5mm.

Generička, objektno orijentisana platforma za korišćenje procedura odlučivanja

Filip Marić, dr Predrag Janičić

Matematički fakultet, Beograd

Motivacija

- Ime ARGO dolazi od *Automated Reasoning GrOup*, kao i od grčkog $A\rho\gamma\omega$, imenu broda kojim su drevni argonauti išli u potragu za zlatnim runom.

Motivacija

- Ime ARGO dolazi od *Automated Reasoning GrOup*, kao i od grčkog $A\rho\gamma\omega$, imenu broda kojim su drevni argonauti išli u potragu za zlatnim runom.
- Rad je započet u novembru 2003. godine i u kontinuitetu traje do danas

ARGOLIB DECISION PROCEDURES

Motivacija

- Ime ARGO dolazi od *Automated Reasoning GrOup*, kao i od grčkog $A\rho\gamma\omega$, imenu broda kojim su drevni argonauti išli u potragu za zlatnim runom.
- Rad je započet u novembru 2003. godine i u kontinuitetu traje do danas
- Platforma je nastala kao odgovor na potrebe za bibliotekom programskih rutina koje omogućavaju i olakšavaju istraživanja na polju automatskog dokazivanja teorema korišćenjem procedura odlučivanja

Mogućnosti platforme

Platforma bi trebalo da raspolaže sa:

- Opštim tehnikama koje omogućuju jednostavnu izgradnju procedura odlučivanja (unifikacija, prezapisivanje, ccc, apstrahovanje, . . .)

Mogućnosti platforme

Platforma bi trebalo da raspolaže sa:

- Opštim tehnikama koje omogućuju jednostavnu izgradnju procedura odlučivanja (unifikacija, prezapisivanje, ccc, apstrahovanje, . . .)
- Procedurama odlučivanja za teorije najkorišćenije u verifikaciji (EUF, PRA, ARRAYS,. . .)

Mogućnosti platforme

Platforma bi trebalo da raspolaže sa:

- Opštim tehnikama koje omogućuju jednostavnu izgradnju procedura odlučivanja (unifikacija, prezapisivanje, ccc, apstrahovanje, . . .)
- Procedurama odlučivanja za teorije najkorišćenije u verifikaciji (EUF, PRA, ARRAYS,. . .)
- Opštim shemama za kombinovanje i proširivanje procedura odlučivanja (Nelson-Oppen-ova shema, Shostak-ova shema, korišćenje hipoteza, . . .)

Dva osnovna cilja

- **Modularnost**

Dva osnovna cilja

- **Modularnost**
  - lako menjanje postojećih funkcionalnosti

Dva osnovna cilja

- **Modularnost**
  - lako menjanje postojećih funkcionalnosti
  - lako dodavanje nove funkcionalnosti

Dva osnovna cilja

- **Modularnost**
  - lako menjanje postojećih funkcionalnosti
  - lako dodavanje nove funkcionalnosti
  - eksperimentisanje i istraživanje novih tehnika

ArgoLib Decision Procedures

Dva osnovna cilja

- **Efikasnost**

Dva osnovna cilja

- **Efikasnost**
  - Prevazilaženje “proof-of-concept” nivoa implementacije

Dva osnovna cilja

- **Efikasnost**
  - Prevazilaženje “proof-of-concept” nivoa implementacije
  - Mogućnost direktnog procesiranja i obrade realnih zadataka.

Dva osnovna cilja

- **Efikasnost**
  - Prevazilaženje “proof-of-concept” nivoa implementacije
  - Mogućnost direktnog procesiranja i obrade realnih zadataka.
  - Rezultate uspešno obavljenih istraživanja na polju procedura odlučivanja je moguće direktno, bez potrebe za reimplementacijom.

Ostali ciljevi

- Prenostivost (portability)

Ostali ciljevi

- Prenostivost (portability)
- Samostalnost (stand-allone)

Ostali ciljevi

- Prenostivost (portability)
- Samostalnost (stand-allone)
- Javna dostupnost

Ostali ciljevi

- Prenostivost (portability)
- Samostalnost (stand-allone)
- Javna dostupnost
  - `http://www.matf.bg.ac.yu/~filip/argo`

Ostali ciljevi

- Prenostivost (portability)
- Samostalnost (stand-allone)
- Javna dostupnost
  - `http://www.matf.bg.ac.yu/~filip/argo`
- Dokumentovanost

Ostali ciljevi

- Prenostivost (portability)
- Samostalnost (stand-allone)
- Javna dostupnost
  - `http://www.matf.bg.ac.yu/~filip/argo`
- Dokumentovanost
- Uporedivost (benchmarking)

SMT-LIB inicijativa

- *Satisfiability Modulo Theory - Library*

SMT-LIB **inicijativa**

- *Satisfiability Modulo Theory - Library*
- Nastala poslednjih godina kao odgovor na potrebu za formiranje biblioteke normiranih testova (*benchmarks*) za evaluaciju sistema koji vrše automatsko rezonovanje koristeći procedure odlučivanja.

SMT-LIB **inicijativa**

- *Satisfiability Modulo Theory - Library*
- Nastala poslednjih godina kao odgovor na potrebu za formiranje biblioteke normiranih testova (*benchmarks*) za evaluaciju sistema koji vrše automatsko rezonovanje koristeći procedure odlučivanja.
- Propisuje format zapisa izraza, testova i opisa korišćenih teorija.

SMT-LIB - primer

```
(:name "RB4"
 :theory "REALS-II"
 :language "Conjunctions of linear equations"
 :extra_funs ((x Real) (y Real) (n Int) (m Int))
 :benchmarks
 (   (:formula
       (:and (= (+ x y) 1)
             (= (- (* 3 x) (* 3 y)) 1)
       }
       :status :sat
     )
     (:formula
       (:and (= (+ m n) 1)
             (= (- (* 3 m) (* 3 n)) 1)
       )
       :status :unsat
     )
 )
)
```

SMT-LIB-XML

- Na konferenciji PDPAR’04. održanoj u Cork-u (Irska), predložili smo prebacivanje SMT-LIB standarda u XML okvir. Inicijativa je naišla na principijelno odobravanje.

SMT-LIB-XML

- Na konferenciji PDPAR’04. održanoj u Cork-u (Irska), predložili smo prebacivanje SMT-LIB standarda u XML okvir. Inicijativa je naišla na principijelno odobravanje.
- Za opis samih matematičkih izraza u okviru benchmarka je predložen format MATH-ML koji postaje standard na polju predstavljanja matematičkih formula, i koji ima prilično rasprostranjenu podršku u postojećem softveru.

SMT-LIB-XML

- Na konferenciji PDPAR'04. održanoj u Cork-u (Irska), predložili smo prebacivanje SMT-LIB standarda u XML okvir. Inicijativa je naišla na principijelno odobravanje.
- Za opis samih matematičkih izraza u okviru benchmarka je predložen format MATH-ML koji postaje standard na polju predstavljanja matematičkih formula, i koji ima prilično rasprostranjenu podršku u postojećem softveru.
- Sva (odnosno skoro sva) sintaksna ograničenja navedena u SMT-LIB specifikaciji se mogu formalno reprezentovati u obliku DTD. Ovo omogućava automatsku validaciju korektnosti benchmarka.

