

# MultumNonMulta at TermComp 2018

Dieter Hofbauer

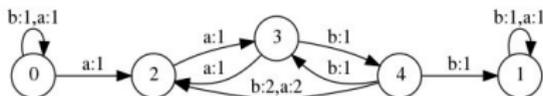
ASW – BA Saarland

16th Workshop on Termination  
Oxford, UK, July 18–19, 2018

## Termination Proof

for input system  
{ a a b b → b b b a a a }

### Step 1



Remains to prove termination of the system  
{ }

The system is trivially terminating.

1 MultumNonMultu

2 TermComp2018

3 Conclusion

1 MultumNonMultum

2 TermComp2018

3 Conclusion

1 MultumNonMultum

2 TermComp2018

3 Conclusion

## MultumNonMult: Mission statement

- Termination and non-termination proofs for **string rewriting**
- Prototype implementation: Demonstrate the power of combining **few but strong** termination proof methods.
- **Termination competition** 2006/'07 + 2016/'17/'18

# MultumNonMulta: Proof method

## Relative termination proofs

- Elimination of rules by **matrix interpretations** over  $\mathbb{N}$  (H/Waldmann'06)
- Synthesizing matrix interpretations by **backward completion** (H'13)
- Synthesizing **weight orderings** (a particular form of matrix interpretations) using the GNU Linear Programming Kit

## MultumNonMulta: Proof method

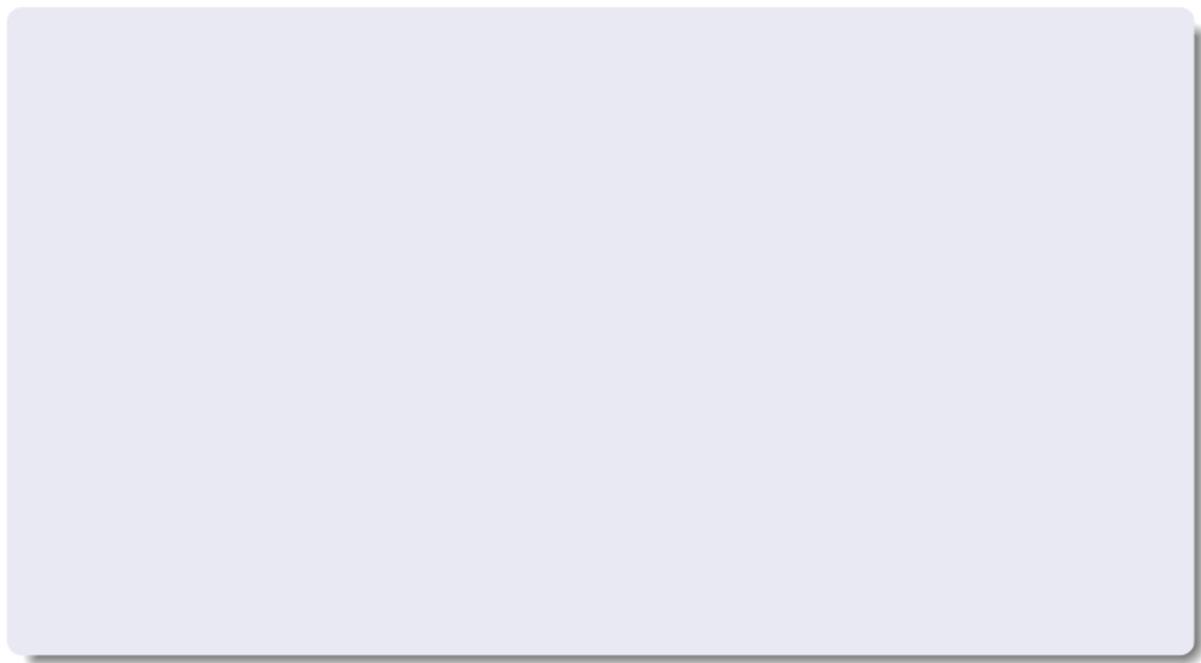
### Relative termination proofs

- Elimination of rules by **matrix interpretations** over  $\mathbb{N}$  (H/Waldmann'06)
- Synthesizing matrix interpretations by **backward completion** (H'13)
- Synthesizing **weight orderings** (a particular form of matrix interpretations) using the GNU Linear Programming Kit

