Multi-Completion with Termination Tools

Haruhiko Sato   Sarah Winkler   Masahito Kurihara   Aart Middeldorp

Institute of Computer Science
University of Innsbruck

August 13, 2008

Outline

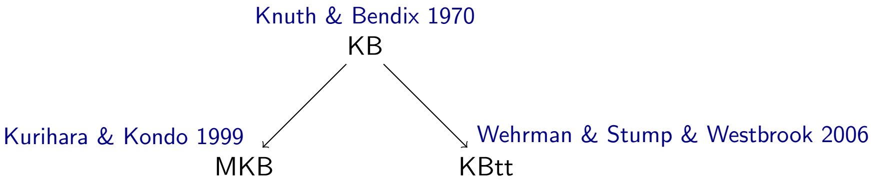
- Completion Inference Systems

Knuth & Bendix 1970

KB

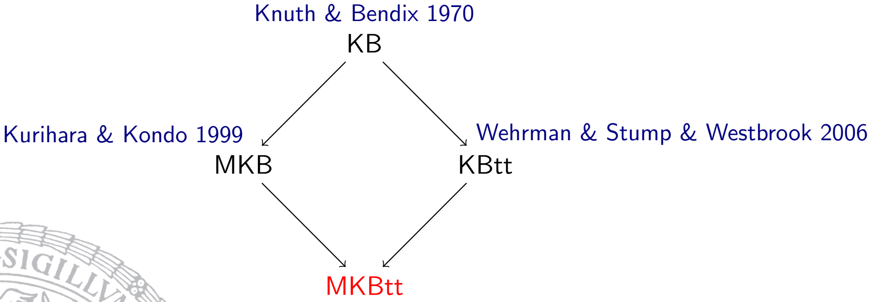
Outline

- Completion Inference Systems



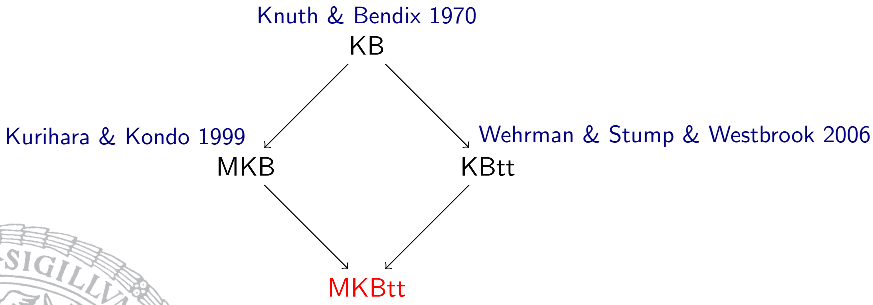
Outline

- Completion Inference Systems



Outline

- Completion Inference Systems



- Implementation
- Experiments

Outline

- Completion Inference Systems
- Implementation
- Experimental Results

KB: Standard Completion

$$\underset{\text{reduction ordering}}{\succ} \;+\; \underset{\text{equations}}{\mathcal{E}} \;\longrightarrow_{KB}\; \underset{\text{rewrite system}}{\mathcal{R}}$$

$\mathcal{R}$ is confluent, terminating, reduced and $\approx_{\mathcal{E}} \; = \; \leftrightarrow^*_{\mathcal{R}}$

KB: Standard Completion

$$\underset{\text{reduction ordering}}{\succ} \quad + \quad \underset{\text{equations}}{\mathcal{E}} \quad \longrightarrow_{KB} \quad \underset{\text{rewrite system}}{\mathcal{R}}$$

$\mathcal{R}$ is confluent, terminating, reduced and $\approx_{\mathcal{E}} \; = \; \leftrightarrow^*_{\mathcal{R}}$

Definition (Orient in KB)

orient $\dfrac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{s \to t\}}$ if $s \succ t$

KB: Standard Completion

$\succ$		$\mathcal{E}$		$\mathcal{R}$
reduction ordering	$+$	equations	$\longrightarrow_{KB}$	rewrite system