SMT-LIB-XML - HTML reprezentacija benchmarka

Zapis benchmarka se lako konvertuje u čitljivi HTML oblik

Name	**Boyer-Moore's example**
Language	Universally quantified
Theories	FOLeq PRA
Extra Signature	delta : Int x Int x Int -> Int maxint : Int
Benchmark 1	
Hypotheses	1. delta $(!x, !y, !z) \leq !y$
$\forall lp \in Int.(\forall lt \in Int.(\forall c \in Int.(\forall i \in Int.(((lp + lt \leq maxint) \wedge (i \leq lt)) \Rightarrow (i + delta(lt, lp, c) \leq maxint)))))$	
status	valid

Planiranje dokaza (Proof planning)

- Kako bi se postigao veliki stepen modularnosti, korišćena je tehnika *planiranja dokaza*

Planiranje dokaza (Proof planning)

- Kako bi se postigao veliki stepen modularnosti, korišćena je tehnika *planiranja dokaza*
- U skladu sa ovim, sistem raspolaže skupom operacija relativno niskog nivoa tzv. *metodima*.

Planiranje dokaza (Proof planning)

- Kako bi se postigao veliki stepen modularnosti, korišćena je tehnika *planiranja dokaza*
- U skladu sa ovim, sistem raspolaže skupom operacija relativno niskog nivoa tzv. *metodima*.
- Operacije višeg nivoa kao i same procedure odlučivanja i sheme za njihovo kombinovanje i proširivanje se dobijaju korišćenjem i kombinovanjem metoda.

Objektno-orijentisan okvir

- Odlučeno je da se implementacija izvrši na objektno-orijentisanom jeziku opšte namene.

Objektno-orijentisan okvir

- Odlučeno je da se implementacija izvrši na objektno-orijentisanom jeziku opšte namene.
- Tokom dizajna su korišćeni objektno-orijentisani projektni obrasci *(design patterns)*

Objektno-orijentisan okvir

- Odlučeno je da se implementacija izvrši na objektno-orijentisanom jeziku opšte namene.
- Tokom dizajna su korišćeni objektno-orijentisani projektni obrasci *(design patterns)*
- Tokom celokupnog razvoja su korišćene tehnike refaktorisanja *(refactoring)*

ARGOLIB DECISION PROCEDURES

Odabir programskog jezika

- Odabran jezik C++

Odabir programskog jezika

- Odabran jezik

Verzije

- Platforma je konstantno menjana i oblikovana kroz mnoge svoje verzije.
- U jednom trenutku je izvršena kompletna reimplementacija i verzija ARGO-LIB 2.0 predstavlja potpuno nezavistan softver od verzija 1.x

Arhitektura sistema

- Predstavljanje formula i termova
  - Deljenje zajedničkih podtermova
  - Čitanje i zapis
- Prezapisivanje
  - Unifikacija
  - Zaustavljanje
  - Upotpunjavanje
- CCC

Arhitektura sistema - nastavak

- Teorije
  - Signature
    - Apstrahovanje stranih termova
  - Procedure odlučivanja
    - EUF
    - PRA
    - superpozicija
- Sheme za kombinovanje i proširivanje procedura odlučivanja
  - Nelson-Oppenova shema
  - Augmentacija

Predstavljanje formula i termova

- Formule i termovi su osnovni objekti kojima platforma manipuliše

Predstavljanje formula i termova

- Formule i termovi su osnovni objekti kojima platforma manipuliše
- Zajednički su predstavljeni klasom `Expression`, odnosno kompleksnom, ali za kranjeg korisnika skrivenom hijerarhijom klasa naslednica klase `ExpressionNode`.

Predstavljanje formula i termova

- Formule i termovi su osnovni objekti kojima platforma manipuliše
- Zajednički su predstavljeni klasom `Expression`, odnosno kompleksnom, ali za kranjeg korisnika skrivenom hijerarhijom klasa naslednica klase `ExpressionNode`.
- Izrazi su predstavljeni drvolikom strukturom, rekurzivnom kompozicijom, po uzorku *Composite*.

Predstavljanje formula i termova

- Formule i termovi su osnovni objekti kojima platforma manipuliše
- Zajednički su predstavljeni klasom `Expression`, odnosno kompleksnom, ali za kranjeg korisnika skrivenom hijerarhijom klasa naslednica klase `ExpressionNode`.
- Izrazi su predstavljeni drvolikom strukturom, rekurzivnom kompozicijom, po uzorku *Composite*.
- Izrazi poseduju samo svedeno ponašanje. Sve kompleksnije operacije za manipulaciju izrazima su smeštene u posebnim klasama operacija.

Predstavljanje formula i termova

- Formule i termovi su osnovni objekti kojima platforma manipuliše
- Zajednički su predstavljeni klasom `Expression`, odnosno kompleksnom, ali za kranjeg korisnika skrivenom hijerarhijom klasa naslednica klase `ExpressionNode`.
- Izrazi su predstavljeni drvolikom strukturom, rekurzivnom kompozicijom, po uzorku *Composite*.
- Izrazi poseduju samo svedeno ponašanje. Sve kompleksnije operacije za manipulaciju izrazima su smeštene u posebnim klasama operacija.
- Interfejs klase `Expression` dovoljno bogat da ovo omogući.

Interfejs klase Expression

- Proizvodni metodi kao npr.:

```
static Expression Variable   (const std::string name);
static Expression Predicate  (const std::string name,
                              const std::vector<Expression> operands);
...
```

- Metodi provere tipa izraza kao npr:

```
bool IsVariable              () const;
bool IsTerm                  () const;
bool IsConjunctionOfLiterals () const;
bool IsGround                () const;
```

- Operatori poređenja jednakosti izraza

```
bool operator== (const Expression& e) const;
bool operator!= (const Expression& e) const;
```

Interfejs klase `Expression`

- Operatori pristupa delovima izraza.

```
std::string          GetName      ()        const;
int                  GetArity     ()        const;
const Expression&    operator[]   (int num) const;
```

- Iteratori za pristup operandima i podizrazima. Npr. iteracija kroz sve podizraze se može ostvariti sa

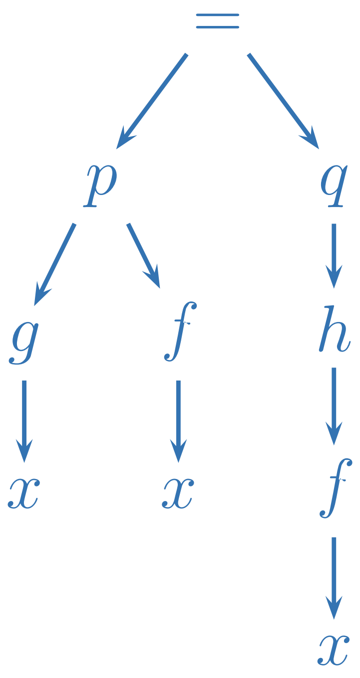
```
Expression::subexpressions_iterator i;
for (i = expression.subexpressions_begin();
     i != expression.subexpressions_end();
     ++i)
                 i->...
```

Interfejs klase Expression

- Metodi za promenu podizraza. Npr.

