A Resolution Decision Procedure for the Guarded Fragment With Transitive Guards

Yevgeny Kazakov
(joint work with Hans de Nivelle)

Objectives

- How to find a decision procedure for a non-trivial fragment of first-order logic?
- How to specify a decision procedure and proof its correctness?

What Is the Guarded Fragment?

- The guarded fragment (Andréka, van Benthem & Németi,1996):
  - Quantifiers should be bounded:

$$\forall\overline{x}.(G\rightarrow F) \qquad \exists\overline{x}.(G\wedge F).$$

- *where G is an atom-**guard** containing all variables of F*

What Is the Guarded Fragment?

- The guarded fragment (Andréka, van Benthem & Németi,1996):
  - Quantifiers should be bounded:

$$\forall\overline{x}.(G\rightarrow F) \qquad \exists\overline{x}.(G\wedge F).$$

- *where G is an atom-**guard** containing all variables of F*

Example:

- Guarded formula:

  Seriality $\equiv \forall x.(\mathrm{V}(x)\rightarrow\exists y.[\mathrm{Edge}(x,y)\wedge V(y)])$

- Non-guarded formula:

  Transitivity $\equiv \forall xyz.[T(x,y)\wedge T(y,z)\rightarrow T(x,z)]$

INFORMATIK

Properties of Guarded Formulas

- ***GF*** is related to many modal-like logics:

$$
\begin{aligned}
\mathcal{ALC} ::= A \mid \\
C_1 \sqcap C_2 \mid \\
C_1 \sqcup C_2 \mid \\
\neg C \mid \\
\forall R.C \mid \\
\exists R.C.
\end{aligned}
$$

INFORMATIK

Properties of Guarded Formulas

- ***GF*** is related to many modal-like logics:

$$
\begin{array}{rcl}
\mathcal{ALC} ::= A & \mid A(x) & =:: \mathcal{FO}[\mathcal{ALC}] \\
C_1 \sqcap C_2 & \mid C_1(x) \wedge C_2(x) & \\
C_1 \sqcup C_2 & \mid C_1(x) \vee C_2(x) & \\
\neg C & \mid \neg C(x) & \\
\forall R.C & \mid \forall y.(R(x,y) \rightarrow C(y)) & \\
\exists R.C. & \exists y.(R(x,y) \wedge C(y)). &
\end{array}
$$

INFORMATIK

Properties of Guarded Formulas

- ***GF*** is related to many modal-like logics:

$$
\begin{array}{rcl}
\mathcal{ALC} ::= A \mid & & \mid A(x) \quad =:: \mathcal{FO}[\mathcal{ALC}] \\
C_1 \sqcap C_2 \mid & & \mid C_1(x) \wedge C_2(x) \\
C_1 \sqcup C_2 \mid & \Rightarrow & \mid C_1(x) \vee C_2(x) \\
\neg C \mid & & \mid \neg C(x) \\
\forall R.C \mid & & \mid \forall y.(R(x,y) \rightarrow C(y)) \\
\exists R.C. & & \exists y.(R(x,y) \wedge C(y)).
\end{array}
$$

- ***GF*** has nice computational properties:
  - A tree-model property,
  - A small model property,
  - Decidability

INFORMATIK

The Guarded Fragment and Transitivity

$$\text{Transitivity} \equiv \forall xyz.[\underline{T(x,y)} \wedge \underline{T(y,z)} \rightarrow T(x,z)]$$

INFORMATIK

The Guarded Fragment and Transitivity

$\text{Transitivity} \equiv \forall xyz.[T(x,y) \wedge T(y,z) \rightarrow T(x,z)]$

- Adding transitivity ruins nice properties of ***GF***:
  - [Grädel,1999]: ***GF³*** *with two transitive predicates is undecidable;*
  - [Ganzinger, Meyer, Veanes,1999]: ***GF²[T]*** *is undecidable.*

The Guarded Fragment and Transitivity

$\text{Transitivity} \equiv \forall xyz.[T(x,y) \wedge T(y,z) \rightarrow T(x,z)]$

- Adding transitivity ruins nice properties of ***GF***:
  - [Grädel,1999]: ***GF³*** *with two transitive predicates is undecidable;*
  - [Ganzinger, Meyer, Veanes,1999]: ***GF²[T]*** *is undecidable.*
- Decidable extensions with transitivity:
  + [Ganzinger at al,1999]: ***monadic-GF²[T]*** is decidable;
  + [Szwast,Tendera,2001]: ***GF[TG]*** is in 2EXPTIME;
  + [Kierionski,2002,2003]: ***monadic-GF²[T]*** is 2EXPTIME-hard.

