Polymorphic P Systems: How Much Win?

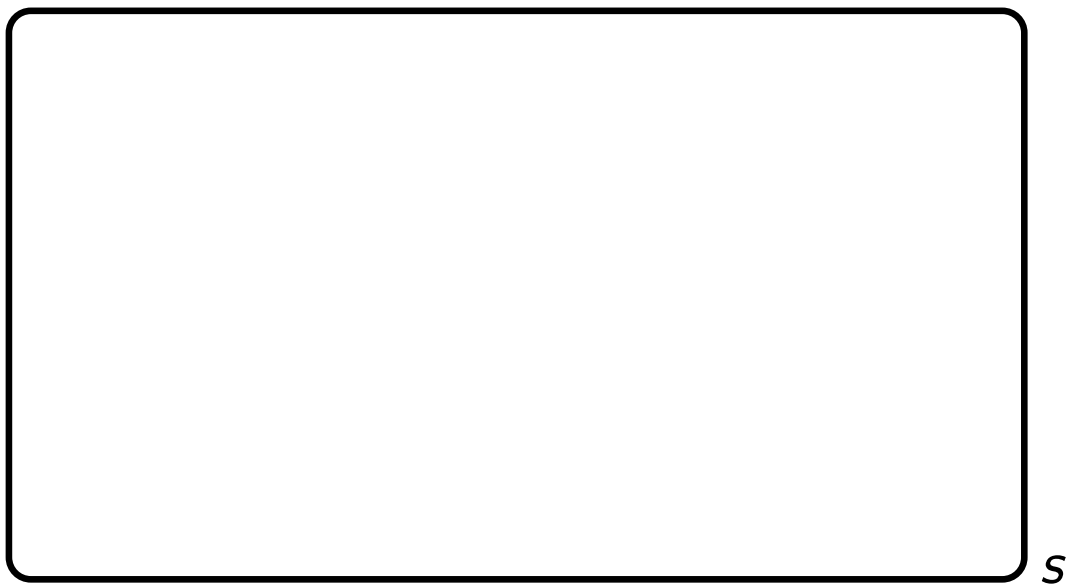
Sergiu Ivanov

LACL, Université Paris Est Créteil, France

BWMC 2015

Polymorphic P Systems [A. Alhazov, S.Ivanov, Yu. Rogozhin, 2010]

The Skin

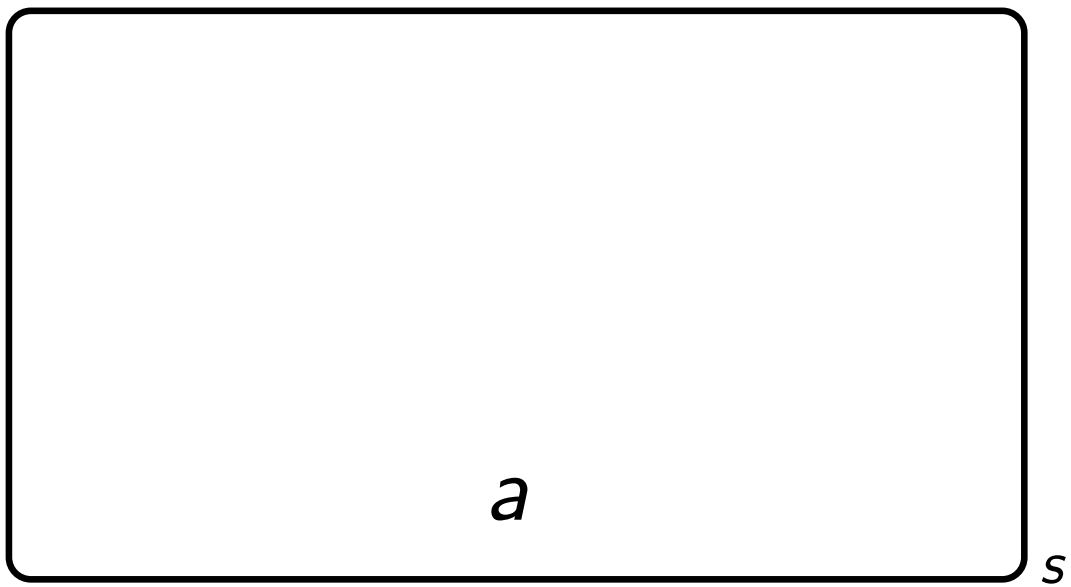


Polymorphic P Systems [A. Alhazov, S.Ivanov, Yu. Rogozhin, 2010]

The Skin

Contents

- multiset



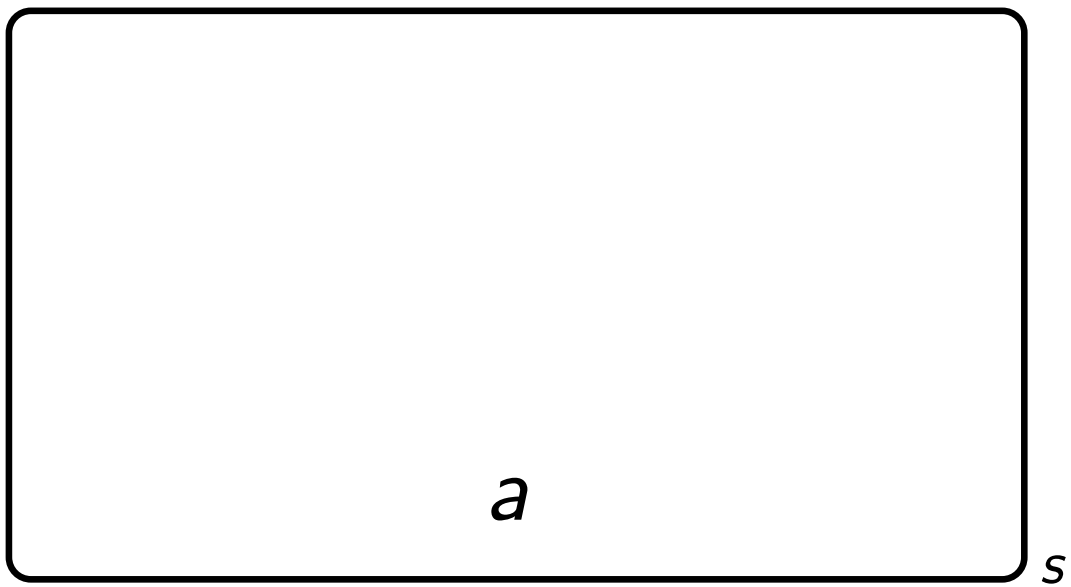
Polymorphic P Systems [A. Alhazov, S.Ivanov, Yu. Rogozhin, 2010]

The Skin

Contents

- multiset

No explicit rules!



Polymorphic P Systems [A. Alhazov, S.Ivanov, Yu. Rogozhin, 2010]

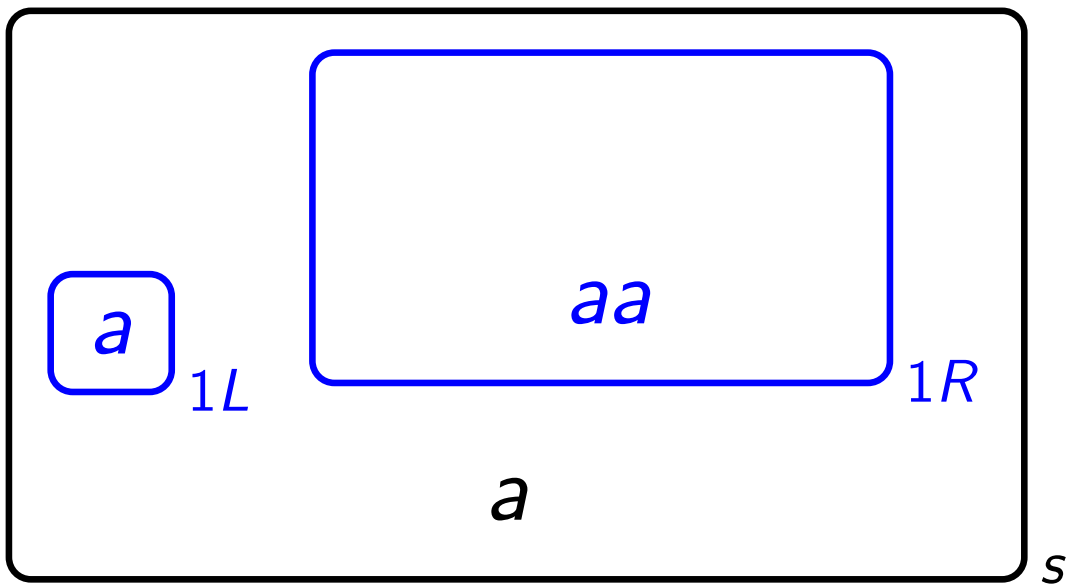
The Skin

Contents

- multiset

No explicit rules!