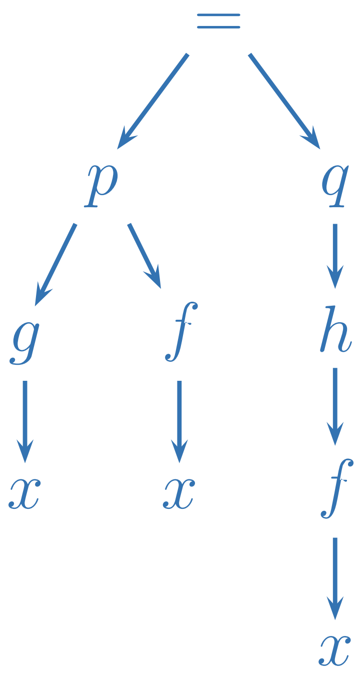
```
void       RemoveOperand    (int num);
void       SetOperand       (int num, const Expression& e);
Expression Substitute       (const Expression& expression,
                             const Expression& replacement) const;
Expression Substitute       (const Position& position,
                             const Expression& replacement) const;
```

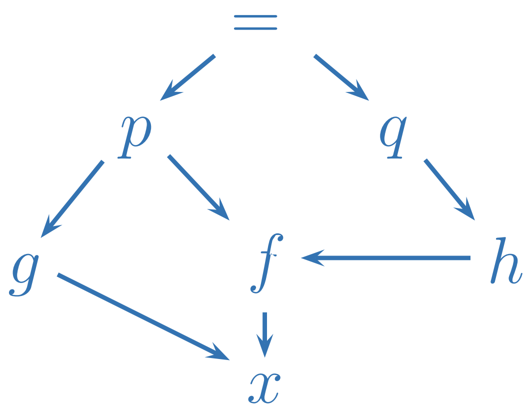
Deljenje zajedničkih podizraza



Deljenje podizraza za term $p(g(x), f(x)) = q(h(f(x)))$

Deljenje zajedničkih podizraza





Deljenje podizraza za term $p(g(x), f(x)) = q(h(f(x)))$

Deljenje zajedničkih podizraza

- Pre svega ušteda memorije
- Donosi i značajno poboljšanje brzine (uklanja *deep-copying*, ubrzava poređenje jednakosti izraza, ...)
- Neophodno je da promena implementacije ostane skrivena od krajnjih korisnika

Ulaz i izlaz

- ARGO-LIB u potpunosti podržava SMT-LIB inicijativu.

Ulaz i izlaz

- ARGO-LIB u potpunosti podržava SMT-LIB inicijativu.
- Koristeći alate LEX i YACC napravljen je parser koji čita izraze iz SMT-LIB zapisa.

Ulaz i izlaz

- ARGO-LIB u potpunosti podržava SMT-LIB inicijativu.
- Koristeći alate LEX i YACC napravljen je parser koji čita izraze iz SMT-LIB zapisa.
- ARGO-LIB nudi mogućnost ispisa formula i rezultata rada svojih metoda u nekoliko različitih formata:
  - *LaTeX format*
  - *SMT format*
  - *XMLFormat*

Prezapisivanje (Rewriting)

- Začetak sistemi za prepisivanje niski

Prezapisivanje (Rewriting)

- Začetak sistemi za prepisivanje niski
- Pravi razvoj kroz radove Knutha i Bendixa

Prezapisivanje (Rewriting)

- Začetak sistemi za prepisivanje niski
- Pravi razvoj kroz radove Knutha i Bendixa
- Koristi se u:
  - jednakosnom rezonovanju
  - različitim dokazivačima teorema
  - implementacijama funkcionalnih programskih jezika
  - prevodilačkim optimizatorima
  - ...

ARGOLIB DECISION PROCEDURES

Prezapisivanje (Rewriting)

Prezapisivanje se zasniva na primeni *pravila prezapisivanja* koja predstavljaju orijentisane jednakosti.

Primer:

- Pravilo prezapisivanja može da bude npr.

$$x + 0 \longrightarrow x$$

ili npr.

$$\varphi \wedge (\psi \vee \theta) \longrightarrow (\varphi \wedge \psi) \vee (\varphi \wedge \theta)$$

Unifikacija

- Osnovni preduslov za primenu prezapisivanja je *unifikacija*

Unifikacija

- Osnovni preduslov za primenu prezapisivanja je *unifikacija*
- Unifikacija je proces pronalaženja supstitucije promenljivih koja čini dva terma jednakim

Unifikacija

- Osnovni preduslov za primenu prezapisivanja je *unifikacija*
- Unifikacija je proces pronalaženja supstitucije promenljivih koja čini dva terma jednakim
- *Primer:* Term $x + 0$ se unifikuje sa termom $2 + 0$, koristeći supstituciju promenljivih $\{x \mapsto 2\}$

Prezapisivanje

Definicija: Termi s i t su u relaciji $s \to t$ ako i samo ako postoji $l \to r$, $p \in Pos(s)$, i supstitucija σ tako da je

$$s|_p = \sigma(l) \quad \text{i} \quad t = s[\sigma(r)]_p$$

Prezapisivanje

Definicija: Termi s i t su u relaciji $s \rightarrow t$ ako i samo ako postoji $l \rightarrow r$, $p \in Pos(s)$, i supstitucija σ tako da je

$$s|_p = \sigma(l) \quad \text{i} \quad t = s[\sigma(r)]_p$$

Primer:

$$x + 0 \quad \rightarrow \quad x$$

$$1 * (2 + 0)$$

Prezapisivanje

Definicija: Termi s i t su u relaciji $s \to t$ ako i samo ako postoji $l \to r$, $p \in Pos(s)$, i supstitucija σ tako da je

$$s|_p = \sigma(l) \quad \text{i} \quad t = s[\sigma(r)]_p$$

Primer:

$$x + 0 \quad \to \quad x$$

$$1 * (2 + 0)$$

$$\{x \mapsto 2\}$$

Prezapisivanje

Definicija: Termi s i t su u relaciji $s \to t$ ako i samo ako postoji $l \to r$, $p \in Pos(s)$, i supstitucija σ tako da je

$$s|_p = \sigma(l) \quad \text{i} \quad t = s[\sigma(r)]_p$$

Primer:

$$\begin{array}{rcl} x + 0 & \to & x \\ 1 * (2 + 0) & \to & 1 * 2 \end{array}$$

$$\{x \mapsto 2\}$$

Osnovna svojstva sistema za prezapisivanje

Zaustavljanje

- *Primer:* Posmatrajmo pravilo $x + y \rightarrow y + x$
- Term $0 + 1$ je moguće prezapisati na sledeći način:

Osnovna svojstva sistema za prezapisivanje

Zaustavljanje

- *Primer:* Posmatrajmo pravilo $x + y \to y + x$
- Term $0 + 1$ je moguće prezapisati na sledeći način:
  $0 + 1$

Osnovna svojstva sistema za prezapisivanje

Zaustavljanje

- *Primer:* Posmatrajmo pravilo $x + y \to y + x$
- Term $0 + 1$ je moguće prezapisati na sledeći način:
  $0 + 1 \ \to 1 + 0$

Osnovna svojstva sistema za prezapisivanje

Zaustavljanje

- *Primer:* Posmatrajmo pravilo $x + y \rightarrow y + x$
- Term $0 + 1$ je moguće prezapisati na sledeći način: $0 + 1 \ \rightarrow 1 + 0 \ \rightarrow 0 + 1$

Osnovna svojstva sistema za prezapisivanje

Zaustavljanje

- *Primer:* Posmatrajmo pravilo $x + y \to y + x$
- Term $0 + 1$ je moguće prezapisati na sledeći način:
  $0 + 1 \;\to 1 + 0 \;\to 0 + 1 \;\to 1 + 0$

Osnovna svojstva sistema za prezapisivanje

Zaustavljanje

- *Primer:* Posmatrajmo pravilo $x + y \to y + x$
- Term $0 + 1$ je moguće prezapisati na sledeći način:
  $0 + 1 \;\to 1 + 0 \;\to 0 + 1 \;\to 1 + 0 \;\to \ldots$

Osnovna svojstva sistema za prezapisivanje

Zaustavljanje

- *Primer:* Posmatrajmo pravilo $x + y \to y + x$
- Term $0 + 1$ je moguće prezapisati na sledeći način:
  $0 + 1 \ \to 1 + 0 \ \to 0 + 1 \ \to 1 + 0 \ \to \ldots$
- Dakle, sistem koji sadrži samo ovo pravilo nije zaustavljajući

Osnovna svojstva sistema za prezapisivanje

Konfluentnost

- *Primer:* Posmatrajmo sistem

$$
\begin{array}{rcl}
x + 0 & \rightarrow & x \\
s(x + y) & \rightarrow & x + s(y)
\end{array}
$$

Osnovna svojstva sistema za prezapisivanje

Konfluentnost

- *Primer:* Posmatrajmo sistem

$$\begin{array}{rcl} x+0 & \rightarrow & x \\ s(x+y) & \rightarrow & x+s(y) \end{array}$$

- Term $s(x+0)$ je moguće prezapisati na sledeće načine:

$$s(x+0)$$

Osnovna svojstva sistema za prezapisivanje

Konfluentnost

- *Primer:* Posmatrajmo sistem

$$\begin{aligned} x + 0 &\rightarrow x \\ s(x+y) &\rightarrow x + s(y) \end{aligned}$$

- Term $s(x+0)$ je moguće prezapisati na sledeće načine:

$$\begin{tikzcd} & s(x+0) \arrow[dl] \\ s(x) & \end{tikzcd}$$

Osnovna svojstva sistema za prezapisivanje

Konfluentnost

- *Primer:* Posmatrajmo sistem

$$
\begin{aligned}
x + 0 &\longrightarrow x \\
s(x + y) &\longrightarrow x + s(y)
\end{aligned}
$$

- Term $s(x + 0)$ je moguće prezapisati na sledeće načine:

$$
\begin{tikzcd}
 & s(x+0) \arrow[dl] \arrow[dr] & \\
s(x) & & x + s(0)
\end{tikzcd}
$$

Osnovna svojstva sistema za prezapisivanje

Konfluentnost

- *Primer:* Posmatrajmo sistem

$$\begin{array}{rcl} x + 0 & \to & x \\ s(x + y) & \to & x + s(y) \end{array}$$

- Term $s(x + 0)$ je moguće prezapisati na sledeće načine:

$$\begin{tikzcd} & s(x+0) \arrow[dl] \arrow[dr] & \\ s(x) & & x + s(0) \end{tikzcd}$$

- Ovi termovi se ne mogu dalje prezapisivati.

ARGOLIB
DECISION PROCEDURES

Osnovna svojstva sistema za prezapisivanje

- *Definicija:* Sistem za prezapisivanje je *zaustavljajući* ukoliko ne postoji beskonačni lanac $a_1 \rightarrow a_2 \rightarrow \ldots$

Osnovna svojstva sistema za prezapisivanje

- *Definicija:* Sistem za prezapisivanje je *zaustavljajući* ukoliko ne postoji beskonačni lanac $a_1 \longrightarrow a_2 \longrightarrow \ldots$
- *Definicija:* Sistem za prezapisivanje je *konfluentan* ukoliko

$$s$$

ARGOLIB DECISION PROCEDURES

Osnovna svojstva sistema za prezapisivanje

- *Definicija:* Sistem za prezapisivanje je *zaustavljajući* ukoliko ne postoji beskonačni lanac $a_1 \rightarrow a_2 \rightarrow \ldots$
- *Definicija:* Sistem za prezapisivanje je *konfluentan* ukoliko

$$\begin{tikzcd} & s \arrow[dl, "*"'] \arrow[dr, "*"] & \\ t_1 & & t_2 \end{tikzcd}$$

Osnovna svojstva sistema za prezapisivanje

- *Definicija:* Sistem za prezapisivanje je *zaustavljajući* ukoliko ne postoji beskonačni lanac $a_1 \to a_2 \to \ldots$

- *Definicija:* Sistem za prezapisivanje je *konfluentan* ukoliko

$$
\begin{tikzcd}
 & s \arrow[dl, "*"'] \arrow[dr, "*"] & \\
t_1 \arrow[dr, dotted, "*"'] & & t_2 \arrow[dl, dotted, "*"] \\
 & t &
\end{tikzcd}
$$

Osnovna svojstva sistema za prezapisivanje

- *Definicija:* Sistem za prezapisivanje je *zaustavljajući* ukoliko ne postoji beskonačni lanac $a_1 \longrightarrow a_2 \longrightarrow \ldots$

- *Definicija:* Sistem za prezapisivanje je *lokalno konfluentan* ukoliko

$$
\begin{tikzcd}
 & s \arrow[dl] \arrow[dr] & \\
t_1 \arrow[dr, dotted, "*"'] & & t_2 \arrow[dl, dotted, "*"] \\
 & t &
\end{tikzcd}
$$

ARGOLIB DECISION PROCEDURES

Osnovna svojstva sistema za prezapisivanje

- *Definicija:* Sistem za prezapisivanje je *zaustavljajući* ukoliko ne postoji beskonačni lanac $a_1 \to a_2 \to \ldots$
- *Definicija:* Sistem za prezapisivanje je *lokalno konfluentan* ukoliko

$$\begin{tikzcd} & s \arrow[dl] \arrow[dr] & \\ t_1 \arrow[dr, dotted, "*"'] & & t_2 \arrow[dl, dotted, "*"] \\ & t & \end{tikzcd}$$

- *Definicija:* Sisteme koji su i zaustavljajući i konfluentni nazivamo *konvergentnim*

Pokazivanje zaustavljanja

- U opštem slučaju pitanje zaustavljanja je neodlučivo.

