Optimizing mkb_{TT}
System Description

Sarah Winkler Haruhiko Sato Aart Middeldorp Masahito Kurihara

Institute of Computer Science
University of Innsbruck

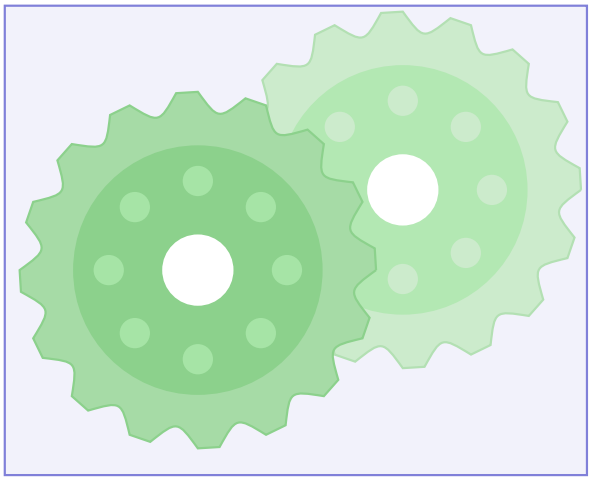
Graduate School of Information Science and Technology
Hokkaido University

21st International Conference on
Rewriting Techniques and Applications

July 13, 2010

Content

mkb$_{TT}$

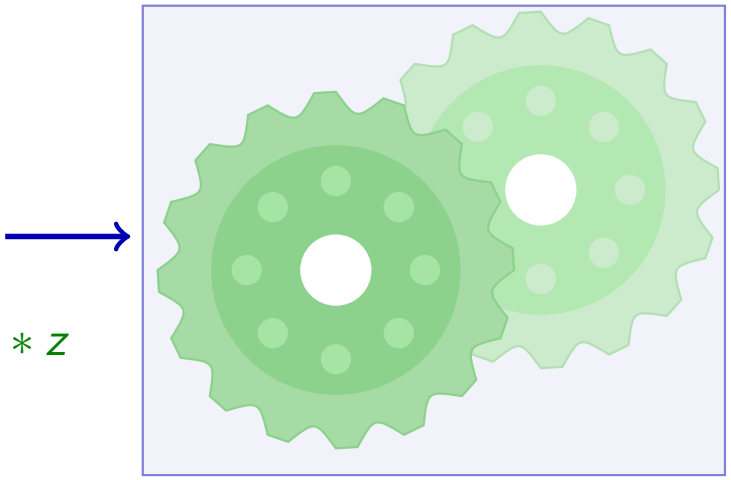


mkb$_{TT}$

mkb$_{TT}$ is Knuth-Bendix completion tool

mkb$_{TT}$

$$x * 1 \approx x$$
$$x * \mathrm{i}(x) \approx 1$$
$$x * (y * z) \approx (x * y) * z$$

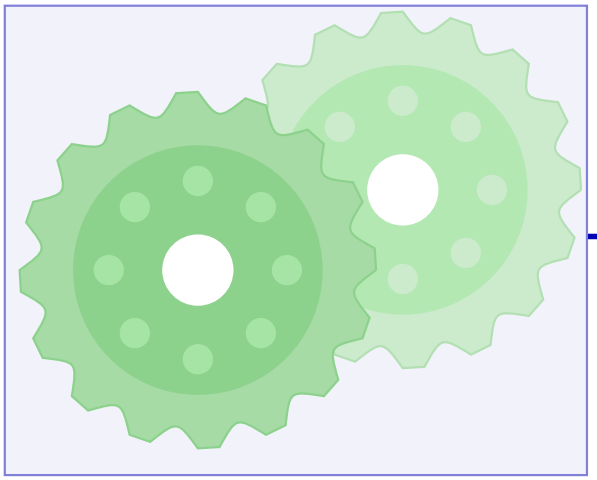


mkb$_{TT}$

mkb$_{TT}$ is Knuth-Bendix completion tool

mkb$_{TT}$

$$x * 1 \approx x$$
$$x * \mathrm{i}(x) \approx 1$$
$$x * (y * z) \approx (x * y) * z$$



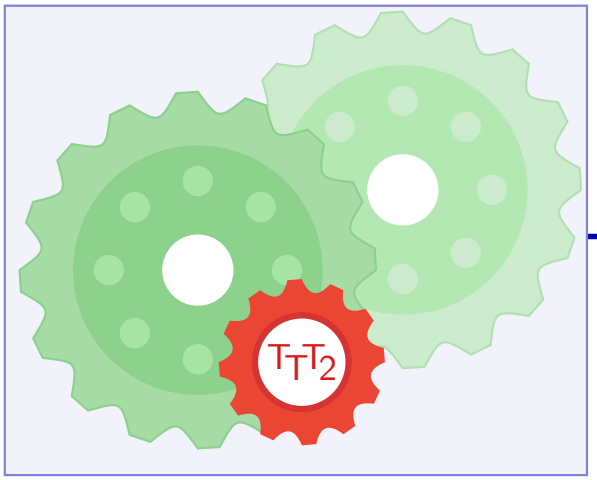
mkb$_{TT}$

$$\mathrm{i}(1) \to 1$$
$$\mathrm{i}(\mathrm{i}(x)) \to x$$
$$x * 1 \to x$$
$$1 * x \to x$$
$$x * \mathrm{i}(x) \to 1$$
$$\mathrm{i}(x) * x \to 1$$
$$\mathrm{i}(x * y) \to \mathrm{i}(y) * \mathrm{i}(x)$$
$$(x * y) * z \to x * (y * z)$$
$$x * (\mathrm{i}(x) * y) \to y$$
$$\mathrm{i}(x) * (x * y) \to y$$

mkb$_{TT}$ is Knuth-Bendix completion tool

mkb$_{TT}$

$$x * 1 \approx x$$
$$x * \mathrm{i}(x) \approx 1$$
$$x * (y * z) \approx (x * y) * z$$



mkb$_{TT}$

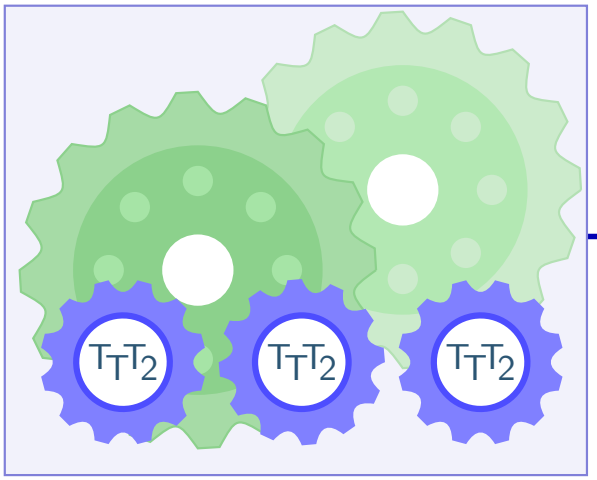
$$\mathrm{i}(1) \to 1$$
$$\mathrm{i}(\mathrm{i}(x)) \to x$$
$$x * 1 \to x$$
$$1 * x \to x$$
$$x * \mathrm{i}(x) \to 1$$
$$\mathrm{i}(x) * x \to 1$$
$$\mathrm{i}(x * y) \to \mathrm{i}(y) * \mathrm{i}(x)$$
$$(x * y) * z \to x * (y * z)$$
$$x * (\mathrm{i}(x) * y) \to y$$
$$\mathrm{i}(x) * (x * y) \to y$$

mkb$_{TT}$ is Knuth-Bendix completion tool combining

- termination tools (*Wehrman, Stump, Westbrook '06*)
  instead of using reduction order

mkb$_{TT}$

$$x * 1 \approx x$$
$$x * \mathrm{i}(x) \approx 1$$
$$x * (y * z) \approx (x * y) * z$$



mkb$_{TT}$

$$\mathrm{i}(1) \to 1$$
$$\mathrm{i}(\mathrm{i}(x)) \to x$$
$$x * 1 \to x$$
$$1 * x \to x$$
$$x * \mathrm{i}(x) \to 1$$
$$\mathrm{i}(x) * x \to 1$$
$$\mathrm{i}(x * y) \to \mathrm{i}(y) * \mathrm{i}(x)$$
$$(x * y) * z \to x * (y * z)$$
$$x * (\mathrm{i}(x) * y) \to y$$
$$\mathrm{i}(x) * (x * y) \to y$$

mkb$_{TT}$ is Knuth-Bendix completion tool combining

- termination tools (*Wehrman, Stump, Westbrook '06*)
  instead of using reduction order
- multi-completion (*Kondo, Kurihara '99*)
  simulating multiple parallel processes

mkb_{TT} in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$

mkb_{TT} in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ

mkb$_{TT}$ in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data)

mkb$_{TT}$ in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data) and process sets $R_0, \ldots, C_1$ (labels)

mkb_{TT} in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data) and process sets $R_0, \ldots, C_1$ (labels)
- mkb_{TT} is described by inference system operating on node set

mkb_{TT} in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data) and process sets $R_0, \ldots, C_1$ (labels)
- mkb_{TT} is described by inference system operating on node set
- run is inference sequence