GF With Transitive Guards

- The guarded fragment with transitive guards ***GF[TG]*** :

$$\forall \overline{x}.(G \rightarrow F) \quad \exists \overline{x}.(G \wedge F).$$

- *where G is an atom-**guard** containing all variables of F*

GF With Transitive Guards

- The guarded fragment with transitive guards ***GF[TG]*** :

$$\forall \overline{x}.(G \rightarrow F) \qquad \exists \overline{x}.(G \wedge F).$$

- *where* G *is an atom-**guard** containing all variables of* F
- *Transitive predicates may occur only as guards.*

GF With Transitive Guards

- The guarded fragment with transitive guards ***GF[TG]*** :

$$\forall\overline{x}.(G \rightarrow F) \qquad \exists\overline{x}.(G \wedge F).$$

- *where G is an atom-**guard** containing all variables of F*
- *Transitive predicates may occur only as guards.*

Example:

- ***GF[TG]*** can express orderings without endpoints:

$$\text{NoEnd} \equiv \forall xy.(\underline{x<y} \rightarrow \exists z.[\underline{y<z}])$$

- ***GF[TG]*** cannot express dense orderings:

$$\text{Density} \equiv \forall xy.(\underline{x < y} \rightarrow \exists z.[\underline{x < z} \wedge \underline{z < y}])$$

INFORMATIK

Two approaches

Decision Procedures for FO-fragments

Two approaches

Model-theoretic
(search for a model)

Proof-theoretic
(Search for a proof)

Decision Procedures for FO-fragments

Two approaches

Model-theoretic
(search for a model)

Proof-theoretic
(Search for a proof)

+ Highly efficient implementations

– Rely on a "good model" property

? Formalization (soundness,completeness, proof/model checking)

Decision Procedures for FO-fragments

Two approaches

Model-theoretic
(search for a model)

+ Highly efficient implementations
– Rely on a "good model" property
? Formalization (soundness,completeness, proof/model checking)

Proof-theoretic
(Search for a proof)

? Highly optimized implementations
+ soundness/completeness is guaranteed
! Correctness is reduced to termination

Resolution Decides *GF*

- [de Nivelle, 1998] *Resolution decides* ***GF*** *without equality.* ***HOW to formalize?***

Resolution Decides *GF*

- [de Nivelle, 1998] *Resolution decides* ***GF*** *without equality.* ***HOW to formalize?***
- Clauses resulted for ***GF*** can be described by the clause schemes:

1. $\widehat{\beta}$ ← Propositional
2. $\neg!\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}), \overline{x}]$ ← **Guarded**

Resolution Decides *GF*

- [de Nivelle, 1998] *Resolution decides* ***GF*** *without equality.* ***HOW to formalize?***
- Clauses resulted for ***GF*** can be described by the clause schemes:

1. $\widehat{\beta}$ ← Propositional
2. $\neg!\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}), \overline{x}]$ ← **Guarded**

Guarded: $\neg a(x, y, x) \vee b(y, f'(x, y));$
$\neg b(x, y) \vee \neg b(y, x);$
$\neg p \vee \neg q(c, c)$

mpi INFORMATIK

Resolution Decides *GF*

- [de Nivelle, 1998] *Resolution decides* ***GF*** *without equality.* ***HOW to formalize?***
- Clauses resulted for ***GF*** can be described by the clause schemes:

1. $\widehat{\beta}$ ← Propositional
2. $\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}), \overline{x}]$ ← **Guarded**

Guarded: $\neg a(x, y, x) \vee b(y, f'(x, y));$

$\neg b(x, y) \vee \neg b(y, x);$

$\neg p \vee \neg q(c, c)$

Resolution Decides *GF*

- [de Nivelle, 1998] *Resolution decides **GF** without equality.* ***HOW to formalize?***
- Clauses resulted for ***GF*** can be described by the clause schemes:

1. $\widehat{\beta}$ ← Propositional
2. $\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}), \overline{x}]$ ← **Guarded**

Non-Guarded:

$\neg a(x, y, x) \vee b(f'(x, y), f'(y, x))$

$\neg b(y, f'(x, y))$

Saturation of the Clause Set

The Ordered Resolution Calculus:

$$\text{OR}: \frac{C \vee \underline{A}^* \quad D \vee \neg \underline{B}^*}{C\sigma \vee D\sigma}$$

$$\text{OF}: \frac{C \vee \underline{A}^* \vee \underline{B}}{C\sigma \vee A\sigma}$$

Saturation of the Clause Set

The Ordered Resolution Calculus:

$$\text{OR}: \frac{C \vee \underline{A}^* \qquad D \vee \neg \underline{B}^*}{C\sigma \vee D\sigma} \qquad \text{OF}: \frac{C \vee \underline{A}^* \vee \underline{B}}{C\sigma \vee A\sigma}$$