Pairs of membranes



Polymorphic P Systems [A. Alhazov, S.Ivanov, Yu. Rogozhin, 2010]

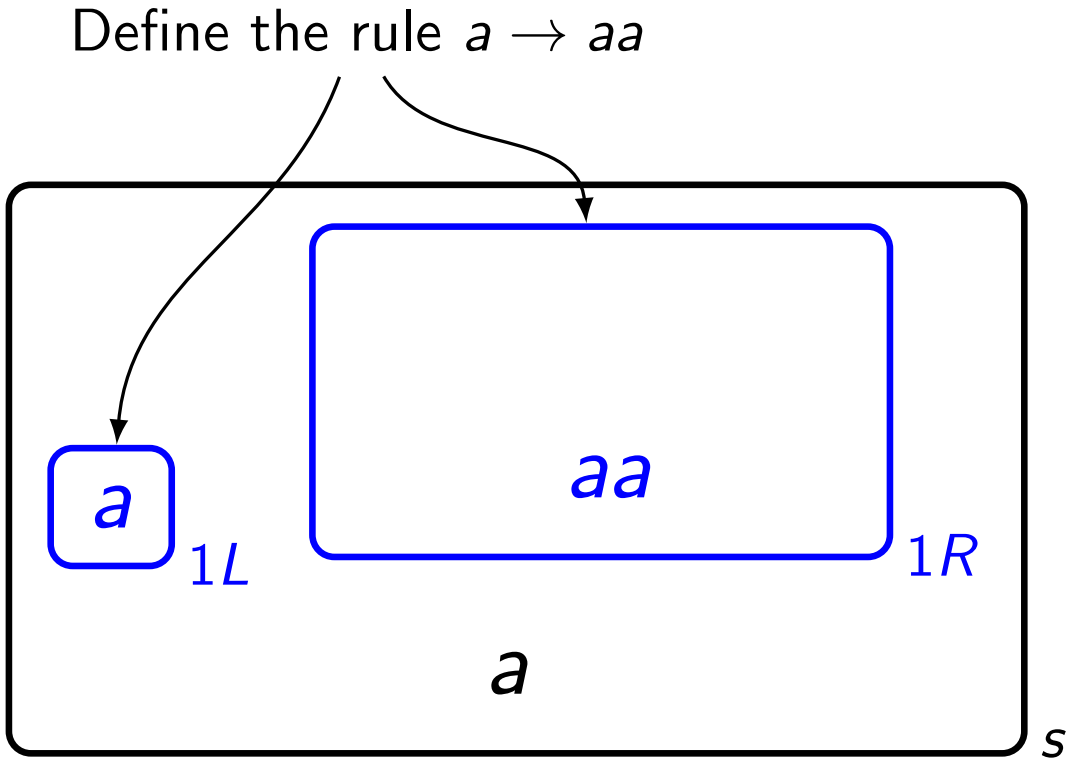
The Skin

Contents

- multiset

No explicit rules!

Pairs of membranes



Polymorphic P Systems [A. Alhazov, S.Ivanov, Yu. Rogozhin, 2010]

The Skin

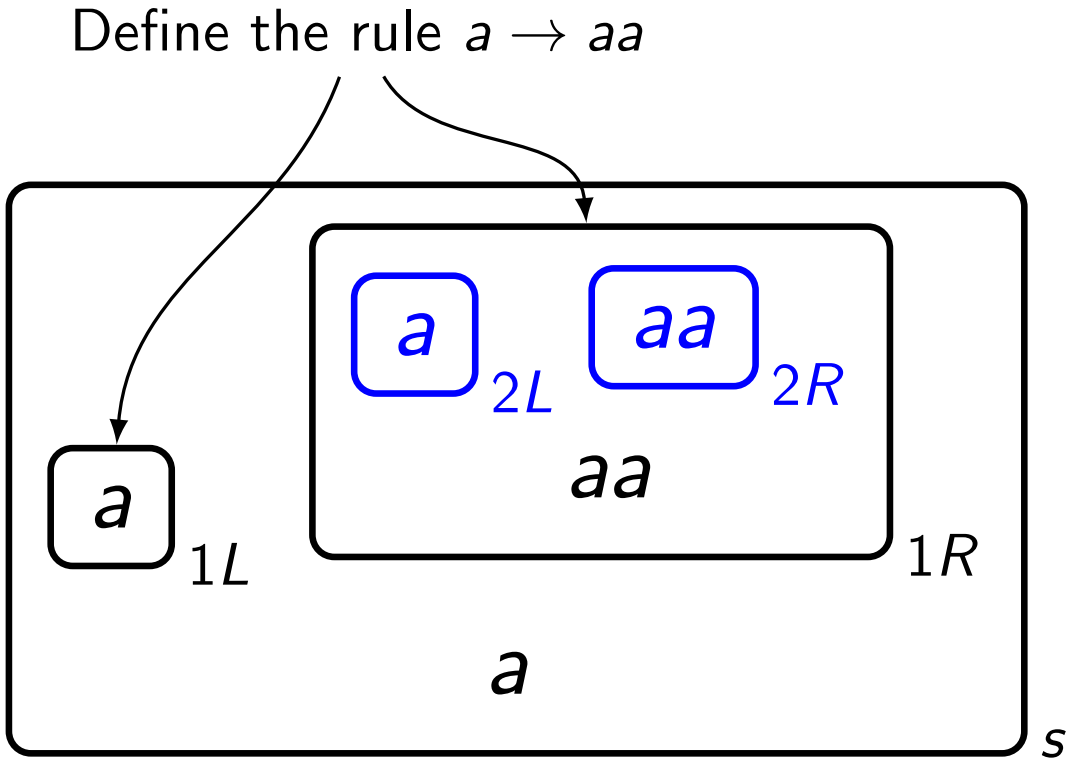
Contents

- multiset

No explicit rules!

Pairs of membranes

Rules modify rules



Polymorphic P Systems [A. Alhazov, S.Ivanov, Yu. Rogozhin, 2010]

