Automating induction for equational reasoning in superposition

Márton Hajdu

September 12, 2022

TECHNISCHE UNIVERSITÄT WIEN

Motivating example

Prove: $\forall x, y. \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(x), y))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y)))$

Motivating example

Prove: $\forall x, y.\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(x), y))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y)))$

---

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]

Motivating example

Prove: $\forall x, y.\texttt{len}(\texttt{rev}(\texttt{app}(f(x), y))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(x), \texttt{len}(y)))$

---

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]

Motivating example

Prove: $\forall x, y.\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(x), y))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y)))$

---

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ [lemma]

Motivating example

Prove: $\forall x, y.\texttt{len}(\texttt{rev}(\texttt{app}(f(x), y))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(x), \texttt{len}(y)))$

1.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$	[axiom]
4.	$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$	[lemma]
5.	$\texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2)) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[Sup 1,4]

Motivating example

Prove: $\forall x, y. \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(x), y))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y)))$

---

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ [lemma]
5. $\mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2)) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [Sup 1,4]
6. $\mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2)) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [Sup 5,3]

Motivating example

Prove: $\forall x, y. \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(x), y))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y)))$

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$	[lemma]
5.	$\mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2)) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[Sup 1,4]
6.	$\mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2)) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[Sup 5,3]
7.	$\mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[Sup 6,2]

Motivating example

Prove: $\forall x, y.\texttt{len}(\texttt{rev}(\texttt{app}(f(x), y))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(x), \texttt{len}(y)))$

1.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$	[axiom]
4.	$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$	[lemma]
5.	$\texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2)) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[Sup 1,4]
6.	$\texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2)) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[Sup 5,3]
7.	$\texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[Sup 6,2]
8.	$\square$	[ER 7]

Motivating example

Prove: $\forall x, y.\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(x), y))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y)))$

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$	[lemma]
5.	$\mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2)) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[Sup 1,4]
6.	$\mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2)) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[Sup 5,3]
7.	$\mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[Sup 6,2]
8.	$\square$	[ER 7]

Observation: The lemma (4.) can be automatically generated by equational reasoning from clauses 1-3.

Motivation

- Superposition – first-order reasoning $\leftrightarrow$ Proof-assistant/user – inductive reasoning

Motivation

- Superposition – first-order reasoning $\leftrightarrow$ Proof-assistant/user – inductive reasoning
- Automated induction in superposition works well for problems where all non-trivial (synthetic) lemmas are supplied as input

Motivation

- Superposition – first-order reasoning $\leftrightarrow$ Proof-assistant/user – inductive reasoning
- Automated induction in superposition works well for problems where all non-trivial (synthetic) lemmas are supplied as input
- But superposition restrictions prohibit deriving even some trivial (analytic) lemmas (modulo equational reasoning)

Motivation

- Superposition – first-order reasoning $\leftrightarrow$ Proof-assistant/user – inductive reasoning
- Automated induction in superposition works well for problems where all non-trivial (synthetic) lemmas are supplied as input
- But superposition restrictions prohibit deriving even some trivial (analytic) lemmas (modulo equational reasoning)
- **Goal:** extend the calculus to derive such trivial lemmas (modulo E) and prove them with induction

Motivation

- Superposition – first-order reasoning $\leftrightarrow$ Proof-assistant/user – inductive reasoning
- Automated induction in superposition works well for problems where all non-trivial (synthetic) lemmas are supplied as input
- But superposition restrictions prohibit deriving even some trivial (analytic) lemmas (modulo equational reasoning)
- **Goal:** extend the calculus to derive such trivial lemmas (modulo E) and prove them with induction
- We restrict the conjectures to universally quantified unit equalities

Superposition vs Paramodulation

Superposition:

$$\frac{l \simeq r \lor C \quad s[l'] \bowtie t \lor D}{(s[r] \bowtie t \lor C \lor D)\theta} \ (Sup)$$

- $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$,
- $r\theta \not\succeq l\theta$,
- $t\theta \not\succeq s[l']\theta$,
- $(l \simeq r)\theta \succ C\theta$ and
- $(s \bowtie t)\theta \succ D\theta$.

Equality resolution:

$$\frac{s \not\simeq t \lor C}{C\theta} \ (ER)$$

where $\theta = mgu(s, t)$.

Equality factoring:

$$\frac{s \simeq t \lor s' \simeq t' \lor C}{(s \simeq t \lor t \not\simeq t' \lor C)\theta} \ (EF)$$

where $\theta = mgu(s, s')$, $t\theta \not\succeq s\theta$, and $t'\theta \not\succeq t\theta$.

Superposition vs Paramodulation

Superposition:

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta} \ (Sup)$$

- $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$,
- $r\theta \not\succeq l\theta$,
- $t\theta \not\succeq s[l']\theta$,
- $(l \simeq r)\theta \succ C\theta$ and
- $(s \bowtie t)\theta \succ D\theta$.

