Unfolding Grammars in Adhesive Categories

Paolo Baldan, Andrea Corradini, **Tobias Heindel**,
Barbara König, Paweł Sobociński

September 10, 2009

Unfolding Grammars in Adhesive Categories

Paolo Baldan, Andrea Corradini, **Tobias Heindel**,
Barbara König, Paweł Sobociński

September 10, 2009

A short history of unfoldings

Petri Nets	Nielsen, Plotkin, Winskel
	McMillan Meseguer, Montanari, Sassone
	Hayman, Winskel

A short history of unfoldings

Petri Nets	Nielsen, Plotkin, Winskel
	McMillan Meseguer, Montanari, Sassone
	Hayman, Winskel
Graph Transformation	
	Baldan, Corradini, Montanari
	Ribeiro

A short history of unfoldings

Petri Nets	Nielsen, Plotkin, Winskel
	McMillan Meseguer, Montanari, Sassone
	Hayman, Winskel
Graph Transformation	
	Baldan, Corradini, Montanari
	Ribeiro
Grammars in adhesive categories	
	Baldan, Corradini, Heindel, König, Sobociński

A short history of unfoldings

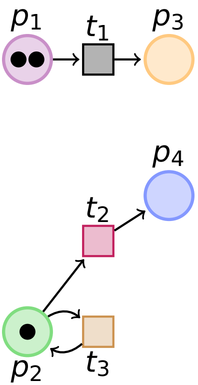
Petri Nets	Nielsen, Plotkin, Winskel
	McMillan Meseguer, Montanari, Sassone
	Hayman, Winskel
Graph Transformation	
	Baldan, Corradini, Montanari
	Ribeiro
Grammars in adhesive categories	
	Baldan, Corradini, Heindel, König, Sobociński

Previous work

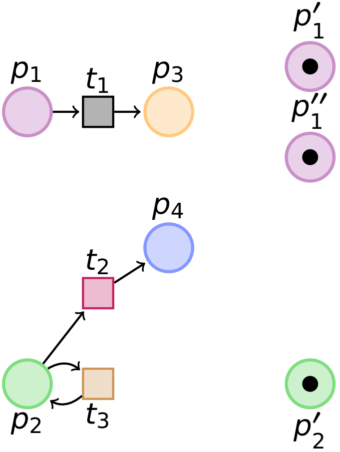
Processes for adhesive rewriting systems
Baldan, Corradini, Heindel, König, Sobociński

