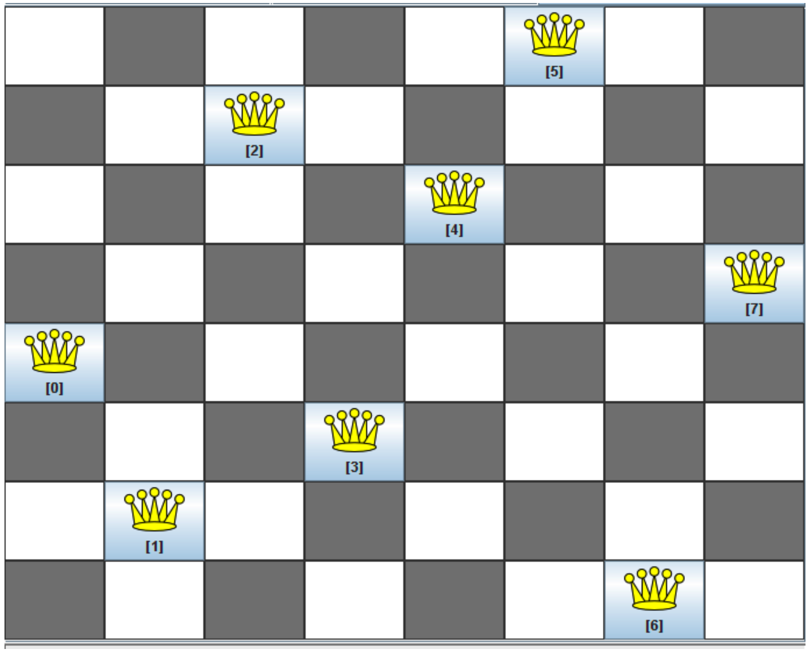
Model Evolution with Equality Modulo Built-In Theories

Peter Baumgartner
NICTA and ANU

Cesare Tinelli
The University of Iowa

Motivating Example: Analysing N-Queens



Task

Prove that

for **all** board sizes n:

if s is a solution, so is mirrored(s)

(PUZ133+2.p, PUZ133=2.p)

- Not a constraint solving task ("for **all** board sizes n...")
- Contains quantifiers: $\forall\,\Phi \models_{\mathbb{Z}} \forall\,\Psi$
- Difficult for SMT-solvers (because of quantifiers)
- Difficult for first-order provers (because of Integers)
- **Needed**: a theorem prover with built-in integer arithmetic

Approaches

Approaches

- **Theorem proving**
  - Theory Resolution/CM/Model Elimination,...
    Hierarchical Superposition [BGW 94], SPASS(LA) [AKW 2009], R+LIA [Korovin&Voronkov 07], Seq+QE [Rümmer 2008], Theory Instantiation [GK 2006], ME(LIA) [BT 2008]

Approaches

- **Theorem proving**
  - Theory Resolution/CM/Model Elimination,... Hierarchical Superposition [BGW 94], SPASS(LA) [AKW 2009], R+LIA [Korovin&Voronkov 07], Seq+QE [Rümmer 2008], Theory Instantiation [GK 2006], ME(LIA) [BT 2008]
- **SMT solvers**
  - Very successful for the quantifier free case, i.e. $\models_{\mathbf{T}} \forall\Phi$
  - Instantiation heuristics for general case, $\forall\Psi \models_{\mathbf{T}} \forall\Phi$

Approaches

- **Theorem proving**
  - Theory Resolution/CM/Model Elimination,... Hierarchical Superposition [BGW 94], SPASS(LA) [AKW 2009], R+LIA [Korovin&Voronkov 07], Seq+QE [Rümmer 2008], Theory Instantiation [GK 2006], ME(LIA) [BT 2008]
- **SMT solvers**
  - Very successful for the quantifier free case, i.e. $\models_T \forall \Phi$
  - Instantiation heuristics for general case, $\forall \Psi \models_T \forall \Phi$
- **Our research plan**
  - Efficient theorem prover for FOL modulo theories
  - Complete (when achievable)
  - Useful for countermodel computation
  - Build on attractive properties of instance based methods (ME, InstGen,...)

ME - Achievements so far

- **FDPLL**: basic ideas, predecessor of ME
- **ME**
  - Universal variables, unit propagation, redundancy criteria
- **ME+Lemmas**
- **Finite model computation**
- **MEE** = ME+Equality
  - Superposition rule, ordering refinements, redundancy criteria
- **MEE+Superposition**
  - Both calculi properly generalized
- **ME+LIA**
- Implementations: **Darwin** [JAIT 2006], E-Darwin, MELIA (new!)
- **This work**: MEE(T) = MEE+Theories (in particular LIA)

Further Plan of This Talk

DPLL

↓

Model Evolution

↓

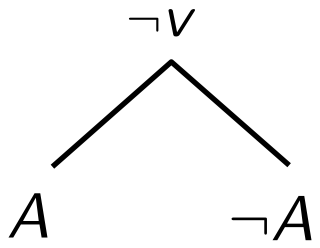
Model Evolution with Equality

↓

Model Evolution with Equality modulo Built-In Theories

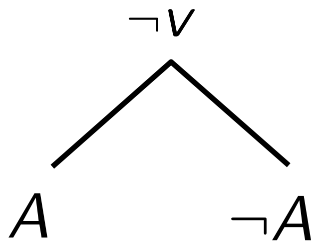
DPLL $\longrightarrow$ Model Evolution (ME)

DPLL



DPLL $\longrightarrow$ Model Evolution (ME)

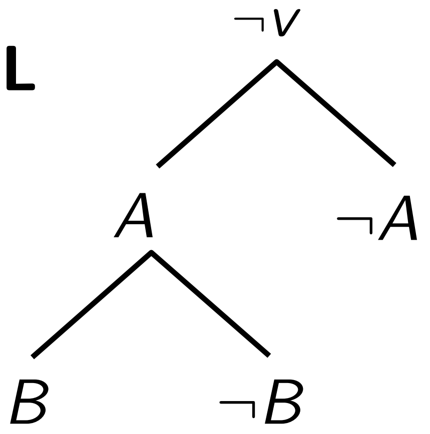
DPLL



$$\{A\} \overset{?}{\models} \neg A \vee B$$ Split

DPLL $\longrightarrow$ Model Evolution (ME)

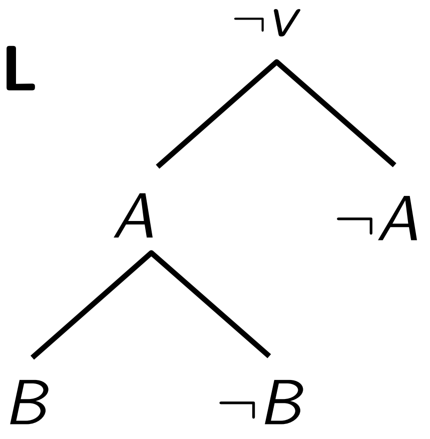
DPLL



$\{A\} \stackrel{?}{\models} \neg A \vee B$ Split

DPLL $\longrightarrow$ Model Evolution (ME)

DPLL

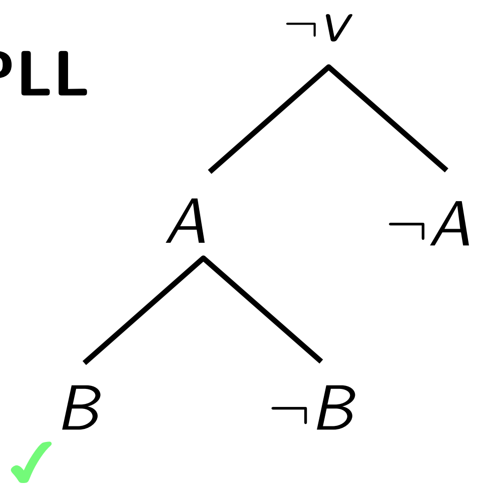


$$\{A\} \overset{?}{\models} \neg A \vee B \qquad \text{Split}$$

$$\{A, B\} \overset{?}{\models} \neg A \vee B$$

DPLL $\longrightarrow$ Model Evolution (ME)

DPLL

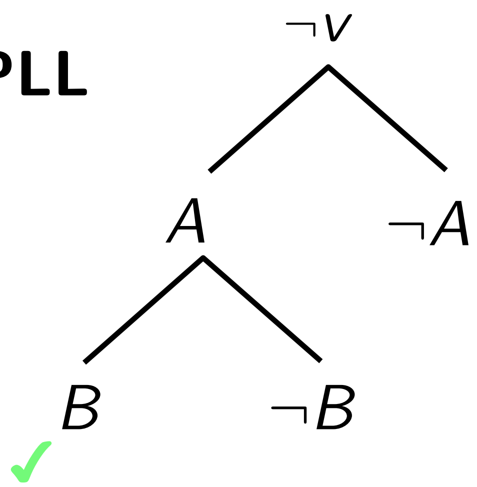


$$\{A\} \stackrel{?}{\models} \neg A \vee B$$ Split

$$\{A, B\} \stackrel{?}{\models} \neg A \vee B$$ ✓

DPLL $\longrightarrow$ Model Evolution (ME)

DPLL



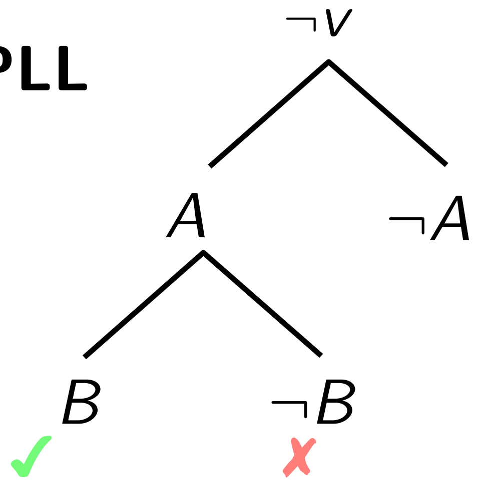
$$\{A\} \stackrel{?}{\models} \neg A \lor B$$ Split

$$\{A, B\} \stackrel{?}{\models} \neg A \lor B$$ ✓

$$\{A, \neg B\} \stackrel{?}{\models} \neg A \lor B$$

DPLL $\longrightarrow$ Model Evolution (ME)