The Skin

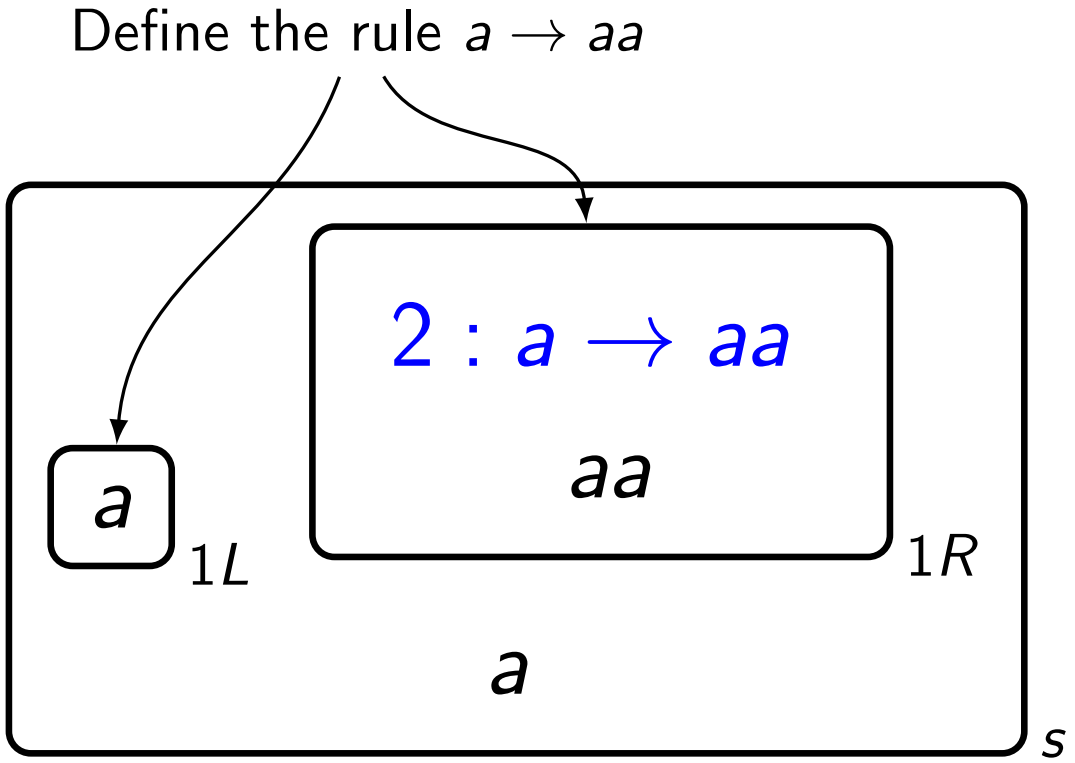
Contents

- multiset

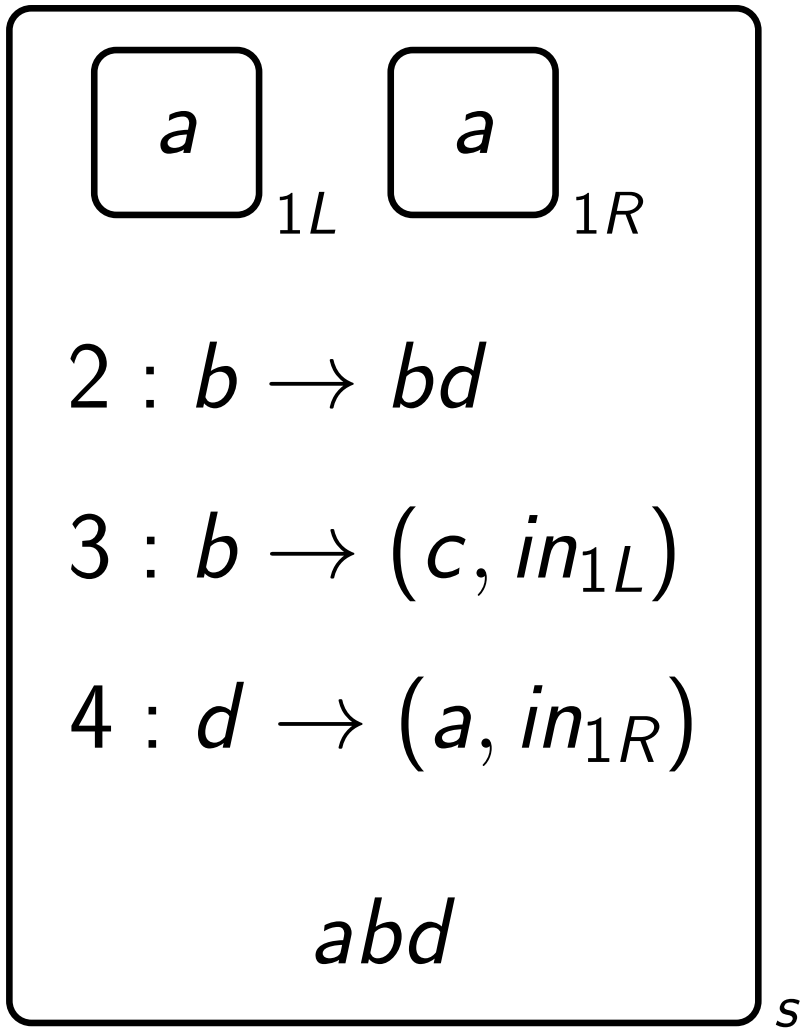
No explicit rules!