$\mathcal{R}$ is confluent, terminating, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (Orient in KB)

orient $\dfrac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{s \to t\}}$ if $s \succ t$

$\succ$ is crucial for success

MKB: Multicompletion

$\mathcal{O} = \{\succ_1, \ldots, \succ_n\}$	$+$	$\mathcal{E}$	$\longrightarrow_{MKB}$	$\mathcal{R}, \succ_i$
reduction orderings		equations		rewrite system

$\mathcal{R}$ is confluent, terminating with $\succ_i$, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

MKB: Multicompletion

$\mathcal{O} = \{\succ_1, \ldots, \succ_n\}$		$\mathcal{E}$		$\mathcal{R}, \succ_i$
reduction orderings	$+$	equations	$\longrightarrow_{MKB}$	rewrite system

$\mathcal{R}$ is confluent, terminating with $\succ_i$, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (MKB node)

node is tuple $\langle s : t, R_0, R_1, E\rangle$ such that

- data $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- labels R_0, R_1, E are disjoint subsets of $\{\succ_1, \ldots, \succ_n\}$

MKB: Multicompletion

$$\underbrace{\mathcal{O} = \{\succ_1, \ldots, \succ_n\}}_{\text{reduction orderings}} + \underbrace{\mathcal{E}}_{\text{equations}} \longrightarrow_{MKB} \underbrace{\mathcal{R}, \succ_i}_{\text{rewrite system}}$$

$\mathcal{R}$ is confluent, terminating with $\succ_i$, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (MKB node)

node is tuple $\langle s : t, R_0, R_1, E \rangle$ such that

- data $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- labels R_0, R_1, E are disjoint subsets of $\{\succ_1, \ldots, \succ_n\}$

Intuition

Simulate parallel completion sequences using $\succ_i$ in state $\mathcal{E}_i, \mathcal{R}_i$:

$$\langle s : t, R_0, R_1, E \rangle$$

- $\succ_i \in R_0$: $s \to t \in \mathcal{R}_i$ (→ R_0)
- $\succ_i \in R_1$: $t \to s \in \mathcal{R}_i$ (→ R_1)
- $\succ_i \in E$: $s \approx t \in \mathcal{E}_i$ (→ E)

MKB: Multicompletion

$$\underbrace{\mathcal{O} = \{\succ_1, \ldots, \succ_n\}}_{\text{reduction orderings}} + \underbrace{\mathcal{E}}_{\text{equations}} \longrightarrow_{MKB} \underbrace{\mathcal{R}, \succ_i}_{\text{rewrite system}}$$

$\mathcal{R}$ is confluent, terminating with $\succ_i$, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (MKB node)

node is tuple $\langle s : t, R_0, R_1, E \rangle$ such that

- data $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- labels R_0, R_1, E are disjoint subsets of $\{\succ_1, \ldots, \succ_n\}$

Intuition

Simulate parallel completion sequences using $\succ_i$ in state $\mathcal{E}_i, \mathcal{R}_i$:

$$\langle s : t, R_0, R_1, E \rangle$$

$\succ_i \in R_0$: $s \to t \in \mathcal{R}_i$ (↗ R_0)

$\succ_i \in R_1$: $t \to s \in \mathcal{R}_i$ (↑ R_1)

$\succ_i \in E$: $s \approx t \in \mathcal{E}_i$ (↖ E)

MKB: Multicompletion

$$\underset{\text{reduction orderings}}{\mathcal{O} = \{\succ_1, \ldots, \succ_n\}} + \underset{\text{equations}}{\mathcal{E}} \longrightarrow_{MKB} \underset{\text{rewrite system}}{\mathcal{R}, \succ_i}$$

$\mathcal{R}$ is confluent, terminating with $\succ_i$, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (MKB node)

node is tuple $\langle s : t, R_0, R_1, E\rangle$ such that

- data $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- labels R_0, R_1, E are disjoint subsets of $\{\succ_1, \ldots, \succ_n\}$