### Non-termination proofs: Loops only

- Loop checker (H'09) enumerating forward closures based on (Geser/Zantema'99)

# MultumNonMult: Transformation methods



## MultumNonMult: Transformation methods

- Basic form of **DP transformation** (Arts/Giesl'00)

# MultumNonMultum: Transformation methods

- Basic form of **DP transformation** (Arts/Giesl'00)
- Simple form of **context closure**:  $R$  over  $\Sigma$  maps to  $\{\text{fold}(xly) \rightarrow \text{fold}(xry) \mid \ell \rightarrow r \in R, x, y \in \Sigma\}$  over  $\Sigma^2$ , where  $\text{fold}(a_1 \dots a_n) = (a_1, a_2) \dots (a_{n-1}, a_n)$ , cf. (Sternagel/Middeldorp'08)

# MultumNonMultum: Transformation methods

- Basic form of **DP transformation** (Arts/Giesl'00)
- Simple form of **context closure**:  $R$  over  $\Sigma$  maps to  $\{\text{fold}(xly) \rightarrow \text{fold}(xry) \mid \ell \rightarrow r \in R, x, y \in \Sigma\}$  over  $\Sigma^2$ , where  $\text{fold}(a_1 \dots a_n) = (a_1, a_2) \dots (a_{n-1}, a_n)$ , cf. (Sternagel/Middeldorp'08)
- **String reversal**: Each rule  $\ell \rightarrow r$  is replaced by  $\text{rev}(\ell) \rightarrow \text{rev}(r)$ , where  $\text{rev}(a_1 \dots a_n) = a_n \dots a_1$

# MultumNonMult: Transformation methods

- Basic form of **DP transformation** (Arts/Giesl'00)
- Simple form of **context closure**:  $R$  over  $\Sigma$  maps to  $\{\text{fold}(xly) \rightarrow \text{fold}(xry) \mid \ell \rightarrow r \in R, x, y \in \Sigma\}$  over  $\Sigma^2$ , where  $\text{fold}(a_1 \dots a_n) = (a_1, a_2) \dots (a_{n-1}, a_n)$ , cf. (Sternagel/Middeldorp'08)
- **String reversal**: Each rule  $\ell \rightarrow r$  is replaced by  $\text{rev}(\ell) \rightarrow \text{rev}(r)$ , where  $\text{rev}(a_1 \dots a_n) = a_n \dots a_1$
- For length-preserving systems, **rule inversion**: Each rule  $\ell \rightarrow r$  is replaced by  $r \rightarrow \ell$

## MultumNonMulta: Non multa

- **No** external constraint solver apart from GLPK
- **No** path orderings, no KBO
- **No** traditional polynomial interpretations
- **No** semantic labelling
- **No** DP refinements (SCC, usable rules, argument filtering ...)
- **No** match-bounds
- **No** other semirings (arctic ...)

## TermComp2018: SRS STANDARD

Don't take the following results too serious

- Almost every solver produces proofs where all others fail.
- Therefore, submitting new problem sets could change the picture completely.

## TermComp2018: SRS STANDARD

Don't take the following results too serious

- Almost every solver produces proofs where all others fail.
- Therefore, submitting new problem sets could change the picture completely.