DPLL



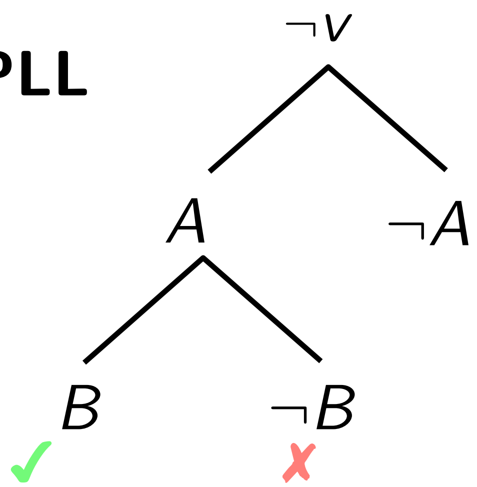
$$\{A\} \overset{?}{\models} \neg A \vee B$$ Split

$$\{A, B\} \overset{?}{\models} \neg A \vee B$$ ✓

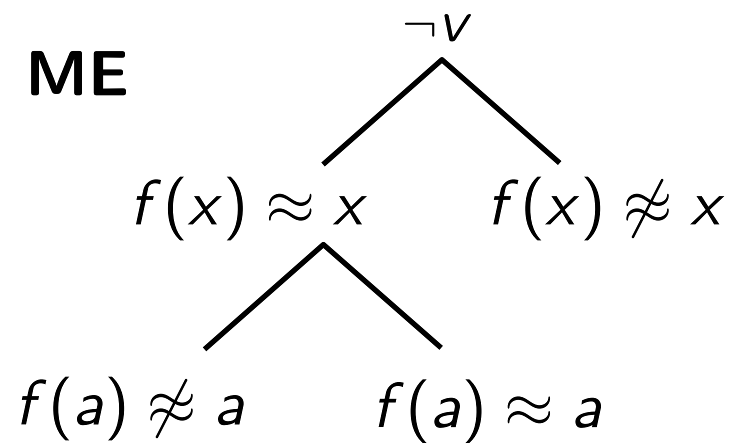
$$\{A, \neg B\} \overset{?}{\models} \neg A \vee B$$ ✗ Close

DPLL → Model Evolution (ME)

DPLL



ME



$\{A\} \overset{?}{\models} \neg A \vee B$ Split

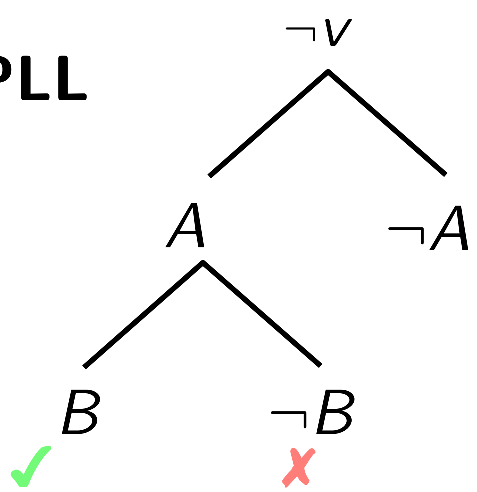
$\{A, B\} \overset{?}{\models} \neg A \vee B$ ✓

$\{A, \neg B\} \overset{?}{\models} \neg A \vee B$ ✗ Close

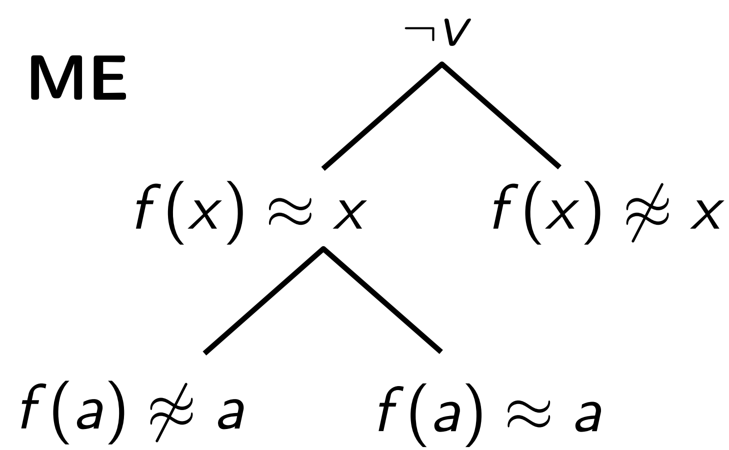
- Branches are called "contexts"
- Context induces interpretation
- **Split** to repair interpretation
- **Close** to abandon interpretation
- **Superposition** for equality reasoning

DPLL $\longrightarrow$ Model Evolution (ME)

DPLL



ME



$\{A\} \overset{?}{\models} \neg A \vee B$ Split

$\{A, B\} \overset{?}{\models} \neg A \vee B$ ✓

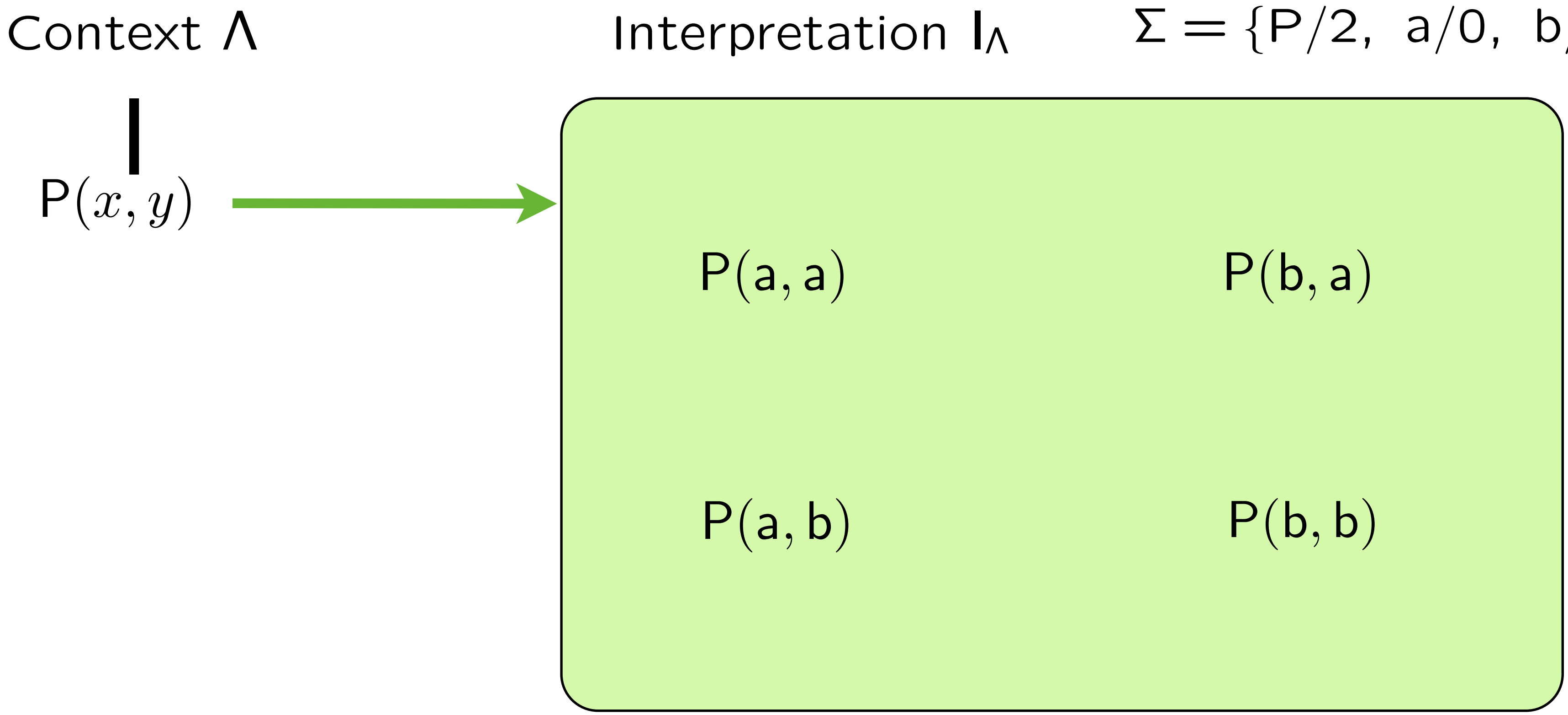
$\{A, \neg B\} \overset{?}{\models} \neg A \vee B$ ✗ Close

- Branches are called "contexts"
- Context induces interpretation
- **Split** to repair interpretation
- **Close** to abandon interpretation
- **Superposition** for equality reasoning