Intuition

Simulate parallel completion sequences using $\succ_i$ in state $\mathcal{E}_i, \mathcal{R}_i$:

$$\langle s : t, R_0, R_1, E\rangle$$

$\succ_i \in R_0$: $s \to t \in \mathcal{R}_i$ (→ R_0)

$\succ_i \in R_1$: $t \to s \in \mathcal{R}_i$ (→ R_1)

$\succ_i \in E$: $s \approx t \in \mathcal{E}_i$ (→ E)

MKB: Multicompletion

$$\underset{\text{reduction orderings}}{\mathcal{O} = \{\succ_1, \ldots, \succ_n\}} \; + \; \underset{\text{equations}}{\mathcal{E}} \; \longrightarrow_{MKB} \; \underset{\text{rewrite system}}{\mathcal{R}, \succ_i}$$

$\mathcal{R}$ is confluent, terminating with $\succ_i$, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (MKB node)

node is tuple $\langle s : t, R_0, R_1, E \rangle$ such that

- data $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- labels R_0, R_1, E are disjoint subsets of $\{\succ_1, \ldots, \succ_n\}$

Definition (Orient in MKB)

$$\text{orient} \quad \frac{\mathcal{N} \uplus \{\langle s : t, R_0, R_1, E \rangle\}}{\mathcal{N} \cup \langle s : t, R_0 \cup O, R_1, E \setminus O \rangle} \qquad \text{if } s \succ_i t \text{ for all } \succ_i \in O \text{ and } O \neq \varnothing$$

MKB: Multicompletion

$$\underbrace{\mathcal{O} = \{\succ_1, \dots, \succ_n\}}_{\text{reduction orderings}} + \underbrace{\mathcal{E}}_{\text{equations}} \longrightarrow_{MKB} \underbrace{\mathcal{R}, \succ_i}_{\text{rewrite system}}$$

$\mathcal{R}$ is confluent, terminating with $\succ_i$, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (MKB node)

node is tuple $\langle s : t, R_0, R_1, E \rangle$ such that

- data $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- labels R_0, R_1, E are disjoint subsets of $\{\succ_1, \dots, \succ_n\}$

Definition (Orient in MKB)

orient $\dfrac{\mathcal{N} \uplus \{\langle s : t, R_0, R_1, E \rangle\}}{\mathcal{N} \cup \langle s : t, R_0 \cup O, R_1, E \setminus O \rangle}$ if $s \succ_i t$ for all $\succ_i \in O$ and $O \neq \varnothing$

- inference system: share steps among parallel sequences

KBtt: Using Termination Tools

$\mathcal{E}$ (equations) $\longrightarrow_{KBtt}$ $\mathcal{R}$ (rewrite system)

$\mathcal{R}$ is confluent, terminating, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

KBtt: Using Termination Tools

$$\underset{\text{equations}}{\mathcal{E}} \quad \longrightarrow_{KBtt} \quad \underset{\text{rewrite system}}{\mathcal{R}}$$

$\mathcal{R}$ is confluent, terminating, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (Orient in KB)

$$\text{orient} \quad \frac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{s \to t\}} \quad \text{if } s \succ t$$

KBtt: Using Termination Tools

$$\underset{\text{equations}}{\mathcal{E}} \quad \longrightarrow_{KBtt} \quad \underset{\text{rewrite system}}{\mathcal{R}}$$

$\mathcal{R}$ is confluent, terminating, reduced and $\approx_{\mathcal{E}} = \leftrightarrow^*_{\mathcal{R}}$

Definition (Orient in KBtt)