Pokazivanje zaustavljanja

- U opštem slučaju pitanje zaustavljanja je neodlučivo.
- Zaustavljanje se najčešće pokazuje korišćenjem posebnih uređenja - *uređenja svođenja*

Pokazivanje zaustavljanja

- U opštem slučaju pitanje zaustavljanja je neodlučivo.
- Zaustavljanje se najčešće pokazuje korišćenjem posebnih uređenja - *uređenja svođenja*
- *Teorema:* Sistem za prezapisivanje termova R je zaustavljajući ako i samo ako postoji uređenje svođenja $>$ tako da je za svako pravilo izvođenja $l > r$ tj.

$$l \rightarrow r \in R \;\Rightarrow\; l > r$$

Pokazivanje zaustavljanja

- *Definicija:* Strogo parcijalno uređenje $>$ na $Term_{\mathcal{L}}$ nazivamo uređenje svođenja ukoliko zadovoljava sledeće uslove:

Pokazivanje zaustavljanja

- *Definicija:* Strogo parcijalno uređenje $>$ na $Term_{\mathcal{L}}$ nazivamo uređenje svođenja ukoliko zadovoljava sledeće uslove:
  1. *(dobro uređenje)* $>$ je dobro uređenje tj. ne postoji beskonačni lanac $t_1 > t_2 > t_3 > \dots$

Pokazivanje zaustavljanja

- *Definicija:* Strogo parcijalno uređenje $>$ na $Term_{\mathcal{L}}$ nazivamo uređenje svođenja ukoliko zadovoljava sledeće uslove:
  1. *(dobro uređenje)* $>$ je dobro uređenje tj. ne postoji beskonačni lanac $t_1 > t_2 > t_3 > \dots$
  2. *(monotono uređenje)* $>$ je u skladu sa Σ funkcijama tj. za svako $f \in \Sigma$ i svako $t_1, t_2 \in Term_{\mathcal{L}}$ važi

$$t_1 > t_2 \Rightarrow f(\dots, t_1, \dots) > f(\dots, t_2, \dots)$$

Pokazivanje zaustavljanja

- *Definicija:* Strogo parcijalno uređenje $>$ na $Term_{\mathcal{L}}$ nazivamo uređenje svođenja ukoliko zadovoljava sledeće uslove:
  1. *(dobro uređenje)* $>$ je dobro uređenje tj. ne postoji beskonačni lanac $t_1 > t_2 > t_3 > \ldots$
  2. *(monotono uređenje)* $>$ je u skladu sa Σ funkcijama tj. za svako $f \in \Sigma$ i svako $t_1, t_2 \in Term_{\mathcal{L}}$ važi

     $$t_1 > t_2 \Rightarrow f(\ldots, t_1, \ldots) > f(\ldots, t_2, \ldots)$$

  3. *(stabilno uređenje)* $>$ je zatvoreno za supstituciju tj. ukoliko je σ supstitucija, tada

     $$t_1 > t_2 \Rightarrow \sigma(t_1) > \sigma(t_2)$$

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*

- *Definicija:* Neka su $l_1 \longrightarrow r_1$ i $l_2 \longrightarrow r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2 .
  Term $l_1[\theta(l$

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*
- *Definicija:* Neka su $l_1 \to r_1$ i $l_2 \to r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2.
  Term $l_1[\theta(l_2)]_p$ određuje kritični par $\langle\theta(r_1), \theta(l_1)[\theta(r_2)]_p\rangle$

Primer:

$$\begin{aligned} s(x+y) &\to x+s(y) \\ x+0 &\to x \end{aligned}$$

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*
- *Definicija:* Neka su $l_1 \rightarrow r_1$ i $l_2 \rightarrow r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2 .
  Term $l_1[\theta(l_2)]_p$ određuje kritični par $\langle \theta(r_1), \theta(l_1)[\theta(r_2)]_p \rangle$

Primer:

$$\begin{aligned} s(x+y) &\rightarrow x+s(y) \\ x+0 &\rightarrow x \end{aligned}$$

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*
- *Definicija:* Neka su $l_1 \rightarrow r_1$ i $l_2 \rightarrow r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2 .
  Term $l_1[\theta(l_2)]_p$ određuje kritični par $\langle \theta(r_1), \theta(l_1)[\theta(r_2)]_p \rangle$

Primer:

$$
\begin{aligned}
s(x + y) &\rightarrow x + s(y) \\
x + 0 &\rightarrow x
\end{aligned}
$$

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*

- *Definicija:* Neka su $l_1 \rightarrow r_1$ i $l_2 \rightarrow r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2 .
  Term $l_1[\theta(l_2)]_p$ određuje kritični par $\langle \theta(r_1), \theta(l_1)[\theta(r_2)]_p \rangle$

Primer:

$$\begin{array}{rcl} s(x+y) & \rightarrow & x+s(y) \\ x+0 & \rightarrow & x \end{array}$$

$\theta = \{y \mapsto 0\}$

ARGOLIB DECISION PROCEDURES

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*
- *Definicija:* Neka su $l_1 \rightarrow r_1$ i $l_2 \rightarrow r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2 .
  Term $l_1[\theta(l_2)]_p$ određuje kritični par $\langle \theta(r_1), \theta(l_1)[\theta(r_2)]_p \rangle$

Primer:

$$\begin{array}{rcl} s(x+y) & \rightarrow & x+s(y) \\ x+0 & \rightarrow & x \end{array} \qquad s(x+0)$$

$\theta = \{y \mapsto 0\}$

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*
- *Definicija:* Neka su $l_1 \rightarrow r_1$ i $l_2 \rightarrow r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2 .
  Term $l_1[\theta(l_2)]_p$ određuje kritični par $\langle \theta(r_1), \theta(l_1)[\theta(r_2)]_p \rangle$

Primer:

$$\begin{array}{rcl} s(x+y) & \rightarrow & x+s(y) \\ x+0 & \rightarrow & x \end{array}$$

$$s(x+0)$$

$$x+s(0)$$

$$\theta = \{y \mapsto 0\}$$

Konfluentnost, kritični parovi, upotpunjavanje

- Za ispitivanje konfluentnosti ključan je koncept *kritčnih parova*
- *Definicija:* Neka su $l_1 \longrightarrow r_1$ i $l_2 \longrightarrow r_2$ dva pravila koja nemaju zajedničkih promenljivih.
  Neka je $p \in Pos(l_1)$ takvo da $l_1|_p$ nije promenljiva i neka je θ najopštiji unifikator termova $l_1|_p$ i l_2 .
  Term $l_1[\theta(l_2)]_p$ određuje kritični par
  $\langle \theta(r_1), \theta(l_1)[\theta(r_2)]_p \rangle$

Primer:

$$
\begin{array}{rcl}
s(x+y) & \longrightarrow & x+s(y) \\
x+0 & \longrightarrow & x
\end{array}
$$

$$
\begin{tikzcd}[column sep=small]
 & s(x+0) \arrow[dl] \arrow[dr] & \\
x+s(0) & & s(x)
\end{tikzcd}
$$

$$\theta = \{y \mapsto 0\}$$

Konfluentnost, kritični parovi, upotpunjavanje

- *Teorema:* Zaustavljajući sistem za prezapisivanje je konfluentan ako i samo ako su mu svi kritični parovi povezivi.