Next: key concept "productivity" to induce interpretation from context

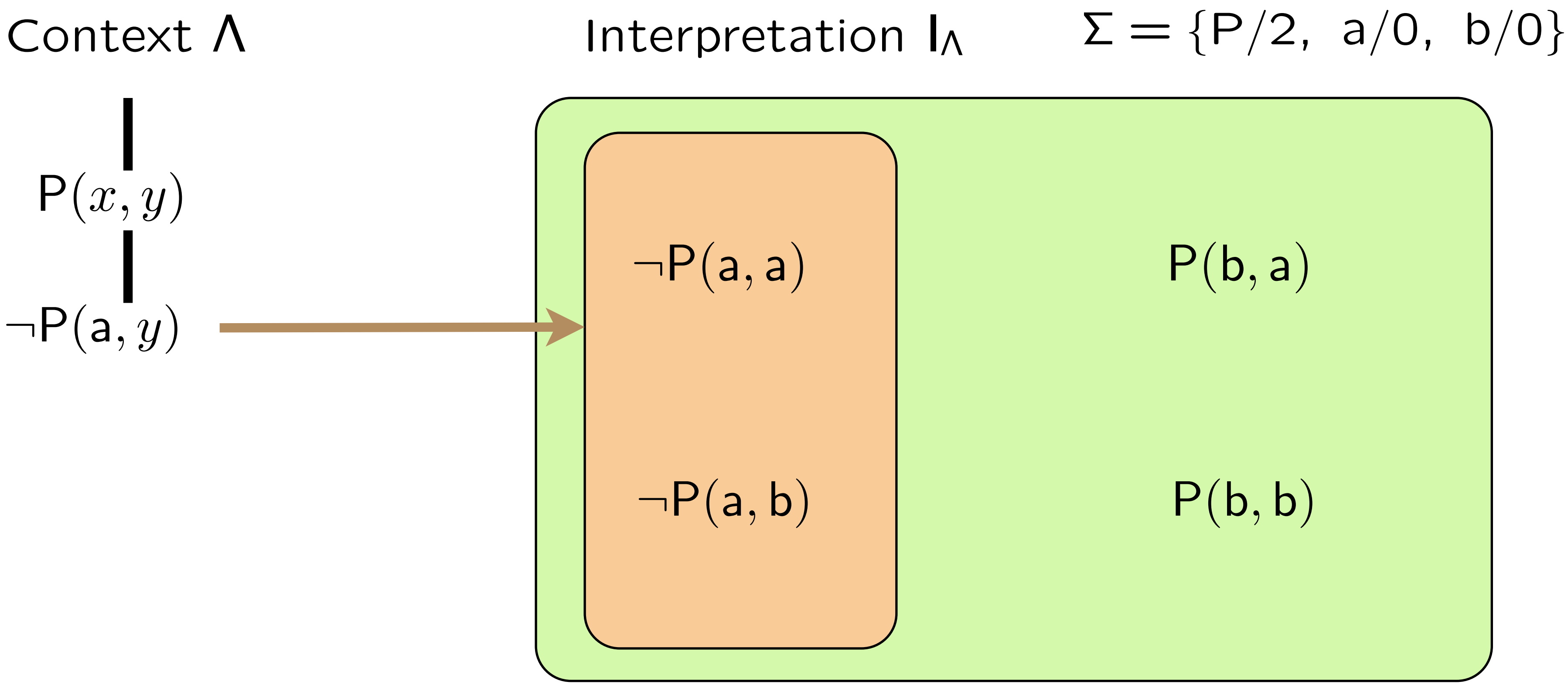
Interpretation Induced by a Context

$\Sigma = \{P/2,\ a/0,\ b/0\}$



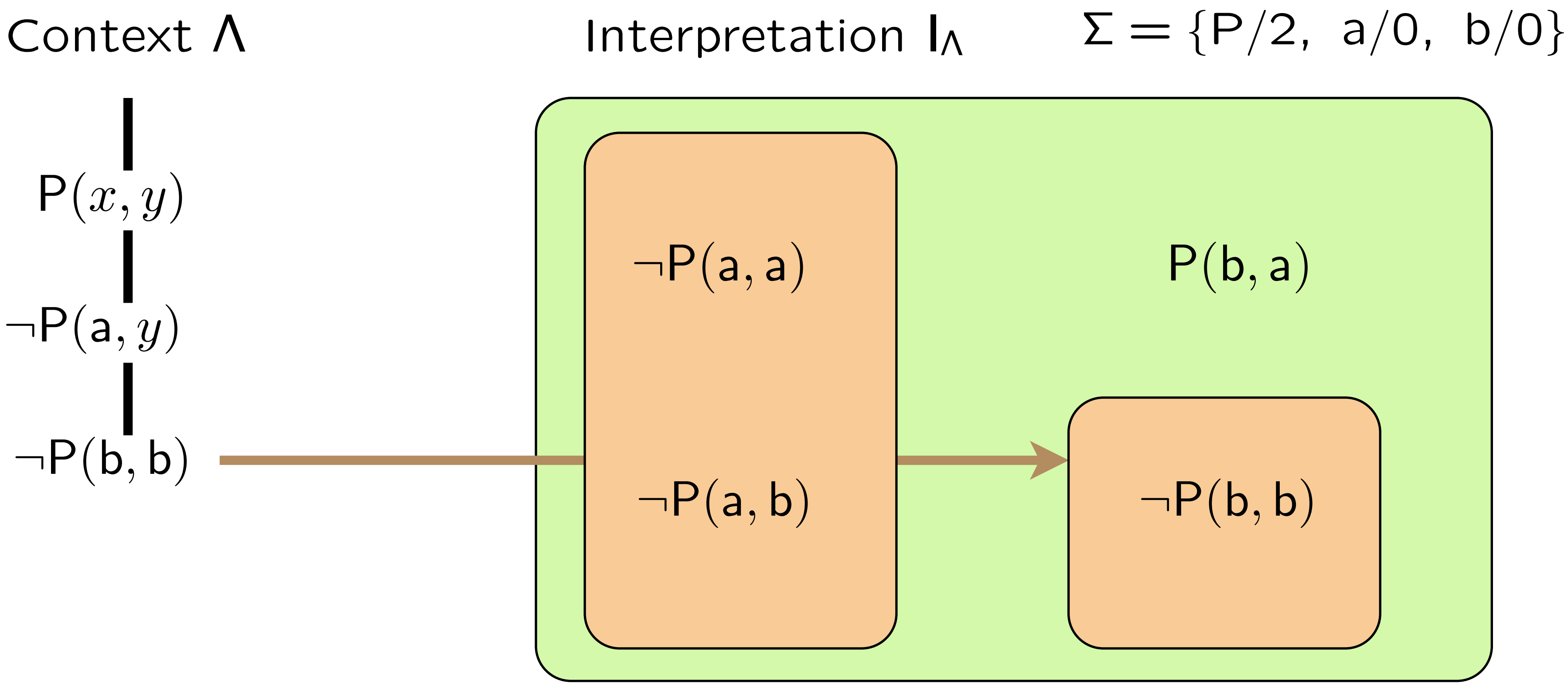
- A context literal specifies a truth value for all its ground instances, unless there is a more specific literal specifying the opposite truth value

Interpretation Induced by a Context



- A context literal specifies a truth value for all its ground instances, unless there is a more specific literal specifying the opposite truth value

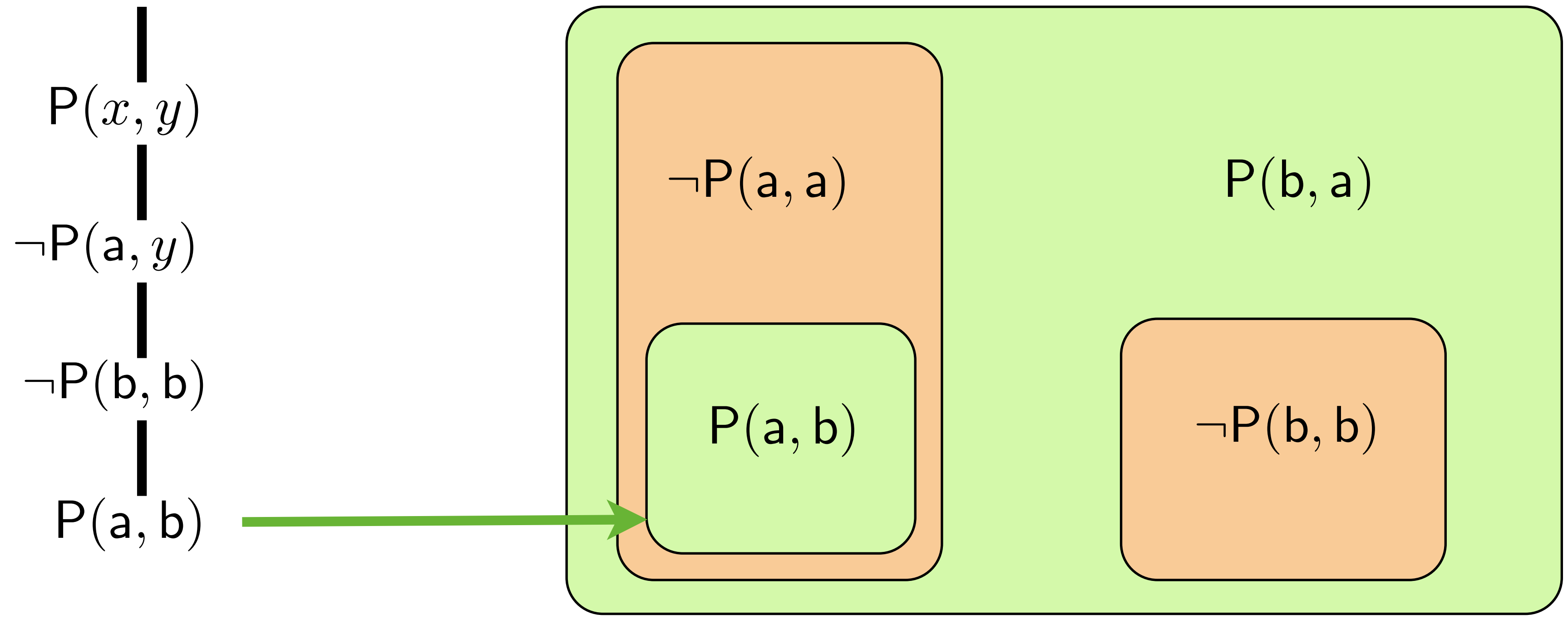
Interpretation Induced by a Branch



- A context literal specifies a truth value for all its ground instances, unless there is a more specific literal specifying the opposite truth value

Interpretation Induced by a Branch

Context Λ Interpretation I_Λ $\Sigma = \{P/2,\ a/0,\ b/0\}$

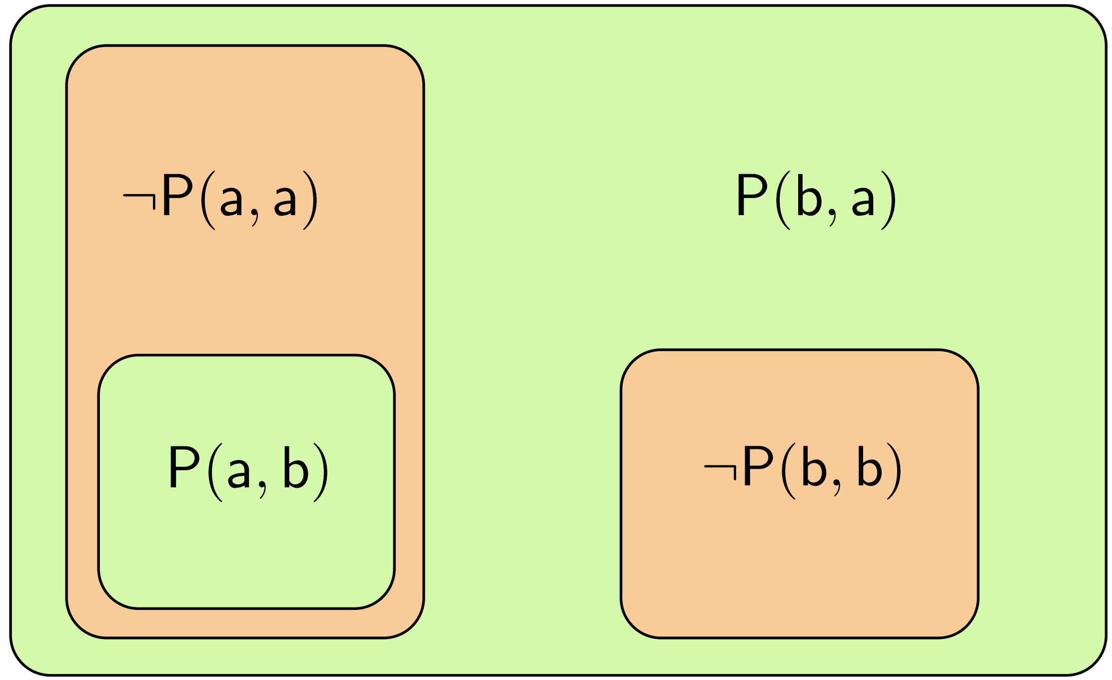


- A context literal specifies a truth value for all its ground instances, unless there is a more specific literal specifying the opposite truth value