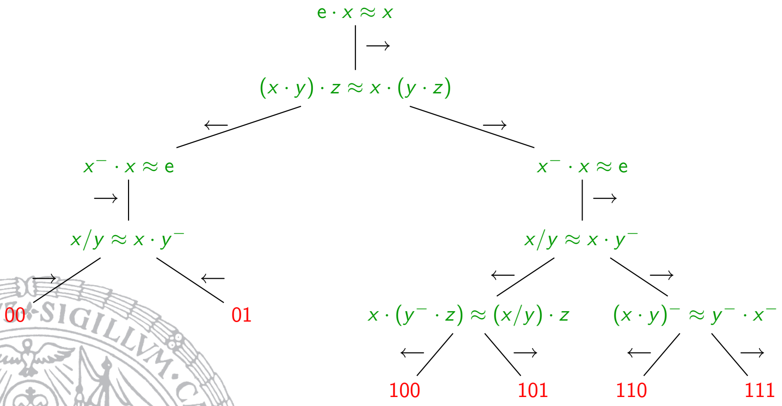
Wehrman/Stump/Westbrook '06

$$\text{orient} \quad \frac{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E}, \mathcal{R} \cup \{s \to t\}, \mathcal{C} \cup \{s \to t\}} \quad \text{if } \mathcal{C} \cup \{s \to t\} \text{ terminates}$$

- implemented in Slothrop

MKBtt: Example

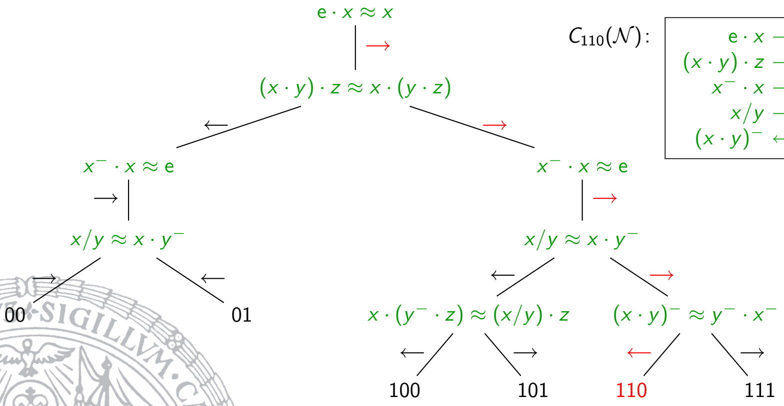
$e \cdot x \approx x \quad (x \cdot y) \cdot z \approx x \cdot (y \cdot z) \quad x^- \cdot x \approx e \quad x/y \approx x \cdot y^- \quad \cdots$



- branches correspond to processes

MKBtt: Example

$\mathsf{e} \cdot x \approx x \quad (x \cdot y) \cdot z \approx x \cdot (y \cdot z) \quad x^{-} \cdot x \approx \mathsf{e} \quad x / y \approx x \cdot y^{-} \quad \cdots$



$C_{110}(\mathcal{N})$:

$$\begin{aligned} \mathsf{e} \cdot x &\to x \\ (x \cdot y) \cdot z &\to x \cdot (y \cdot z) \\ x^{-} \cdot x &\to \mathsf{e} \\ x / y &\to x \cdot y^{-} \\ (x \cdot y)^{-} &\leftarrow y^{-} \cdot x^{-} \end{aligned}$$

- branches correspond to processes
- orientations on branch p determine rules in $C_p(\mathcal{N})$

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- R_0, R_1, E, C_0, C_1 are sets of processes

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- R_0, R_1, E, C_0, C_1 are sets of processes
- $(R_0 \cup C_0) \cap (R_1 \cup C_1) = \varnothing$, $E \cap (R_0 \cup C_0 \cup R_1 \cup C_1) = \varnothing$

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- R_0, R_1, E, C_0, C_1 are sets of processes
- $(R_0 \cup C_0) \cap (R_1 \cup C_1) = \varnothing$, $E \cap (R_0 \cup C_0 \cup R_1 \cup C_1) = \varnothing$