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \ldots$$

mkb$_{TT}$ in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data) and process sets $R_0, \ldots, C_1$ (labels)
- mkb$_{TT}$ is described by inference system operating on node set
- run is inference sequence

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \ldots$$

- initial node set for equations $\mathcal{E}$ is
  $\mathcal{N}_{\mathcal{E}} = \{\langle s : t, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \mid s \approx t \in \mathcal{E}\}$

mkb$_{TT}$ in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data) and process sets $R_0, \ldots, C_1$ (labels)
- mkb$_{TT}$ is described by inference system operating on node set
- run is inference sequence

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \ldots$$

- initial node set for equations $\mathcal{E}$ is $\mathcal{N}_{\mathcal{E}} = \{\langle s : t, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \mid s \approx t \in \mathcal{E}\}$
- persistent nodes $\mathcal{N}_\omega = \bigcup_{i \geqslant 0} \bigcap_{j \geqslant i} \mathcal{N}_j$

mkb$_{TT}$ in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data) and process sets $R_0, \ldots, C_1$ (labels)
- mkb$_{TT}$ is described by inference system operating on node set
- run is inference sequence

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \ldots$$

- initial node set for equations $\mathcal{E}$ is $\mathcal{N}_{\mathcal{E}} = \{\langle s : t, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \mid s \approx t \in \mathcal{E}\}$
- persistent nodes $\mathcal{N}_\omega = \bigcup_{i \geqslant 0} \bigcap_{j \geqslant i} \mathcal{N}_j$
- projection of node set $\mathcal{N}$ to process p yields equations $E_p(\mathcal{N})$, rules $R_p(\mathcal{N})$ and constraints $C_p(\mathcal{N})$

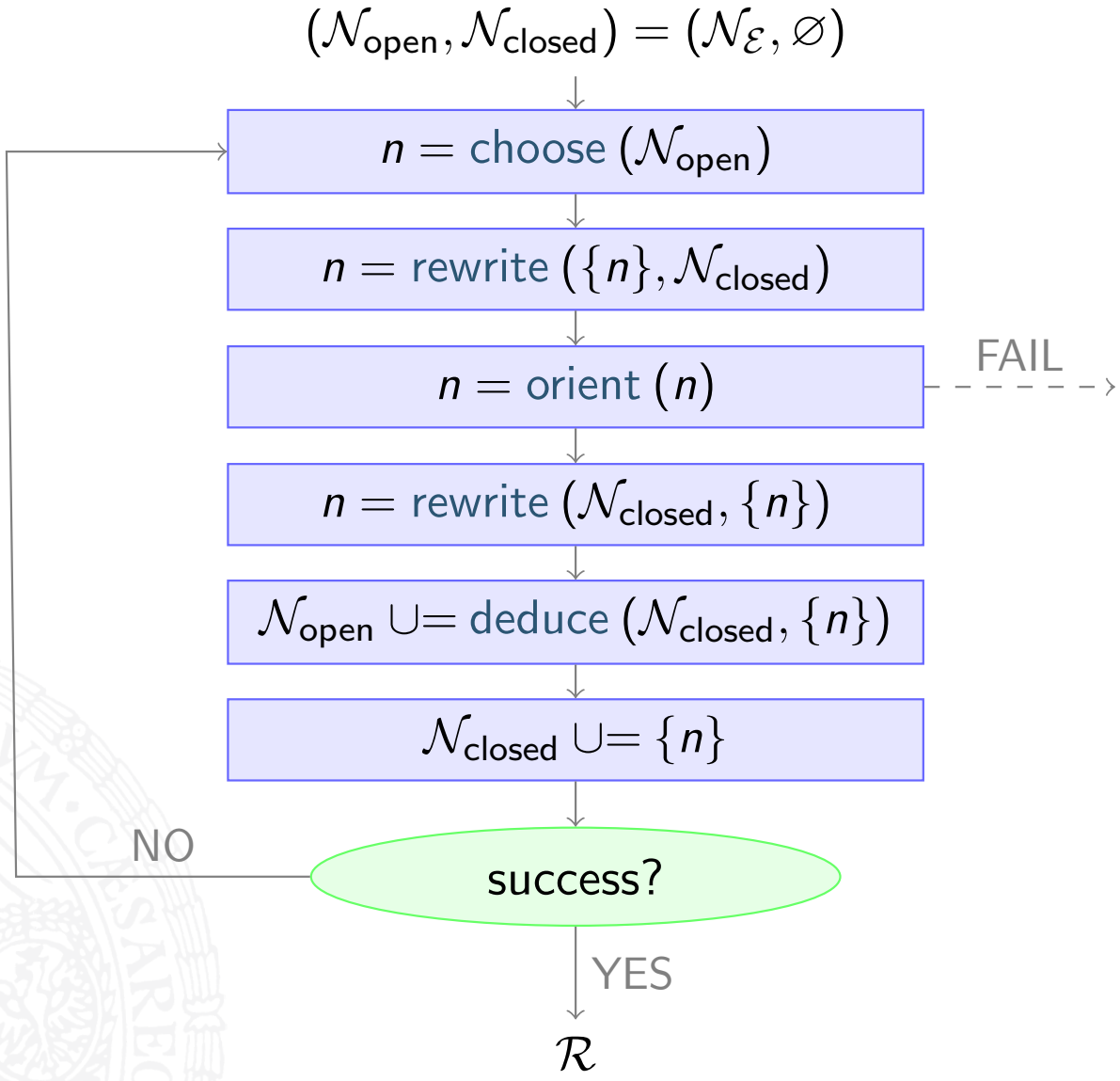
mkb$_{TT}$ in a Nutshell

- processes are strings in $\mathcal{L}((0+1)^*)$, initial process is ϵ
- node is tuple $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ of term pair $s : t$ (data) and process sets $R_0, \ldots, C_1$ (labels)
- mkb$_{TT}$ is described by inference system operating on node set
- run is inference sequence