1 $\hat{\beta} \vee \beta^*$	2 $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}]$
1.1 $\hat{\beta} \vee \underline{b}^*$:OR	2.1 $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \beta[!f(\bar{x}), \bar{x}]^*$
1.2 $\hat{\beta} \vee \neg \underline{b}^*$:OR	2.1.1 $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \underline{b[!f(\bar{x}), \bar{x}]}^*$:OR
1.3 $\hat{\beta} \vee \underline{b_1}^* \vee \underline{b_2}$:OF	2.1.2 $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \neg \underline{b[!f(\bar{x}), \bar{x}]}^*$:OR
OR[1.1;1.2]: $\hat{\beta}$:1	2.1.3 $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \underline{b[!f(\bar{x}), \bar{x}]}^* \vee \underline{b[f(\bar{x}), \bar{x}]}$:OF
OF[1.3] : $\hat{\beta} \vee b_1$:1	OR[2.1.1;2.1.2]: $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}]$:2
	OF[2.1.3] : $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee b[!f(\bar{x}), \bar{x}]$:2
	2.2 $\neg \underline{g[!\bar{x}]}^{\#} \vee \neg \hat{g}[!\bar{x}] \vee \hat{\beta}[\bar{x}]$:OR
	OR[1.1;2.2] : $\hat{\beta}$:1
	OR[2.1.1;2.2]: $\neg !\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}]$:2

Saturation of the Clause Set

The Ordered Resolution Calculus:

$$\text{OR}: \frac{C \vee \underline{A}^* \qquad D \vee \neg \underline{B}^*}{C\sigma \vee D\sigma} \qquad\qquad \text{OF}: \frac{C \vee \underline{A}^* \vee \underline{B}}{C\sigma \vee A\sigma}$$

1 $\hat{\beta} \vee \beta^*$	2 $\neg!\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}]$
1.1 $\hat{\beta} \vee \underline{b}^*$:OR	2.1 $\neg!\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \beta[!f(\bar{x}), \bar{x}]^*$
1.2 $\hat{\beta} \vee \neg \underline{b}^*$:OR	2.1.1 $\neg!\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \underline{b[!f(\bar{x}), \bar{x}]}^*$:OR
1.3 $\hat{\beta} \vee \underline{b_1}^* \vee \underline{b_2}$:OF	2.1.2 $\neg!\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \neg \underline{b[!f(\bar{x}), \bar{x}]}^*$:OR
OR[1.1;1.2]: $\hat{\beta}$:1	2.1.3 $\neg!\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}] \vee \underline{b[!f(\bar{x}), \bar{x}]}^* \vee \underline{b[f(\bar{x}), \bar{x}]}$:OF
OF[1.3] : $\hat{\beta} \vee b_1$:1	OR[2.1.1;2.1.2]: $\neg!\hat{g}[!\bar{x}] \vee \hat{\beta}[f(\bar{x}), \bar{x}]$:2
	… $, \bar{x}]$:2
	:1
	OR[2.1.1;2.2]: $\neg!\hat{g}[!\bar{x}] \vee \beta[f(\bar{x}), \bar{x}] \vee \beta[f(\bar{x}), \bar{x}]$:2

Resolution generates finitely many clauses for every input!

Why Resolution Terminates for ***GF*** ?

2.1.1 $\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}), \overline{x}] \vee \underline{b[!f(\overline{x}), \overline{x}]^{*}}$: OR

2.2 $\neg \underline{g[!\overline{x}]}^{\#} \vee \neg \widehat{g}[!\overline{x}] \vee \widehat{\beta}[\overline{x}]$: OR

OR[2.1.1;2.2]: $\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}), \overline{x}] \vee \widehat{\beta}[f(\overline{x}), \overline{x}]$:2

Why Resolution Terminates for *GF* ?

2.1.1 $\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}),\overline{x}] \vee \underline{b[!f(\overline{x}),\overline{x}]}^{*}$: OR

2.2 $\underline{\neg g[!\overline{x}]}^{\#} \vee \neg \widehat{g}[!\overline{x}] \vee \widehat{\beta}[\overline{x}]$: OR

OR[2.1.1;2.2]: $\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[f(\overline{x}),\overline{x}] \vee \widehat{\beta}[f(\overline{x}),\overline{x}]$: 2

1. Unified expressions contain all variables;
   - Number of variables does not grow.
2. Every variable occurs in a deepest position
   - Clause depth does not grow.