Equality resolution:

$$\frac{s \not\simeq t \vee C}{C\theta} \ (ER)$$

where $\theta = mgu(s, t)$.

Equality factoring:

$$\frac{s \simeq t \vee s' \simeq t' \vee C}{(s \simeq t \vee t \not\simeq t' \vee C)\theta} \ (EF)$$

where $\theta = mgu(s, s')$, $t\theta \not\succeq s\theta$, and $t'\theta \not\succeq t\theta$.

$\rightarrow$ efficient if properly implemented but not consequence-complete

Superposition vs Paramodulation

Paramodulation:

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta} \; (Par)$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$.

Superposition vs Paramodulation

Paramodulation:

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta} \ (Par)$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$.

$\rightarrow$ very prolific together with equality axioms (reflexivity, symmetry, transitivity, congruence) but can derive much more

Induction and superposition

$$\frac{\overline{L}[t] \vee C}{\texttt{cnf}(\neg F) \vee C}\ (\texttt{Ind}),$$

where $\overline{L}[t]$ is ground and $F \rightarrow \forall x.L[x]$ is a valid induction formula.

Induction and superposition

$$\frac{\overline{L}[t] \vee C}{\texttt{cnf}(\neg F) \vee C} \ (\texttt{Ind}),$$

where $\overline{L}[t]$ is ground and $F \rightarrow \forall x.L[x]$ is a valid induction formula.

A lemma $\forall \overline{x}.G[\overline{x}]$ is generatable from a set of clauses $\mathcal{C}$ if the negation of one of its instances $\neg G[\overline{t}]$ can be inferred from $\mathcal{C}$ with equality reasoning.

$$\mathcal{C} \vdash_E \neg G[\overline{t}]$$

Induction and superposition

$$\frac{\overline{L}[t] \vee C}{\mathtt{cnf}(\neg F) \vee C} \; (\mathtt{Ind}),$$

where $\overline{L}[t]$ is ground and $F \to \forall x.L[x]$ is a valid induction formula.

A lemma $\forall \overline{x}.G[\overline{x}]$ is generatable from a set of clauses $\mathcal{C}$ if the negation of one of its instances $\neg G[\overline{t}]$ can be inferred from $\mathcal{C}$ with equality reasoning.

$$\mathcal{C} \vdash_E \neg G[\overline{t}]$$

By generating lemma $\forall x.L[x]$, we informally mean that $(\mathtt{Ind})$ is used with a valid induction formula $F \to \forall x.L[x]$.

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

---

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

$$\left(\forall y, z. \left(\begin{array}{c} \texttt{len}(\texttt{rev}(\texttt{app}(\texttt{nil}, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\texttt{nil}), \texttt{len}(\sigma_2)) \\ \texttt{len}(\texttt{rev}(\texttt{app}(z, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(z), \texttt{len}(\sigma_2)) \rightarrow \\ \texttt{len}(\texttt{rev}(\texttt{app}(\texttt{cons}(y, z), \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\texttt{cons}(y, z), \texttt{len}(\sigma_2)) \end{array} \right) \right)$$

$$\rightarrow \forall x.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(\sigma_2))$$

1. l ... [...conj.]
2. ∀ ... [axiom]
3. ∀ ... [axiom]
4. l ... [...ar 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [Par 4,3]

---

6. $\texttt{len}(\texttt{rev}(\texttt{app}(\texttt{nil}, \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(\texttt{nil}), \texttt{len}(\sigma_2))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$ [Ind 5]
7. $\texttt{len}(\texttt{rev}(\texttt{app}(\texttt{nil}, \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(\texttt{nil}), \texttt{len}(\sigma_2))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\texttt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(\texttt{cons}(\sigma_3, \sigma_4)), \texttt{len}(\sigma_2))$ [Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{nil}), \mathtt{len}(\sigma_2))$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{nil}), \mathtt{len}(\sigma_2))$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_3, \sigma_4)), \mathtt{len}(\sigma_2))$	[Ind 5]

$\mathtt{len}(\mathtt{nil}) = 0$

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

---

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [Par 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [Par 4,3]

---

6. $\texttt{len}(\texttt{rev}(\texttt{app}(\texttt{nil}, \sigma_2))) \not\simeq \texttt{plus}(0, \texttt{len}(\sigma_2))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$ [Ind 5]
7. $\texttt{len}(\texttt{rev}(\texttt{app}(\texttt{nil}, \sigma_2))) \not\simeq \texttt{plus}(0, \texttt{len}(\sigma_2))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\texttt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(\texttt{cons}(\sigma_3, \sigma_4)), \texttt{len}(\sigma_2))$ [Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{plus}(0, \mathtt{len}(\sigma_2))$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{plus}(0, \mathtt{len}(\sigma_2))$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_3, \sigma_4)), \mathtt{len}(\sigma_2))$	[Ind 5]