FOSSACS'06

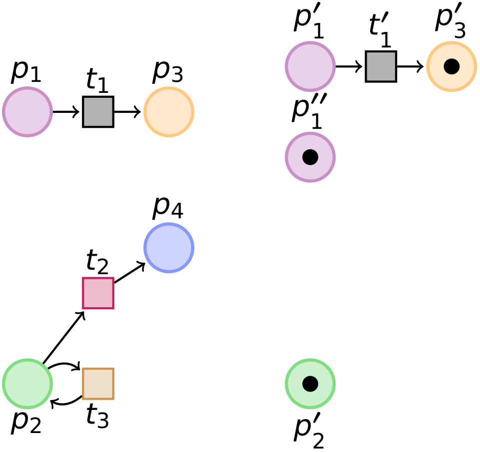
Unfolding as *“complete net behaviour”*



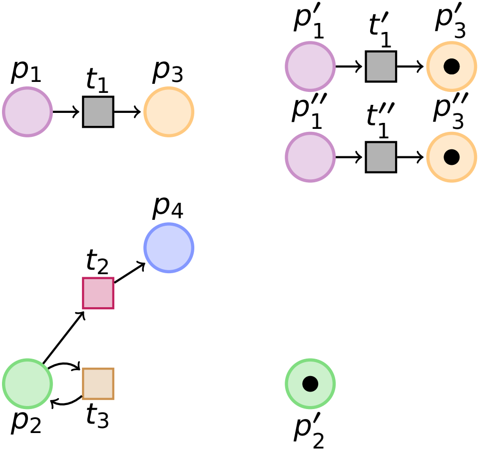
Unfolding as *“complete net behaviour”*



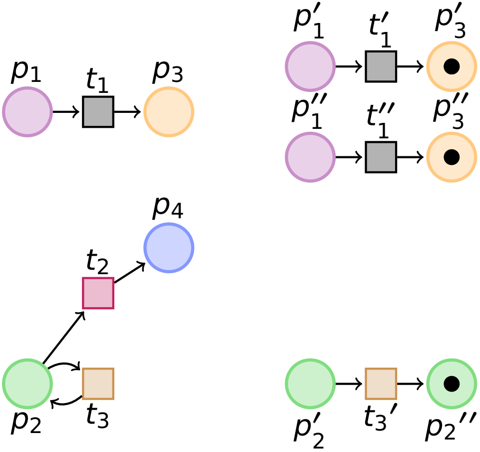
Unfolding as *"complete net behaviour"*



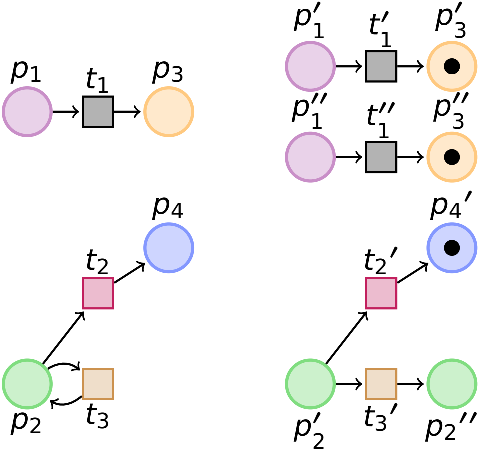
Unfolding as *"complete net behaviour"*



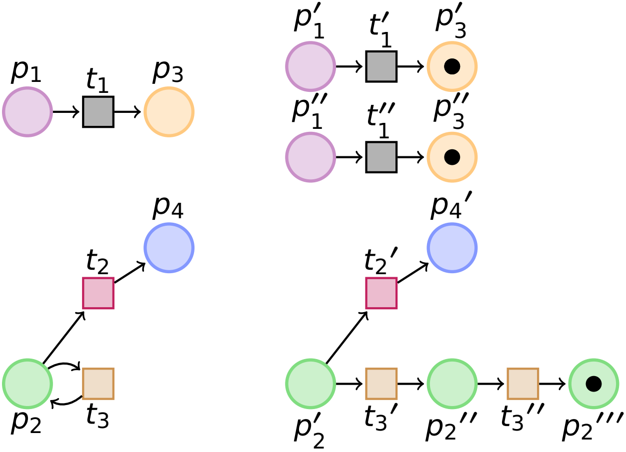
Unfolding as *“complete net behaviour”*



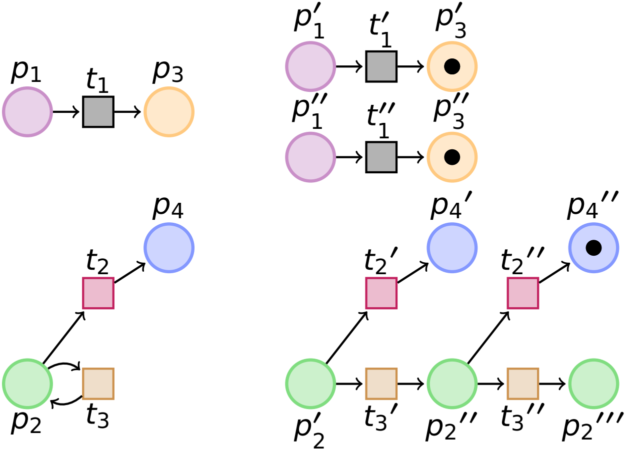
Unfolding as *"complete net behaviour"*



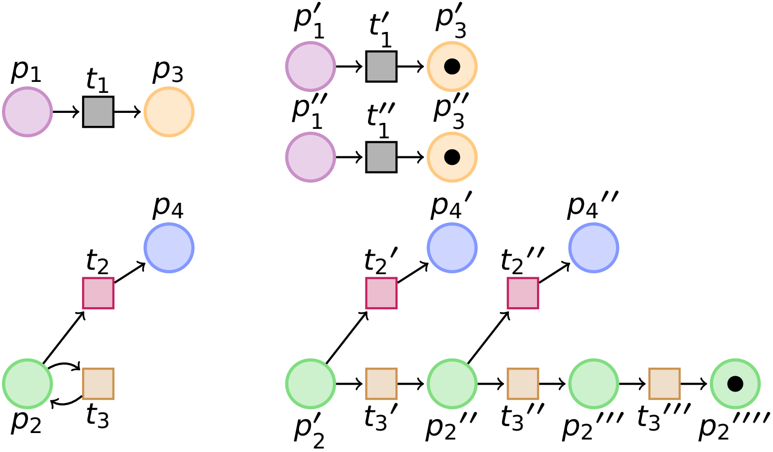
Unfolding as *“complete net behaviour”*



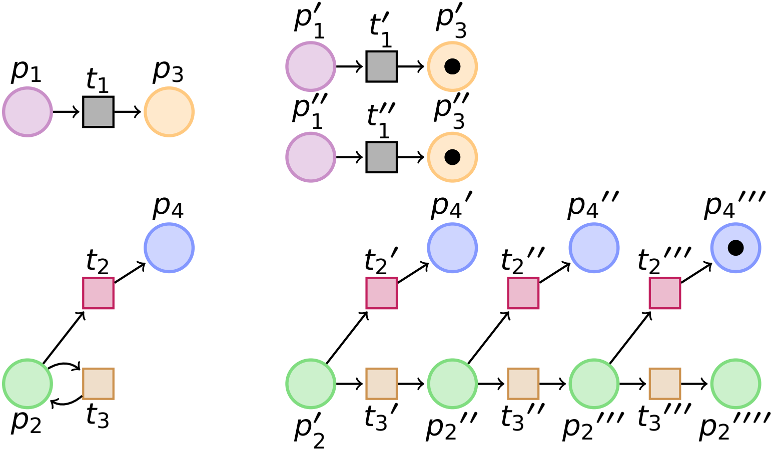
Unfolding as *"complete net behaviour"*



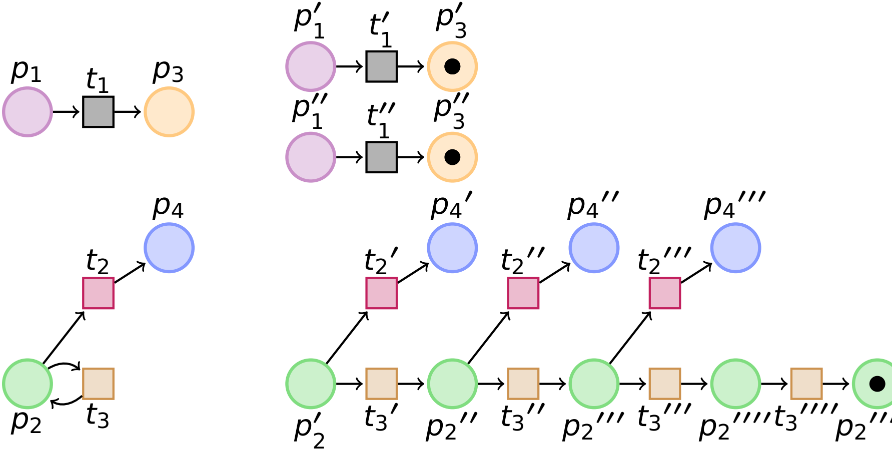
Unfolding as *“complete net behaviour”*



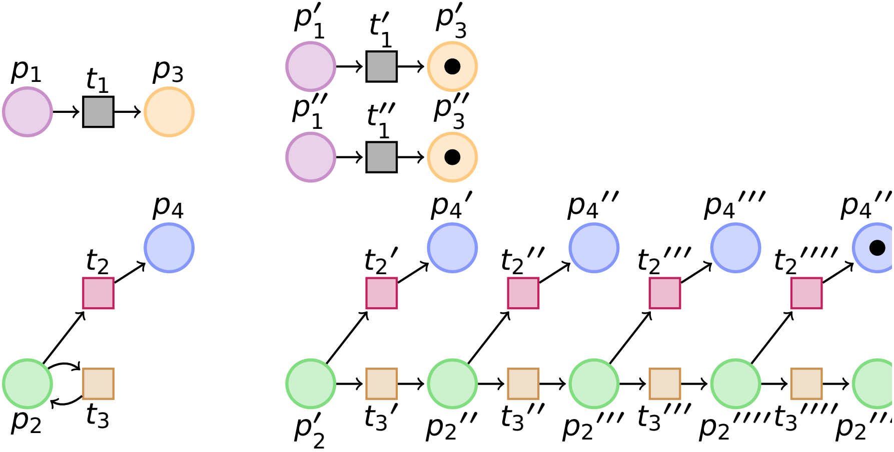
Unfolding as *"complete net behaviour"*



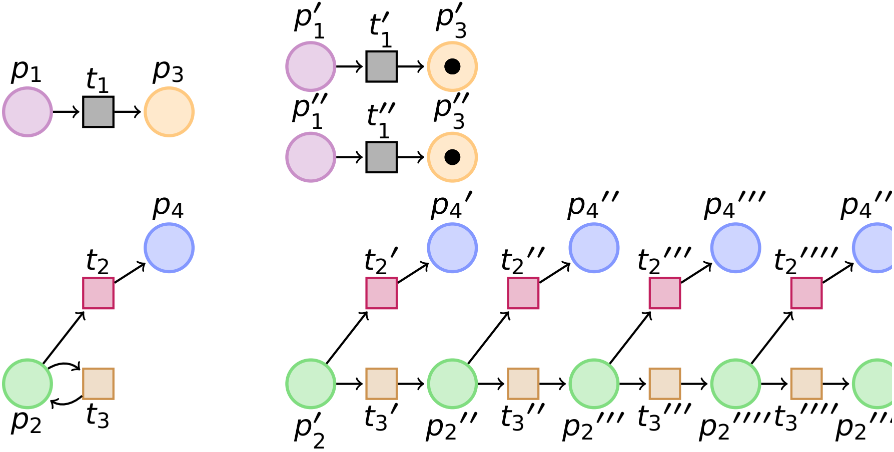