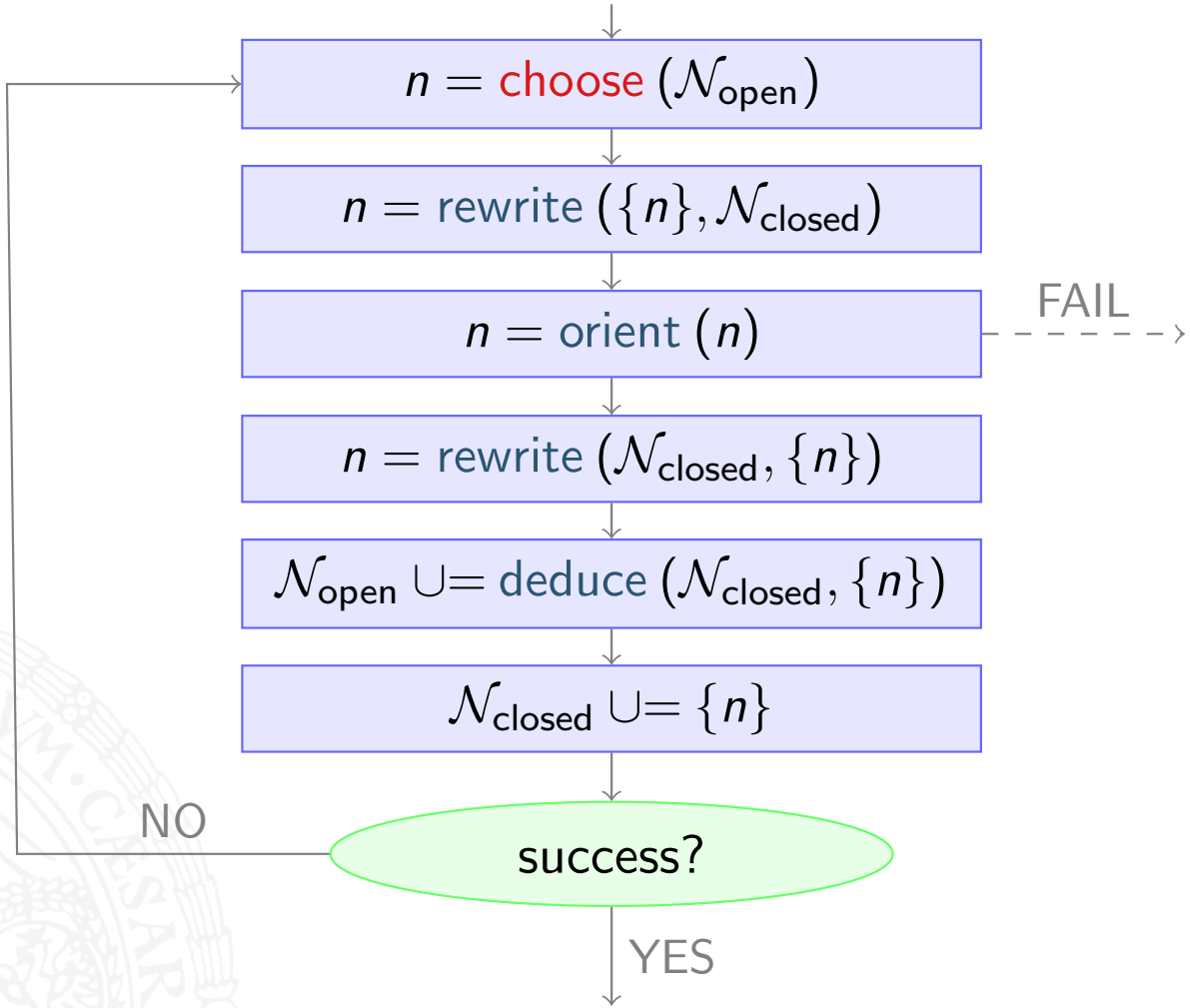
$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \ldots$$

- initial node set for equations $\mathcal{E}$ is $\mathcal{N}_{\mathcal{E}} = \{\langle s : t, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \mid s \approx t \in \mathcal{E}\}$
- persistent nodes $\mathcal{N}_\omega = \bigcup_{i \geqslant 0} \bigcap_{j \geqslant i} \mathcal{N}_j$
- projection of node set $\mathcal{N}$ to process p yields equations $E_p(\mathcal{N})$, rules $R_p(\mathcal{N})$ and constraints $C_p(\mathcal{N})$
- finite run is fair if all critical pairs with respect to $\mathcal{N}_\omega$ are deduced for some process p

The Control Loop



Improvement 1: Selection Strategies



- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$\begin{aligned} \textit{strategy} ::= \quad & \texttt{?} \mid \texttt{(}\textit{node\_property}\texttt{,}\textit{strategy}\texttt{)} \\ & \mid \textit{float}\texttt{(}\textit{strategy}\texttt{:}\textit{strategy}\texttt{)} \end{aligned}$$

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$\begin{aligned} \textit{strategy} ::= \quad & ? \mid (\textit{node\_property}, \textit{strategy}) \\ & \mid \textit{float}(\textit{strategy} : \textit{strategy}) \end{aligned}$$

- random

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process
- mkb_{TT} 2.0: strategy language

$$\begin{aligned} \textit{strategy} ::= \quad & ? \mid (\textit{node\_property}, \textit{strategy}) \\ & \mid \textit{float}(\textit{strategy} : \textit{strategy}) \end{aligned}$$

- tuple of node property and strategy, compared lexicographically

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$\begin{array}{rl} \textit{strategy} ::= & ? \mid (\textit{node\_property}, \textit{strategy}) \\ & \mid \textit{float}(\textit{strategy} : \textit{strategy}) \end{array}$$

- $r(s_1, s_2)$ combines s_1 and s_2 according to ratio r

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$
\begin{aligned}
\mathit{strategy} &::= \ \texttt{?} \mid \texttt{(}\mathit{node\_property}\texttt{,}\mathit{strategy}\texttt{)} \\
&\quad \mid \mathit{float}\texttt{(}\mathit{strategy}\texttt{:}\mathit{strategy}\texttt{)} \\
\mathit{node\_property} &::= \ \texttt{*} \mid \texttt{data(}\mathit{termpair\_property}\texttt{)} \mid \texttt{el(}\mathit{pset\_property}\texttt{)}
\end{aligned}
$$

- age

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$
\begin{aligned}
\textit{strategy} &::= \texttt{?} \mid \texttt{(}\textit{node\_property}\texttt{,}\textit{strategy}\texttt{)} \\
&\quad \mid \textit{float}\texttt{(}\textit{strategy}\texttt{:}\textit{strategy}\texttt{)} \\
\textit{node\_property} &::= \texttt{*} \mid \texttt{data(}\textit{termpair\_property}\texttt{)} \mid \texttt{el(}\textit{pset\_property}\texttt{)}
\end{aligned}
$$

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$
\begin{aligned}
\textit{strategy} &::= \texttt{?} \mid \texttt{(}\textit{node\_property}\texttt{,}\textit{strategy}\texttt{)} \\
&\quad \mid \textit{float}\texttt{(}\textit{strategy}\texttt{:}\textit{strategy}\texttt{)} \\
\textit{node\_property} &::= \texttt{*} \mid \texttt{data(}\textit{termpair\_property}\texttt{)} \mid \texttt{el(}\textit{pset\_property}\texttt{)}
\end{aligned}
$$

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$
\begin{array}{rcl}
\mathit{strategy} & ::= & \texttt{?} \mid \texttt{(}\mathit{node\_property}\texttt{,}\mathit{strategy}\texttt{)} \\
 & & \mid \mathit{float}\texttt{(}\mathit{strategy}\texttt{:}\mathit{strategy}\texttt{)} \\
\mathit{node\_property} & ::= & \texttt{*} \mid \texttt{data(}\mathit{termpair\_property}\texttt{)} \mid \texttt{el(}\mathit{pset\_property}\texttt{)} \\
 & & \mid\texttt{-}\ \mathit{node\_property} \mid \mathit{node\_property}\ \texttt{+}\ \mathit{node\_property} \\
\mathit{pset\_property} & ::= & \texttt{\#} \mid \texttt{sum(}\mathit{process\_property}\texttt{)} \mid \texttt{min(}\mathit{process\_property}\texttt{)} \\
\mathit{process\_property} & ::= & \texttt{e(}\mathit{eqs\_property}\texttt{)} \mid \texttt{r(}\mathit{trs\_property}\texttt{)} \mid \texttt{c(}\mathit{trs\_property}\texttt{)} \\
 & & \mid \mathit{process\_property}\ \texttt{+}\ \mathit{process\_property} \\
\mathit{trs\_property} & ::= & \texttt{sum(}\mathit{termpair\_property}\texttt{)} \mid \texttt{cp(}\mathit{eqs\_property}\texttt{)} \mid \texttt{\#} \\
\mathit{eqs\_property} & ::= & \texttt{sum(}\mathit{termpair\_property}\texttt{)} \mid \texttt{\#} \\
\mathit{termpair\_property} & ::= & \texttt{sizemax} \mid \texttt{sizesum}
\end{array}
$$