Konfluentnost, kritični parovi, upotpunjavanje

- *Teorema:* Zaustavljajući sistem za prezapisivanje je konfluentan ako i samo ako su mu svi kritični parovi povezivi.

- Prethodna teorema inspiriše *proceduru upotpunjavanja*

Konfluentnost, kritični parovi, upotpunjavanje

- *Teorema:* Zaustavljajući sistem za prezapisivanje je konfluentan ako i samo ako su mu svi kritični parovi povezivi.
- Prethodna teorema inspiriše *proceduru upotpunjavanja*
- Procedura upotpunjavanja omogućava da se određeni sistemi za prezapisivanje dodavanjem izvesnog broja pravila učine konvergentnim.

Procedure odlučivanja

- ARGO-LIB raspolaže procedurama odlučivanja za nekoliko značajnih teorija (PRA, EUF, LISTS, ARRAYS, . . .)

Procedure odlučivanja

- ARGO-LIB raspolaže procedurama odlučivanja za nekoliko značajnih teorija (PRA, EUF, LISTS, ARRAYS, . . .)
- Procedure odlučivanja su definisane za konjunkcije literala

Procedure odlučivanja

- ARGO-LIB raspolaže procedurama odlučivanja za nekoliko značajnih teorija (PRA, EUF, LISTS, ARRAYS, . . .)
- Procedure odlučivanja su definisane za konjunkcije literala
- Od procedure se, takođe, zahteva da može da prijavi sve implicitne jednakosti među promenljivima koje se javljaju u formuli.

Inkrementalnost

- Zahtev koji je poželjan za svaku proceduru odlučivanja je *inkrementalnost*. Inkrementalnost dozvoljava dodavanje jednog po jednog literala, bez potrebe za pokretanjem procedure iz početka.

Inkrementalnost

- Zahtev koji je poželjan za svaku proceduru odlučivanja je *inkrementalnost*. Inkrementalnost dozvoljava dodavanje jednog po jednog literala, bez potrebe za pokretanjem procedure iz početka.
- Inkremetalne procedure odlučivanja se odlikuju i mogućnošću povlačenja određenog broja dodatih literala.

Inkrementalnost

- Zahtev koji je poželjan za svaku proceduru odlučivanja je *inkrementalnost*. Inkrementalnost dozvoljava dodavanje jednog po jednog literala, bez potrebe za pokretanjem procedure iz početka.
- Inkremetalne procedure odlučivanja se odlikuju i mogućnošću povlačenja određenog broja dodatih literala.
- Inkrementalnost poboljšava efikasnost shema za kombinovanje procedura odlučivanja, kao i sheme za kombinovanje sa iskaznim rezonovanjem.

- Kao primer navodimo proceduru odlučivanja za *Prezburgerovu racionalnu aritmetiku*.

PRA

- Kao primer navodimo proceduru odlučivanja za *Prezburgerovu racionalnu aritmetiku*.
- Procedura se zasniva na *Fourier-Motzkin-ovom* algoritmu eliminacije promenljivih.

PRA

- Kao primer navodimo proceduru odlučivanja za *Prezburgerovu racionalnu aritmetiku*.
- Procedura se zasniva na *Fourier-Motzkin-ovom* algoritmu eliminacije promenljivih.
- Osnovni korak je eliminacija zajedničke promenljive iz dva literala.

PRA

- Literali se čuvaju u normalizovanom obliku.

Definicija:

$$a_m x_m \odot \sum_{k=1}^{n} a_k x_k + c, \quad \odot \in \{<, >, \leq, \geq\}$$

pri čemu je $x_m \succ x_k$ za svako k i $a_m > 0$.

PRA

- Literali se čuvaju u normalizovanom obliku.

  Definicija:

  $$a_m x_m \odot \sum_{k=1}^{n} a_k x_k + c, \quad \odot \in \{<, >, \leq, \geq\}$$

  pri čemu je $x_m \succ x_k$ za svako k i $a_m > 0$.

- Na literale u normalnoj formi se primenjuje operacija eliminacije

PRA

- Literali se čuvaju u normalizovanom obliku.

  Definicija:

$$a_m x_m \odot \sum_{k=1}^{n} a_k x_k + c, \quad \odot \in \{<, >, \leq, \geq\}$$

  pri čemu je $x_m \succ x_k$ za svako k i $a_m > 0$.

- Na literale u normalnoj formi se primenjuje operacija eliminacije

- *Primer:* Literali $x > z - y$ i $x \leq y + z$ postupkom eliminacije daju literal $z - y < y + z$ tj. $y > 0$.

PRA

- Literali se čuvaju u normalizovanom obliku.

  Definicija:
  $$a_m x_m \odot \sum_{k=1}^{n} a_k x_k + c, \quad \odot \in \{<, >, \leq, \geq\}$$
  pri čemu je $x_m \succ x_k$ za svako k i $a_m > 0$.

- Na literale u normalnoj formi se primenjuje operacija eliminacije

- *Primer:* Literali $x > z - y$ i $x \leq y + z$ postupkom eliminacije daju literal $z - y < y + z$ tj. $y > 0$.

- *Teorema:* Skup literala $\mathcal{A}$ je nezadovoljiv u teoriji PRA je ako i samo ako se postupkom eliminacije promenljivih iz njegovih literala može dobiti kontradiktoran literal.