Unfolding as *"complete net behaviour"*



Unfolding as *“complete net behaviour”*



Unfolding as *"complete net behaviour"*



Goal: Unfolding as a coreflector

- N any net

N

Goal: Unfolding as a coreflector

- N any net
- $\mathcal{U}(N)$ the unfolded net

$\mathcal{U}(N)$

N

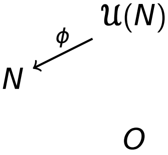
Goal: Unfolding as a coreflector

- N any net
- $\mathcal{U}(N)$ the unfolded net
- $N \xleftarrow{\phi} \mathcal{U}(N)$ a simulation

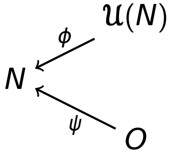
$$N \xleftarrow{\phi} \mathcal{U}(N)$$

Goal: Unfolding as a coreflector

- N any net
- $\mathcal{U}(N)$ the unfolded net
- $N \xleftarrow{\phi} \mathcal{U}(N)$ a simulation
- O a “net behaviour” *(occurrence net)*

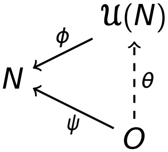


Goal: Unfolding as a coreflector



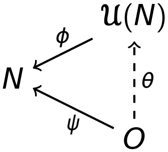
- N any net
- $\mathfrak{U}(N)$ the unfolded net
- $N \xleftarrow{\phi} \mathfrak{U}(N)$ a simulation
- O a "net behaviour" *(occurrence net)*
- $N \xleftarrow{\psi} O$ the "occurrence" in the net N

Goal: Unfolding as a coreflector



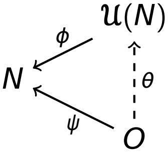
- N any net
- $\mathfrak{U}(N)$ the unfolded net
- $N \xleftarrow{\phi} \mathfrak{U}(N)$ a simulation
- O a "net behaviour" *(occurrence net)*
- $N \xleftarrow{\psi} O$ the "occurrence" in the net N
- $\begin{array}{c}\mathfrak{U}(N)\\ \uparrow\theta \\ O\end{array}$ "occurrence" in the unfolding

Goal: Unfolding as a coreflector



- N any net
- $\mathfrak{U}(N)$ the unfolded net
- $N \xleftarrow{\phi} \mathfrak{U}(N)$ a simulation
- O a "net behaviour" *(occurrence net)*
- $N \xleftarrow{\psi} O$ the "occurrence" in the net N
- $\begin{array}{c} \mathfrak{U}(N) \\ \uparrow\theta \\ O \end{array}$ "occurrence" in the unfolding
- θ should be unique
  . . . or at least essentially unique . . .