YES / NO

MnM	802	174
AProVE	855	116
TTT2	707	37
NaTT		202
muterm		135

## TermComp2018: SRS STANDARD

## YES / NO

MnM	802	174
AProVE	855	116
TTT2	707	37
NaTT		202
muterm		135

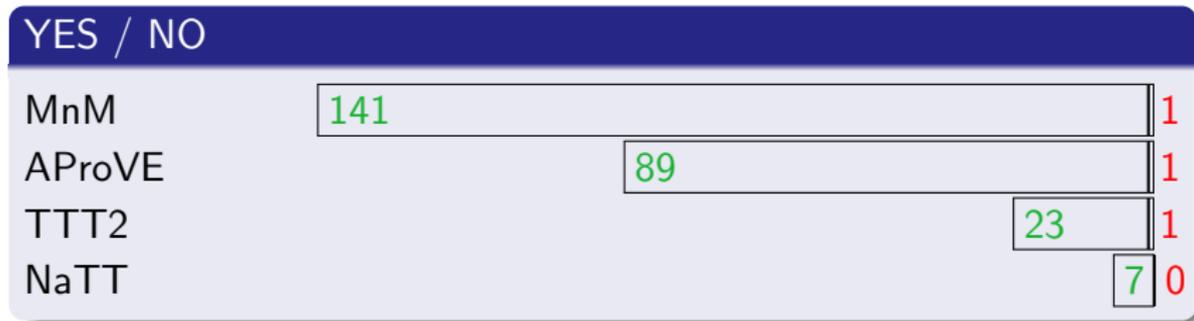
## With / without starexec runscript error

MnM	802	174
MnM	817	174

Howto: Get job output from starexec, then

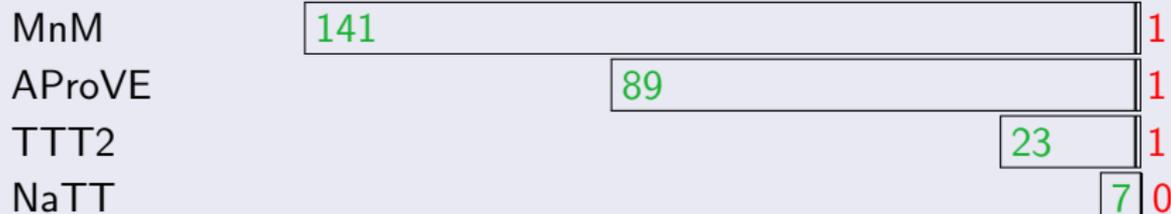
```
head -7 */solver/*/*.txt | grep "YES" | wc -l
```

## TermComp2018: SRS RELATIVE



## TermComp2018: SRS RELATIVE

YES / NO



With / without starexec runscript error



## Variant: Twice the time

MnM 300 vs 600 sec: SRS STANDARD

991 = 817 174

1031 = 853 178

## Variant: Twice the time

## MnM 300 vs 600 sec: SRS STANDARD

991 =	817	174
1031 =	853	178

## MnM 300 vs 600 sec: SRS RELATIVE

149 =	148	1
152 =	151	1

## Variant: Twice the time

## MnM 300 vs 600 sec: SRS STANDARD

991 =	817	174
1031 =	853	178

## MnM 300 vs 600 sec: SRS RELATIVE

149 =	148	1
152 =	151	1

Time helps – a bit.

## Variant: Twice the time

## MnM 300 vs 600 sec: SRS STANDARD

991 =	817	174
1031 =	853	178

## MnM 300 vs 600 sec: SRS RELATIVE

149 =	148	1
152 =	151	1

Time helps – a bit.

PS: Space helps – a lot.

## Variant: The dependency pairs dependency

MnM with

DP: SRS STANDARD

991 =

817

174

## Variant: The dependency pairs dependency

MnM with/without DP: SRS STANDARD

991 =	817	174
937 =	759	178

## Variant: The dependency pairs dependency

## MnM with/without DP: SRS STANDARD

991 =	817	174
937 =	759	178

## Some competitors with DP: SRS STANDARD

AProVE	855	116
TTT2	707	37

## Variant: The dependency pairs dependency

MnM with/without DP: SRS STANDARD

991 =	817	174
937 =	759	178

DP helps, but takes time from other threads.

# Conclusion

- 2017 → 2018: No methods added, but different combination

# Conclusion

- 2017 → 2018: No methods added, but different combination
- 2019: Implement **match-bounds**?

# Conclusion

- **2017 → 2018**: No methods added, but different combination
- 2019: Implement **match-bounds**?
- **CPF output**: Done for loops, but CeTA fails for (this basic form of) matrix interpretations

# Conclusion

- **2017 → 2018**: No methods added, but different combination
- 2019: Implement **match-bounds**?
- **CPF output**: Done for loops, but CeTA fails for (this basic form of) matrix interpretations
- Improve modular **design**