PRA

- Prikažimo rad procedure na konjunkciji

$$x \le y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$

PRA

- Prikažimo rad procedure na konjunkciji
$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$
- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju
$$x \leq y + z$$

PRA

- Prikažimo rad procedure na konjunkciji

$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$

- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$

$$x > z - y$$

PRA

- Prikažimo rad procedure na konjunkciji

$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$

- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$\begin{array}{l} x \leq y + z \\ x > z - y \\ y > 0 \end{array}$$

PRA

- Prikažimo rad procedure na konjunkciji
$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$
- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$

$$x > z - y$$

$$y > 0$$

$$x > y$$

PRA

- Prikažimo rad procedure na konjunkciji

$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$

- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$

$$x > z - y$$

$$y > 0$$

$$x > y$$

$$z > 0$$

PRA

- Prikažimo rad procedure na konjunkciji
$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$
- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$

$$x > z - y$$

$$y > 0$$

$$x > y$$

$$z > 0$$

$$x < -1$$

PRA

- Prikažimo rad procedure na konjunkciji
$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$
- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$

$$x > z - y$$

$$y > 0$$

$$x > y$$

$$z > 0$$

$$x < -1$$

$$y > z + 1$$

PRA

- Prikažimo rad procedure na konjunkciji
$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$
- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$

$$x > z - y$$

$$y > 0$$

$$x > y$$

$$z > 0$$

$$x < -1$$

$$y > z + 1$$

$$y < -1$$

PRA

- Prikažimo rad procedure na konjunkciji
$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$
- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$
$$x > z - y$$
$$y > 0$$
$$x > y$$
$$z > 0$$
$$x < -1$$
$$y > z + 1$$
$$y < -1$$
$$0 < -1$$

PRA

- Prikažimo rad procedure na konjunkciji
$$x \leq y + z \wedge z < y + x \wedge x - y > 0 \wedge x < -1$$
- Inkrementalno dodajemo jedan po jedan normalizovan literal i vršimo eliminaciju

$$x \leq y + z$$

$$x > z - y$$

$$y > 0$$

$$x > y$$

$$z > 0$$

$$x < -1$$

$$y > z + 1$$

$$y < -1$$

$$0 < -1$$

Superpozicija

- Upotpunjavanje je osnova za izgradnju mnogih procedura odlučivanja

Superpozicija

- Upotpunjavanje je osnova za izgradnju mnogih procedura odlučivanja
- *Primer:* Teorija lista LISTS se aksiomatizuje putem sledećih aksioma

$$
\begin{aligned}
cons(car(l), cdr(l)) &= l \\
car(cons(a, l)) &= a \\
cdr(cons(a, l)) &= l
\end{aligned}
$$

ARGOLIB DECISION PROCEDURES

Superpozicija

$$\begin{aligned} cons(car(l), cdr(l)) &\rightarrow l \\ car(cons(a, l)) &\rightarrow a \\ cdr(cons(a, l)) &\rightarrow l \end{aligned}$$

- Ispitajmo zadovoljivost formule

$$cons(x, z) = cons(y, z) \wedge x \neq y$$

Superpozicija

$$\begin{aligned} cons(car(l), cdr(l)) &\rightarrow l \\ car(cons(a, l)) &\rightarrow a \\ cdr(cons(a, l)) &\rightarrow l \end{aligned} \qquad cons(x, z) \rightarrow cons(y, z)$$

- Ispitajmo zadovoljivost formule

$$cons(x, z) = cons(y, z) \wedge x \neq y$$

- Pridođamo orijentisanu jednakost $cons(x, z) \rightarrow cons(y, z)$ sistemu za prezapisivanje dobijenom od aksioma.

Superpozicija

$$\begin{aligned} cons(car(l), cdr(l)) &\rightarrow l \\ car(cons(a, l)) &\rightarrow a \\ cdr(cons(a, l)) &\rightarrow l \end{aligned}$$

$$cons(x, z) \rightarrow cons(y, z)$$

- Ispitajmo zadovoljivost formule

$$cons(x, z) = cons(y, z) \wedge x \neq y$$

- Pridođamo orijentisanu jednakost $cons(x, z) \rightarrow cons(y, z)$ sistemu za prezapisivanje dobijenom od aksioma.
- Dobijeni sistem nije konvergentan i na njega se može primeniti procedura upotpunjavanja.

Superpozicija

$$\begin{array}{rcl} cons(car(l), cdr(l)) & \rightarrow & l \\ car(cons(a,l)) & \rightarrow & a \\ cdr(cons(a,l)) & \rightarrow & l \end{array}$$

$$cons(x,z) \rightarrow cons(y,z)$$

$$x \rightarrow y$$

- Ispitajmo zadovoljivost formule

$$cons(x,z) = cons(y,z) \wedge x \neq y$$

- Pridođamo orijentisanu jednakost $cons(x,z) \rightarrow cons(y,z)$ sistemu za prezapisivanje dobijenom od aksioma.
- Dobijeni sistem nije konvergentan i na njega se može primeniti procedura upotpunjavanja.
- Upotpunjavanjem se dobija konvergentni sistem

ARGOLIB DECISION PROCEDURES

Superpozicija

$$
\begin{aligned}
cons(car(l), cdr(l)) &\rightarrow l \\
car(cons(a, l)) &\rightarrow a \\
cdr(cons(a, l)) &\rightarrow l \\
cons(x, z) &\rightarrow cons(y, z) \\
x &\rightarrow y
\end{aligned}
$$

- Ispitajmo zadovoljivost formule

$$cons(x, z) = cons(y, z) \wedge x \neq y$$

ARGOLIB DECISION PROCEDURES

Superpozicija

$$\begin{array}{rcl} cons(car(l), cdr(l)) & \rightarrow & l \\ car(cons(a, l)) & \rightarrow & a \\ cdr(cons(a, l)) & \rightarrow & l \\ cons(x, z) & \rightarrow & cons(y, z) \\ x & \rightarrow & y \end{array}$$

- Ispitajmo zadovoljivost formule

$$cons(x, z) = cons(y, z) \wedge x \neq y$$

- Iskoristimo dobijeni sistem da normalizujemo levu odnosno desnu stranu različitosti $x \neq y$.

Superpozicija

$$
\begin{aligned}
cons(car(l), cdr(l)) &\longrightarrow l \\
car(cons(a, l)) &\longrightarrow a \\
cdr(cons(a, l)) &\longrightarrow l \\
cons(x, z) &\longrightarrow cons(y, z) \\
x &\longrightarrow y
\end{aligned}
$$

- Ispitajmo zadovoljivost formule

$$cons(x, z) = cons(y, z) \wedge x \neq y$$

- Iskoristimo dobijeni sistem da normalizujemo levu odnosno desnu stranu različitosti $y \neq y$.

Superpozicija

$$
\begin{array}{rcl}
cons(car(l), cdr(l)) & \longrightarrow & l \\
car(cons(a, l)) & \longrightarrow & a \\
cdr(cons(a, l)) & \longrightarrow & l \\
cons(x, z) & \longrightarrow & cons(y, z) \\
x & \longrightarrow & y
\end{array}
$$

- Ispitajmo zadovoljivost formule

$$cons(x, z) = cons(y, z) \wedge x \neq y$$

- Iskoristimo dobijeni sistem da normalizujemo levu odnosno desnu stranu različitosti $y \neq y$.
- Dobijena kontradiktornost ukazuje na nezadovoljivost polazne formule

Kombinovanje procedura odlučivanja

- U realnim verifikacijskim problemima se često javljaju formule definisane nad jezicima više teorija.

Kombinovanje procedura odlučivanja

- U realnim verifikacijskim problemima se često javljaju formule definisane nad jezicima više teorija.
- *Primer:* Formula

$$z = f(x - y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

sadrži aritmetičke simbole $+$ i $-$, ali i neinterpretirani funkcijski simbol f tako da je definisana nad kombinacijama teorija PRA i EUF.