Goal: Unfolding as a coreflector

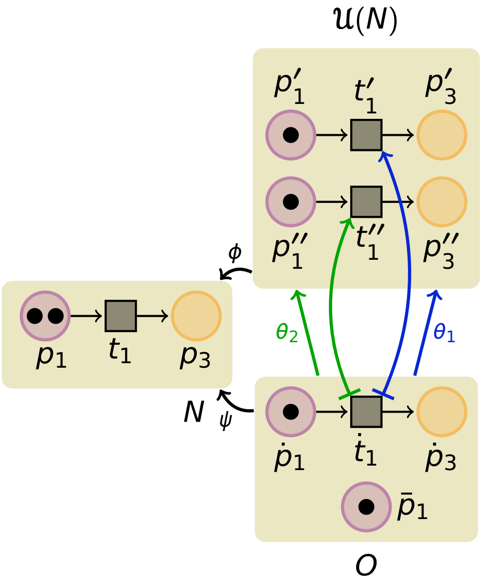


- N any net
- $\mathfrak{U}(N)$ the unfolded net
- $N \xleftarrow{\phi} \mathfrak{U}(N)$ a simulation
- O a “net behaviour” *(occurrence net)*
- $N \xleftarrow{\psi} O$ the “occurrence” in the net N
- $\begin{array}{c}\mathfrak{U}(N)\\ \uparrow\theta\\ O\end{array}$ “occurrence” in the unfolding
- θ should be unique

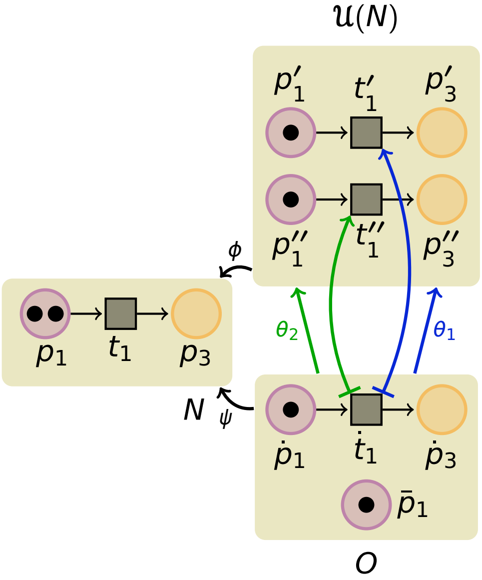
  ...or at least essentially unique ...

Motivation: Coreflection for “distributed” unfolding ($\rightsquigarrow$ $\mathfrak{U}$ preserves limits)

Unfolding – a coreflector or not?

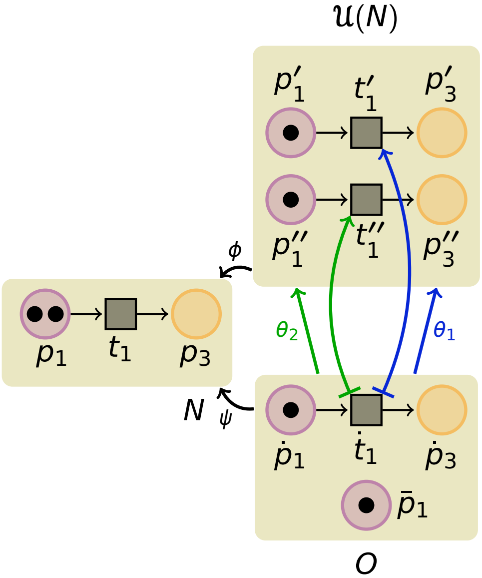


Unfolding – a coreflector or not?



Solutions

Unfolding – a coreflector or not?

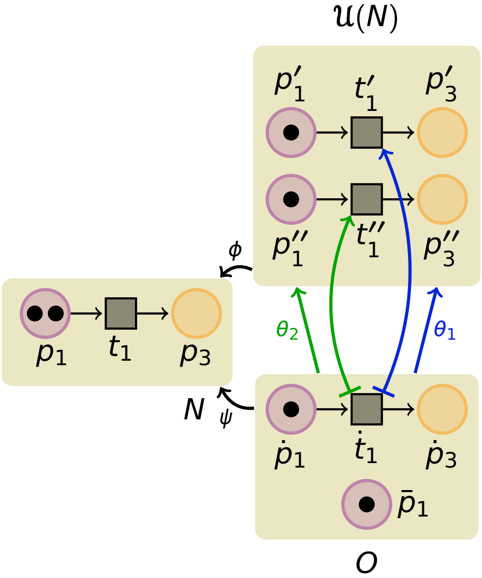


Solutions

- N is bad ⇝ safe / semi-weighted nets

NPW / MMS

Unfolding – a coreflector or not?