$\forall y.\mathtt{plus}(0, y) = y$

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_3, \sigma_4)), \mathtt{len}(\sigma_2))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{nil}, \sigma_2))) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_3, \sigma_4)), \mathtt{len}(\sigma_2))$	[Ind 5]

$\forall y. \mathtt{app}(\mathtt{nil}, y) = y$

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_3, \sigma_4)), \mathtt{len}(\sigma_2))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}($ [illegible] $\mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_3, \sigma_4)), \mathtt{len}(\sigma_2))$	[Ind 5]

$\forall x, y.\mathtt{len}(\mathtt{cons}(x, y)) = \mathtt{s}(\mathtt{len}(y))$

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_4)), \mathtt{len}(\sigma_2))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

---

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{l}$
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [Par 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [Par 4,3]

---

6. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\lor\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$ [Ind 5]
7. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\lor\ \texttt{len}(\texttt{rev}(\texttt{app}(\texttt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_4)), \texttt{len}(\sigma_2))$ [Ind 5]

Note: clause 2 is oriented the other way!
(compared to the inference of 4)

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2)))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}($ …	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\mathtt{cons}(\sigma_3, \sigma_4), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2)))$	[Ind 5]

$\forall x, y, z. \mathtt{app}(\mathtt{cons}(x, y), z) = \mathtt{cons}(x, \mathtt{app}(y, z))$

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{cons}(\sigma_3, \mathtt{app}(\sigma_4, \sigma_5)))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2)))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$	
	$\vee\ \mathtt{len}$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$	
	$\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{cons}(\sigma_3, \mathtt{app}(\sigma_4, \sigma_5)))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2)))$	[Ind 5]

$\forall x, y. \mathtt{rev}(\mathtt{cons}(x, y)) = \mathtt{app}(\mathtt{rev}(x), \mathtt{cons}(y, \mathtt{nil}))$

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

1.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$	[axiom]
4.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$	[Par 1,2]
5.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$	[Par 4,3]
6.	$\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$ $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$	[Ind 5]
7.	$\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$ $\vee\ \texttt{len}(\texttt{app}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2)), \texttt{cons}(\sigma_3, \texttt{nil}))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2)))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

$$\left(\begin{array}{c} \texttt{len}(\texttt{rev}(\texttt{nil})) \simeq \texttt{len}(\texttt{nil}) \\ \forall y, z.\ \left(\begin{array}{c} \texttt{len}(\texttt{rev}(z)) \simeq \texttt{len}(z) \rightarrow \\ \texttt{len}(\texttt{rev}(\texttt{cons}(y, z))) \simeq \texttt{len}(\texttt{cons}(y, z)) \end{array}\right) \end{array}\right)$$
$$\rightarrow \forall x.\ \texttt{len}(\texttt{rev}(x)) \simeq \texttt{len}(x)$$

1. [neg.conj.]
2. [axiom]
3. [axiom]
4. [Par 1,2]
5. [Par 4,3]

---

6. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$ [Ind 5]
7. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\vee\ \texttt{len}(\texttt{app}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2)), \texttt{cons}(\sigma_3, \texttt{nil}))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2)))$ [Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

---

1. [neg.conj.]
2. [axiom]
3. [axiom]
4. [Par 1,2]
5. [Par 4,3]

$$\left(\begin{array}{c} \texttt{len}(\texttt{rev}(\texttt{nil})) \simeq \texttt{len}(\texttt{nil}) \\ \forall y, z. \left(\begin{array}{c} \texttt{len}(\texttt{rev}(z)) \simeq \texttt{len}(z) \rightarrow \\ \texttt{len}(\texttt{rev}(\texttt{cons}(y, z))) \simeq \texttt{len}(\texttt{cons}(y, z)) \end{array} \right) \end{array} \right) \rightarrow \forall x.\ \texttt{len}(\texttt{rev}(x)) \simeq \texttt{len}(x)$$

and one more induction...