Resolution With Transitivity Axioms

- Resolution with transitivity axioms may produce larger clauses:

1. $\neg(xTy) \vee \neg(yTz) \vee \underline{xTz}^{*}$;
2. $\neg(\underline{xTz})^{*} \vee \neg(zTu) \vee xTu$;

Resolution With Transitivity Axioms

- Resolution with transitivity axioms may produce larger clauses:

1. $\neg(xTy) \vee \neg(yTz) \vee \underline{xTz}^*;$
2. $\neg\underline{(xTz)}^* \vee \neg(zTu) \vee xTu;$

OR[1;2]: 3. $\neg(xTy) \vee \neg(yTz) \vee \neg(zTu) \vee xTu;$

MPI INFORMATIK

Resolution With Transitivity Axioms

- Resolution with transitivity axioms may produce larger clauses:

1. $\neg(xTy) \vee \neg(yTz) \vee \underline{xTz}^{*}$;
2. $\neg(\underline{xTz})^{*} \vee \neg(zTu) \vee xTu$;

OR[1;2]: 3. $\neg(xTy) \vee \neg(yTz) \vee \neg(zTu) \vee xTu$;

- Solution: use a selection function:

1. $\neg(\underline{xTy})^{\#} \vee \neg(yTz) \vee xTz$;
2. $\neg(xTy) \vee \neg(\underline{yTz})^{\#} \vee xTz$;

OR[1;2]: —

INFORMATIK

Resolution With Transitivity Axioms

- Selection does not help avoiding increase of variables in clauses:

1. $\neg\underline{(xTy)}^{\#} \vee \neg(yTz) \vee xTz;$
2. $\alpha(x) \vee \underline{f(x)Tx^{*}};$

OR[2;1]: 3. $\alpha(x) \vee \neg(xTz) \vee \underline{f(x)Tz^{*}};$

OR[3;1]: 4. $\alpha(x) \vee \neg(xTz) \vee \neg(zTz_1) \vee \underline{f(x)Tz_1^{*}};$

Resolution With Transitivity Axioms

- Selection does not help avoiding increase of variables in clauses:

1. $\neg(\underline{xTy})^{\#} \vee \neg(yTz) \vee xTz;$
2. $\alpha(x) \vee \underline{f(x)Tx}^{*};$

OR[2;1]: 3. $\alpha(x) \vee \neg(xTz) \vee \underline{f(x)Tz}^{*};$

OR[3;1]: 4. $\alpha(x) \vee \neg(xTz) \vee \neg(zTz_1) \vee \underline{f(x)Tz_1}^{*};$

- Or increase of functional depth:

1. $\neg(\underline{xTy})^{\#} \vee \neg(\underline{yTz})^{\#} \vee xTz;$
2. $\alpha(x) \vee \underline{f(x)Tx}^{*};$

HR[2,2;1]: 3. $\alpha(x) \vee \underline{ff(x)Tx}^{*};$

HR[3,2;1]: 4. $\alpha(x) \vee \underline{fff(x)Tx}^{*};$

Resolution With Transitivity Axioms

- Selection does not help avoiding increase of v…

1. $\neg(\underline{xTy})$…
2. $\alpha(x) \vee f$…

OR[2;1]: 3. α…

OR[3;1]: 4. α…

- Harmless situations:

1. $\neg(\underline{yTz})^{\#} \vee \neg(\underline{yTz})^{\#} \vee xTz;$
2. $\alpha(x) \vee \underline{f(x)Tx^{*}};$
3. $\alpha'(x) \vee \underline{xTx^{*}};$

HR[2,3;1]: 4. $\alpha(x) \vee \alpha'(x) \vee \underline{f(x)Tx^{*}};$

- Or increase of functional depth:

1. $\neg(\underline{xTy})^{\#} \vee \neg(\underline{yTz})^{\#} \vee xTz;$
2. $\alpha(x) \vee \underline{f(x)Tx^{*}};$

HR[2,2;1]: 3. $\alpha(x) \vee \underline{ff(x)Tx^{*}};$

HR[3,2;1]: 4. $\alpha(x) \vee \underline{fff(x)Tx^{*}};$

Constraint Clauses

- "Smart" selection strategies can be realized through constraint clauses (~ Chaining calculus):

$$
\begin{array}{lll}
T.1 & \neg(\underline{xTy})^{\#} \vee \neg(yTz) \vee xTz & \mid x \succ max(y,z) \\
T.2 & \neg(xTy) \vee \neg(\underline{yTz})^{\#} \vee xTz & \mid z \succ max(y,x) \\
T.3 & \neg(\underline{xTy})^{\#} \vee \neg(\underline{yTz})^{\#} \vee xTz & \mid \text{otherwise}
\end{array}
$$

MPI INFORMATIK

Constraint Clauses

- "Smart" selection strategies can be realized through constraint clauses (~ Chaining calculus):

$$T.1 \quad \neg\underline{(xTy)}^{\#} \vee \neg(yTz) \vee xTz \quad \mid x \succ max(y,z)$$

$$T.2 \quad \neg(xTy) \vee \neg\underline{(yTz)}^{\#} \vee xTz \quad \mid z \succ max(y,x)$$

$$T.3 \quad \neg\underline{(xTy)}^{\#} \vee \neg\underline{(yTz)}^{\#} \vee xTz \quad \mid \text{otherwise}$$