Interpretation Induced by a Branch

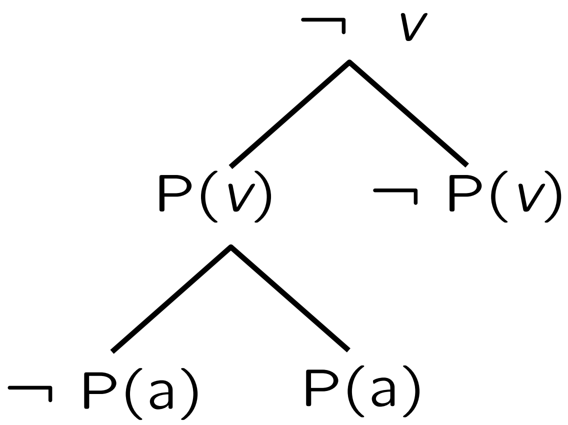
Context Λ | Interpretation I_Λ | $\Sigma = \{P/2,\ a/0,\ b/0\}$

$$\{\ P(x, y),\ \neg P(a, y),\ \neg P(b, b),\ P(a, b)\ \}$$



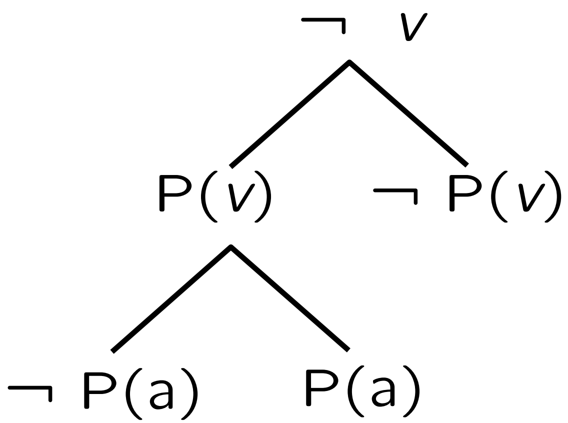
- A context literal specifies a truth value for all its ground instances, unless there is a more specific literal specifying the opposite truth value
- **The order of the context literals is irrelevant**

Inference Rule: Split



$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



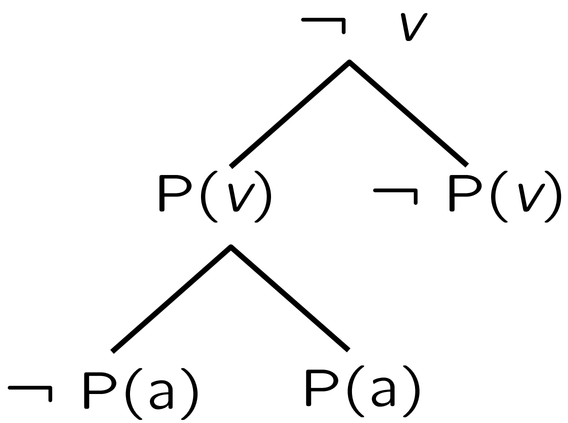
Context: $\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a})\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

$$\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



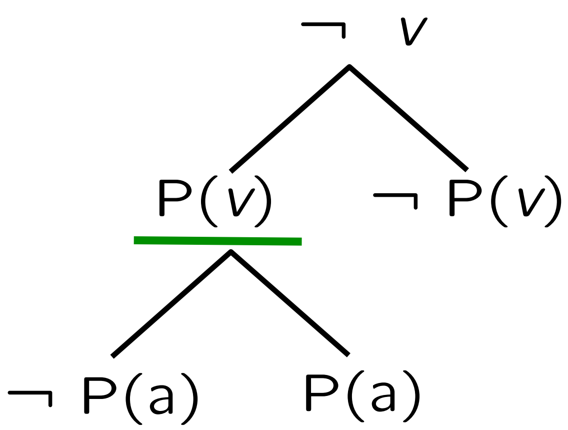
Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$

False: $\neg\mathsf{P}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{b})$

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



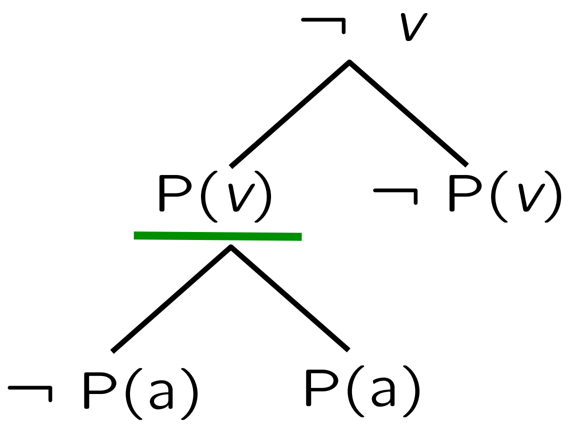
Context: $\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$

False: $\neg\mathsf{P}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{b})$

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



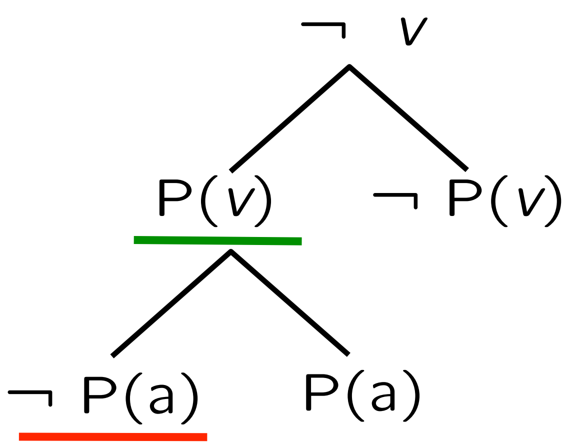
Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



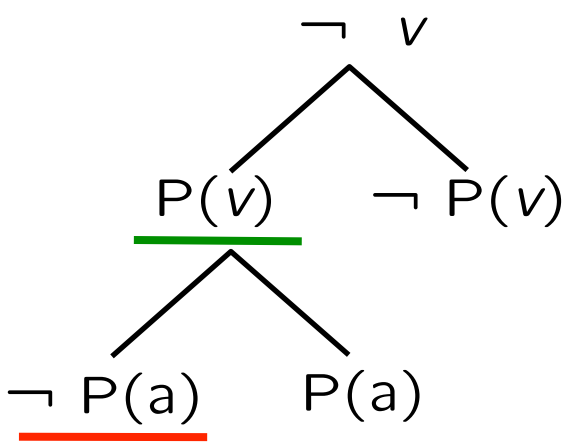
Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$

False: $\neg\mathsf{P}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{b})$

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



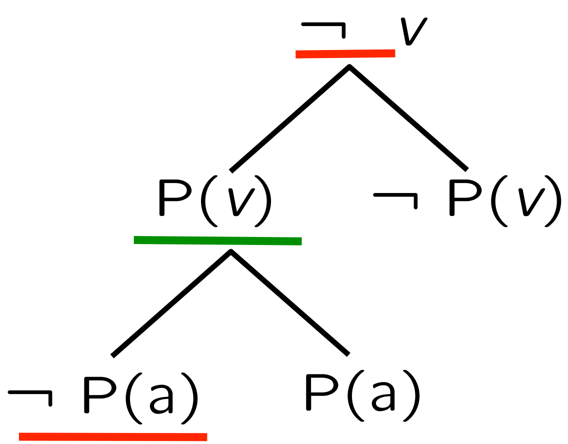
Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



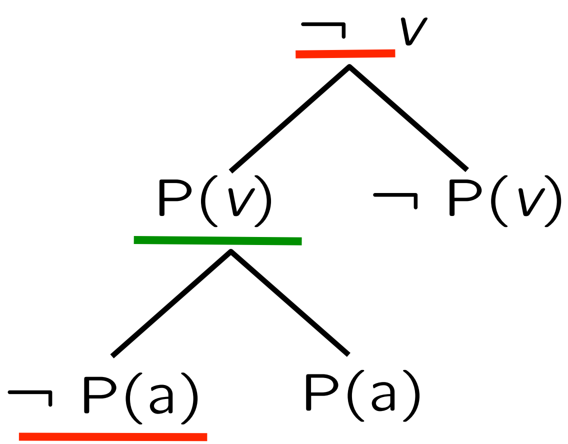
Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$

False: $\neg\mathsf{P}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{b})$

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



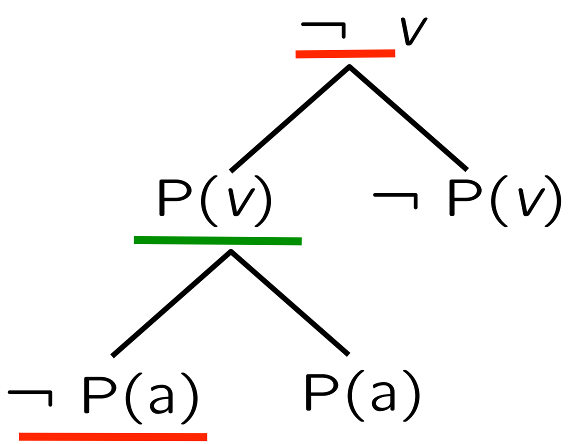
Context: $\{\neg v, P(v), \neg P(a)\}$

True: $P(b)$

False: $\neg P(a)$, $\neg Q(a)$, $\neg Q(b)$

$$\{\neg v, P(v), \neg P(a)\} \overset{?}{\models} P(x) \lor Q(x) \quad ✗$$

Inference Rule: Split



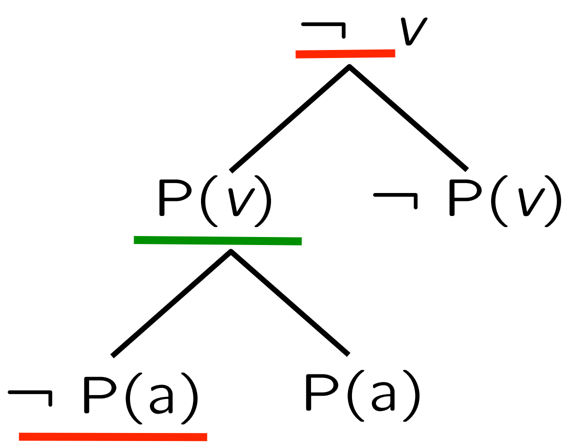
Context: $\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(a)\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

$$\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x) \quad ✗ \quad \xrightarrow{\text{Context Unifier}} \quad \mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{a})$$