---

6. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$ [Ind 5]
7. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\vee\ \texttt{len}(\texttt{app}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2)), \texttt{cons}(\sigma_3, \texttt{nil}))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2)))$ [Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$	[Par 4,3]
6.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2))$	[Ind 5]
7.	$\mathtt{len}(\mathtt{rev}(\sigma_2)) \not\simeq \mathtt{len}(\sigma_2)$ $\vee\ \mathtt{len}(\mathtt{app}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2)), \mathtt{cons}(\sigma_3, \mathtt{nil}))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_4), \mathtt{len}(\sigma_2)))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [Par 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [Par 4,3]

---

6. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$ [Ind 5]
7. $\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$
   $\vee\ \texttt{len}(\texttt{app}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2)), \texttt{cons}(\sigma_3, \texttt{nil}))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2)))$ [Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \texttt{len}(\texttt{rev}(\texttt{app}(x, y))) \simeq \texttt{plus}(\texttt{len}(x), \texttt{len}(y))$ is generatable from 1-3.:

1.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$	[axiom]
3.	$\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$	[axiom]
4.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$	[Par 1,2]
5.	$\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$	[Par 4,3]
6.	$\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$ $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))) \simeq \texttt{plus}(\texttt{len}(\sigma_4), \texttt{len}(\sigma_2))$	[Ind 5]
7.	$\texttt{len}(\texttt{rev}(\sigma_2)) \not\simeq \texttt{len}(\sigma_2)$ $\vee\ \texttt{len}(\texttt{app}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2)), \texttt{cons}(\sigma_3, \texttt{nil}))) \not\simeq \texttt{s}(\texttt{len}(\texttt{rev}(\texttt{app}(\sigma_4, \sigma_2))))$	[Ind 5]

Motivating example revisited – lemma generation

$\forall x, y.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(x, y))) \simeq \mathtt{plus}(\mathtt{len}(x), \mathtt{len}(y))$ is generatable from 1-3.:

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}($ [Par 1,2]
5. $\mathtt{len}($ [Par 4,3]

---

6. $\mathtt{len}($
   $\vee$ [Ind 5]
7. $\mathtt{len}($
   $\vee\ \mathtt{len}(\mathtt{app}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2)), \mathtt{cons}(\sigma_3, \mathtt{nil}))) \not\simeq \mathtt{s}(\mathtt{len}(\mathtt{rev}(\mathtt{app}(\sigma_4, \sigma_2))))$ [Ind 5]

$$\left(\begin{array}{c} \mathtt{len}(\mathtt{app}(\mathtt{nil}, \mathtt{cons}(\sigma_3, \mathtt{nil}))) \simeq \mathtt{s}(\mathtt{len}(\mathtt{nil})) \\ \forall y, z.\ \left(\begin{array}{c} \mathtt{len}(\mathtt{app}(z, \mathtt{cons}(\sigma_3, \mathtt{nil}))) \simeq \mathtt{s}(\mathtt{len}(z)) \rightarrow \\ \mathtt{len}(\mathtt{app}(\mathtt{cons}(y, z), \mathtt{cons}(\sigma_3, \mathtt{nil}))) \simeq \mathtt{s}(\mathtt{len}(\mathtt{cons}(y, z))) \end{array}\right) \end{array}\right)$$
$$\rightarrow \forall x.\ \mathtt{len}(\mathtt{app}(x, \mathtt{cons}(\sigma_3, \mathtt{nil}))) \simeq \mathtt{s}(\mathtt{len}(x))$$

Paramodulation

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$.

Paramodulation

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$.

This inference is too coarse-grained:

Paramodulation

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$.

This inference is too coarse-grained:

- It overlaps with superposition inferences

Paramodulation

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$.

This inference is too coarse-grained:

- It overlaps with superposition inferences
- For efficiency reasons, we want to differentiate between $r\theta \not\succeq l\theta$ and $r\theta \succeq l\theta$

Paramodulation - split

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$

Paramodulation - split

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$

Restrictions:

(1) $r\theta \not\succeq l\theta$

(2) $t\theta \not\succeq s[l']\theta$

(3) $(l \simeq r)\theta \succ C\theta$ and $(s \bowtie t)\theta \succ D\theta$

Paramodulation - split

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$

Restrictions:

(1) $r\theta \not\succeq l\theta$

(2) $t\theta \not\succeq s[l']\theta$

(3) $(l \simeq r)\theta \succ C\theta$ and $(s \bowtie t)\theta \succ D\theta$

The following 3 inferences cover (Par):

- Superposition (Sup): (1)-(3) hold.

Paramodulation - split

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$

Restrictions:

(1) $r\theta \not\succeq l\theta$

(2) $t\theta \not\succeq s[l']\theta$

(3) $(l \simeq r)\theta \succ C\theta$ and $(s \bowtie t)\theta \succ D\theta$

The following 3 inferences cover (Par):

- Superposition (Sup): (1)-(3) hold.
- Forward co-superposition $(FCSup)$: (1) holds and at least one of (2) and (3) does not hold.