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

$$
\begin{array}{rcl}
\mathit{strategy} & ::= & \texttt{?} \mid \texttt{(}\mathit{node\_property}\texttt{,}\mathit{strategy}\texttt{)} \\
 & & \mid \mathit{float}\texttt{(}\mathit{strategy}\texttt{:}\mathit{strategy}\texttt{)} \\
\mathit{node\_property} & ::= & \texttt{*} \mid \texttt{data(}\mathit{termpair\_property}\texttt{)} \mid \texttt{el(}\mathit{pset\_property}\texttt{)} \\
 & & \mid\texttt{-}\ \mathit{node\_property} \mid \mathit{node\_property}\ \texttt{+}\ \mathit{node\_property} \\
\mathit{pset\_property} & ::= & \texttt{\#} \mid \texttt{sum(}\mathit{process\_property}\texttt{)} \mid \texttt{min(}\mathit{process\_property}\texttt{)} \\
\mathit{process\_property} & ::= & \texttt{e(}\mathit{eqs\_property}\texttt{)} \mid \texttt{r(}\mathit{trs\_property}\texttt{)} \mid \texttt{c(}\mathit{trs\_property}\texttt{)} \\
 & & \mid \mathit{process\_property}\ \texttt{+}\ \mathit{process\_property} \\
\mathit{trs\_property} & ::= & \texttt{sum(}\mathit{termpair\_property}\texttt{)} \mid \texttt{cp(}\mathit{eqs\_property}\texttt{)} \mid \texttt{\#} \\
\mathit{eqs\_property} & ::= & \texttt{sum(}\mathit{termpair\_property}\texttt{)} \mid \texttt{\#} \\
\mathit{termpair\_property} & ::= & \texttt{sizemax} \mid \texttt{sizesum}
\end{array}
$$

Example (size-age ratio)

choose mostly small, sometimes old node:

```
0.9((data(sumsize),?):(*,?))
```

- mkb_{TT} 1.0:
  - select process for which $|E_p(\mathcal{N})| + |R_p(\mathcal{N})|$ is minimal
  - select mostly small, sometimes old node for process

- mkb_{TT} 2.0: strategy language

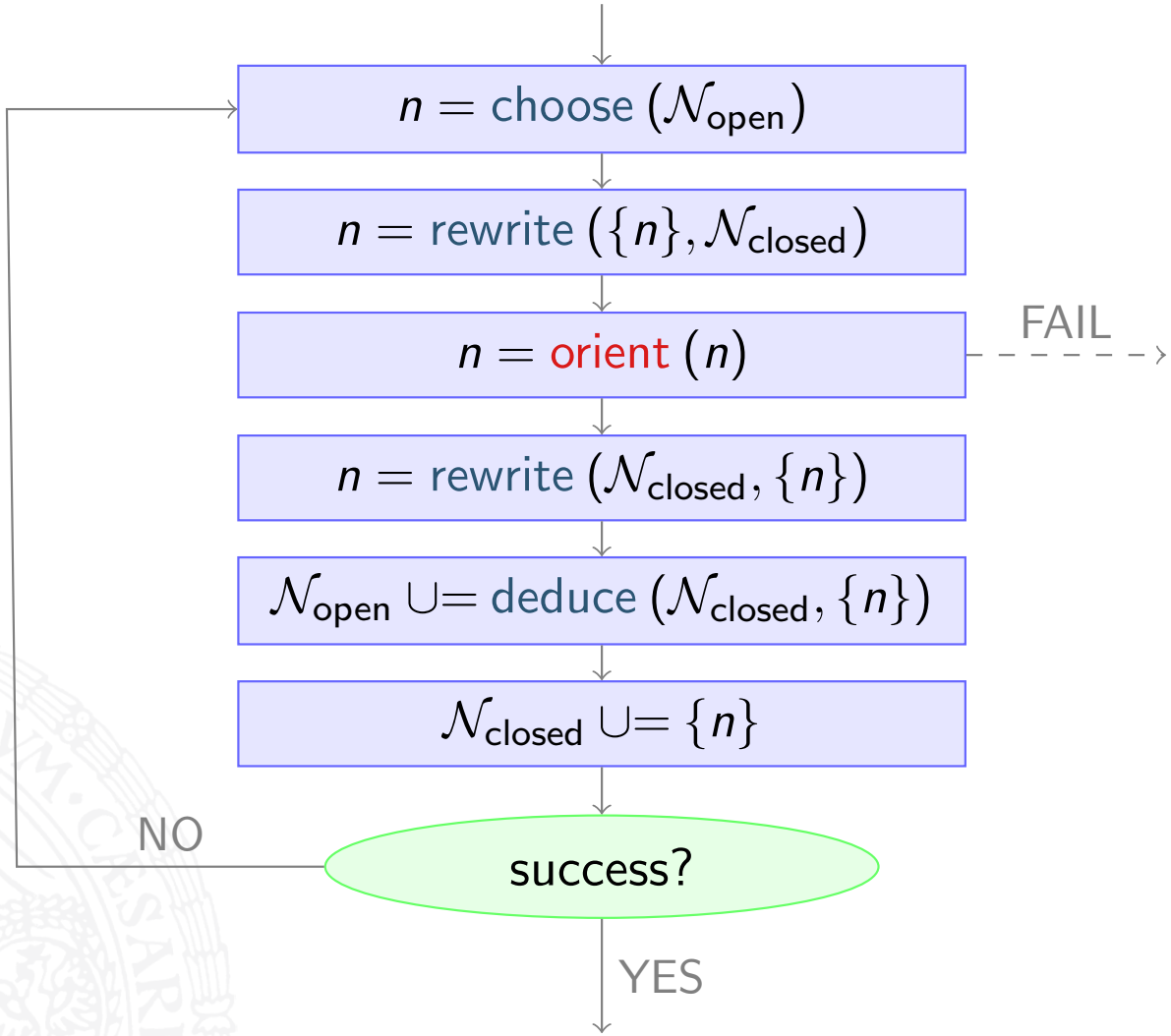
$$
\begin{array}{rcl}
\textit{strategy} & ::= & \texttt{?} \mid \texttt{(}\textit{node\_property}\texttt{,}\textit{strategy}\texttt{)} \\
 & & \mid \textit{float}\texttt{(}\textit{strategy}\texttt{:}\textit{strategy}\texttt{)} \\
\textit{node\_property} & ::= & \texttt{*} \mid \texttt{data(}\textit{termpair\_property}\texttt{)} \mid \texttt{el(}\textit{pset\_property}\texttt{)} \\
 & & \mid\texttt{-}\ \textit{node\_property} \mid \textit{node\_property}\ \texttt{+}\ \textit{node\_property} \\
\textit{pset\_property} & ::= & \texttt{\#} \mid \texttt{sum(}\textit{process\_property}\texttt{)} \mid \texttt{min(}\textit{process\_property}\texttt{)} \\
\textit{process\_property} & ::= & \texttt{e(}\textit{eqs\_property}\texttt{)} \mid \texttt{r(}\textit{trs\_property}\texttt{)} \mid \texttt{c(}\textit{trs\_property}\texttt{)} \\
 & & \mid \textit{process\_property}\ \texttt{+}\ \textit{process\_property} \\
\textit{trs\_property} & ::= & \texttt{sum(}\textit{termpair\_property}\texttt{)} \mid \texttt{cp(}\textit{eqs\_property}\texttt{)} \mid \texttt{\#} \\
\textit{eqs\_property} & ::= & \texttt{sum(}\textit{termpair\_property}\texttt{)} \mid \texttt{\#} \\
\textit{termpair\_property} & ::= & \texttt{sizemax} \mid \texttt{sizesum}
\end{array}
$$

Example (*sum*)

process with small $E_p(\mathcal{N})$ and $C_p(\mathcal{N})$, node with small terms and big $|E|$

```
(el(min(e(sum(sizesum))+c(sum(sizesum)))),(data(sizesum),(-el(#),?))))
```

Improvement 2: Process Isomorphisms



Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

$\mathcal{N}_0$

$$E_\epsilon = \begin{cases} (x * y) * z \approx x * (y * z) & \mathsf{f}(x * y) \approx \mathsf{f}(x) * \mathsf{f}(y) \\ x * 1 \approx x & \mathsf{g}(x * y) \approx \mathsf{g}(x) * \mathsf{g}(y) \\ \mathsf{i}(x) * x \approx 1 & \mathsf{g}(x) * \mathsf{f}(y) \approx \mathsf{f}(y) * \mathsf{g}(x) \end{cases} \qquad R_\epsilon = C_\epsilon = \varnothing$$

Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

$$\mathcal{N}_0 \vdash \mathcal{N}_1$$

Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2$$

Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \quad \cdots \quad \vdash \mathcal{N}_i$$

$$E_p = \begin{cases} (x * y) * z \approx x * (y * z) \\ x * 1 \;\; x \\ \mathrm{f}(1) \approx 1 \\ \mathrm{g}(1) \approx 1 \\ \mathrm{g}(x) * \mathrm{f}(y) \approx \mathrm{f}(y) * \mathrm{g}(x) \end{cases} \qquad R_p = C_p = \begin{cases} \mathrm{i}(x) * x \to 1 \\ \mathrm{f}(x * y) \to \mathrm{f}(x) * \mathrm{f}(y) \\ \mathrm{g}(x * y) \to \mathrm{g}(x) * \mathrm{g}(y) \end{cases}$$

Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

$$\text{orient}\quad \frac{\langle \mathrm{g}(x) * \mathrm{f}(y) : \mathrm{f}(y) * \mathrm{g}(x), \varnothing, \varnothing, \{p\}, \varnothing, \varnothing \rangle}{}$$

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \quad \cdots \quad \vdash \mathcal{N}_i \vdash$$

$$E_p = \begin{cases} (x * y) * z \approx x * (y * z) \\ x * 1 \;\; x \\ \mathrm{f}(1) \approx 1 \\ \mathrm{g}(1) \approx 1 \\ \mathrm{g}(x) * \mathrm{f}(y) \approx \mathrm{f}(y) * \mathrm{g}(x) \end{cases} \qquad R_p = C_p = \begin{cases} \mathrm{i}(x) * x \to 1 \\ \mathrm{f}(x * y) \to \mathrm{f}(x) * \mathrm{f}(y) \\ \mathrm{g}(x * y) \to \mathrm{g}(x) * \mathrm{g}(y) \end{cases}$$

Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

$$\text{orient}\quad \frac{\langle \mathrm{g}(x) * \mathrm{f}(y) : \mathrm{f}(y) * \mathrm{g}(x), \varnothing, \varnothing, \{p\}, \varnothing, \varnothing\rangle}{\langle \mathrm{g}(x) * \mathrm{f}(y) : \mathrm{f}(y) * \mathrm{g}(x), \{p0\}, \{p1\}, \varnothing, \varnothing, \varnothing\rangle}$$

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \quad \cdots \quad \vdash \mathcal{N}_i \vdash$$

$$E_p = \begin{cases} (x * y) * z \approx x * (y * z) \\ x * 1 \;\; x \\ \mathrm{f}(1) \approx 1 \\ \mathrm{g}(1) \approx 1 \\ \mathrm{g}(x) * \mathrm{f}(y) \approx \mathrm{f}(y) * \mathrm{g}(x) \end{cases} \qquad R_p = C_p = \begin{cases} \mathrm{i}(x) * x \to 1 \\ \mathrm{f}(x * y) \to \mathrm{f}(x) * \mathrm{f}(y) \\ \mathrm{g}(x * y) \to \mathrm{g}(x) * \mathrm{g}(y) \end{cases}$$

Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

orient $\dfrac{\langle \mathsf{g}(x) * \mathsf{f}(y) : \mathsf{f}(y) * \mathsf{g}(x), \varnothing, \varnothing, \{p\}, \varnothing, \varnothing\rangle}{\langle \mathsf{g}(x) * \mathsf{f}(y) : \mathsf{f}(y) * \mathsf{g}(x), \{p0\}, \{p1\}, \varnothing, \varnothing, \varnothing\rangle}$

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \quad \cdots \quad \vdash \mathcal{N}_i \vdash \mathcal{N}_{i+1}$$

$$E_{p0} = \begin{cases} (x * y) * z \approx x * (y * z) \\ x * 1 \approx x \\ \mathsf{f}(1) \approx 1 \\ \mathsf{g}(1) \approx 1 \end{cases} \qquad R_{p0} = C_{p0} = \begin{cases} \mathsf{i}(x) * x \to 1 \\ \mathsf{f}(x * y) \to \mathsf{f}(x) * \mathsf{f}(y) \\ \mathsf{g}(x * y) \to \mathsf{g}(x) * \mathsf{g}(y) \\ \mathsf{g}(x) * \mathsf{f}(y) \to \mathsf{f}(y) * \mathsf{g}(x) \end{cases}$$

$$E_{p1} = \begin{cases} (x * y) * z \approx x * (y * z) \\ x * 1 \approx x \\ \mathsf{f}(1) \approx 1 \\ \mathsf{g}(1) \approx 1 \end{cases} \qquad R_{p1} = C_{p1} = \begin{cases} \mathsf{i}(x) * x \to 1 \\ \mathsf{f}(x * y) \to \mathsf{f}(x) * \mathsf{f}(y) \\ \mathsf{g}(x * y) \to \mathsf{g}(x) * \mathsf{g}(y) \\ \mathsf{f}(y) * \mathsf{g}(x) \to \mathsf{g}(x) * \mathsf{f}(y) \end{cases}$$

Example (Renaming Isomorphism)

- mkb_{TT} run on system of commuting group endomorphisms CGE_2

$$\mathcal{N}_0 \vdash \mathcal{N}_1 \vdash \mathcal{N}_2 \vdash \quad \cdots \quad \vdash \mathcal{N}_i \vdash \mathcal{N}_{i+1}$$

$$E_{p0} = \begin{cases} (x * y) * z \approx x * (y * z) \\ x * 1 \approx x \\ \mathrm{f}(1) \approx 1 \\ \mathrm{g}(1) \approx 1 \end{cases} \qquad R_{p0} = C_{p0} = \begin{cases} \mathrm{i}(x) * x \to 1 \\ \mathrm{f}(x * y) \to \mathrm{f}(x) * \mathrm{f}(y) \\ \mathrm{g}(x * y) \to \mathrm{g}(x) * \mathrm{g}(y) \\ \mathrm{g}(x) * \mathrm{f}(y) \to \mathrm{f}(y) * \mathrm{g}(x) \end{cases}$$

$$E_{p1} = \begin{cases} (x * y) * z \approx x * (y * z) \\ x * 1 \approx x \\ \mathrm{f}(1) \approx 1 \\ \mathrm{g}(1) \approx 1 \end{cases} \qquad R_{p1} = C_{p1} = \begin{cases} \mathrm{i}(x) * x \to 1 \\ \mathrm{f}(x * y) \to \mathrm{f}(x) * \mathrm{f}(y) \\ \mathrm{g}(x * y) \to \mathrm{g}(x) * \mathrm{g}(y) \\ \mathrm{f}(y) * \mathrm{g}(x) \to \mathrm{g}(x) * \mathrm{f}(y) \end{cases}$$

- states of $p0$ and $p1$ are identical up to renaming function symbols

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

processes p, q are isomorphic in node set $\mathcal{N}$ if for some θ

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

processes p, q are isomorphic in node set $\mathcal{N}$ if for some θ

$$R_p(\mathcal{N}) \cong_\theta R_q(\mathcal{N})$$

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

processes p, q are isomorphic in node set $\mathcal{N}$ if for some θ

$$R_p(\mathcal{N}) \cong_\theta R_q(\mathcal{N}) \qquad C_p(\mathcal{N}) \cong_\theta C_q(\mathcal{N})$$

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

processes p, q are isomorphic in node set $\mathcal{N}$ if for some θ

$$R_p(\mathcal{N}) \cong_\theta R_q(\mathcal{N}) \qquad C_p(\mathcal{N}) \cong_\theta C_q(\mathcal{N}) \qquad E_p(\mathcal{N}) \cong_\theta E_q(\mathcal{N})$$

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

processes p, q are isomorphic in node set $\mathcal{N}$ if for some θ

$$R_p(\mathcal{N}) \cong_\theta R_q(\mathcal{N}) \qquad C_p(\mathcal{N}) \cong_\theta C_q(\mathcal{N}) \qquad E_p(\mathcal{N}) \cong_\theta E_q(\mathcal{N})$$

Lemma

Assume p, q are isomorphic in $\mathcal{N}$

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

processes p, q are isomorphic in node set $\mathcal{N}$ if for some θ

$$R_p(\mathcal{N}) \cong_\theta R_q(\mathcal{N}) \qquad C_p(\mathcal{N}) \cong_\theta C_q(\mathcal{N}) \qquad E_p(\mathcal{N}) \cong_\theta E_q(\mathcal{N})$$

Lemma

Assume p, q are isomorphic in $\mathcal{N}$ and $\exists$ fair $\mathcal{N} \vdash^* \mathcal{N}'$ with $E_p(\mathcal{N}') = \varnothing$.

