункцију без покретања целог поступка из почетка)
- ефикасна реконструкција претходног стања (за потребе враћања уназад)

# $DPLL(\mathcal{T})$

$DPLL(\mathcal{T})$  заснован SMT решавач



## Структура

- SMT решавач има модуларну структуру, компоненте су јасно одвојене и комуницирају путем прецизно дефинисаног интерфејса
- Оваква архитектура омогућава да се  $\mathcal{T}$ -решавач замени  $\mathcal{T}'$ -решавачем за неку другу теорију  $\mathcal{T}'$  без икаквих промена на SAT решавачу
- $DPLL(X) + \mathcal{T}\text{-решавач} = DPLL(\mathcal{T})$

## Пример интерфејса теоријског решавача

### Интерфејс процедуре

- *newLevel()* – успостављање новог нивоа одлучивања
- *backtrack( $m$ )* – враћање уназад на ниво  $m$
- *assert( $l$ )* – додавање литерала  $l$  на стек
- *checkConflict( $E$ )* – провера конфликта у теорији
- *checkPropagate( $L$ )* – детекција теоријских пропагација
- *explainLiteral( $l, E$ )* – објашњавање пропагираног литерала

# SMT-LIB

## SMT-LIB

- Познати SMT решавачи: Z3, Yices, CVC, MathSAT, OpenSMT, BarcelogicTools
- Циљ SMT-LIB иницијативе: боља координација у развоју и лакше поређење SMT решавача
- Стандард SMT-LIB (текућа верзија 2.5): улазно-излазни језик, логички оквир, теорије
- Велики скуп инстанци за тестирање и поређење
- <http://smtlib.cs.uiowa.edu/>