- Saturation with constraint clauses:

1. $\neg\underline{(xTy)}^{*} \vee \neg(yTz) \vee xTz; \quad \mid x \succ max(y,z)$
2. $\alpha(x) \vee \underline{f(x)Tx}^{*};$

OR[2; 1]: 3. $\alpha(x) \vee \neg(xTz) \vee \underline{f(x)Tz}^{*}; \quad \mid f(x) \succ z$

OR[3; 1]: 4. $\alpha(x) \vee \neg(xTz) \vee \neg(zTz_1) \vee \underline{f(x)Tz_1}^{*}; \quad \mid f(x) \succ max(z, z_1)$

:

Constraint Clauses

- "Smart" selection strategies can be realized through constraint clauses (~ Chaining calculus):

$$T.1 \quad \neg\underline{(xTy)}^{\#} \vee \neg(yTz) \vee xTz \;\; | \; x \succ max(y,z)$$
$$T.2 \quad \neg(xTy) \vee \neg\underline{(yTz)}^{\#} \vee xTz \;\; | \; z \succ max(y,x)$$
$$T.3 \quad \neg\underline{(xTy)}^{\#} \vee \neg\underline{(yTz)}^{\#} \vee xTz \;\; | \; \text{otherwise}$$

- Saturation with constraint clauses:

1. $\neg\underline{(xTy)}^{*} \vee \neg(yTz) \vee xTz; \;\; | \; x \succ max(y,z)$
2. $\alpha(x) \vee \underline{f(x)Tx}^{*};$

OR[2; 1]: 3. $\alpha(x) \vee \neg(xTz) \vee \underline{f(x)Tz}^{*}; \;\; | \; f(x) \succ z$

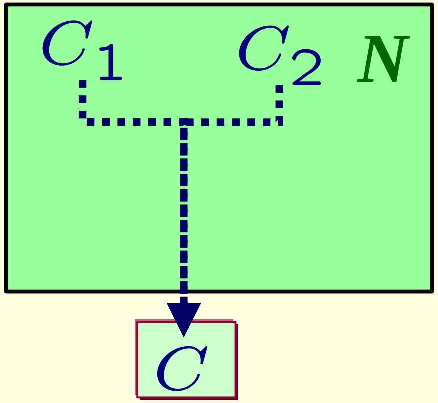
~~OR[3; 1]: 4. $\alpha(x) \vee \neg(xTz) \vee \neg(zTz_1) \vee \underline{f(x)Tz_1}^{*}; \;\; | \; f(x) \succ max(z, z_1)$~~

$\vdots$

MPI INFORMATIK

Redundancy of Inferences

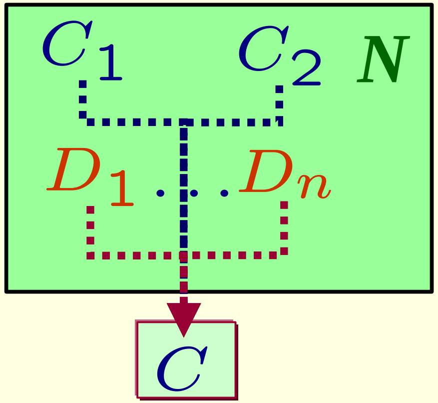
- Abstract notion of redundancy [Bachmair,Ganzinger,1990]:



- An inference $C_1, C_2 \vdash C$ is redundant in N if $N_{\prec \max(C_1, C_2)} \vdash C$

Redundancy of Inferences

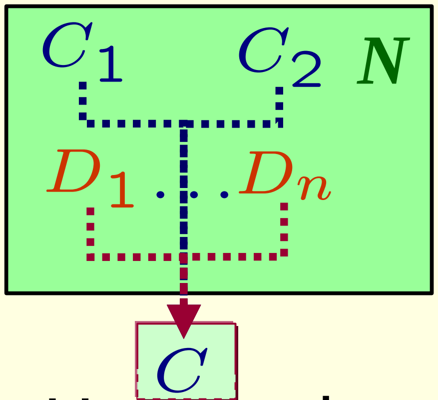
- Abstract notion of redundancy [Bachmair,Ganzinger,1990]:



- An inference $C_1, C_2 \vdash C$ is redundant in N if $N_{\prec \max(C_1, C_2)} \vdash C$