Definition

$$\mathcal{R} \cong_\theta \mathcal{R}'$$

i.e., rewrite systems $\mathcal{R}, \mathcal{R}'$ are isomorphic via $\theta \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ if

- $\mathcal{R}' = \{\theta(l) \to \theta(r) \mid l \to r \in \mathcal{R}\}$,
- $\forall s, t \qquad s \to_{\mathcal{R}} t \quad$ if and only if $\quad \theta(s) \to_{\mathcal{R}'} \theta(t)$

Definition

processes p, q are isomorphic in node set $\mathcal{N}$ if for some θ

$$R_p(\mathcal{N}) \cong_\theta R_q(\mathcal{N}) \qquad C_p(\mathcal{N}) \cong_\theta C_q(\mathcal{N}) \qquad E_p(\mathcal{N}) \cong_\theta E_q(\mathcal{N})$$

Lemma

Assume p, q are isomorphic in $\mathcal{N}$ and $\exists$ fair $\mathcal{N} \vdash^* \mathcal{N}'$ with $E_p(\mathcal{N}') = \varnothing$.
Then $\exists\, \mathcal{N}''$ such that $\mathcal{N} \vdash^* \mathcal{N}''$ is fair and $E_q(\mathcal{N}'') = \varnothing$.

Example (Function Symbol Renamings)

- let ρ be arity-preserving permutation of $\mathcal{F}$

Example (Function Symbol Renamings)

- let ρ be arity-preserving permutation of $\mathcal{F}$
- renaming isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ & \end{cases}$$

Example (Function Symbol Renamings)

- let ρ be arity-preserving permutation of $\mathcal{F}$
- renaming isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ \rho(f)(\theta(t_1), \ldots, \theta(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Example (Function Symbol Renamings)

- let ρ be arity-preserving permutation of $\mathcal{F}$
- renaming isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ \rho(f)(\theta(t_1), \ldots, \theta(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Example (Argument Permutations)

- for $f \in \mathcal{F}$ with arity $n > 0$ choose permutation π_f of $\{1, \ldots, n\}$

Example (Function Symbol Renamings)

- let ρ be arity-preserving permutation of $\mathcal{F}$
- renaming isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ \rho(f)(\theta(t_1), \ldots, \theta(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Example (Argument Permutations)

- for $f \in \mathcal{F}$ with arity $n > 0$ choose permutation π_f of $\{1, \ldots, n\}$
- argument permutation isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ & \end{cases}$$

Example (Function Symbol Renamings)

- let ρ be arity-preserving permutation of $\mathcal{F}$
- renaming isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ \rho(f)(\theta(t_1), \ldots, \theta(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Example (Argument Permutations)

- for $f \in \mathcal{F}$ with arity $n > 0$ choose permutation π_f of $\{1, \ldots, n\}$
- argument permutation isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f(\theta(t_{\pi_f(1)}), \ldots, \theta(t_{\pi_f(n)})) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Example (Function Symbol Renamings)

- let ρ be arity-preserving permutation of $\mathcal{F}$
- renaming isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ \rho(f)(\theta(t_1), \ldots, \theta(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Example (Argument Permutations)

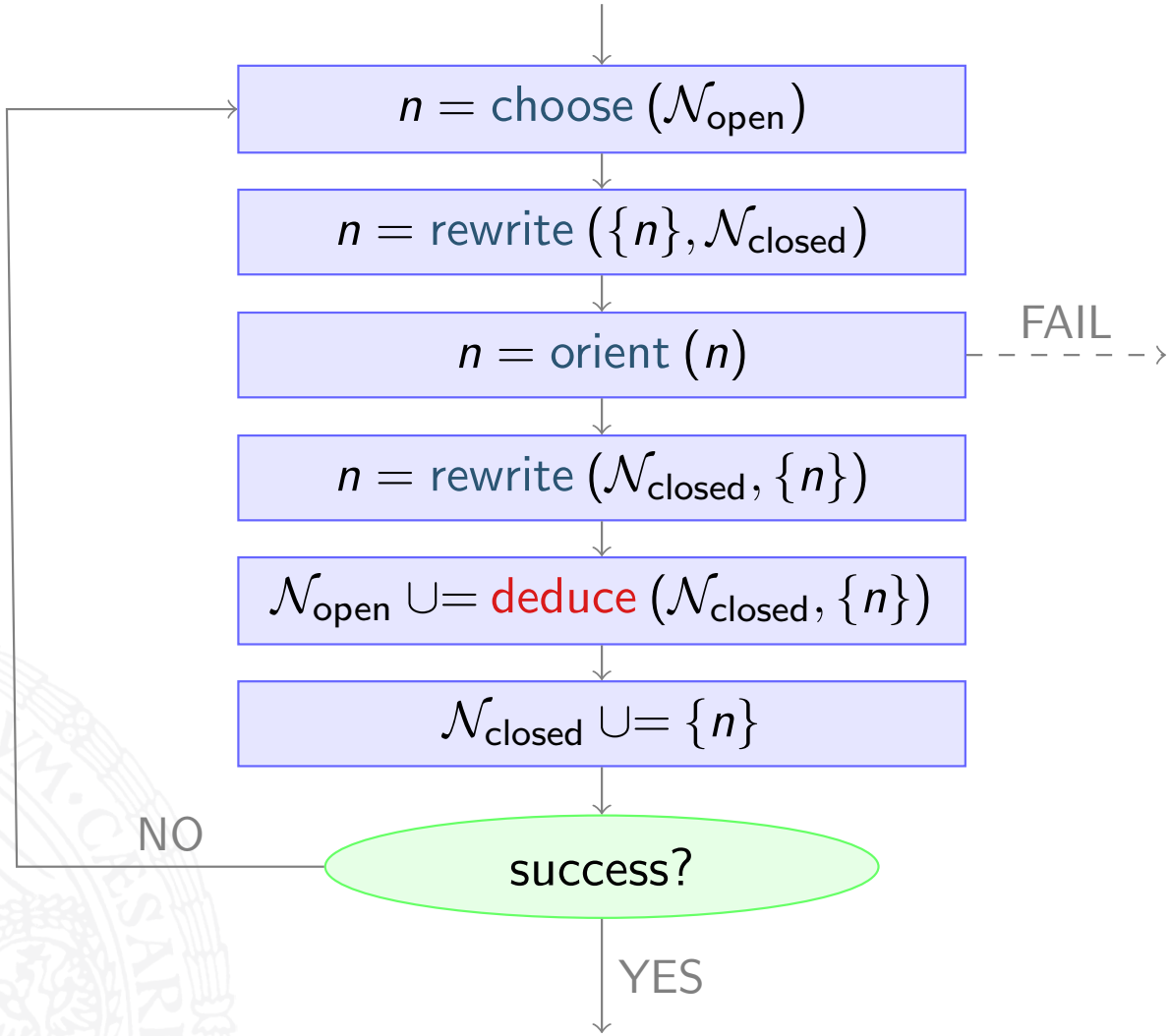
- for $f \in \mathcal{F}$ with arity $n > 0$ choose permutation π_f of $\{1, \ldots, n\}$
- argument permutation isomorphism is given by

$$\theta(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f(\theta(t_{\pi_f(1)}), \ldots, \theta(t_{\pi_f(n)})) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Implementation

- check before process splits in orient, or repeatedly

Improvement 3: Critical Pair Criteria



Definition (Deduce in mkb_{TT})

$$\text{deduce} \quad \frac{\mathcal{N}}{\text{if } \langle l : r, R, \ldots \rangle, \langle l' : r', R', \ldots \rangle \in \mathcal{N}}$$

Definition (Deduce in mkb_{TT})

deduce $\dfrac{\mathcal{N}}{}$

if $\langle l : r, R, \ldots \rangle, \langle l' : r', R', \ldots \rangle \in \mathcal{N}$
such that $s \xleftarrow{l \to r} u \xrightarrow{l' \to r'} t$ and $s \approx t$ is critical pair

Definition (Deduce in mkb_{TT})

deduce
$$\frac{\mathcal{N}}{\mathcal{N} \cup \langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing, \varnothing \rangle}$$

if $\langle l : r, R, \ldots \rangle, \langle l' : r', R', \ldots \rangle \in \mathcal{N}$
such that $s \xleftarrow{l \to r} u \xrightarrow{l' \to r'} t$ and $s \approx t$ is critical pair