Definition

orient
$$\frac{\mathcal{N} \cup \{\langle s : t, R_0, R_1, E, C_0, C_1 \rangle\}}{split_{E_{rl} \cap E_{lr}}(\mathcal{N}) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl} \rangle\}}$$

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- R_0, R_1, E, C_0, C_1 are sets of processes
- $(R_0 \cup C_0) \cap (R_1 \cup C_1) = \varnothing$, $E \cap (R_0 \cup C_0 \cup R_1 \cup C_1) = \varnothing$

Definition

orient
$$\frac{\mathcal{N} \cup \{\langle s : t, R_0, R_1, E, C_0, C_1 \rangle\}}{\mathit{split}_{E_{rl} \cap E_{lr}}(\mathcal{N}) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl} \rangle\}}$$

- where $E_{lr}, E_{rl} \subseteq E$ such that $E_{lr} \cup E_{rl} \neq \varnothing$,

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- $R_0,\ R_1,\ E,\ C_0,\ C_1$ are sets of processes
- $(R_0 \cup C_0) \cap (R_1 \cup C_1) = \varnothing,\ E \cap (R_0 \cup C_0 \cup R_1 \cup C_1) = \varnothing$

Definition

orient
$$\frac{\mathcal{N} \cup \{\langle s : t, R_0, R_1, E, C_0, C_1 \rangle\}}{split_{E_{rl} \cap E_{lr}}(\mathcal{N}) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl} \rangle\}}$$

- where $E_{lr}, E_{rl} \subseteq E$ such that $E_{lr} \cup E_{rl} \neq \varnothing$,
- $C[\mathcal{N}, p] \cup \{s \to t\}$ terminates for all $p \in E_{lr}$ and
  $C[\mathcal{N}, p] \cup \{t \to s\}$ terminates for all $p \in E_{rl}$,

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- R_0, R_1, E, C_0, C_1 are sets of processes
- $(R_0 \cup C_0) \cap (R_1 \cup C_1) = \varnothing$, $E \cap (R_0 \cup C_0 \cup R_1 \cup C_1) = \varnothing$

Definition

orient
$$\frac{\mathcal{N} \cup \{\langle s : t, R_0, R_1, E, C_0, C_1 \rangle\}}{\mathit{split}_{E_{rl} \cap E_{lr}}(\mathcal{N}) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl} \rangle\}}$$

- where $E_{lr}, E_{rl} \subseteq E$ such that $E_{lr} \cup E_{rl} \neq \varnothing$,
- $\mathrm{C}[\mathcal{N}, p] \cup \{s \to t\}$ terminates for all $p \in E_{lr}$ and $\mathrm{C}[\mathcal{N}, p] \cup \{t \to s\}$ terminates for all $p \in E_{rl}$,
- $E' = E \setminus (E_{lr} \cup E_{rl})$,

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- R_0, R_1, E, C_0, C_1 are sets of processes
- $(R_0 \cup C_0) \cap (R_1 \cup C_1) = \varnothing$, $E \cap (R_0 \cup C_0 \cup R_1 \cup C_1) = \varnothing$

Definition

orient
$$\frac{\mathcal{N} \cup \{\langle s : t, R_0, R_1, E, C_0, C_1 \rangle\}}{\mathit{split}_{E_{rl} \cap E_{lr}}(\mathcal{N}) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl} \rangle\}}$$

- where $E_{lr}, E_{rl} \subseteq E$ such that $E_{lr} \cup E_{rl} \neq \varnothing$,
- $\mathrm{C}[\mathcal{N}, p] \cup \{s \to t\}$ terminates for all $p \in E_{lr}$ and
  $\mathrm{C}[\mathcal{N}, p] \cup \{t \to s\}$ terminates for all $p \in E_{rl}$,
- $E' = E \setminus (E_{lr} \cup E_{rl})$,
- $R_{lr} = (E_{lr} \setminus E_{rl}) \cup \{p0 \mid p \in (E_{rl} \cap E_{lr})\}$ and
  $R_{rl} = (E_{rl} \setminus E_{lr}) \cup \{p1 \mid p \in (E_{rl} \cap E_{lr})\}$,

split processes

Definition (MKBtt node)

node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that

- $R_0,\ R_1,\ E,\ C_0,\ C_1$ are sets of processes
- $(R_0 \cup C_0) \cap (R_1 \cup C_1) = \varnothing,\ E \cap (R_0 \cup C_0 \cup R_1 \cup C_1) = \varnothing$