Inference Rule: Split



Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

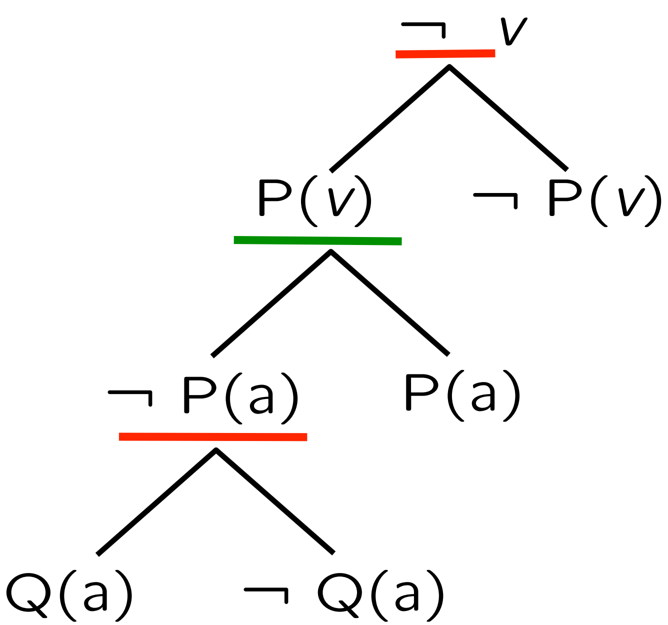
True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$ ✗ $\xrightarrow{\text{Context Unifier}}$ $\mathsf{P}(\mathsf{a}) \vee \underline{\mathsf{Q}(\mathsf{a})}$

Split

Inference Rule: Split



Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

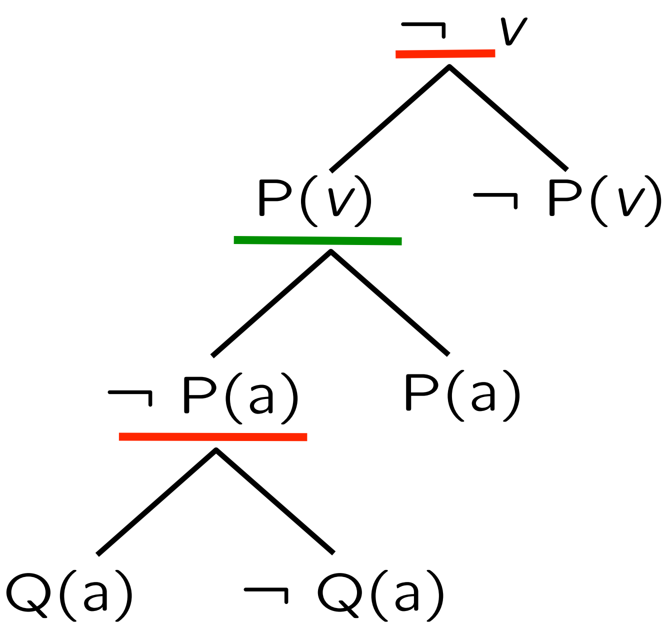
True: $\mathsf{P}(\mathsf{b})$

False: $\neg\mathsf{P}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{b})$

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x) \quad ✗ \quad \xrightarrow{\text{Context Unifier}} \quad \mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{a})$$

Split

Inference Rule: Split



Context: $\{\neg v, P(v), \neg P(a)\}$

True: $P(b)$

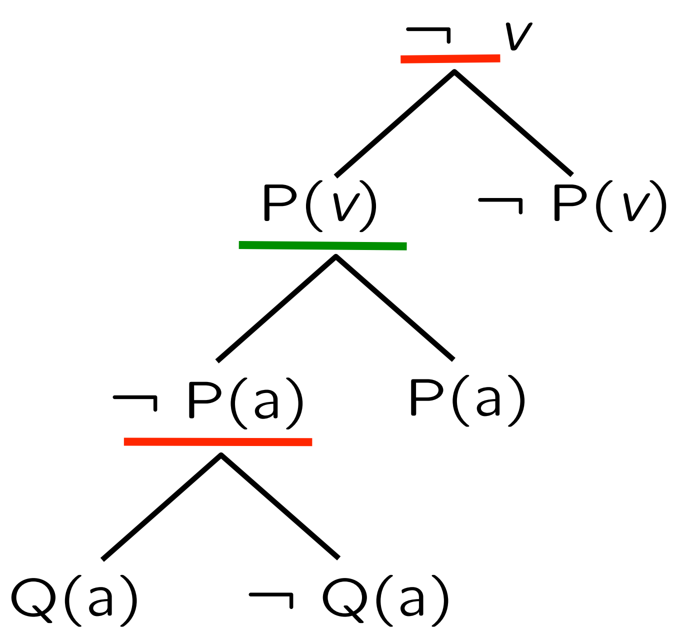
False: $\neg P(a)$, $\neg Q(a)$, $\neg Q(b)$

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \lor \mathsf{Q}(x)$$ ✗ —Context Unifier→ $\mathsf{P}(\mathsf{a}) \lor \mathsf{Q}(\mathsf{a})$

Split

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \lor \mathsf{Q}(x)$$

Inference Rule: Split



Context: $\{\neg v, P(v), \neg P(a)\}$

True: $P(b)$

False: $\neg P(a)$, $\neg Q(a)$, $\neg Q(b)$

Context: $\{\neg v, P(v), \neg P(a), Q(a)\}$

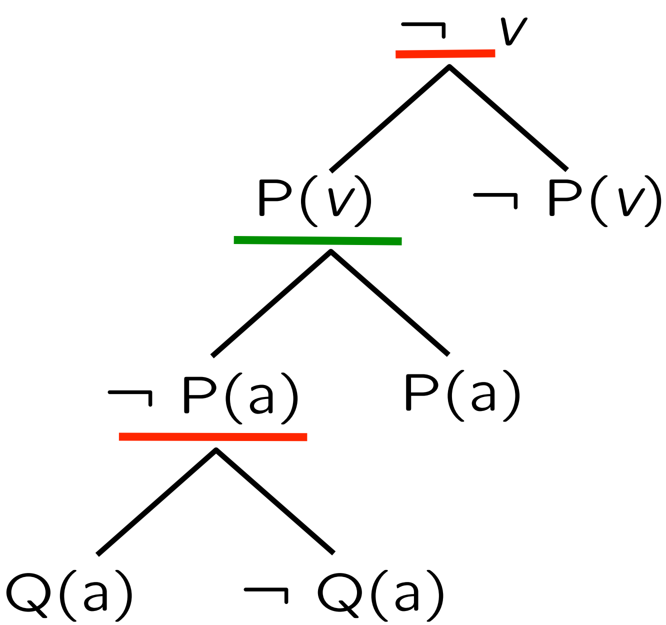
True: $P(b)$, $Q(a)$

False: $\neg P(a)$, $\neg Q(b)$

$$\{\neg v, P(v), \neg P(a)\} \stackrel{?}{\models} P(x) \vee Q(x)$$ ✗ $\xrightarrow{\text{Context Unifier}}$ $P(a) \vee Q(a)$ — Split

$$\{\neg v, P(v), \neg P(a), Q(a)\} \stackrel{?}{\models} P(x) \vee Q(x)$$

Inference Rule: Split



Context: $\{\neg v, P(v), \neg P(a)\}$

True: $P(b)$

False: $\neg P(a)$, $\neg Q(a)$, $\neg Q(b)$

Context: $\{\neg v, P(v), \neg P(a), Q(a)\}$

True: $P(b)$, $Q(a)$

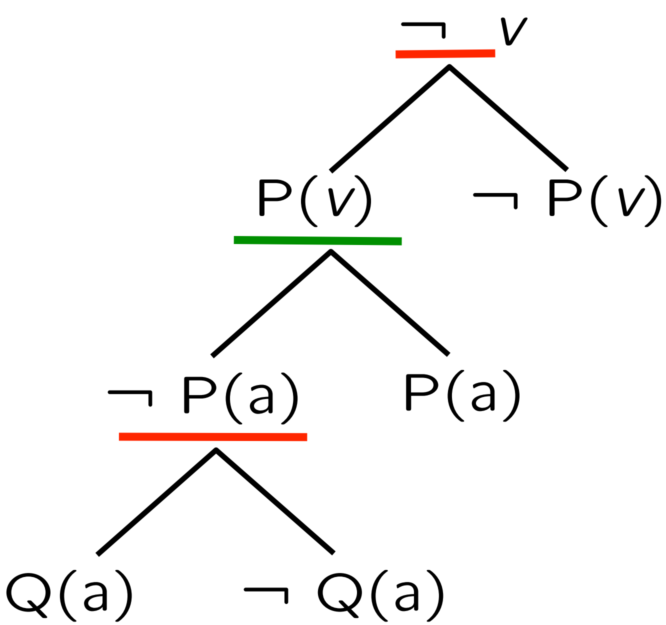
False: $\neg P(a)$, $\neg Q(b)$

$$\{\neg v, P(v), \neg P(a)\} \overset{?}{\models} P(x) \vee Q(x) \quad ✗ \quad \xrightarrow{\text{Context Unifier}} \quad P(a) \vee \underline{Q(a)}$$

Split

$$\{\neg v, P(v), \neg P(a), Q(a)\} \overset{?}{\models} P(x) \vee Q(x)$$

Inference Rule: Split



Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\}$

True: P(b), Q(a)