Definition (Deduce in mkb_{TT})

deduce
$$\frac{\mathcal{N}}{\mathcal{N} \cup \langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing, \varnothing \rangle}$$

if $\langle l : r, R, \ldots \rangle, \langle l' : r', R', \ldots \rangle \in \mathcal{N}$
such that $s \xleftarrow{l \to r} u \xrightarrow{l' \to r'} t$ and $s \approx t$ is critical pair

Example

$\mathcal{N}$:

$$\langle \sqrt{-x+x} : 0,\ \{0,1\}, \ldots \rangle \quad (1)$$
$$\langle -0+0 : 0,\ \{0,1\}, \ldots \rangle \quad (2)$$
$$\langle -0 : 0,\ \{0,1\}, \ldots \rangle \quad (3)$$

$\mathrm{CP}(\mathcal{N})$:

$$\langle \sqrt{0} : 0,\ \varnothing, \varnothing,\ \{0,1\}, \ldots \rangle \quad \text{from } \langle (1), 1, (2) \rangle$$
$$\langle \sqrt{0+0} : 0,\ \varnothing, \varnothing,\ \{0,1\}, \ldots \rangle \quad \text{from } \langle (1), 11, (3) \rangle$$
$$\langle \sqrt{0} : 0,\ \varnothing, \varnothing,\ \{0,1\}, \ldots \rangle \quad \text{from } \langle (2), 1, (3) \rangle$$

Definition (Deduce in mkb_{TT})

deduce
$$\frac{\mathcal{N}}{\mathcal{N} \cup \langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing, \varnothing \rangle}$$

if $\langle l : r, R, \ldots \rangle, \langle l' : r', R', \ldots \rangle \in \mathcal{N}$
such that $s \xleftarrow{l \to r} u \xrightarrow{l' \to r'} t$ and $s \approx t$ is critical pair

Example

all critical pairs required?

$\mathcal{N}$:

$$\langle \sqrt{-x+x} : 0,\ \{0,1\}, \ldots \rangle \quad (1)$$
$$\langle -0+0 : 0,\ \{0,1\}, \ldots \rangle \quad (2)$$
$$\langle -0 : 0,\ \{0,1\}, \ldots \rangle \quad (3)$$

$\mathrm{CP}(\mathcal{N})$:

$$\langle \sqrt{0} : 0,\ \varnothing, \varnothing,\ \{0,1\}, \ldots \rangle \quad \text{from } \langle (1), 1, (2) \rangle$$
$$\langle \sqrt{0+0} : 0,\ \varnothing, \varnothing,\ \{0,1\}, \ldots \rangle \quad \text{from } \langle (1), 11, (3) \rangle$$
$$\langle \sqrt{0} : 0,\ \varnothing, \varnothing,\ \{0,1\}, \ldots \rangle \quad \text{from } \langle (2), 1, (3) \rangle$$

Definition (Deduce in mkb_{TT})

deduce
$$\frac{\mathcal{N}}{\mathcal{N} \cup \langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing, \varnothing \rangle}$$

if $\langle l : r, R, \ldots \rangle, \langle l' : r', R', \ldots \rangle \in \mathcal{N}$
such that $s \xleftarrow{l \to r} u \xrightarrow{l' \to r'} t$ and $s \approx t$ is critical pair

Critical Pair Criteria in mkb_{TT}

- primality criterion PCP — Kapur et al '88
- blocking criterion BCP — Bachmair/Dershowitz '88
- connectedness criterion CCP — Küchlin '85

Definition (Deduce in mkb_{TT})

deduce
$$\frac{\mathcal{N}}{\mathcal{N} \cup \langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing, \varnothing \rangle}$$

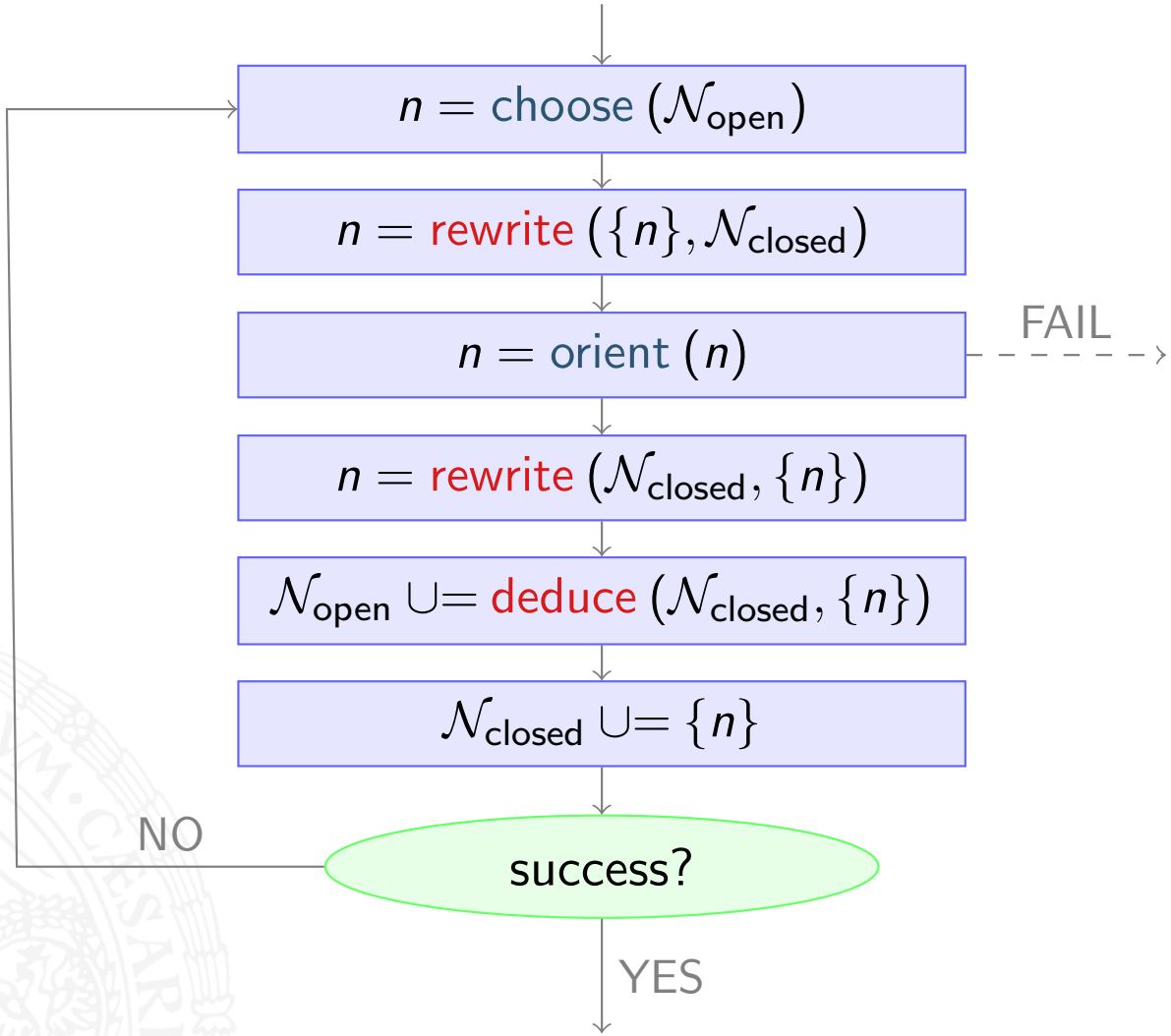
if $\langle l : r, R, \ldots \rangle, \langle l' : r', R', \ldots \rangle \in \mathcal{N}$
such that $s \xleftarrow{l \to r} u \xrightarrow{l' \to r'} t$ and $s \approx t$ is critical pair

Critical Pair Criteria in mkb_{TT}

exploit sharing

- primality criterion PCP — Kapur et al '88
- blocking criterion BCP — Bachmair/Dershowitz '88
- connectedness criterion CCP — Küchlin '85