Definition

orient
$$\frac{\mathcal{N} \cup \{\langle s : t, R_0, R_1, E, C_0, C_1 \rangle\}}{\mathit{split}_{E_{rl} \cap E_{lr}}(\mathcal{N}) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl} \rangle\}}$$

- where $E_{lr}, E_{rl} \subseteq E$ such that $E_{lr} \cup E_{rl} \neq \varnothing$,
- $\mathrm{C}[\mathcal{N}, p] \cup \{s \to t\}$ terminates for all $p \in E_{lr}$ and $\mathrm{C}[\mathcal{N}, p] \cup \{t \to s\}$ terminates for all $p \in E_{rl}$,
- $E' = E \setminus (E_{lr} \cup E_{rl})$,
- $R_{lr} = (E_{lr} \setminus E_{rl}) \cup \{p0 \mid p \in (E_{rl} \cap E_{lr})\}$ and $R_{rl} = (E_{rl} \setminus E_{lr}) \cup \{p1 \mid p \in (E_{rl} \cap E_{lr})\}$,
- and $\mathit{split}_P(\mathcal{N})$ replaces every $p \in P$ by $p0,\ p1$.

Outline

- Completion Inference Systems
- Implementation
- Experimental Results

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathsf{T_TT_2}$ internally

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathrm{T_TT_2}$ internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathsf{T_{T}T_{2}}$ internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

global time limit (in seconds)

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses TTT2 internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

time limit (in seconds) for each call to termination prover

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathsf{T_TT_2}$ internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

termination prover

- mkbTT is written in OCaml (≈ 5000 lines of code)
- interfaces arbitrary termination prover, or uses T_TT_2 internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

input system (in TPDB format)

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathsf{T_{T}T_{2}}$ internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

output statistics

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathsf{T_TT_2}$ internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

output completed system

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathsf{T_TT_2}$ internally
- simple command-line interface

```
mkbtt -t 3600 -T 1     -st -tp ttt2fast cge2.trs
```

```
SUCCESS

186.70 (total time)

STATISTICS

number of inference steps: 77

total time:    186.70
orient:        169.70
rewrite:        14.16
deduce:          2.06
termination:   162.88

external termination prover: ttt2fast
calls to termination prover: 1072 (yes: 933, timeouts: 0)
time limit per call:         1.0
```

- mkbTT is written in OCaml ($\approx$ 5000 lines of code)
- interfaces arbitrary termination prover, or uses $\mathsf{T_TT_2}$ internally
- simple command-line interface

```
mkbtt -t 3600 -T 1 -ct -st -tp ttt2fast cge2.trs
```

- http://cl-informatik.uibk.ac.at/mkbtt/

Outline

- Completion Inference Systems
- Implementation
- Experimental Results

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds

TRS	Slothrop		mkbTT		
	(1)	(2)	(1)	(2)	(3)
SK90_3.01	800.37	326	85.30	51	29

(1) total time in seconds

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds

TRS	Slothrop		mkbTT		
	(1)	(2)	(1)	(2)	(3)
SK90_3.01	800.37	326	85.30	51	29

(2) number of calls to termination prover

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds

TRS	Slothrop (1)	Slothrop (2)	mkbTT (1)	mkbTT (2)	mkbTT (3)
SK90_3.01	800.37	326	85.30	51	29