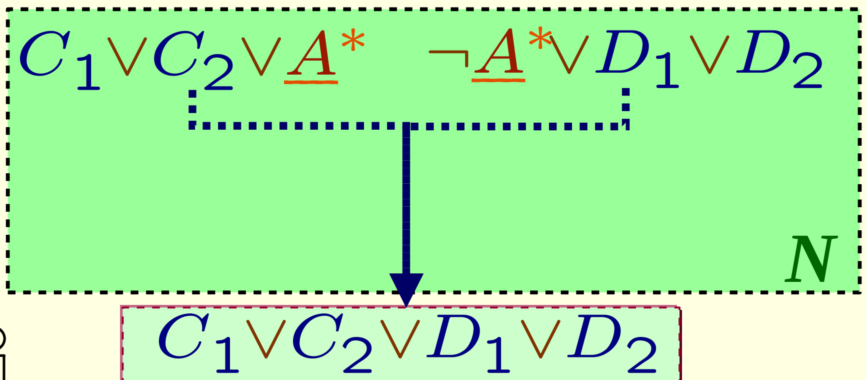
Redundancy of Inferences

- Abstract notion of redundancy [Bachmair,Ganzinger,1990]:



- An inference $C_1, C_2 \vdash C$ is redundant in N if $N_{\prec \max(C_1,C_2)} \vdash C$

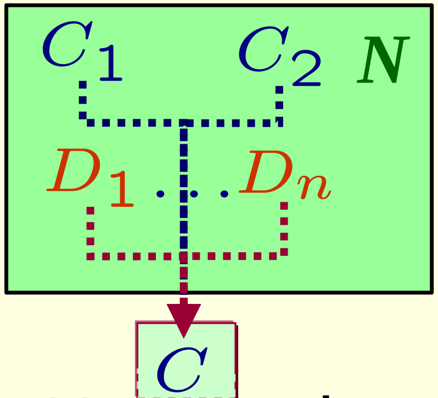
- How to show that inference is redundant?



A resolution inference is redundant in N

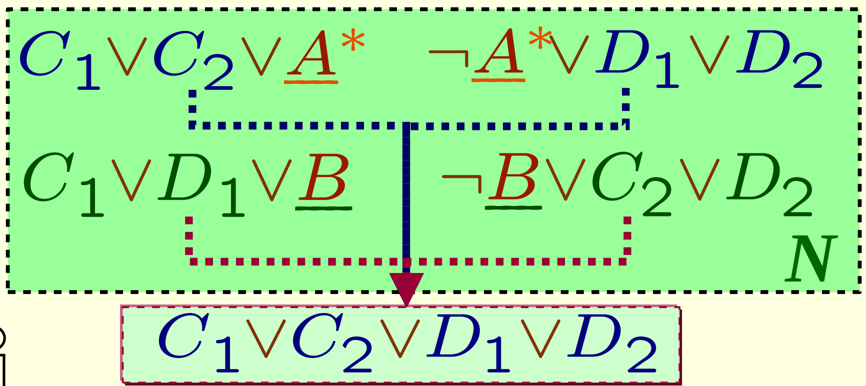
Redundancy of Inferences

- Abstract notion of redundancy [Bachmair,Ganzinger,1990]:



- An inference $C_1,C_2 \vdash C$ is **redundant** in N if $N_{\prec \max(C_1,C_2)} \vdash C$

- How to show that inference is redundant?



A resolution inference is **redundant** in N if $A \succ B$

Redundancy in practice

- The clause 4 can be obtained differently by resolving on smaller literals:

$$
\begin{array}{ll}
1. & \neg\underline{(xTy)}^{*} \vee \neg(yTz) \vee xTz; \mid x \succ max(y,z) \\
2. & \alpha(c) \vee \underline{f(c)Tc}^{*}; \\
\textsc{or}[2;1]: 3. & \alpha(c) \vee \neg(cTz) \vee \underline{f(c)Tz}^{*}; \mid f(a) \succ z \\
\textsc{or}[3;1]: 4. & \alpha(c) \vee \neg(cTz) \vee \neg(zTz_1) \vee \underline{f(c)Tz_1}^{*}; \\
\vdots & \mid f(c) \succ max(z,z_1)
\end{array}
$$

MPI INFORMATIK

Redundancy in practice

- The clause 4 can be obtained differently by resolving on smaller literals:

$$
\begin{array}{ll}
1. & \neg(xTy)^{*} \vee \neg(yTz) \vee xTz; \;\mid\; x \succ max(y,z) \\
2. & \alpha(c) \vee f(c)Tc^{*}; \\
\text{OR}[2;1]: 3. & \alpha(c) \vee \neg(cTz) \vee f(c)Tz^{*}; \;\mid\; f(a) \succ z \\
\text{OR}[3;1]: 4. & \alpha(c) \vee \neg(cTz) \vee \neg(zTz_1) \vee f(c)Tz_1{}^{*}; \\
\vdots & \mid\; f(c) \succ max(z,z_1)
\end{array}
$$

$$
\begin{array}{ll}
T. & \neg(xTz) \vee \neg(zTz_1) \vee xTz_1; \\
\text{OR}[2;1]: 3. & \alpha(c) \vee \neg(cTz_1) \vee f(c)Tz; \;\mid\; f(c) \succ z_1 \\
\text{OR}[3;1]: 4. & \alpha(c) \vee \neg(cTz) \vee \neg(zTz_1) \vee f(c)Tz_1; \\
\vdots & \mid\; f(c) \succ z_1
\end{array}
$$