Improvement 4: Term Indexing



Term Indexing

Given

- set of terms L
- binary relation R on terms
- term t

Term Indexing

Given

- set of terms L
- binary relation R on terms
- term t

identify all $s \in L$ with $s \; R \; t$

Term Indexing

Given

- set of terms L — index
- binary relation R on terms — retrieval condition
- term t — query term

identify all $s \in L$ with $s \; R \; t$ — candidate terms

Term Indexing

Given

- set of terms L — index
- binary relation R on terms — retrieval condition
- term t — query term

identify all $s \in L$ with $s \; R \; t$ — candidate terms

Example

mkb_{TT} faces term indexing problem for

- retrieval of variants and encompassments in rewrite_1 and rewrite_2

Term Indexing

Given

- set of terms L — index
- binary relation R on terms — retrieval condition
- term t — query term

identify all $s \in L$ with $s \; R \; t$ — candidate terms

Example

mkb_{TT} faces term indexing problem for

- retrieval of variants and encompassments in rewrite_1 and rewrite_2
- retrieval of unifiable terms in deduce

Term Indexing

Given

- set of terms L — index
- binary relation R on terms — retrieval condition
- term t — query term

identify all $s \in L$ with $s \; R \; t$ — candidate terms

Example

mkb_{TT} faces term indexing problem for

- retrieval of variants and encompassments in rewrite_1 and rewrite_2
- retrieval of unifiable terms in deduce

Implementation

- path indexing and discrimination trees for unifiable terms
- additionally also code trees for encompassments and variants

Experiments

hardware:	AMD Opteron® 885, 2.6GHz, 64GB memory
database:	101 systems collected from various papers
settings:	600 seconds timeout, termination checks with $T_T T_2$

Experiments

hardware:	AMD Opteron® 885, 2.6GHz, 64GB memory
database:	101 systems collected from various papers
settings:	600 seconds timeout, termination checks with $T_T T_2$

Overall Result

- 74 (mkb_{TT} 2.0) instead of 48 (mkb_{TT} 1.0) systems solved
- speedup of 40%

Experiments

hardware:	AMD Opteron® 885, 2.6GHz, 64GB memory
database:	101 systems collected from various papers
settings:	600 seconds timeout, termination checks with $T_T T_2$

Overall Result

- 74 (mkb_{TT} 2.0) instead of 48 (mkb_{TT} 1.0) systems solved
- speedup of 40%

New!

First automatical completion of CGE_4 system

$$\begin{aligned} \mathrm{e} \cdot x &\approx x \\ x^{-} \cdot x &\approx \mathrm{e} \\ (x \cdot y) \cdot z &\approx x \cdot (y \cdot z) \end{aligned} \qquad \begin{aligned} \mathrm{f_i}(x \cdot y) &\approx \mathrm{f_i}(x) \cdot \mathrm{f_i}(y) \quad 1 \leq i \leq 4 \\ \mathrm{f_i}(x) \cdot \mathrm{f_j}(y) &\approx \mathrm{f_j}(y) \cdot \mathrm{f_i}(x) \quad 1 \leq i < j \leq 4 \end{aligned}$$

into a 38 rule convergent TRS in 622 seconds

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8

time in seconds

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞

time in seconds

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞
SK3.4	75	2	3	38
GRP484-1	252	∞	∞	∞

time in seconds

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞
SK3.4	75	2	3	38
GRP484-1	252	∞	∞	∞
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
# successes	74	71	69	66

time in seconds

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞
SK3.4	75	2	3	38
GRP484-1	252	∞	∞	∞
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
# successes	74	71	69	66
time	22.2	12.8	38.9	23.5

time in seconds

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞
SK3.4	75	2	3	38
GRP484-1	252	∞	∞	∞
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
# successes	74	71	69	66
time	22.2	12.8	38.9	23.5

time in seconds

Isomorphisms

- renaming isomorphisms

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞
SK3.4	75	2	3	38
GRP484-1	252	∞	∞	∞
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
# successes	74	71	69	66
time	22.2	12.8	38.9	23.5

time in seconds

Isomorphisms

- renaming isomorphisms
  - CGE_2: 4 instead of 138 seconds, CGE_3: 30 instead of 192 seconds

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞
SK3.4	75	2	3	38
GRP484-1	252	∞	∞	∞
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
# successes	74	71	69	66
time	22.2	12.8	38.9	23.5

time in seconds

Isomorphisms

- renaming isomorphisms
  - CGE_2: 4 instead of 138 seconds, CGE_3: 30 instead of 192 seconds
  - on database: number of processes and time decreased by 15%

Selection Strategies

	sum	max	slothrop	old
CGE_2	138	9	16	8
CGE_3	∞	190	343	∞
SK3.4	75	2	3	38
GRP484-1	252	∞	∞	∞
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
# successes	74	71	69	66
time	22.2	12.8	38.9	23.5

time in seconds

Isomorphisms

- renaming isomorphisms
  - CGE_2: 4 instead of 138 seconds, CGE_3: 30 instead of 192 seconds
  - on database: number of processes and time decreased by 15%
- argument permutations
  - no improvement

Critical Pair Criteria

	none	PCP		BCP		CCP		all	
	(1)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
Chr89-A_2	126	133	70	134	51	168	25	137	75
GRP463-1	8	5	24	7	24	9	9	6	27

(1) time in seconds (2) redundant critical pairs for successful process

Critical Pair Criteria

	none	PCP		BCP		CCP		all	
	(1)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
Chr89-A_2	126	133	70	134	51	168	25	137	75
GRP463-1	8	5	24	7	24	9	9	6	27
BGK94-D_8	∞	550	28	550	28	∞		549	28

(1) time in seconds (2) redundant critical pairs for successful process

Critical Pair Criteria

	none	PCP		BCP		CCP		all	
	(1)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
Chr89-A_2	126	133	70	134	51	168	25	137	75
GRP463-1	8	5	24	7	24	9	9	6	27
BGK94-D_8	∞	550	28	550	28	∞		549	28
WS06-1	138	139	0	140	0	139	0	138	0

(1) time in seconds (2) redundant critical pairs for successful process

Critical Pair Criteria

	none	PCP		BCP		CCP		all	
	(1)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
Chr89-A_2	126	133	70	134	51	168	25	137	75
GRP463-1	8	5	24	7	24	9	9	6	27
BGK94-D_8	∞	550	28	550	28	∞		549	28
WS06-1	138	139	0	140	0	139	0	138	0
⋮	⋮	⋮		⋮		⋮		⋮	
successes	70	71		71		70		71	

(1) time in seconds (2) redundant critical pairs for successful process

Critical Pair Criteria

	none	PCP		BCP		CCP		all	
	(1)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
Chr89-A_2	126	133	70	134	51	168	25	137	75
GRP463-1	8	5	24	7	24	9	9	6	27
BGK94-D_8	∞	550	28	550	28	∞		549	28
WS06-1	138	139	0	140	0	139	0	138	0
⋮	⋮	⋮		⋮		⋮		⋮	
successes	70	71		71		70		71	

(1) time in seconds (2) redundant critical pairs for successful process

Term Indexing

percentage of retrieval time compared to naive search

	path indexing	discrimination trees	code trees
encompassments	89	39	27
variants	19	6	6
unifiable terms	90	30	

Critical Pair Criteria

	none	PCP		BCP		CCP		all	
	(1)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
Chr89-A_2	126	133	70	134	51	168	25	137	75
GRP463-1	8	5	24	7	24	9	9	6	27
BGK94-D_8	∞	550	28	550	28	∞		549	28
WS06-1	138	139	0	140	0	139	0	138	0
⋮	⋮	⋮		⋮		⋮		⋮	
successes	70	71		71		70		71	

(1) time in seconds (2) redundant critical pairs for successful process

Term Indexing

percentage of retrieval time compared to naive search

	path indexing	discrimination trees	code trees
encompassments	89	39	27
variants	19	6	6
unifiable terms	90	30	
execution time	95	83	78

Conclusion

mkb_{TT} is automatic completion tool with

- indexing techniques (pay off)
- selection strategies (considerable impact – optimal one?)
- critical pair criteria (tiny improvements)
- isomorphisms (renamings are useful for special systems)

Conclusion

mkb_{TT} is automatic completion tool with

- indexing techniques (pay off)
- selection strategies (considerable impact – optimal one?)
- critical pair criteria (tiny improvements)
- isomorphisms (renamings are useful for special systems)

mkb_{TT} Online

various options can be controlled via web interface:
http://cl-informatik.uibk.ac.at/software/mkbtt