Solutions

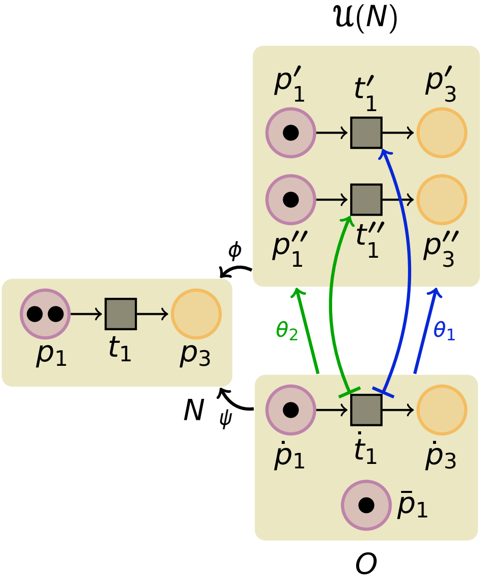
- N is bad ⇝ safe / semi-weighted nets

  NPW / MMS

- ϕ is bad ⇝ "disjoint" morphisms

  MMS

Unfolding – a coreflector or not?



Solutions

- N is bad ⇝ safe / semi-weighted nets

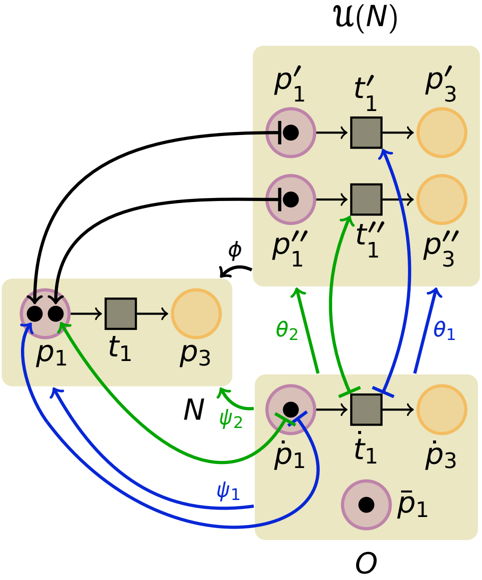
  NPW / MMS

- ϕ is bad ⇝ "disjoint" morphisms

  MMS

- θ_1 "=" θ_2 are (essentially) the same ⇝ bi-coreflection HW

Unfolding – a coreflector or not?



Solutions

- N is bad ⇝ safe / semi-weighted nets NPW / MMS
- ϕ is bad ⇝ "disjoint" morphisms MMS
- θ_1 "=" θ_2 are (essentially) the same ⇝ bi-coreflection HW
- "individual" tokens B & morphisms with "extra information"

↳ application of our work

Overview

Motivation Unfolding as a Coreflector ✓

Overview

Motivation Unfolding as a Coreflector ✓

From Petri Nets to Grammars

- pushout rewriting
- a category of grammars

Overview

Motivation Unfolding as a Coreflector ✓

From Petri Nets to Grammars

- pushout rewriting
- a category of grammars

Unfolding as a Coreflector

- revisiting Petri nets
- theorem

Overview

Motivation Unfolding as a Coreflector ✓

From Petri Nets to Grammars

- pushout rewriting
- a category of grammars

Unfolding as a Coreflector

- revisiting Petri nets
- theorem

Van Kampen colimits . . .

Overview

Motivation Unfolding as a Coreflector ✓

From Petri Nets to Grammars

- pushout rewriting
- a category of grammars

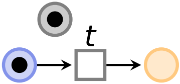
Unfolding as a Coreflector

- revisiting Petri nets
- theorem

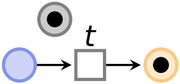
Van Kampen colimits . . .

Conclusion

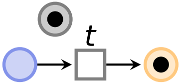
From Petri nets to pushout rewriting



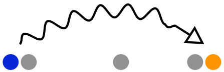
From Petri nets to pushout rewriting



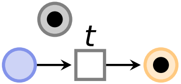
From Petri nets to pushout rewriting



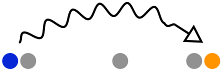
$${}^{\bullet}t \oplus m \;[t\rangle\; m \oplus t^{\bullet}$$



From Petri nets to pushout rewriting



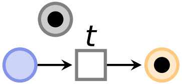
$$^{\bullet}t \oplus m \;\; [t\rangle \;\; m \oplus t^{\bullet}$$