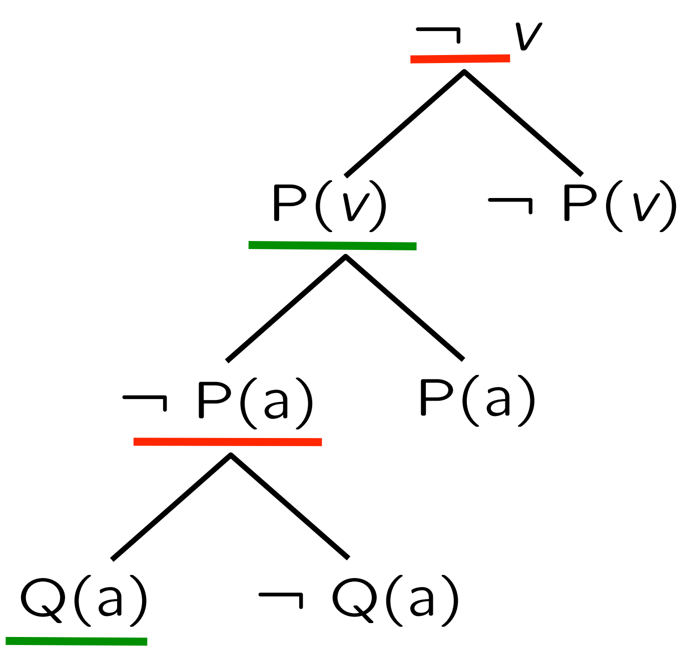
False: ¬P(a), ¬Q(b)

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x) \quad ✗ \quad \xrightarrow{\text{Context Unifier}} \quad \mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{a})$$

Split

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



Context: $\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$

False: $\neg \mathsf{P}(\mathsf{a})$, $\neg \mathsf{Q}(\mathsf{a})$, $\neg \mathsf{Q}(\mathsf{b})$

Context: $\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$, $\mathsf{Q}(\mathsf{a})$

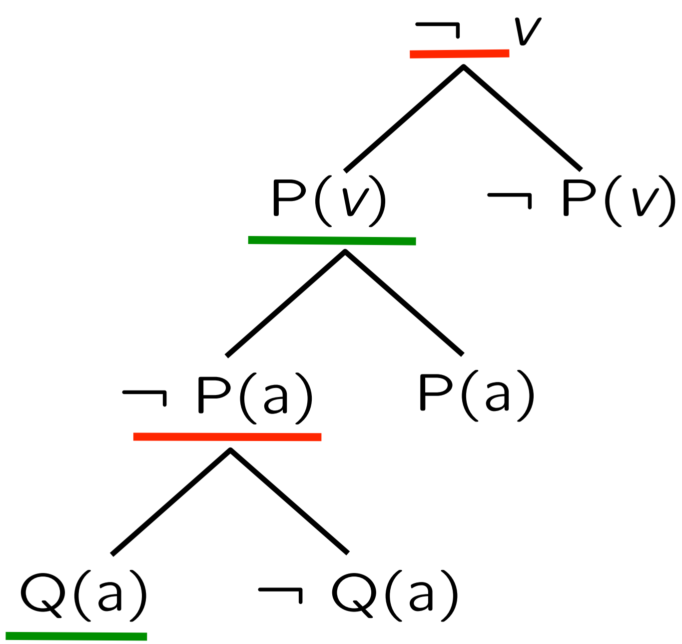
False: $\neg \mathsf{P}(\mathsf{a})$, $\neg \mathsf{Q}(\mathsf{b})$

$$\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x) \quad ✗ \quad \xrightarrow{\text{Context Unifier}} \quad \mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{a})$$

Split

$$\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$$

Inference Rule: Split



Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$

False: $\neg\mathsf{P}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{b})$

Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\}$

True: $\mathsf{P}(\mathsf{b})$, $\mathsf{Q}(\mathsf{a})$

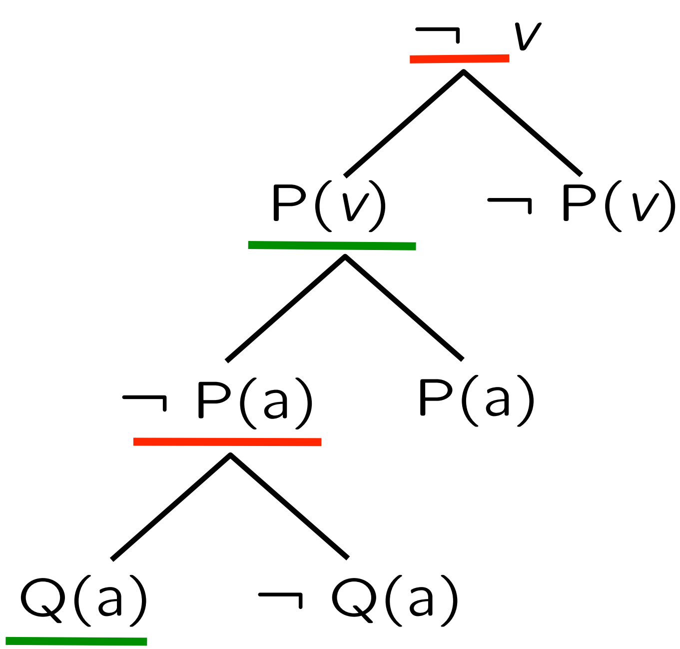
False: $\neg\mathsf{P}(\mathsf{a})$, $\neg\mathsf{Q}(\mathsf{b})$

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \lor \mathsf{Q}(x) \quad ✗ \quad \xrightarrow{\text{Context Unifier}} \quad \mathsf{P}(\mathsf{a}) \lor \underline{\mathsf{Q}(\mathsf{a})}$$

Split

$$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \lor \mathsf{Q}(x)$$

Inference Rule: Split



Context: $\{\neg v, P(v), \neg P(a)\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

Context: $\{\neg v, P(v), \neg P(a), Q(a)\}$

True: P(b), Q(a)

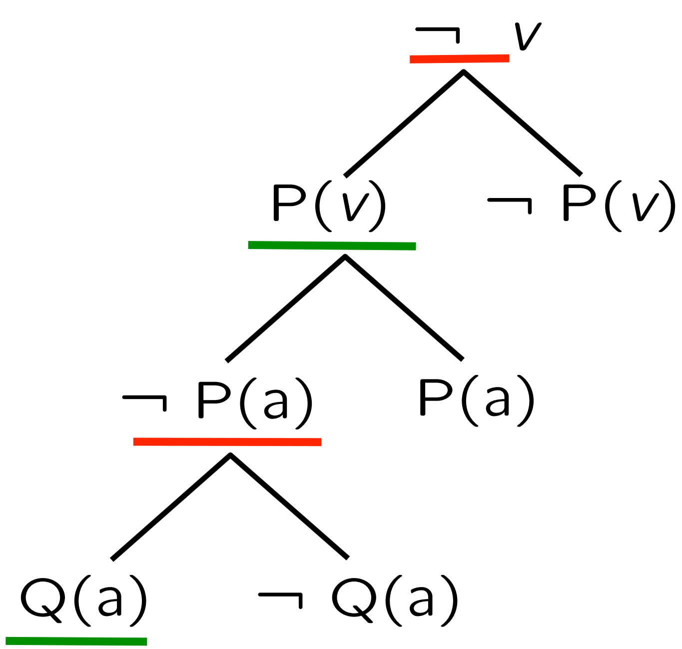
False: ¬P(a), ¬Q(b)

$$\{\neg v, P(v), \neg P(a)\} \overset{?}{\models} P(x) \vee Q(x)$$ ✗ —Context Unifier→ $P(a) \vee Q(a)$

$$\{\neg v, P(v), \neg P(a), Q(a)\} \overset{?}{\models} P(x) \vee Q(x)$$ ✓

Split

Inference Rule: Split



Context: $\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a})\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

Context: $\{\neg v, \mathsf{P}(v), \neg \mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\}$

True: P(b), Q(a)

False: ¬P(a), ¬Q(b)

$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$ ✗ —Context Unifier→ $\mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{a})$

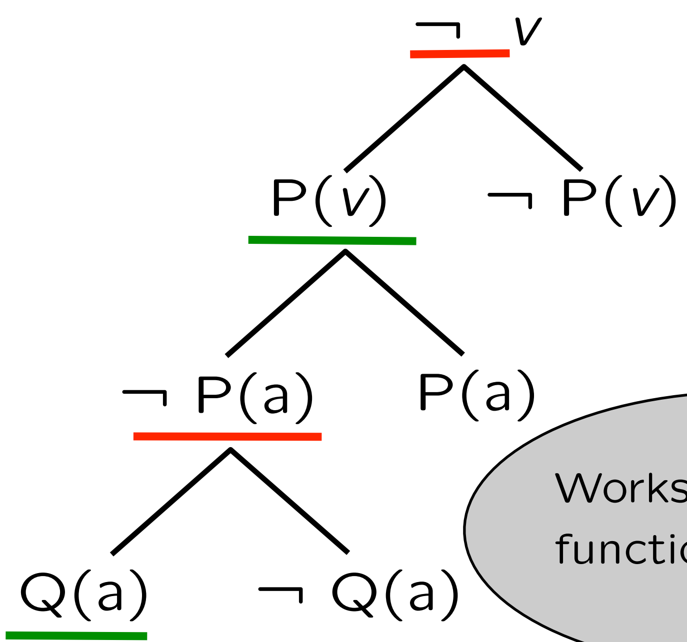
Split

$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$ ✓

Split - detect falsified instances and repair interpretation

Additional rules: Close, Assert, Compact, Resolve, Subsume

Inference Rule: Split



Works also with function symbols

Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\}$

True: P(b)

False: ¬P(a), ¬Q(a), ¬Q(b)

Context: $\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\}$

True: P(b), Q(a)

False: ¬P(a), ¬Q(b)

$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$ ✗ $\xrightarrow{\text{Context Unifier}}$ $\mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{a})$

Split

$\{\neg v, \mathsf{P}(v), \neg\mathsf{P}(\mathsf{a}), \mathsf{Q}(\mathsf{a})\} \overset{?}{\models} \mathsf{P}(x) \vee \mathsf{Q}(x)$ ✓

Split - detect falsified instances and repair interpretation
Additional rules: Close, Assert, Compact, Resolve, Subsume

ME $\rightarrow$ MEE $\rightarrow$ MEE(T) - Data Structures

- **ME**
  - Context Λ induces Herbrand interpretation I_Λ via productivity
  - Clauses $C \in \Phi$ are evaluated in I_Λ to drive derivation
- **MEE** (ME with equality)
  - Context Λ induces rewrite system $\mathfrak{R}_\Lambda$ via productivity and Bachmair-Ganzinger-like model construction
  - Constraint clauses $(C \leftarrow R) \in \Phi$
    - Paramodulation from context literals Λ into C

$$\frac{\mathsf{f(a)} \approx \mathsf{a} \qquad\qquad \mathsf{P(f(}x\mathsf{))} \lor \mathsf{Q(}x, y\mathsf{)} \leftarrow}{\mathsf{P(a)} \lor \mathsf{Q(a,} y\mathsf{)} \leftarrow \mathsf{f(a)} \approx \mathsf{a}}$$