Pairs of membranes

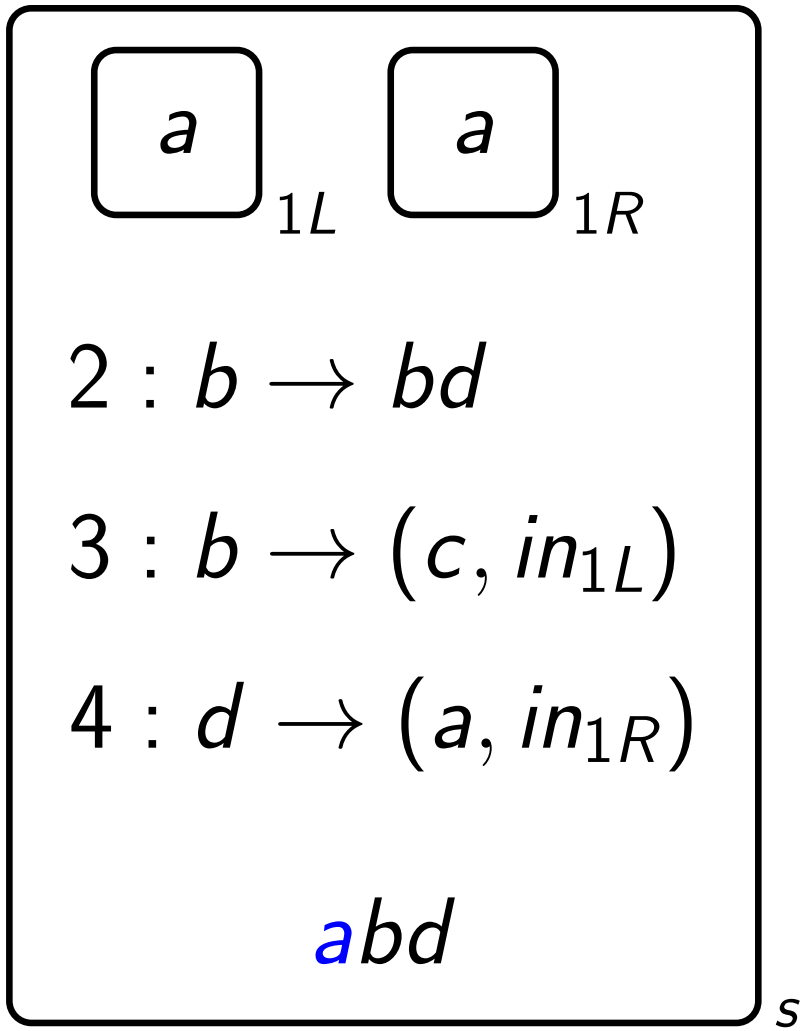
Rules modify rules



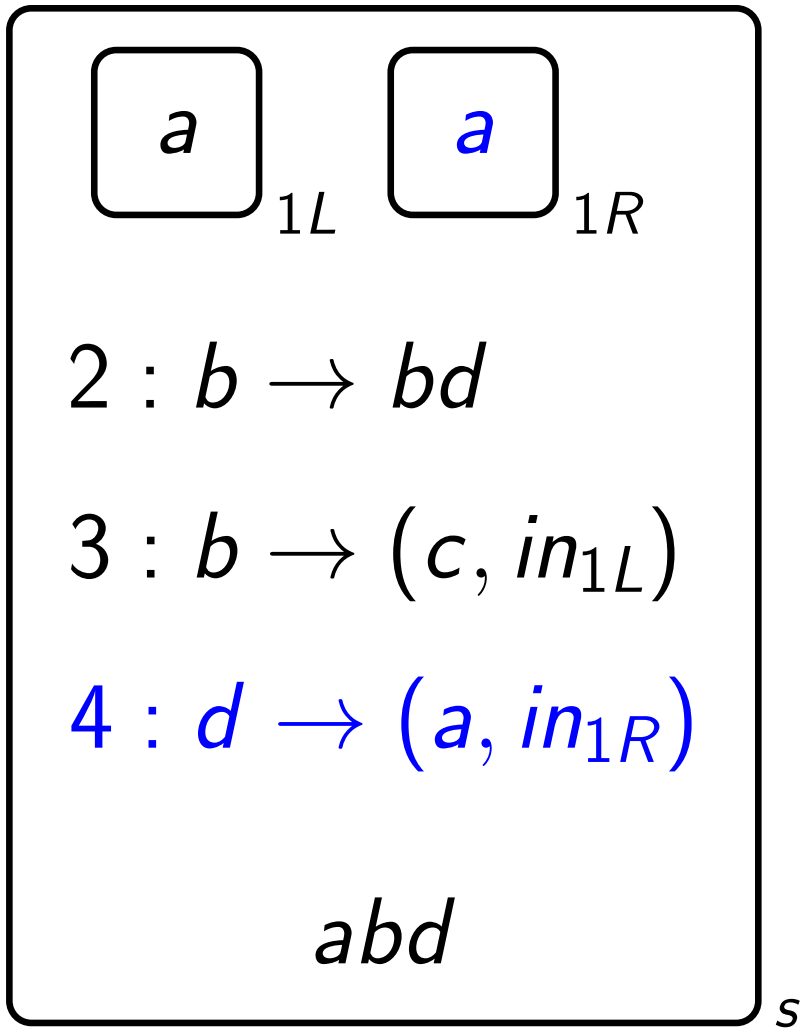
Generating $n!$



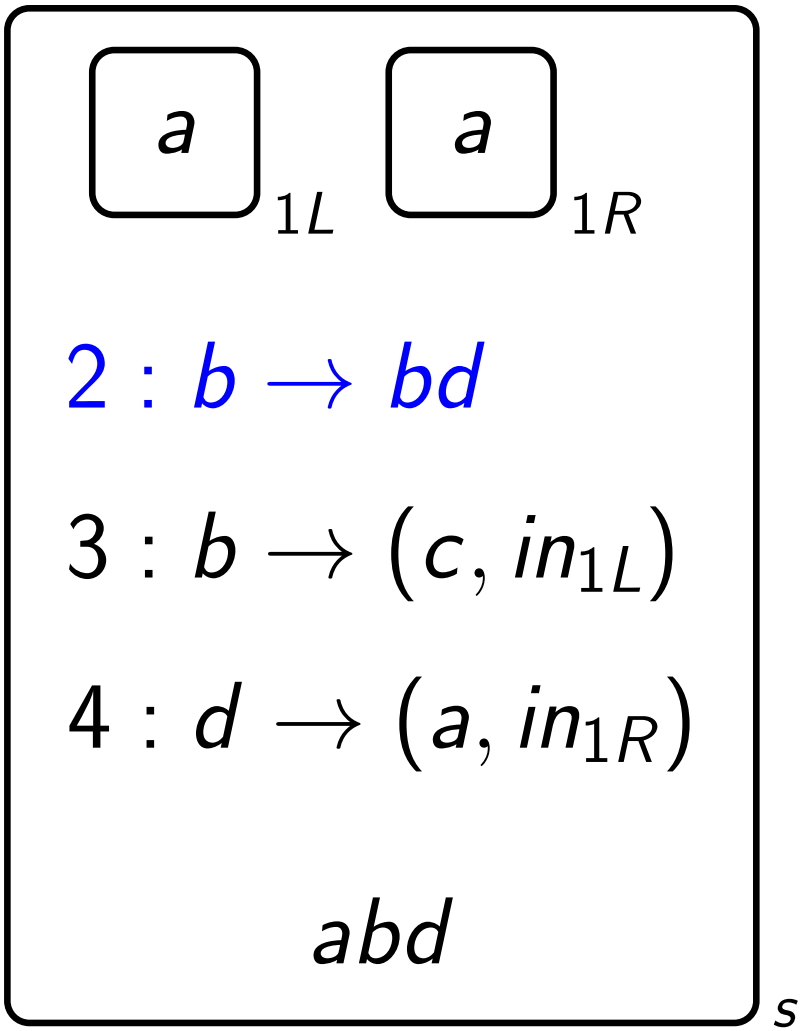
Generating $n!$



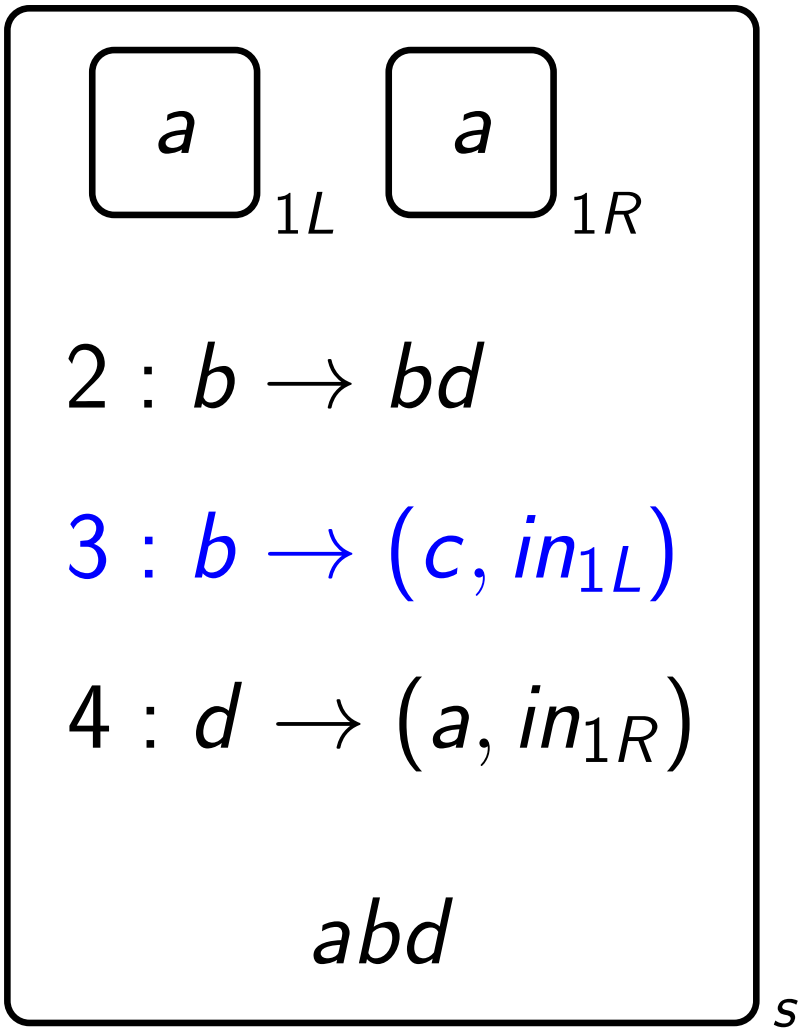
Generating $n!$



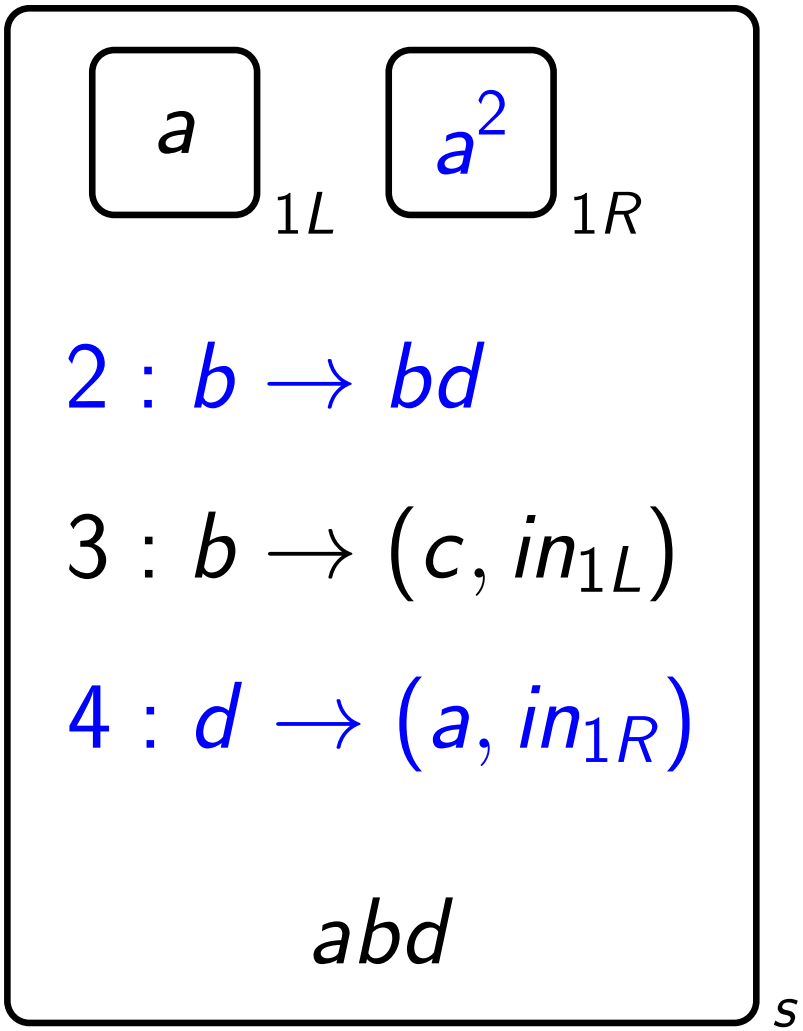
Generating $n!$



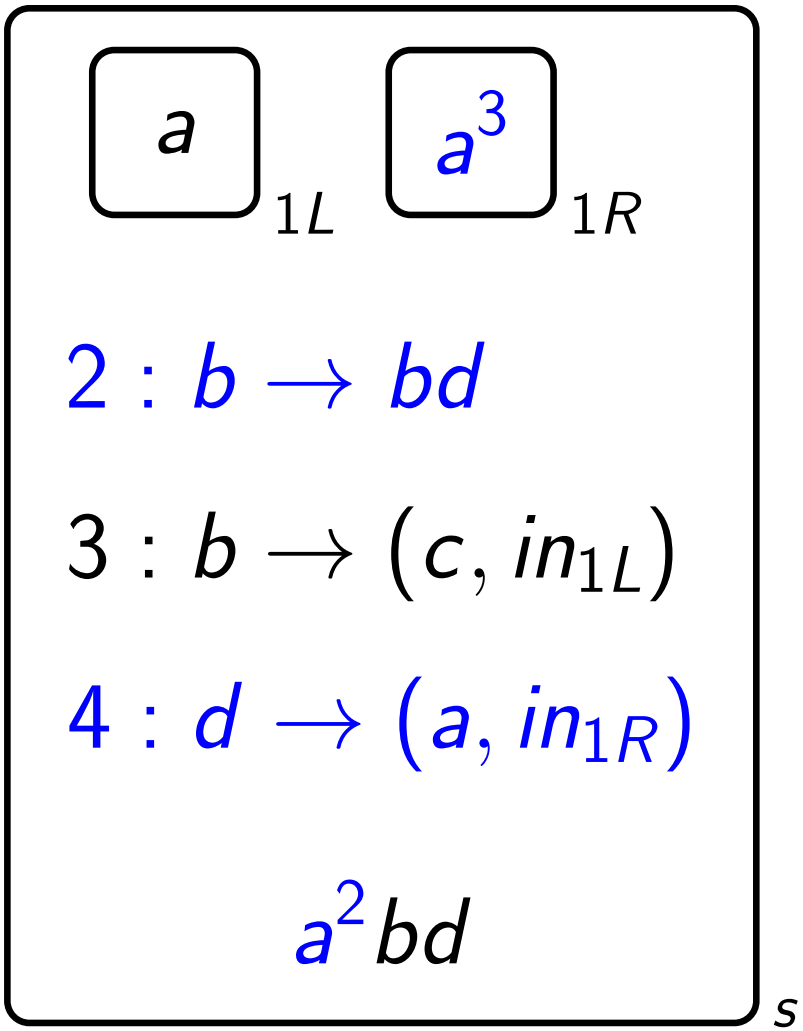
Generating $n!$



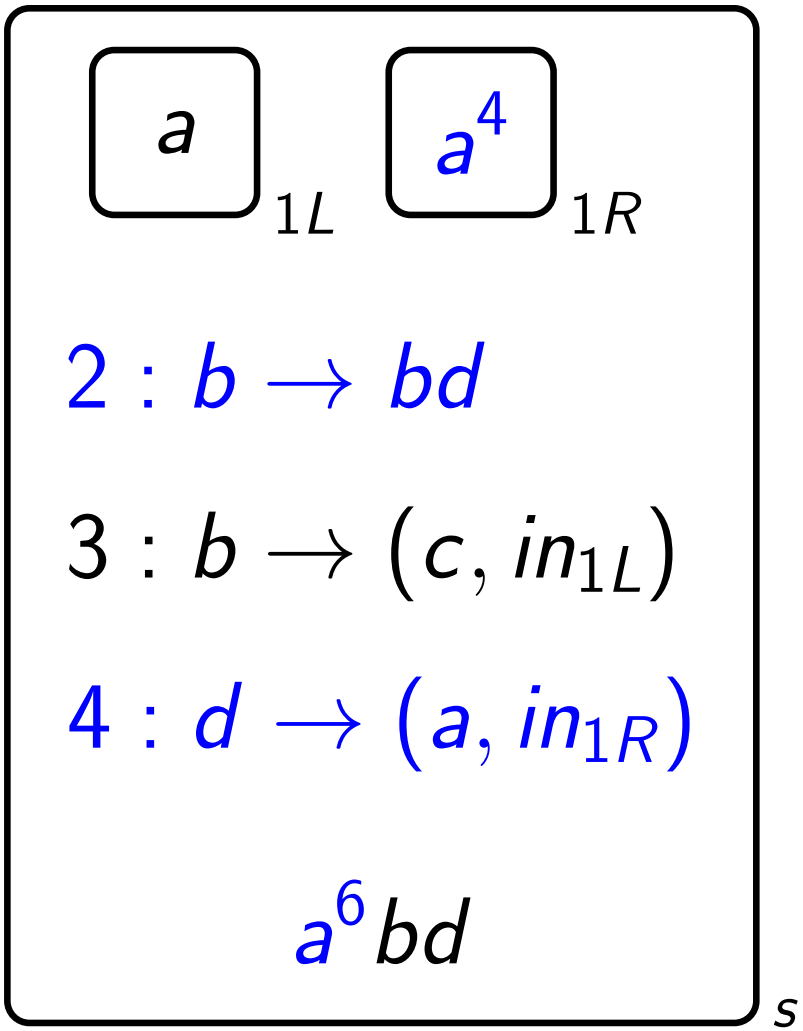
Generating $n!$



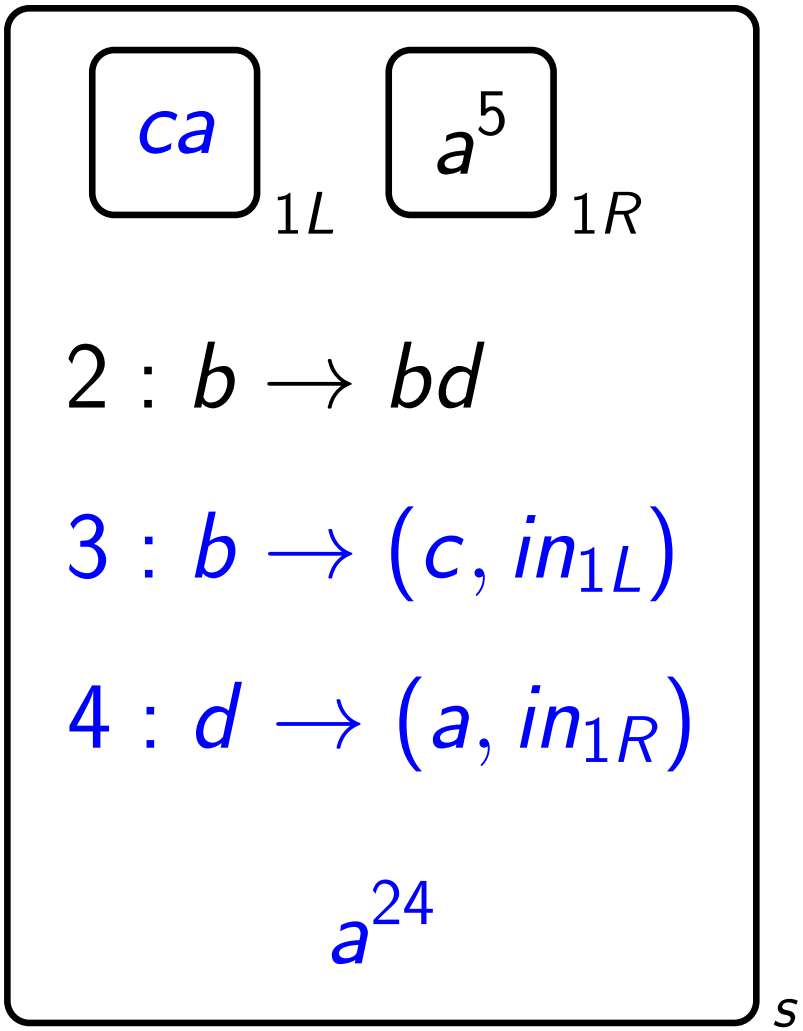
Generating $n!$



Generating $n!$



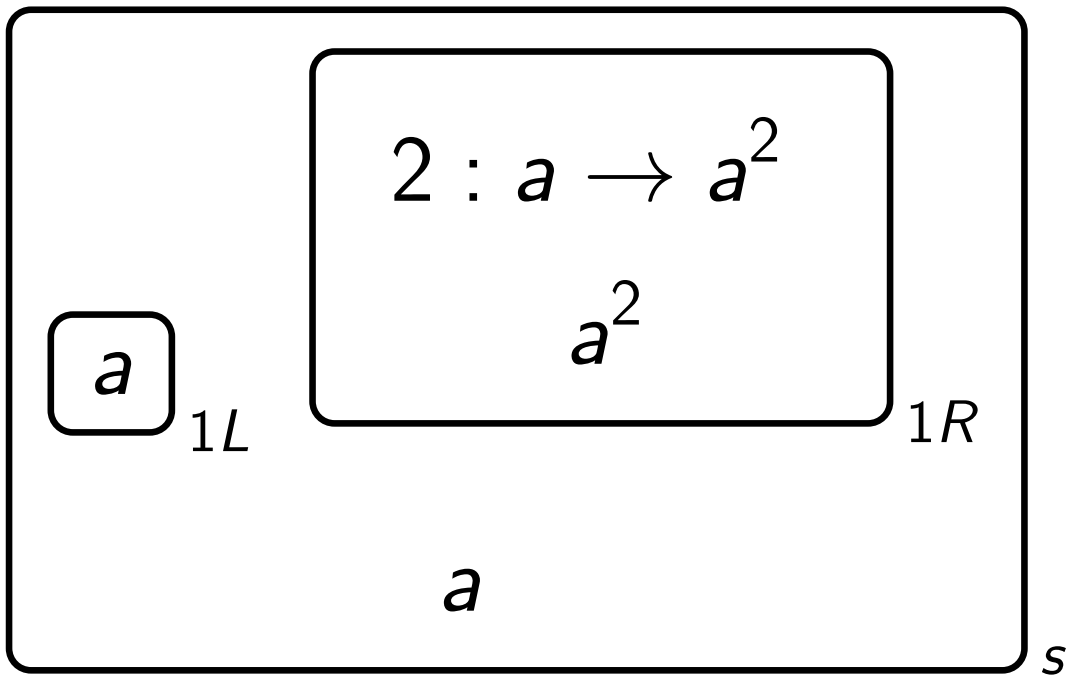
Generating $n!$



Superexponential Growth

Skin contents

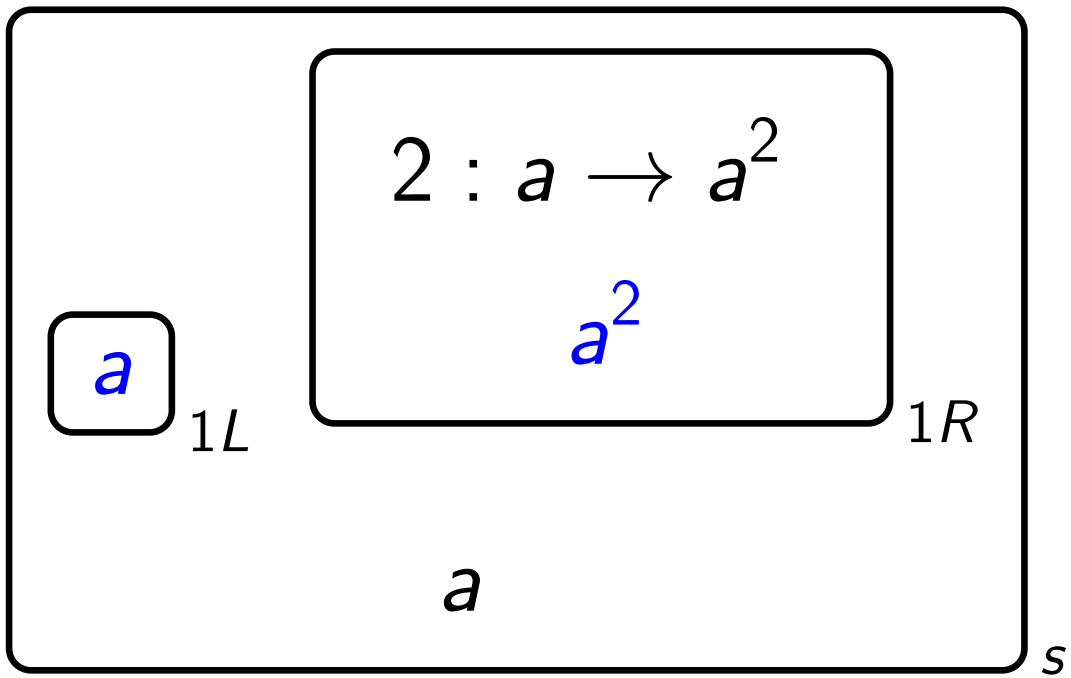
0. a



Superexponential Growth

Skin contents

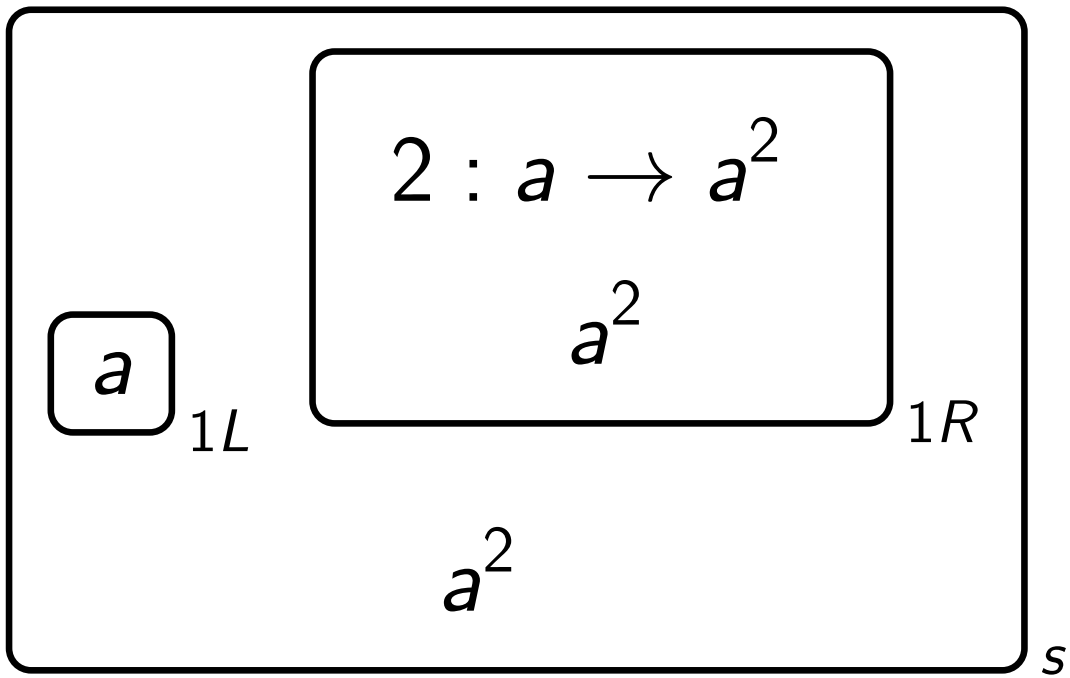
0. a



Superexponential Growth