MPI INFORMATIK

Redundancy in practice

- The clause 4 can be obtained differently by resolving on smaller literals:

1. $\neg(xTy)^{*} \vee \neg(yTz) \vee xTz; \mid x \succ max(y,z)$
2. $\alpha(c) \vee f(c)Tc^{*};$

OR[2; 1]: 3. $\alpha(c) \vee \neg(cTz) \vee f(c)Tz^{*}; \mid f(a) \succ z$

OR[3; 1]: 4. $\alpha(c) \vee \neg(cTz) \vee \neg(zTz_1) \vee f(c)Tz_1{}^{*};$

$\vdots$ $f(c)Tz \succ cTz_1$ $\mid f(c) \succ max(z, z_1)$

T. $\neg(xTz) \vee \neg(zTz_1) \vee xTz_1;$

OR[2;1]: 3. $\alpha(c) \vee \neg(cTz_1) \vee f(c)Tz; \mid f(c) \succ z_1$

OR[3;1]: 4. $\alpha(c) \vee \neg(cTz) \vee \neg(zTz_1) \vee f(c)Tz_1;$

$\vdots$ $\mid f(c) \succ z_1$

Redundancy in practice

- The clause 4 can be obtained differently by resolving on smaller literals:

1. $\neg\underline{(xTy)}^{*} \vee \neg(yTz) \vee xTz; \mid x \succ max(y,z)$
2. $\alpha(c) \vee \underline{f(c)Tc}^{*};$

OR[2; 1]: 3. $\alpha(c) \vee \neg(cTz) \vee \underline{f(c)Tz}^{*}; \mid f(a) \succ z$

~~OR[3; 1]: 4. $\alpha(c) \vee \neg(cTz) \vee \neg(zTz_1) \vee \underline{f(c)Tz_1}^{*};$~~

$\vdots$ $f(c)Tz \succ cTz_1$ ~~$\mid f(c) \succ max(z, z_1)$~~

$T.$ $\neg(xTz) \vee \neg(zTz_1) \vee \underline{xTz_1};$

OR[2;1]: 3. $\alpha(c) \vee \neg(\underline{cTz_1}) \vee f(c)Tz; \mid f(c) \succ z_1$

OR[3;1]: 4. $\alpha(c) \vee \neg(cTz) \vee \neg(zTz_1) \vee f(c)Tz_1;$

$\vdots$ $\mid f(c) \succ z_1$

More Troublesome Inferences

- Resolving negative occurrences of transitive predicates may yield problems:

1. $\alpha(x) \vee \underline{f(x)Tx}^{*}$;
2. $\neg \underline{(xTy)}^{*} \vee a(x) \vee \beta(y)$;

OR[1;T.1]: 3. $\alpha(x) \vee \neg(xTz) \vee \underline{f(x)Tz}^{*}$;

OR[2;3] : 4. $\alpha(x) \vee \neg(xTz) \vee \underline{a(f(x))}^{*} \vee \beta(z)$;

- The variable *z* does not occur in a deepest position.
- How to make the inference OR[5;2] redundant?

Auxiliary Inference Rule

- Add a sound inference rule:

Transitive Recursion

$$TR: \frac{\neg(x\hat{T}y)^{*} \vee \alpha[x] \vee \beta[y]}{\begin{array}{r}\neg(x\hat{T}y) \vee \alpha[x] \vee u(y) \\ \neg(x\hat{T}y) \vee \neg u(x) \vee u(y) \\ \neg u(y) \vee \beta[y]\end{array}}$$

1. $\alpha(c) \vee f(c)Tc^{*}$;
2. $\neg(xTy)^{*} \vee a(x) \vee \beta(y)$;

OR$[1; T.1]$: 3. $\alpha(c) \vee \neg(cTz) \vee f(c)Tz^{*}$;

OR$[3; 2]$: 4. $\alpha(c) \vee \neg(cTz) \vee a(f(c)) \vee \beta(z)$;

Auxiliary Inference Rule

- Add a sound inference rule:

Transitive Recursion

$$TR: \frac{\neg(x\hat{T}y)^* \vee \alpha[x] \vee \beta[y]}{\begin{array}{r}\neg(x\hat{T}y) \vee \alpha[x] \vee u(y) \\ \neg(x\hat{T}y) \vee \neg u(x) \vee u(y) \\ \neg u(y) \vee \beta[y]\end{array}}$$

1. $\alpha(c) \vee f(c)Tc^*$;
2. $\neg(xTy)^* \vee a(x) \vee \beta(y)$;

OR[1; T.1]: 3. $\alpha(c) \vee \neg(cTz) \vee f(c)Tz^*$;