(3) number of inference steps

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds

TRS	Slothrop (1)	Slothrop (2)	mkbTT (1)	mkbTT (2)	mkbTT (3)
SK90_3.01	800.37	326	85.30	51	29
SK90_3.04	∞		∞		
SK90_3.05	∞		913.77	225	94
SK90_3.19	84.25	43	112.65	65	11
SK90_3.27	∞		∞		
SL-cge2	∞		2793.21	1220	77
SL-cge3	∞		∞		
SL-endo	246.56	101	218.96	135	45
SL-ep	∞		∞		

∞ timeout

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds

TRS	Slothrop (1)	Slothrop (2)	mkbTT (1)	mkbTT (2)	mkbTT (3)
SK90_3.01	800.37	326	85.30	51	29
SK90_3.04	∞		∞		
SK90_3.05	∞		913.77	225	94
SK90_3.19	84.25	43	112.65	65	11
SK90_3.27	∞		∞		
SL-cge2	∞		2793.21	1220	77
SL-cge3	∞		∞		
SL-endo	246.56	101	218.96	135	45
SL-ep	∞		∞		
⋮	⋮			⋮	
27 systems	∅ 2061	#12	∅ 1489	#18	

∅ average execution time

- hardware: Intel® Pentium^{TM} M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds

TRS	Slothrop (1)	Slothrop (2)	mkbTT (1)	mkbTT (2)	mkbTT (3)
SK90_3.01	800.37	326	85.30	51	29
SK90_3.04	∞		∞		
SK90_3.05	∞		913.77	225	94
SK90_3.19	84.25	43	112.65	65	11
SK90_3.27	∞		∞		
SL-cge2	∞		2793.21	1220	77
SL-cge3	∞		∞		
SL-endo	246.56	101	218.96	135	45
SL-ep	∞		∞		
⋮	⋮			⋮	
27 systems	∅ 2061	#12	∅ 1489	#18	

number of successes

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds, $\mathsf{T_TT_2}$ 1 second

TRS	AProVE Slothrop (1)	(2)	AProVE mkbTT (1)	(2)	(3)	$\mathsf{T_TT_2}$ Slothrop (1)	(2)	$\mathsf{T_TT_2}$ mkbTT (1)	(2)	(3)
SK90_3.01	800.37	326	85.30	51	29	71.52	304	4.45	51	29
SK90_3.04	∞		∞							
SK90_3.05	∞		913.77	225	94					
SK90_3.19	84.25	43	112.65	65	11					
SK90_3.27	∞		∞							
SL-cge2	∞		2793.21	1220	77					
SL-cge3	∞		∞							
SL-endo	246.56	101	218.96	135	45					
SL-ep	∞		∞							
⋮	⋮			⋮						
27 systems	∅ 2061	#12	∅ 1489	#18						

- hardware: Intel® Pentium™ M processor, 2 GHz, 1 GB
- global time limit: 1 hour
- termination time limit: AProVE 5 seconds, $\mathsf{T_TT_2}$ 1 second

	AProVE					$\mathsf{T_TT_2}$				
	Slothrop		mkbTT			Slothrop		mkbTT		
TRS	(1)	(2)	(1)	(2)	(3)	(1)	(2)	(1)	(2)	(3)
SK90_3.01	800.37	326	85.30	51	29	71.52	304	4.45	51	29
SK90_3.04	∞		∞			∞		508.24	658	140
SK90_3.05	∞		913.77	225	94	78.48	258	46.45	220	92
SK90_3.19	84.25	43	112.65	65	11	3.39	43	5.01	65	11
SK90_3.27	∞		∞			73.61	70	118.56	143	37
SL-cge2	∞		2793.21	1220	77	665.29	1384	246.64	1072	77
SL-cge3	∞		∞			∞		∞		
SL-endo	246.56	101	218.96	135	45	12.64	105	7.87	135	45
SL-ep	∞		∞			54.47	266	230.06	1101	26
⋮	⋮			⋮		⋮			⋮	
27 systems	∅ 2061	#12	∅ 1489	#18		∅ 2334	#20	∅ 473	#24	