Paramodulation - split

$$\frac{l \simeq r \vee C \quad s[l'] \bowtie t \vee D}{(s[r] \bowtie t \vee C \vee D)\theta}$$

where $\bowtie \in \{\simeq, \not\simeq\}$ and $\theta = mgu(l, l')$

Restrictions:

(1) $r\theta \not\succeq l\theta$

(2) $t\theta \not\succeq s[l']\theta$

(3) $(l \simeq r)\theta \succ C\theta$ and $(s \bowtie t)\theta \succ D\theta$

The following 3 inferences cover (Par):

- Superposition (Sup): (1)-(3) hold.
- Forward co-superposition $(FCSup)$: (1) holds and at least one of (2) and (3) does not hold.
- Backward paramodulation $(BPar)$: (1) does not hold.

General paramodulation - split

With the 3 inferences (*Sup*, *FCSup*, *BPar*) we can perform the inductive reasoning **goal-oriented**:

General paramodulation - split

With the 3 inferences (*Sup*, *FCSup*, *BPar*) we can perform the inductive reasoning **goal-oriented**:

- *Sup* is used for first-order reasoning

General paramodulation - split

With the 3 inferences (*Sup*, *FCSup*, *BPar*) we can perform the inductive reasoning **goal-oriented**:

- *Sup* is used for first-order reasoning
- *FCSup* and *BPar* are used only on inductive goals (e.g. negative, ground, unit clauses)

General paramodulation - split

With the 3 inferences (*Sup*, *FCSup*, *BPar*) we can perform the inductive reasoning **goal-oriented**:

- *Sup* is used for first-order reasoning
- *FCSup* and *BPar* are used only on inductive goals (e.g. negative, ground, unit clauses)
- most conclusions from *BPar* are redundant w.r.t. their premises, so they need special handling to avoid deletion

General paramodulation - split

With the 3 inferences (*Sup*, *FCSup*, *BPar*) we can perform the inductive reasoning **goal-oriented**:

- *Sup* is used for first-order reasoning
- *FCSup* and *BPar* are used only on inductive goals (e.g. negative, ground, unit clauses)
- most conclusions from *BPar* are redundant w.r.t. their premises, so they need special handling to avoid deletion
- some chains of inferences are preferably avoided, e.g.

$$L[l'] \vee D \leadsto (L[r] \vee C \vee D)\theta \leadsto (L[l] \vee C \vee D)\theta \leadsto (L[r] \vee C \vee D)\theta$$

where $l \simeq r \vee C$ is a clause, $\theta = mgu(l, l')$ and $r\theta \not\succeq l\theta$

General paramodulation - split

With the 3 inferences (*Sup*, *FCSup*, *BPar*) we can perform the inductive reasoning **goal-oriented**:

- *Sup* is used for first-order reasoning
- *FCSup* and *BPar* are used only on inductive goals (e.g. negative, ground, unit clauses)
- most conclusions from *BPar* are redundant w.r.t. their premises, so they need special handling to avoid deletion
- some chains of inferences are preferably avoided, e.g.

2. clause subsumes 4. clause!

$$L[l'] \vee D \rightsquigarrow (L[r] \vee C \vee D)\theta \rightsquigarrow (L[l] \vee C \vee D)\theta \rightsquigarrow (L[r] \vee C \vee D)\theta$$

where $l \simeq r \vee C$ is a clause, $\theta = mgu(l, l')$ and $r\theta \not\succeq l\theta$

Handling *BPar* conclusions

- **Idea:** a *BPar* conclusion can be only used by *BPar* and *Ind*

Handling *BPar* conclusions

- **Idea:** a *BPar* conclusion can be only used by *BPar* and *Ind*
- Inferences can be ordered such that *BPar*s are the last generating inferences on a clause before an *Ind* inference

Handling *BPar* conclusions

- **Idea:** a *BPar* conclusion can be only used by *BPar* and *Ind*
- Inferences can be ordered such that *BPar*s are the last generating inferences on a clause before an *Ind* inference
- This can be done for ground clauses, e.g.

$$\cfrac{\cfrac{\cfrac{\cdots}{L[s][l] \vee C} \qquad \cfrac{\cdots}{t \simeq s \vee D}}{L[t][l] \vee C \vee D} \qquad \cfrac{\cdots}{l \simeq r \vee D'}}{\cfrac{L[t][r] \vee C \vee D \vee D'}{\cdots}} \Longrightarrow \cfrac{\cfrac{\cfrac{\cdots}{L[s][l] \vee C} \qquad \cfrac{\cdots}{l \simeq r \vee D'}}{L[s][r] \vee C \vee D'} \qquad \cfrac{\cdots}{t \simeq s \vee D}}{\cfrac{L[t][r] \vee C \vee D \vee D'}{\cdots}}$$

where $t \succ s$ and $l \succ r$.