OR[3; 2] : 4. $\alpha(c) \vee \neg(cTz) \vee a(f(c)) \vee \beta(z)$;

TR[2] : 5. $\neg(xTy)^* \vee a(x) \vee u(y)$;

: 6. $\neg(xTz)^* \vee \neg u(x) \vee u(z)$;

OR[1; 5]: 7. $\alpha(c) \vee a(f(c)) \vee u(c)$;

Auxiliary Inference Rule

- Add a sound inference rule:

Transitive Recursion

$$TR: \frac{\neg(x\hat{T}y)^{*} \vee \alpha[x] \vee \beta[y]}{\begin{array}{r}\neg(x\hat{T}y) \vee \alpha[x] \vee u(y) \\ \neg(x\hat{T}y) \vee \neg u(x) \vee u(y) \\ \neg u(y) \vee \beta[y]\end{array}}$$

1. $\alpha(c) \vee f(c)Tc^{*}$;
2. $\neg(xTy)^{*} \vee a(x) \vee \beta(y)$;

OR[1;T.1]: 3. $\alpha(c) \vee \neg(cTz) \vee f(c)Tz^{*}$;

OR[3; 2] : 4. $\alpha(c) \vee \neg(cTz) \vee a(f(c)) \vee \beta(z)$;

TR[2] : 5. $\neg(xTy)^{*} \vee a(x) \vee u($

$f(c)Tz \succ u(c)$

: 6. $\neg(xTz)^{*} \vee \neg u(x) \vee u(z)$;

OR[1;5]: 7. $\alpha(c) \vee a(f(c)) \vee u(c)$;

INFORMATIK

Auxiliary Inference Rule

- Add a sound inference rule:

Transitive Recursion

$$TR: \frac{\neg(x\hat{T}y)^* \vee \alpha[x] \vee \beta[y]}{\begin{array}{l}\neg(x\hat{T}y) \vee \alpha[x] \vee u(y) \\ \neg(x\hat{T}y) \vee \neg u(x) \vee u(y) \\ \neg u(y) \vee \beta[y]\end{array}}$$

1. $\alpha(c) \vee f(c)Tc^*$;
2. $\neg(xTy)^* \vee a(x) \vee \beta(y)$;

OR[1; T.1]: 3. $\alpha(c) \vee \neg(cTz) \vee f(c)Tz^*$;

~~OR[3; 2] : 4. $\alpha(c) \vee \neg(cTz) \vee a(f(c)) \vee \beta(z)$;~~

TR[2] : 5. $\neg(xTy)^* \vee a(x) \vee u(y)$;

: 6. $\neg(xTz)^* \vee \neg u(x) \vee u(z)$;

OR[1; 5]: 7. $\alpha(c) \vee a(f(c)) \vee u(c)$;

$f(c)Tz \succ u(c)$

MPI INFORMATIK

Extended Guarded Clauses

- Extended guarded clauses for the ***GF[TG]***:

$$\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[\overline{x}] \vee \gamma[!\overline{x}] \vee$$
$$\vee \neg \widehat{T}[!f(\overline{x}), \overline{x}] \vee \gamma[!f(\overline{x}), \overline{x}]$$

MPI INFORMATIK

Extended Guarded Clauses

- Extended guarded clauses for the ***GF[TG]***:

$$\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[\overline{x}] \vee \gamma[!\overline{x}] \vee$$
$$\vee \neg \widehat{T}[!f(\overline{x}), \overline{x}] \vee \gamma[!f(\overline{x}), \overline{x}]$$

- The fragment is closed under inference rules of ordered resolution:
  - using constraints and redundancy elimination;
  - all cases can be schematically described;

Extended Guarded Clauses

- Extended guarded clauses for the ***GF[TG]***:

$$\neg !\widehat{g}[!\overline{x}] \vee \widehat{\beta}[\overline{x}] \vee \gamma[!\overline{x}] \vee$$
$$\vee \neg \widehat{T}[!f(\overline{x}), \overline{x}] \vee \gamma[!f(\overline{x}), \overline{x}]$$

- The fragment is closed under inference rules of ordered resolution:
  - using constraints and redundancy elimination;
  - all cases can be schematically described;
- procedure has an optimal complexity and captures the complexities of simpler sub-fragments (***S,SHI,SHIb***).

INFORMATIK

Conclusions

- A decision procedure for ***GF[TG]*** is given which make use of advanced refinements of the resolution calculus;
- The procedure has an optimal complexity and scalable to sub-fragments;
- Surprisingly: the clause class captures even a larger fragment than ***GF[TG]*** : it allows the conjunction of transitive relations as guards.
- Current work: extend to the case with equality (integrating the chaining calculus), compositional binary relations, theories of linear and branching orderings.

Thank You!

Thank you !

INFORMATIK