- mkbTT with $\mathsf{T_{T}T_{2}}$micro manages to complete CGE$_3$

$$
\begin{array}{ccc}
\mathsf{e} \cdot x \approx x & \mathsf{f}(x \cdot y) \approx \mathsf{f}(x) \cdot \mathsf{f}(y) & \mathsf{f}(x) \cdot \mathsf{g}(y) \approx \mathsf{g}(y) \cdot \mathsf{f}(x) \\
x^{-} \cdot x \approx \mathsf{e} & \mathsf{g}(x \cdot y) \approx \mathsf{g}(x) \cdot \mathsf{g}(y) & \mathsf{f}(x) \cdot \mathsf{h}(y) \approx \mathsf{h}(y) \cdot \mathsf{f}(x) \\
(x \cdot y) \cdot z \approx x \cdot (y \cdot z) & \mathsf{h}(x \cdot y) \approx \mathsf{h}(x) \cdot \mathsf{h}(y) & \mathsf{g}(x) \cdot \mathsf{h}(y) \approx \mathsf{h}(y) \cdot \mathsf{g}(x)
\end{array}
$$

into a 28 rule convergent TRS in 6136 seconds

- mkbTT with $\mathsf{T_TT_2}$micro manages to complete CGE_3

$$\begin{aligned} \mathsf{e} \cdot x &\approx x & \mathsf{f}(x \cdot y) &\approx \mathsf{f}(x) \cdot \mathsf{f}(y) & \mathsf{f}(x) \cdot \mathsf{g}(y) &\approx \mathsf{g}(y) \cdot \mathsf{f}(x) \\ x^- \cdot x &\approx \mathsf{e} & \mathsf{g}(x \cdot y) &\approx \mathsf{g}(x) \cdot \mathsf{g}(y) & \mathsf{f}(x) \cdot \mathsf{h}(y) &\approx \mathsf{h}(y) \cdot \mathsf{f}(x) \\ (x \cdot y) \cdot z &\approx x \cdot (y \cdot z) & \mathsf{h}(x \cdot y) &\approx \mathsf{h}(x) \cdot \mathsf{h}(y) & \mathsf{g}(x) \cdot \mathsf{h}(y) &\approx \mathsf{h}(y) \cdot \mathsf{g}(x) \end{aligned}$$

into a 28 rule convergent TRS in 6136 seconds

- Waldmeister is able to produce a 70 rule ground convergent TRS using ordered completion (Thomas Hillenbrand – RTA 2008 invited talk)

- mkbTT with $\mathsf{T_TT_2}$micro manages to complete CGE_3

$$
\begin{array}{rccc}
e \cdot x \approx x & f(x \cdot y) \approx f(x) \cdot f(y) & f(x) \cdot g(y) \approx g(y) \cdot f(x) \\
x^{-} \cdot x \approx e & g(x \cdot y) \approx g(x) \cdot g(y) & f(x) \cdot h(y) \approx h(y) \cdot f(x) \\
(x \cdot y) \cdot z \approx x \cdot (y \cdot z) & h(x \cdot y) \approx h(x) \cdot h(y) & g(x) \cdot h(y) \approx h(y) \cdot g(x)
\end{array}
$$

  into a 28 rule convergent TRS in 6136 seconds

- Waldmeister is able to produce a 70 rule ground convergent TRS using ordered completion (Thomas Hillenbrand – RTA 2008 invited talk)

- breaking news: new version of mkbTT needs only 3277 seconds

Current Work

- indexing techniques
- critical pair criteria
- tighter coupling with $\mathsf{T_TT_2}$

Current Work

- indexing techniques
- critical pair criteria
- tighter coupling with $\mathsf{T_TT_2}$

Future Work

- ordered multi-completion with termination tools
- multi-completion modulo theories (AC)
- multi-completion for theorem proving (CASC – UEQ)