Handling *BPar* conclusions

- **Idea:** a *BPar* conclusion can be only used by *BPar* and *Ind*
- Inferences can be ordered such that *BPar*s are the last generating inferences on a clause before an *Ind* inference
- This can be done for ground clauses, e.g.

$$\dfrac{\dfrac{\overset{\cdots}{L[s][l] \vee C} \qquad \overset{\cdots}{t \simeq s \vee D}}{L[t][l] \vee C \vee D} \qquad \overset{\cdots}{l \simeq r \vee D'}}{\underset{\cdots}{L[t][r] \vee C \vee D \vee D'}} \Longrightarrow \dfrac{\dfrac{\overset{\cdots}{L[s][l] \vee C} \qquad \overset{\cdots}{l \simeq r \vee D'}}{L[s][r] \vee C \vee D'} \qquad \overset{\cdots}{t \simeq s \vee D}}{\underset{\cdots}{L[t][r] \vee C \vee D \vee D'}}$$

  where $t \succ s$ and $l \succ r$.
- As a side effect, we have to perform *FCSup* inferences on all clauses (not just goal-clauses)

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

$\ldots$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

   ...

8. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\mathtt{nil})), \mathtt{len}(\sigma_2))$
   $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_6), \sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(f(\sigma_6)), \mathtt{len}(\sigma_2))$ [Ind 5]
9. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\mathtt{nil})), \mathtt{len}(\sigma_2))$
   $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{cons}(\sigma_5, \sigma_6)), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\mathtt{cons}(\sigma_5, \sigma_6))), \mathtt{len}(\sigma_2))$ [Ind 5]

Redundant induction formulas

1.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1),\sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1),\mathtt{len}(\sigma_2)))$	[neg.conj.]
2.	$\forall x,y.\ \mathtt{plus}(\mathtt{s}(x),y) \simeq \mathtt{s}(\mathtt{plus}(x,y))$	[axiom]
3.	$\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$	[axiom]
4.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1),\sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)),\mathtt{len}(\sigma_2))$	[BPar 1,2]
5.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1),\sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)),\mathtt{len}(\sigma_2))$	[BPar 4,3]
	...	
8.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}),\sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\mathtt{nil})),\mathtt{len}(\sigma_2))$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_6),\sigma_2))) \simeq \mathtt{plus}(\mathtt{len}(f(\sigma_6)),\mathtt{len}(\sigma_2))$	[Ind 5]
9.	$\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}),\sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\mathtt{nil})),\mathtt{len}(\sigma_2))$ $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{cons}(\sigma_5,\sigma_6)),\sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\mathtt{cons}(\sigma_5,\sigma_6))),\mathtt{len}(\sigma_2))$	[Ind 5]

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

   ...

8. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\mathtt{nil})), \mathtt{len}(\sigma_2))$
   $\lor\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_6), \sigma_2))) \simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_6)), \mathtt{len}(\sigma_2))$ [Ind 5]
9. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\mathtt{nil})), \mathtt{len}(\sigma_2))$
   $\lor\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{cons}(\sigma_5, \sigma_6)), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\mathtt{cons}(\sigma_5, \sigma_6))), \mathtt{len}(\sigma_2))$ [Ind 5]

Redundant induction formulas

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 4,3]

   ...

8. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\texttt{nil})), \texttt{len}(\sigma_2))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_6), \sigma_2))) \simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_6)), \texttt{len}(\sigma_2))$ [Ind 5]
9. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\texttt{nil})), \texttt{len}(\sigma_2))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{cons}(\sigma_5, \sigma_6)), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\texttt{cons}(\sigma_5, \sigma_6))), \texttt{len}(\sigma_2))$ [Ind 5]

Redundant induction formulas

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 4,3]

   ...

8. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{nil}), \texttt{len}(\sigma_2)))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_6), \sigma_2))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_6), \texttt{len}(\sigma_2)))$ [Ind 5]
9. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{nil}), \texttt{len}(\sigma_2)))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{cons}(\sigma_5, \sigma_6)), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{cons}(\sigma_5, \sigma_6)), \texttt{len}(\sigma_2)))$ [Ind 5]

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. [axiom]
3. [axiom]
4. Par 1,2]
5. Par 4,3]

$$\left(\forall y, z. \left(\begin{array}{c} \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2))) \wedge \\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(z), \sigma_2))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(z), \mathtt{len}(\sigma_2))) \rightarrow \\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{cons}(y,z)), \sigma_2))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\mathtt{cons}(y,z)), \mathtt{len}(\sigma_2))) \end{array}\right)\right)$$
$$\rightarrow \forall x.\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(x), \sigma_2))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(x), \mathtt{len}(\sigma_2)))$$

8. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\mathtt{nil}), \mathtt{len}(\sigma_2)))$
   $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_6), \sigma_2))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_6), \mathtt{len}(\sigma_2)))$ [Ind 5]
9. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\mathtt{nil}), \mathtt{len}(\sigma_2)))$
   $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{cons}(\sigma_5, \sigma_6)), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_5, \sigma_6)), \mathtt{len}(\sigma_2)))$ [Ind 5]

10. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\mathtt{nil}), \mathtt{len}(\sigma_2)))$
    $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_8), \sigma_2))) \simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_8), \mathtt{len}(\sigma_2)))$ [Ind 1]
11. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{nil}), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\mathtt{nil}), \mathtt{len}(\sigma_2)))$
    $\vee\ \mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\mathtt{cons}(\sigma_7, \sigma_8)), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\mathtt{cons}(\sigma_7, \sigma_8))), \mathtt{len}(\sigma_2)))$ [Ind 1]

Redundant induction formulas

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1),\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1),\texttt{len}(\sigma_2)))$ [neg.conj.]
2. [axiom]
3. [axiom]
4. Par 1,2]
5. Par 4,3]

$$\left(\forall y,z.\left(\begin{array}{c}\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}),\sigma_2))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1),\texttt{len}(\sigma_2)))\wedge \\ \texttt{len}(\texttt{rev}(\texttt{app}(f(z),\sigma_2))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(z),\texttt{len}(\sigma_2))) \rightarrow \\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{cons}(y,z)),\sigma_2))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{cons}(y,z)),\texttt{len}(\sigma_2)))\end{array}\right)\right)$$
$$\rightarrow \forall x.\ \texttt{len}(\texttt{rev}(\texttt{app}(f(x),\sigma_2))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(x),\texttt{len}(\sigma_2)))$$

8. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}),\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{nil}),\texttt{len}(\sigma_2)))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_6),\sigma_2))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_6),\texttt{len}(\sigma_2)))$
9. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}),\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{nil}),\texttt{len}(\sigma_2)))$
   $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{cons}(\sigma_5,\sigma_6)),\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{cons}(\sigma_5,\sigma_6)),\texttt{len}(\sigma_2)))$ [Ind 5]

These pairs of clauses are equivalent (due to the freshness of Skolems)!

10. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}),\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{nil}),\texttt{len}(\sigma_2)))$
    $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_8),\sigma_2))) \simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_8),\texttt{len}(\sigma_2)))$ [Ind 1]
11. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{nil}),\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{nil}),\texttt{len}(\sigma_2)))$
    $\vee\ \texttt{len}(\texttt{rev}(\texttt{app}(f(\texttt{cons}(\sigma_7,\sigma_8)),\sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\texttt{cons}(\sigma_7,\sigma_8))),\texttt{len}(\sigma_2)))$ [Ind 1]

Redundant induction formulas

Checking such equivalences may require arbitrarily many simplifications (depending on *BPar* depth) and checking clause equivalence modulo Skolems.

Redundant induction formulas

Checking such equivalences may require arbitrarily many simplifications (depending on *BPar* depth) and checking clause equivalence modulo Skolems.

Alternatively, modify induction rule:

$$\frac{\overline{L}[t] \vee C}{\mathtt{cnf}(\neg F) \vee C}\ (\mathtt{Ind}),$$

where $\overline{L}[t]$ is ground and $F \rightarrow \forall x. L[x]$ is a valid induction formula and there is no clause $l \simeq r$ s.t.

Redundant induction formulas

Checking such equivalences may require arbitrarily many simplifications (depending on *BPar* depth) and checking clause equivalence modulo Skolems.

Alternatively, modify induction rule:

$$\frac{\overline{L}[t] \vee C}{\texttt{cnf}(\neg F) \vee C}\ (\texttt{Ind}),$$

where $\overline{L}[t]$ is ground and $F \to \forall x. L[x]$ is a valid induction formula and there is no clause $l \simeq r$ s.t.

- $l\theta \lhd \overline{L}[c]$ for some θ and fresh constant c and
- $(l \succ r)\theta$ and
- $\texttt{cnf}(\neg F) \vee C$ contains no positive unit depending on $\overline{L}[\cdot]$ or (let $\overline{L}[c] = (s[l\theta] \bowtie u)$) $l\theta \neq s$ or θ is not a renaming. [IJCAR22]

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

   ...

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \to L[\mathsf{cons}(y, z)])\big) \to \forall x.L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \to L[\mathsf{cons}(y, z)])\big) \to \forall x.L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \rightarrow L[\mathrm{cons}(y, z)])\big) \rightarrow \forall x.L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \to L[\mathrm{cons}(y, z)])\big) \to \forall x.L[x]$$

Redundant induction formulas

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\texttt{nil}] \land \forall y, z.(L[z] \to L[\mathsf{cons}(y, z)])\big) \to \forall x.L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \to L[\mathsf{cons}(y, z)])\big) \to \forall x.L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \wedge \forall y, z.(L[z] \rightarrow L[\mathsf{cons}(y, z)])\big) \rightarrow \forall x.L[x]$$

Redundant induction formulas

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 4,3]