    - Splitting with a literal from $C \cup \neg R$
    - Semantics (ground): $\Lambda \models C \leftarrow R$ iff $({\mathfrak{R}_\Lambda}^* \models C$ or $R \not\subseteq \mathfrak{R}_\Lambda)$

MEE(T) - Data Structures

- Purified constraint clauses $C \leftarrow R \cdot c$
  - Ordinary clause C over foreground operators
  - Restriction R as above over foreground operators
  - Constraint c over background signature
  - Communication via shared variables (only)

$$\mathrm{select}(\mathrm{store}(a,i,e),j) \approx \mathrm{select}(a,j) \leftarrow \emptyset \cdot i \neq j$$ OK

$$\mathrm{a}_2 \approx \mathrm{store}(\mathrm{a}_1,\mathrm{n},5) \leftarrow \emptyset \cdot \emptyset$$ not OK

$$\mathrm{a}_2 \approx \mathrm{store}(\mathrm{a}_1,\mathrm{n},e) \leftarrow \emptyset \cdot e = 5$$ OK (n is foreground)

$$\mathrm{a}_2 \approx \mathrm{store}(\mathrm{a}_1,n,e) \leftarrow \emptyset \cdot e = 5 \wedge n = \mathrm{n}$$ OK (n is background)

MEE(T) - Data Structures

- $C \leftarrow R \cdot c$ stands for all instances $(C \leftarrow R \cdot c)\gamma$ where γ maps var(c) to symbolic constants ("rigid variables")

$$\mathrm{select}(\mathrm{store}(a, i, e), j) \approx \mathrm{select}(a, j) \ \leftarrow \ \emptyset \cdot i \neq j$$

MEE(T) - Data Structures

- $C \leftarrow R \cdot c$ stands for all instances $(C \leftarrow R \cdot c)\gamma$ where γ maps var(c) to symbolic constants ("rigid variables")

$$\mathrm{select}(\mathrm{store}(a, i, e), j) \approx \mathrm{select}(a, j) \ \leftarrow \ \emptyset \cdot i \neq j$$

$$\Big\downarrow \text{ inst } \gamma$$

$$\mathrm{select}(\mathrm{store}(a, \mathrm{r}_1, e), \mathrm{r}_1) \approx \mathrm{select}(a, \mathrm{r}_1) \ \leftarrow \ \emptyset \cdot \mathrm{r}_1 \neq \mathrm{r}_1$$

MEE(T) - Data Structures

- $C \leftarrow R \cdot c$ stands for all instances $(C \leftarrow R \cdot c)\gamma$ where γ maps var(c) to symbolic constants ("rigid variables")

$$\mathrm{select}(\mathrm{store}(a, i, e), j) \approx \mathrm{select}(a, j) \;\leftarrow\; \emptyset \cdot i \neq j$$

$$\Big\downarrow \text{ inst } \gamma$$

$$\mathrm{select}(\mathrm{store}(a, \mathrm{r}_1, e), \mathrm{r}_1) \approx \mathrm{select}(a, \mathrm{r}_1) \;\leftarrow\; \emptyset \cdot \mathrm{r}_1 \neq \mathrm{r}_1$$

$$\mathrm{select}(\mathrm{store}(a, \mathrm{r}_1, e), \mathrm{r}_2) \approx \mathrm{select}(a, \mathrm{r}_2) \;\leftarrow\; \emptyset \cdot \mathrm{r}_1 \neq \mathrm{r}_2$$

$$\dots$$

MEE(T) - Data Structures

- $C \leftarrow R \cdot c$ stands for all instances $(C \leftarrow R \cdot c)\gamma$
  where γ maps var(c) to symbolic constants ("rigid variables")

$$\mathrm{select}(\mathrm{store}(a, i, e), j) \approx \mathrm{select}(a, j) \;\leftarrow\; \emptyset \cdot i \neq j$$

$$\Big\downarrow \text{ inst } \gamma$$

$$\mathrm{select}(\mathrm{store}(a, \mathrm{r}_1, e), \mathrm{r}_1) \approx \mathrm{select}(a, \mathrm{r}_1) \;\leftarrow\; \emptyset \cdot \mathrm{r}_1 \neq \mathrm{r}_1$$

$$\mathrm{select}(\mathrm{store}(a, \mathrm{r}_1, e), \mathrm{r}_2) \approx \mathrm{select}(a, \mathrm{r}_2) \;\leftarrow\; \emptyset \cdot \mathrm{r}_1 \neq \mathrm{r}_2$$

$$\ldots$$

- The role of rigid variables
  - For foreground calculus: uninterpreted constants
  - For background reasoner: existentially quantified variables (however same r can be shared across constraint clauses)
- -> Instantiation via γ enables separation of reasoning

Main Inference Rule: Paramodulation

- Constraint clauses $C \leftarrow R \cdot c$ are
  - paramodulated into C by context equations Λ

$$\frac{\underline{f(r_1)} \approx a \qquad\qquad P(\underline{f(x)}) \vee Q(x, y) \leftarrow \emptyset \cdot x + y > r_2}{P(a) \vee Q(r_1, y) \leftarrow f(r_1) \approx a \cdot r_1 + y > r_2}$$

  - used for Splitting if
    - C consists of positive literals only, and
    - $(C \leftarrow R \cdot c)$ is (potentially) falsified in $\Lambda \cdot \Gamma$ where Γ is the "global background context" (next slide)

Main Inference Rule: Splitting

$\boldsymbol{\Lambda}$

$\neg v$

$f(\mathrm{r}_1) \approx a$

$\boldsymbol{\Phi}$

$$\mathrm{P(a)} \lor \mathrm{Q}(\mathrm{r}_1, y) \leftarrow \mathrm{f}(\mathrm{r}_1) \approx \mathrm{a} \cdot \mathrm{r}_1 + y > \mathrm{r}_2$$

Main Inference Rule: Splitting

A global background context Γ collects instantiated constraints

$\boldsymbol{\Lambda}$	$\boldsymbol{\Gamma}$	$\boldsymbol{\Phi}$
$\neg v$	$r_1 \neq r_2$	$P(a) \lor Q(r_1, y) \leftarrow f(r_1) \approx a \cdot r_1 + y > r_2$
$f(r_1) \approx a$		

Main Inference Rule: Splitting

A global background context Γ collects instantiated constraints

Λ	Γ	Φ
$\neg v$	$\mathsf{r}_1 \neq \mathsf{r}_2$	$\mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{r}_1, y) \leftarrow \mathsf{f}(\mathsf{r}_1) \approx \mathsf{a} \cdot \mathsf{r}_1 + y > \mathsf{r}_2$
$f(\mathsf{r}_1) \approx a$		context unifier σ
		$\mathsf{P}(\mathsf{a}) \vee \mathsf{Q}(\mathsf{r}_1, y) \leftarrow \mathsf{f}(\mathsf{r}_1) \approx \mathsf{a} \cdot \mathsf{r}_1 + y > \mathsf{r}_2$

Main Inference Rule: Splitting

A global background context Γ collects instantiated constraints

Λ	**Γ**	**Φ**
$\neg v$	$\mathrm{r}_1 \neq \mathrm{r}_2$	$\mathrm{P(a)} \vee \mathrm{Q}(\mathrm{r}_1, y) \leftarrow \mathrm{f}(\mathrm{r}_1) \approx \mathrm{a} \cdot \mathrm{r}_1 + y > \mathrm{r}_2$
$f(\mathrm{r}_1) \approx a$		↓ context unifier σ
		$\mathrm{P(a)} \vee \mathrm{Q}(\mathrm{r}_1, y) \leftarrow \mathrm{f}(\mathrm{r}_1) \approx \mathrm{a} \cdot \mathrm{r}_1 + y > \mathrm{r}_2$
		↓ inst γ
		$\mathrm{P(a)} \vee \mathrm{Q}(\mathrm{r}_1, \mathrm{r}_1) \leftarrow \mathrm{f}(\mathrm{r}_1) \approx \mathrm{a} \cdot \mathrm{r}_1 + r_1 > \mathrm{r}_2$

Main Inference Rule: Splitting

A global background context Γ collects instantiated constraints

Λ	**Γ**	**Φ**
$\neg v$	$r_1 \neq r_2$	$P(a) \vee Q(r_1, y) \leftarrow f(r_1) \approx a \cdot r_1 + y > r_2$
		↓ context unifier σ
$f(r_1) \approx a$		$P(a) \vee Q(r_1, y) \leftarrow f(r_1) \approx a \cdot r_1 + y > r_2$
		↓ inst γ
		$P(a) \vee Q(r_1, r_1) \leftarrow f(r_1) \approx a \cdot r_1 + r_1 > r_2$

$f(r_1) \approx a \in \Re_\Lambda$?

Γ consistent with $r_1 + r_1 > r_2$?

Main Inference Rule: Splitting

A global background context Γ collects instantiated constraints

Λ **Γ** **Φ**

$\neg v$ $\quad$ $r_1 \neq r_2$ $\quad$ $P(a) \vee Q(r_1, y) \leftarrow f(r_1) \approx a \cdot r_1 + y > r_2$

$f(r_1) \approx a$

context unifier σ

$P(a) \vee Q(r_1, y) \leftarrow f(r_1) \approx a \cdot r_1 + y > r_2$

inst γ

$\neg v$ $\quad$ $r_1 \neq r_2$

$r_1 + r_1 > r_2$ $\quad$ $P(a) \vee Q(r_1, r_1) \leftarrow f(r_1) \approx a \cdot r_1 + r_1 > r_2$

$f(r_1) \approx a$

$Q(r_1, r_1)$ $\quad$ $\neg Q(r_1, r_1)$

Split

$f(r_1) \approx a \in \Re_\Lambda$?

Γ consistent with $r_1 + r_1 > r_2$?