Kombinovanje procedura odlučivanja

- U realnim verifikacijskim problemima se često javljaju formule definisane nad jezicima više teorija.
- *Primer:* Formula

$$z = f(x-y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

  sadrži aritmetičke simbole $+$ i $-$, ali i neinterpretirani funkcijski simbol f tako da je definisana nad kombinacijama teorija PRA i EUF.
- Kako bi se rezonovalo o ovakvim formulama, koriste se *sheme za kombinovanje procedura odlučivanja*

Nelson-Oppenova shema

- Najkorišćenija shema za kombinovanje procedura odlučivanja je *Nelson-Oppen-va* shema

Nelson-Oppenova shema

- Najkorišćenija shema za kombinovanje procedura odlučivanja je *Nelson-Oppen-va* shema
- Shema se zasniva na prosleđivanju implicitnih jednakosti koje važe među promenljivima

Nelson-Oppenova shema

- Najkorišćenija shema za kombinovanje procedura odlučivanja je *Nelson-Oppen-va* shema
- Shema se zasniva na prosleđivanju implicitnih jednakosti koje važe među promenljivima
- Definisana je isključivo za konjunkcije literala

Nelson-Oppenova shema

- Najkorišćenija shema za kombinovanje procedura odlučivanja je *Nelson-Oppen-va* shema
- Shema se zasniva na prosleđivanju implicitnih jednakosti koje važe među promenljivima
- Definisana je isključivo za konjunkcije literala
- Teorije koje se kombinuju moraju da zadovoljavaju određena svojstva

Nelson-Oppenova shema - primer

$$z = f(x - y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

Nelson-Oppenova shema - primer

$$z = f(x - y) \land x = y + z \land x \neq f(f(z)) + y$$

PRA	EUF
$v_1 = x - y$	$z = f(v_1)$
$x = y + z$	
$x \neq v_2 + y$	$v_2 = f(f(z))$

Nelson-Oppenova shema - primer

$$z = f(x-y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

PRA	EUF
$v_1 = x - y$	$z = f(v_1)$
$x = y + z$	
$x \neq v_2 + y$	$v_2 = f(f(z))$

$v_1 = z$

Nelson-Oppenova shema - primer

$$z = f(x - y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

PRA	EUF
$v_1 = x - y$	$z = f(v_1)$
$x = y + z$	
$x \neq v_2 + y$	$v_2 = f(f(z))$

$$v_1 = z$$

ARGOLIB DECISION PROCEDURES

Nelson-Oppenova shema - primer

$$z = f(x - y) \land x = y + z \land x \neq f(f(z)) + y$$

PRA	EUF
$z = x - y$	$z = f(z)$
$x = y + z$	
$x \neq v_2 + y$	$v_2 = f(f(z))$

$$v_1 = z$$

Nelson-Oppenova shema - primer

$$z = f(x - y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

PRA	EUF
$z = x - y$	$z = f(z)$
$x = y + z$	
$x \neq v_2 + y$	$v_2 = f(f(z))$

$$v_1 = z$$

$$v_2 = z$$

Nelson-Oppenova shema - primer

$$z = f(x - y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

PRA	EUF
$z = x - y$	$z = f(z)$
$x = y + z$	
$x \neq v_2 + y$	$v_2 = f(f(z))$

$$v_1 = z$$

$$v_2 = z$$

Nelson-Oppenova shema - primer

$$z = f(x - y) \land x = y + z \land x \neq f(f(z)) + y$$

PRA	EUF
$z = x - y$	$z = f(z)$
$x = y + z$	
$x \neq z + y$	$z = f(f(z))$

$$v_1 = z$$

$$v_2 = z$$

Nelson-Oppenova shema - primer

$$z = f(x - y) \wedge x = y + z \wedge x \neq f(f(z)) + y$$

PRA	EUF
$z = x - y$	$z = f(z)$
$x = y + z$	
$x \neq z + y$	$v_2 = f(f(z))$

$$v_1 = z$$

$$v_2 = z$$

$$\perp$$

Zaključci

- Verujemo da smo izgradili platformu koja u potpunosti zadovoljava sve ciljeve koje smo na početku zacrtali

Zaključci

- **Verujemo da smo izgradili platformu koja u potpunosti zadovoljava sve ciljeve koje smo na početku zacrtali**
- Efikasnost softvera prevazilazi za više od reda veličine efikasnost ranije PROLOG implementacije.

Zaključci

- **Verujemo da smo izgradili platformu koja u potpunosti zadovoljava sve ciljeve koje smo na početku zacrtali**
- Efikasnost softvera prevazilazi za više od reda veličine efikasnost ranije PROLOG implementacije.
- Na testiranim instancama, efikasnost samih procedura odlučivanja je uporediva sa relevantnim sistemima CVC, TECTON, . . .

Zaključci

- **Verujemo da smo izgradili platformu koja u potpunosti zadovoljava sve ciljeve koje smo na početku zacrtali**
- Efikasnost softvera prevazilazi za više od reda veličine efikasnost ranije PROLOG implementacije.
- Na testiranim instancama, efikasnost samih procedura odlučivanja je uporediva sa relevantnim sistemima CVC, TECTON, . . .
- Sistem je u potpunosti spreman za kombinovanje sa sistemima koji vrše iskazno rezonovanje (SAT). Dokazivači zasnovani na ovakvom kombinovanom pristupu daju ubedljivo najbolje rezultate u poslednje vreme.

Dalji rad

- Uz pomoć platforme, su već vršeni eksperimenti na polju automatskog učenja procedura odlučivanja. Planiramo da u ovom smeru i nastavimo.

Dalji rad

- Uz pomoć platforme, su već vršeni eksperimenti na polju automatskog učenja procedura odlučivanja. Planiramo da u ovom smeru i nastavimo.
- Planirano je kombinovanje sa sistemom za iskazno rezonovanje TSAT++

Dalji rad

- Uz pomoć platforme, su već vršeni eksperimenti na polju automatskog učenja procedura odlučivanja. Planiramo da u ovom smeru i nastavimo.
- Planirano je kombinovanje sa sistemom za iskazno rezonovanje TSAT++
- Planirana je implementacija procedure odlučivanja za nove teorije, kao i prilagodjavanje postojećih procedura zahtevima novih, efikasnijih, algoritama kombinovanja sa iskaznim rezonovanjem.

Objavljeni radovi

- ARGO-LIB: *A Generic Platform for Decision Procedures*, F. Marić, P. Janičić, IJCAR-04, Lecture Notes in Artificial Intelligence, 3097, Springer, 2004.

- SMT-LIB *in* XML *clothes*, F. Marić, P. Janičić, Proceedings of the 2nd Workshop on Pragmatics of Decision Procedures in Automated Reasoning (PDPAR 2004), Dublin, July 2004.