   ...

$$\big(L[\texttt{nil}] \land \forall y, z.(L[z] \to L[\texttt{cons}(y, z)])\big) \to \forall x. L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \to L[\mathrm{cons}(y, z)])\big) \to \forall x.L[x]$$

Redundant induction formulas

1. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{s}(\texttt{plus}(\texttt{len}(\sigma_1), \texttt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \texttt{plus}(\texttt{s}(x), y) \simeq \texttt{s}(\texttt{plus}(x, y))$ [axiom]
3. $\forall y.\ \texttt{len}(f(y)) \simeq \texttt{s}(\texttt{len}(y))$ [axiom]
4. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{s}(\texttt{len}(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 1,2]
5. $\texttt{len}(\texttt{rev}(\texttt{app}(f(\sigma_1), \sigma_2))) \not\simeq \texttt{plus}(\texttt{len}(f(\sigma_1)), \texttt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\texttt{nil}] \land \forall y, z.(L[z] \rightarrow L[\texttt{cons}(y, z)])\big) \rightarrow \forall x. L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \rightarrow L[\mathtt{cons}(y, z)])\big) \rightarrow \forall x.L[x]$$

Redundant induction formulas

1. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{s}(\mathtt{plus}(\mathtt{len}(\sigma_1), \mathtt{len}(\sigma_2)))$ [neg.conj.]
2. $\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$ [axiom]
3. $\forall y.\ \mathtt{len}(f(y)) \simeq \mathtt{s}(\mathtt{len}(y))$ [axiom]
4. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{s}(\mathtt{len}(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 1,2]
5. $\mathtt{len}(\mathtt{rev}(\mathtt{app}(f(\sigma_1), \sigma_2))) \not\simeq \mathtt{plus}(\mathtt{len}(f(\sigma_1)), \mathtt{len}(\sigma_2))$ [BPar 4,3]

...

$$\big(L[\mathtt{nil}] \land \forall y, z.(L[z] \to L[\mathrm{cons}(y, z)])\big) \to \forall x.L[x]$$

Optimizing the simplification ordering

Couldn't we use a simplification ordering where *BPar* is not needed?

Optimizing the simplification ordering

Couldn't we use a simplification ordering where *BPar* is not needed?

- Different orderings can make a huge difference in performance and deriving power

Optimizing the simplification ordering

Couldn't we use a simplification ordering where *BPar* is not needed?

- Different orderings can make a huge difference in performance and deriving power
- E.g. with LPO one can always orient non-mutually recursive function definition axioms s.t. the intended orientation is preserved

Optimizing the simplification ordering

Couldn't we use a simplification ordering where *BPar* is not needed?

- Different orderings can make a huge difference in performance and deriving power
- E.g. with LPO one can always orient non-mutually recursive function definition axioms s.t. the intended orientation is preserved
- Still, some need to be used in both orientations, e.g.

$$\forall x, y.\ \mathtt{plus}(\mathtt{s}(x), y) \simeq \mathtt{s}(\mathtt{plus}(x, y))$$

Conclusion & Future work

Conclusion

- Lemma generation for induction is done for equational reasoning with 2 special inferences

Conclusion & Future work

Conclusion

- Lemma generation for induction is done for equational reasoning with 2 special inferences
- Induction rule is modified to check for redundant inductions efficiently

Conclusion & Future work

Conclusion

- Lemma generation for induction is done for equational reasoning with 2 special inferences
- Induction rule is modified to check for redundant inductions efficiently

Conclusion & Future work

Conclusion

- Lemma generation for induction is done for equational reasoning with 2 special inferences
- Induction rule is modified to check for redundant inductions efficiently

Future work

- The demodulation check can be extended to subsumption demodulation

Conclusion & Future work

Conclusion

- Lemma generation for induction is done for equational reasoning with 2 special inferences
- Induction rule is modified to check for redundant inductions efficiently

Future work

- The demodulation check can be extended to subsumption demodulation
- For full equational reasoning, function congruence has to be added

Conclusion & Future work

Conclusion

- Lemma generation for induction is done for equational reasoning with 2 special inferences
- Induction rule is modified to check for redundant inductions efficiently

Future work

- The demodulation check can be extended to subsumption demodulation
- For full equational reasoning, function congruence has to be added
- Binary resolution, predicate congruence, etc. could be incorporated for non-equational problems

Conclusion & Future work

Conclusion

- Lemma generation for induction is done for equational reasoning with 2 special inferences
- Induction rule is modified to check for redundant inductions efficiently

Future work

- The demodulation check can be extended to subsumption demodulation
- For full equational reasoning, function congruence has to be added
- Binary resolution, predicate congruence, etc. could be incorporated for non-equational problems
- Heuristics are needed to reduce the search space

Thank you!