The Bigger Picture

- **Derivations** on top of the above (and other) inference rules
- **Model construction** as before but parameteric in models of Γ
- **Backround reasoner**
  - Satisfiability checker for quantifier-free constraints
  - Quantifier elimination if free background constants allowed
- **Improvements** (some not in the paper)
  - Ordering refinements for paramodulation
  - Selection function to focus paramodulation inference
  - Simplification by context literals (gives DPLL)
  - Simplification by rewriting with unit clauses (gives KB)
  - Proper subsumption
  - Universal context literals
    $\forall \boldsymbol{x}\ \mathrm{f}(x) \approx x$ no exceptions allowed

Soundness and Completeness Under Conditions

- **Fairness**
  - standard: non-redundant inferences must become redundant eventually
  - specific: Γ must persistently represent all solutions of constraints of non-redundant clauses, see below
- **Completeness**
  - Free background constants must be finitely bounded

    $$P(0) \qquad P(x) \Rightarrow P(x{+}1) \qquad \neg P(n)$$

    is not OK. (No complete calculus in this case anyway.)
  - Possible "fix": add input formula $0 \leq n \leq 100$
    - This is much better than using $n \approx 0 \vee \ldots \vee n \approx 100$
    - But gives a (fixable) soundness problem, see below

Soundness and Completeness Under Conditions

- **Soundness Problem and Fix**
  - Satisfiable clause set, n is free background const

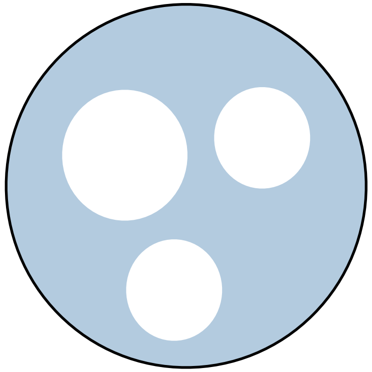
$$P(x) \leftarrow x<50 \qquad \neg P(n) \qquad 0 \leq n \leq 100$$

  - Will find a "refutation" with $\Gamma = \{ r_1<50,\ r_1=n \}$
  - Problem is that unsatisfiability has been established for **some** values for n, not for **all** values
  - Fix:
    - Eliminate rigid variables, giving $\Gamma 1 = \{ n<50 \}$
    - Start new derivation, this time with a $\Gamma = \{ \neg(n<50) \}$
    - Will find a model in that derivation
- Exhausts finitely in the bounded case
- **Conflict-driven**

Soundness and Completeness Under Conditions

- **Soundness Problem and Fix**
  - Satisfiable clause set, n is free background const

    $P(x) \leftarrow x{<}50$ $\neg P(n)$ $0 \leq n \leq 100$



  - Will find a "refutation" with $\Gamma = \{ r_1{<}50, r_1{=}n \}$
  - Problem is that unsatisfiability has been established for **some** values for n, not for **all** values
  - Fix:
    - Eliminate rigid variables, giving $\Gamma 1 = \{ n{<}50 \}$
    - Start new derivation, this time with a $\Gamma = \{ \neg(n{<}50) \}$
    - Will find a model in that derivation
- Exhausts finitely in the bounded case
- **Conflict-driven**

Fairness Problem

$\mathsf{P}(x) \leftarrow 0 < x \quad (1) \qquad \neg \mathsf{P}(x) \vee \mathsf{Q}(x) \leftarrow 3 < x \quad (2)$

$\neg \mathsf{P}(x) \leftarrow x = 2 \quad (3)$

A derivation might always prefer (1) and (2) over (3):

	Λ	Γ
(1)	P(r1)	0 < r1
(2)		

Problem: Solution of x = 2 in (3) is not persistently represented

Fairness Problem

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

A derivation might always prefer (1) and (2) over (3):

	Λ	Γ
(1)	P(r1)	0 < r1
(2)		3 < r1

Problem: Solution of x = 2 in (3) is not persistently represented

Fairness Problem

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

A derivation might always prefer (1) and (2) over (3):

	Λ	Γ
(1)	P(r1)	$0 < r1$
(2)	Q(r1)	$3 < r1$

Problem: Solution of $x = 2$ in (3) is not persistently represented

Fairness Problem

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

A derivation might always prefer (1) and (2) over (3):

	Λ	Γ
(1)	P(r1)	$0 < r1$
(2)	Q(r1)	$3 < r1$

(3) does not close as r1=2 not consistent with Γ

Problem: Solution of $x = 2$ in (3) is not persistently represented

Fairness Problem

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

A derivation might always prefer (1) and (2) over (3):

	Λ	Γ
(1)	P(r1)	$0 < \text{r1}$
(2)	Q(r1)	$3 < \text{r1}$
		(3) does not close as r1=2 not consistent with Γ
(1)	P(r2)	$0 < \text{r2}$

Problem: Solution of $x = 2$ in (3) is not persistently represented

Fairness Problem

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

A derivation might always prefer (1) and (2) over (3):

	Λ	Γ
(1)	P(r1)	0 < r1
(2)	Q(r1)	3 < r1
		(3) does not close as r1=2 not consistent with Γ
(1)	P(r2)	0 < r2
	⋮	

Problem: Solution of x = 2 in (3) is not persistently represented

Fairness Problem - Solution

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

Fairness Problem - Solution

$P(x) \leftarrow 0 < x$ (1) $\qquad$ $\neg P(x) \vee Q(x) \leftarrow 3 < x$ (2)

$\neg P(x) \leftarrow x = 2$ (3)

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1) 0<r1 $\qquad$ r1 ∊ [**1** .. **5**]

Fairness Problem - Solution

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1) 0<r1 r1 ∈ [**1** .. **5**]

(2) 3<r1 ⇒ r1 ∈ [**4** .. **5**]

Fairness Problem - Solution

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1) 0<r1 r1 $\in$ [**1** .. **5**]

(2) 3<r1 $\Rightarrow$ r1 $\in$ [**4** .. **5**]

(1) 0<r2 r2 $\in$ [**1** .. **5**]

Fairness Problem - Solution

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1)	0<r1		r1 $\in$ [**1** .. **5**]
(2)	3<r1	$\Rightarrow$	r1 $\in$ [**4** .. **5**]
(1)	0<r2		r2 $\in$ [**1** .. **5**]
(2)	3<r2	$\Rightarrow$	r2 $\in$ [**4** .. **5**]

Fairness Problem - Solution

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1)	0<r1		r1 ∈ [**1** .. **5**]
(2)	3<r1	⇒	r1 ∈ [**4** .. **5**]
(1)	0<r2		r2 ∈ [**1** .. **5**]
(2)	3<r2	⇒	r2 ∈ [**4** .. **5**]
(1)	0<r3		r3 ∈ [**1** .. **5**]

Fairness Problem - Solution

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1) 0<r1 $r1 \in [\mathbf{1} .. \mathbf{5}]$

(2) 3<r1 $\Rightarrow r1 \in [\mathbf{4} .. \mathbf{5}]$

(1) 0<r2 $r2 \in [\mathbf{1} .. \mathbf{5}]$

(2) 3<r2 $\Rightarrow r2 \in [\mathbf{4} .. \mathbf{5}]$

(1) 0<r3 $r3 \in [\mathbf{1} .. \mathbf{5}]$

(2) 3<r3 $\Rightarrow r3 \in [\mathbf{4} .. \mathbf{5}]$

Fairness Problem - Solution

$P(x) \leftarrow 0 < x \quad (1) \qquad \neg P(x) \vee Q(x) \leftarrow 3 < x \quad (2)$

$\neg P(x) \leftarrow x = 2 \quad (3)$

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1)	0<r1		r1 ∈ [**1** .. **5**]
(2)	3<r1	⇒	r1 ∈ [**4** .. **5**]
(1)	0<r2		r2 ∈ [**1** .. **5**]
(2)	3<r2	⇒	r2 ∈ [**4** .. **5**]
(1)	0<r3		r3 ∈ [**1** .. **5**]
(2)	3<r3	⇒	r3 ∈ [**4** .. **5**]

Impossible:
r1, r2, r3 ∈ [**4** .. **5**] with
r1 ≠ r2, r1 ≠ r3, r2 ≠ r3

Fairness Problem - Solution

$P(x) \leftarrow 0 < x$ (1) $\qquad \neg P(x) \vee Q(x) \leftarrow 3 < x$ (2)

$\neg P(x) \leftarrow x = 2$ (3)

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1)	0<r1		r1 ∈ [**1** .. **5**]
(2)	3<r1	⇒	r1 ∈ [**4** .. **5**]
(1)	0<r2		r2 ∈ [**1** .. **5**]
(2)	3<r2	⇒	r2 ∈ [**4** .. **5**]
(1)	0<r3		r3 ∈ [**1** .. **5**]
~~(2)~~	~~3<r3~~	~~⇒~~	~~r3 ∈ [**4** .. **5**]~~
(3)	r3=2	⇒	r3 ∈ [**2** .. **2**]

Impossible:
r1, r2, r3 ∈ [**4** .. **5**] with
r1 ≠ r2, r1 ≠ r3, r2 ≠ r3

Fairness Problem - Solution

$P(x) \leftarrow 0 < x$ (1) $\qquad \neg P(x) \lor Q(x) \leftarrow 3 < x$ (2)

$\neg P(x) \leftarrow x = 2$ (3)

- Slice BG domain into finite segments [1 .. 5] [6 .. 10] [11 .. 15] ...
- Fix one such domain for each rigid variable
- Provide enough rigid vars eventually to fully cover each segment

(1)	$0<r1$	$r1 \in [\mathbf{1} .. \mathbf{5}]$
(2)	$3<r1$	$\Rightarrow r1 \in [\mathbf{4} .. \mathbf{5}]$
(1)	$0<r2$	$r2 \in [\mathbf{1} .. \mathbf{5}]$
(2)	$3<r2$	$\Rightarrow r2 \in [\mathbf{4} .. \mathbf{5}]$
(1)	$0<r3$	$r3 \in [\mathbf{1} .. \mathbf{5}]$
~~(2)~~	~~$3<r3$~~	~~$\Rightarrow r3 \in [\mathbf{4} .. \mathbf{5}]$~~
(3)	$r3=2$	$\Rightarrow r3 \in [\mathbf{2} .. \mathbf{2}]$

Impossible:
$r1, r2, r3 \in [\mathbf{4} .. \mathbf{5}]$ with
$r1 \neq r2, r1 \neq r3, r2 \neq r3$

now close with (3)

Conclusions

- Sketched the new MEE(T) calculus
- Intended for application in software verification, analysis of constraint problems, analysis of business rules and process models
- If it terminates without a refutation it provides a model
  - Useful for countermodel finding
  - Decides extensions of Bernays-Schoenfinkel fragment
- Issues/Future work
  - Lack of guidance for instantiation substitution γ
  - Expensive background satisfiability check called frequently
  - Background-sorted foreground operators, as in $\mathrm{car}(\mathrm{cons}(x, l)) \approx x \ \leftarrow\ \emptyset \cdot \emptyset$
- Download implementation: see my home page