Skin contents

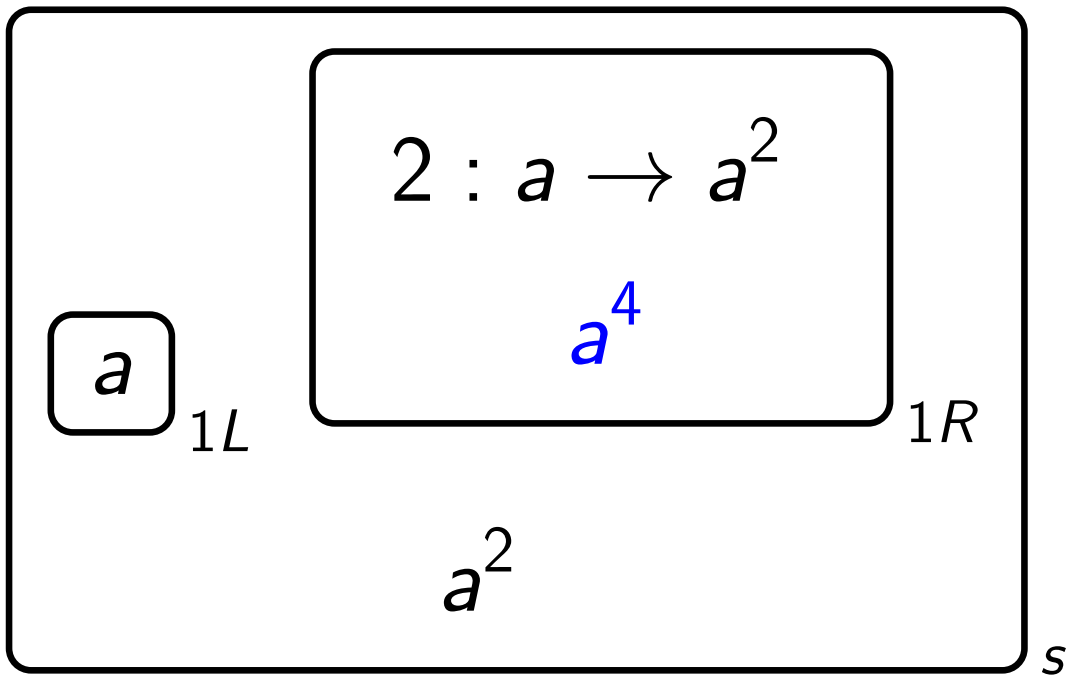
0. a
1. a^2



Superexponential Growth

Skin contents

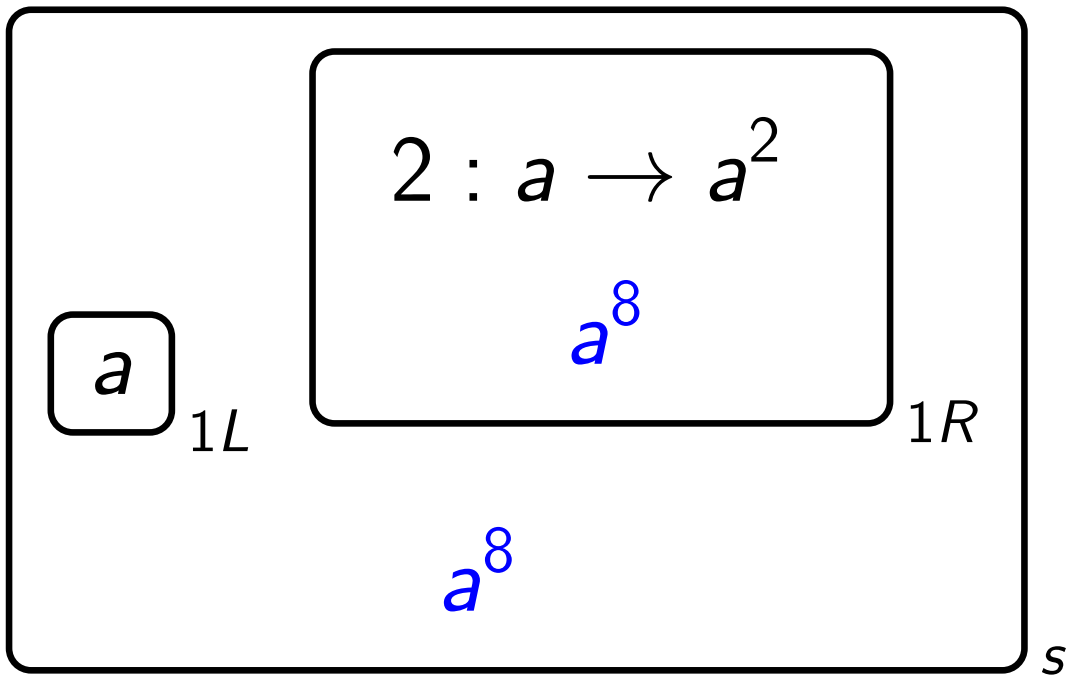
0. a
1. a^2



Superexponential Growth

Skin contents

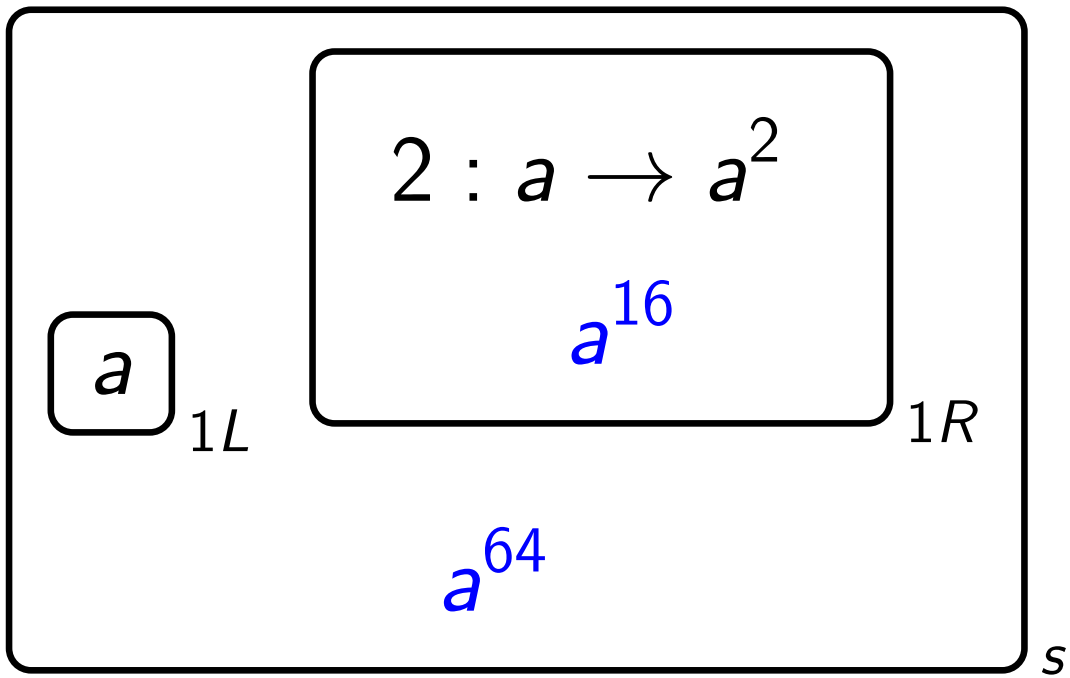
0. a
1. a^2
2. a^8



Superexponential Growth

Skin contents

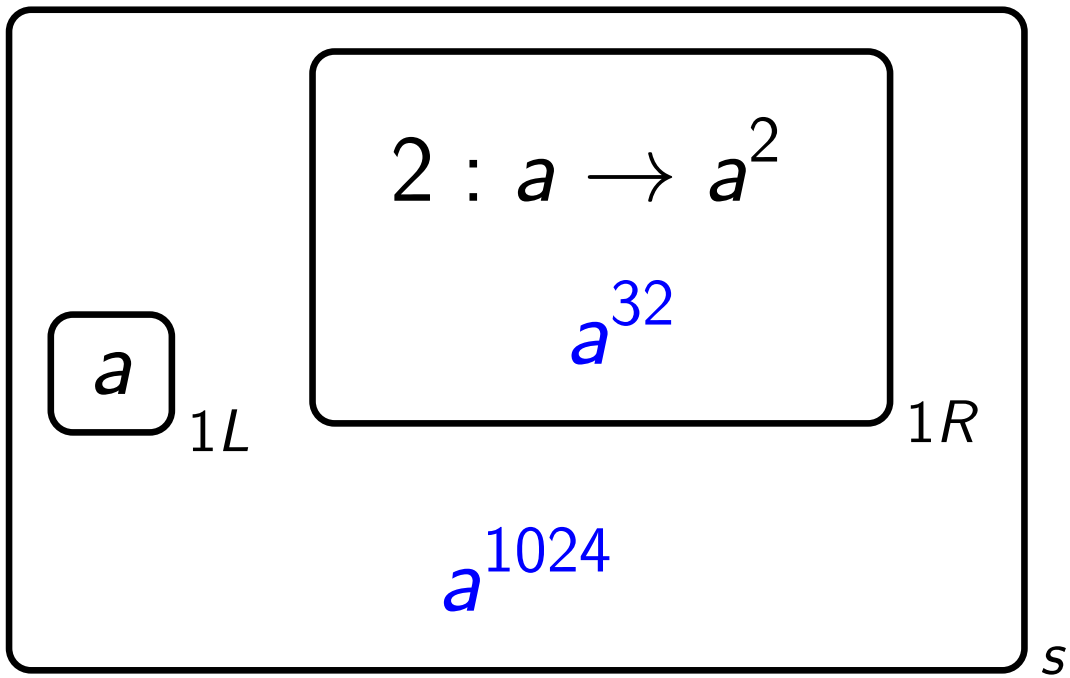
0. a
1. a^2
2. a^8
3. a^{64}



Superexponential Growth

Skin contents

0. a
1. a^2
2. a^8
3. a^{64}
4. a^{1024}



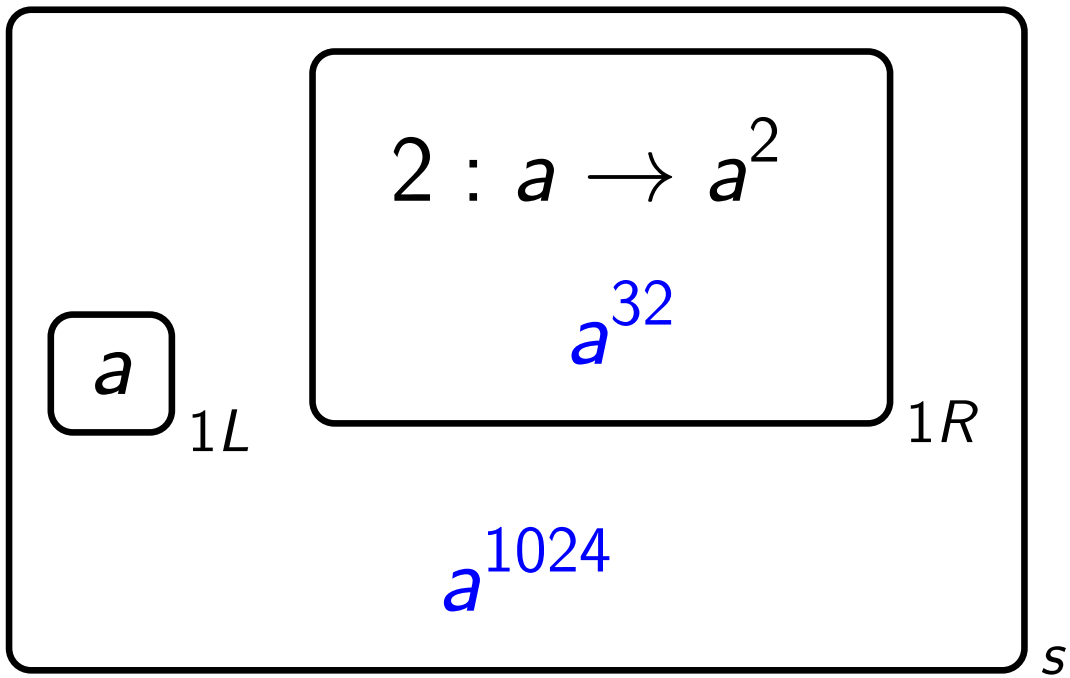
Superexponential Growth

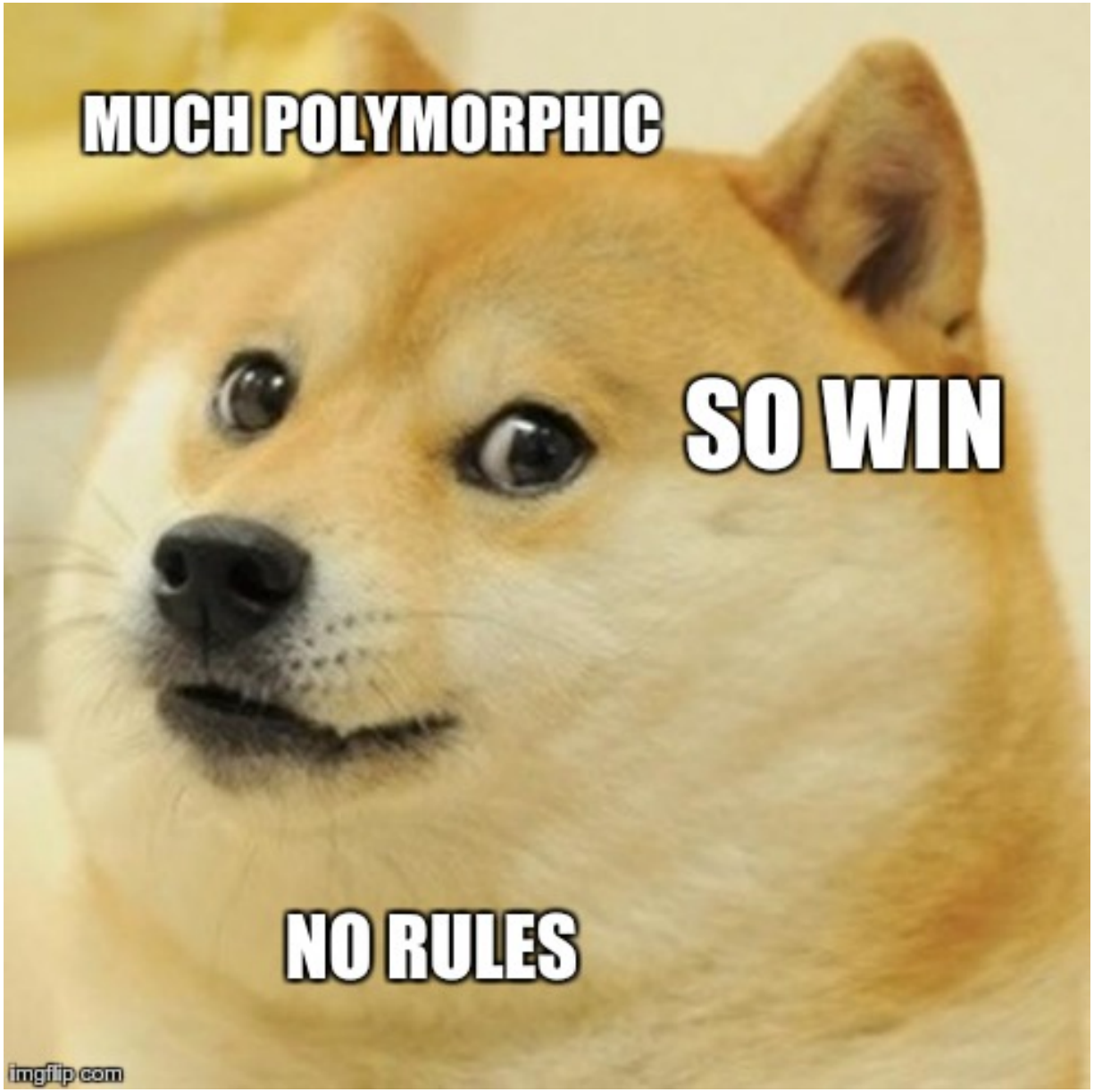
Skin contents

0. a
1. a^2
2. a^8
3. a^{64}
4. a^{1024}

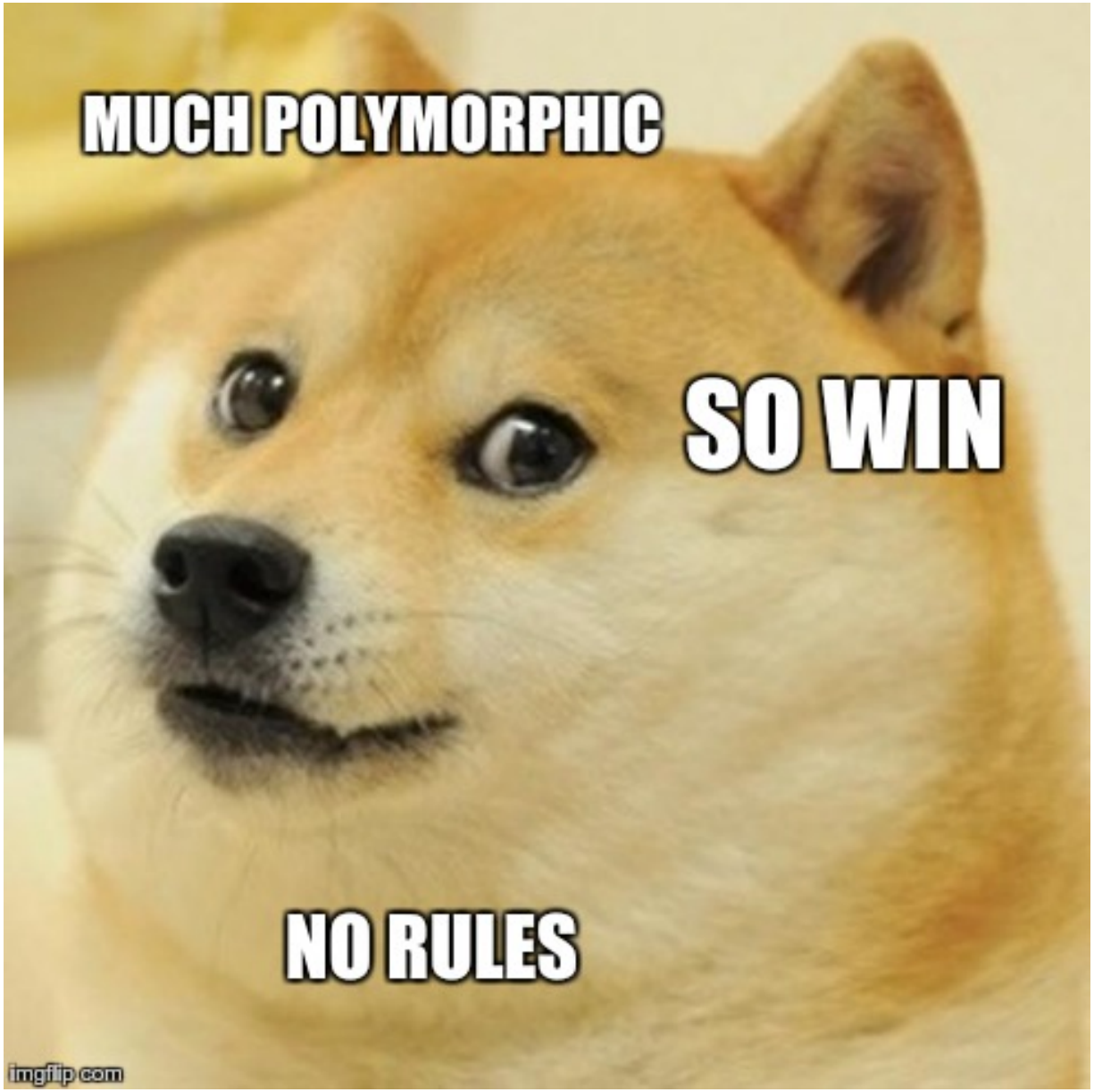
---

n. $a^{2^{\frac{n(n+1)}{2}}}$



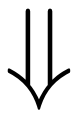

MUCH POLYMORPHIC