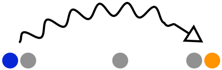
coloured sets $\mathbf{Set}{\downarrow}P$

$$L \uplus X \;\; \supseteq \;\; X \;\; \subseteq \;\; X \uplus R$$

From Petri nets to pushout rewriting



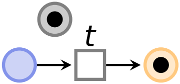
$$^{\bullet}t \oplus m \; [t\rangle \; m \oplus t^{\bullet}$$



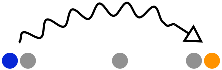
coloured sets $\mathbf{Set}{\downarrow}P$

$$\begin{array}{ccccc} L & & & & R \\ \downarrow & & & & \downarrow \\ L + X & \longleftarrow & X & \longrightarrow & X + R \end{array}$$

From Petri nets to pushout rewriting



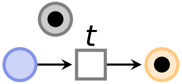
$$ {}^{\bullet}\boldsymbol{t} \oplus m \ [t\rangle \ m \oplus \boldsymbol{t}^{\bullet} $$



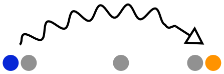
coloured sets $\mathbf{Set}\!\downarrow\! P$

$$
\begin{array}{ccccc}
\boldsymbol{L} & \longleftarrow & 0 & \longrightarrow & \boldsymbol{R} \\
\downarrow & & \downarrow & & \downarrow \\
\boldsymbol{L} + \boldsymbol{X} & \longleftarrow & \boldsymbol{X} & \longrightarrow & \boldsymbol{X} + \boldsymbol{R}
\end{array}
$$

From Petri nets to pushout rewriting



$${}^{\bullet}t \oplus m \;[t\rangle\; m \oplus t^{\bullet}$$



coloured sets $\mathbf{Set}{\downarrow}P$

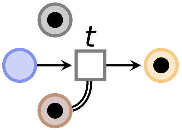
$$\begin{array}{ccccc} L & \longleftarrow & 0 & \longrightarrow & R \\ \downarrow & & \downarrow & & \downarrow \\ L + X & \longleftarrow & X & \longrightarrow & X + R \end{array}$$

rule $q =$ $L \longleftarrow K \longrightarrow R$

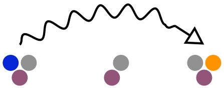
match m

$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ \downarrow & & \downarrow & & \downarrow \\ L +_K X & \longleftarrow & X & \longrightarrow & X +_K R \end{array}$$

From Petri nets to pushout rewriting



$$\underset{\sim}{t} \oplus {}^{\bullet}t \oplus m \;\; [t\rangle \;\; m \oplus t^{\bullet} \oplus \underset{\sim}{t}$$



coloured sets $\mathbf{Set}{\downarrow}P$

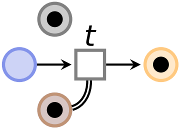
rule $q = L \longleftarrow K \longrightarrow R$

match m

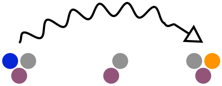
$$L +_K X \longleftarrow X \longrightarrow X +_K R$$

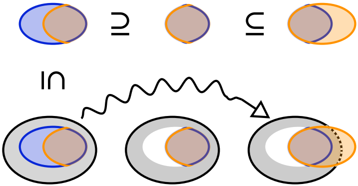
$\Longrightarrow^{\langle q,m\rangle}$ $=\!\langle q,m\rangle\!\Rightarrow$

From Petri nets to pushout rewriting



$\underset{\sim}{t} \oplus {}^{\bullet}t \oplus m \; [t\rangle \; m \oplus t^{\bullet} \oplus \underset{\sim}{t}$





coloured sets $\mathbf{Set}{\downarrow}P$

"topological" objects

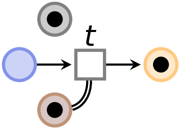
rule $q = L \longleftarrow K \longrightarrow R$

match m

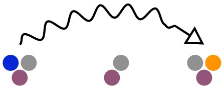
$L +_K X \longleftarrow X \longrightarrow X +_K R$