SO WIN
NO RULES
imgflip.com

How Much Win?



Intuitive Reasoning

Efficient solutions beyond P



Fast exhaustive search

Intuitive Reasoning

Efficient solutions beyond P

$\Downarrow$ (unless P = NP)

Fast exhaustive search

Intuitive Reasoning

Efficient solutions beyond P

$\Downarrow$ (unless P = NP)

Fast exhaustive search

- Adleman's experiment
- SAT with P systems

Intuitive Reasoning

Efficient solutions beyond P

$\Downarrow$ (unless P = NP)

Fast exhaustive search

- Adleman's experiment
- SAT with P systems

Candidates must be tracked

Questions

Polymorphism can replicate symbols a lot

Questions

Polymorphism can replicate symbols a lot

How about identities?

- interpretations of symbols

Questions

Polymorphism can replicate symbols a lot

How about identities?

- interpretations of symbols

Conjecture: No.

Questions

Polymorphism can replicate symbols a lot

How about identities?

- interpretations of symbols

Conjecture: No.

- identity traded for quantity

Questions

Polymorphism can replicate symbols a lot

How about identities?

- interpretations of symbols

Conjecture: No.

- identity traded for quantity

If No, how much speed-up?

Relevance for HPC

Program = Data

- Programs working in parallel on programs :-)

Relevance for HPC

Program = Data

- Programs working in parallel on programs :-)
- Self-modification vs. self-duplication

Relevance for HPC

Program = Data

- Programs working in parallel on programs :-)
- Self-modification vs. self-duplication
  - suffices for speed-up

Relevance for HPC

Program = Data

- Programs working in parallel on programs :-)
- Self-modification vs. self-duplication
  - suffices for speed-up
- Simple protocol of self-modification

Relevance for HPC

Program = Data

- Programs working in parallel on programs :-)
- Self-modification vs. self-duplication
  - suffices for speed-up
- Simple protocol of self-modification

Easier implementable?

Relevance for HPC

Program = Data

- Programs working in parallel on programs :-)
- Self-modification vs. self-duplication
  - suffices for speed-up
- Simple protocol of self-modification

Easier implementable?

- easily?

Questions

Polymorphism can replicate symbols a lot

How about identities?

- interpretations of symbols

Conjecture: No.

- identity traded for quantity

If No, how much speed-up?