$=\!\langle q,m\rangle\!\Rightarrow$

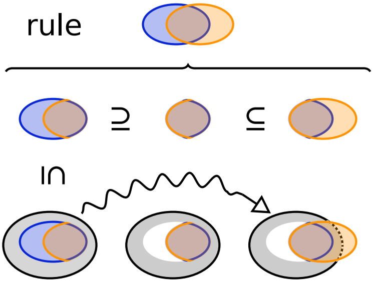
From Petri nets to pushout rewriting



$\underset{\sim}{t} \oplus {}^{\bullet}t \oplus m \;\; [t\rangle \;\; m \oplus t^{\bullet} \oplus \underset{\sim}{t}$



coloured sets $\mathbf{Set}{\downarrow}P$



"topological" objects

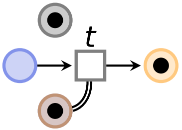
rule $q = L \longleftarrow K \longrightarrow R$

match m

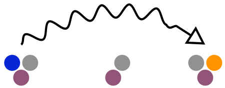
$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ \downarrow & & \downarrow & & \downarrow \\ L +_K X & \longleftarrow & X & \longrightarrow & X +_K R \end{array}$$

$\overset{\langle q,m\rangle}{\Longrightarrow}$

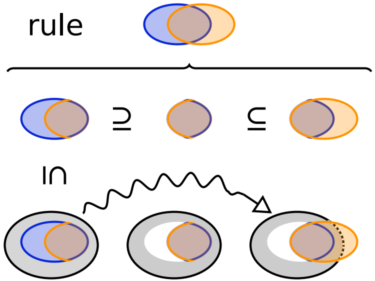
From Petri nets to pushout rewriting



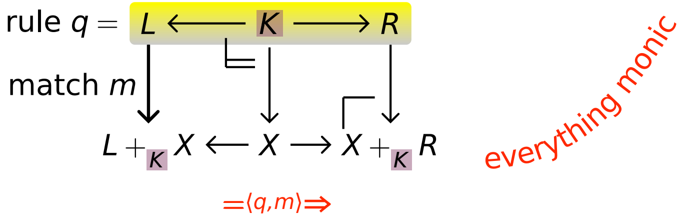
$\underset{\sim}{t} \oplus {}^{\bullet}t \oplus m \; [t\rangle \; m \oplus t^{\bullet} \oplus \underset{\sim}{t}$



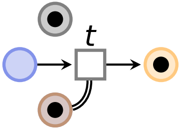
coloured sets $\mathbf{Set}{\downarrow}P$



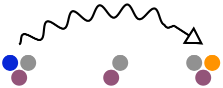
“topological” objects



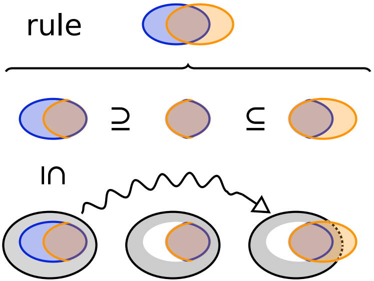
From Petri nets to pushout rewriting



$\underset{\sim}{t} \oplus {}^{\bullet}t \oplus m \ [t\rangle \ m \oplus t^{\bullet} \oplus \underset{\sim}{t}$



coloured sets **Set**$\downarrow P$



"topological" objects

$$\text{rule } q = L \longleftarrow K \longrightarrow R$$

match m

$$L +_K X \longleftarrow X \longrightarrow X +_K R$$

$=\langle q,m\rangle\Rightarrow$

adhesive categories

everything monic

Grammars: a generalization of Petri nets

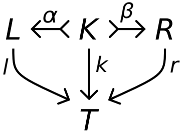
T

Definition (T-typed grammar ($T \in \mathbf{C}$))

Petri net $N = (P, T, {}^\bullet(\_), (\_)^\bullet)$
with
initial marking $m \in P^\oplus$

$P \triangleq T$

Grammars: a generalization of Petri nets



$\cdots$

Definition (T-typed grammar ($T \in \mathbf{C}$))

A *T-typed grammar* is a pair $G = (Q, s)$ with

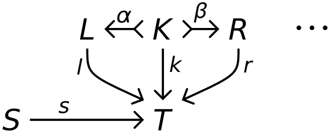
1. Q a set of rules $l \leftarrow\!\alpha\!\prec k \succ\!\beta\!\rightarrow r$ in $\mathbf{C}{\downarrow}T$,

Petri net $N = (P, T, {}^\bullet(\_), (\_)^\bullet)$
with
initial marking $m \in P^\oplus$

$P \triangleq T$

$\{({}^\bullet t, t^\bullet) \mid t \in T\} \triangleq Q$

Grammars: a generalization of Petri nets



Definition (T-typed grammar ($T \in \mathbf{C}$))

A *T-typed grammar* is a pair $G = (Q, s)$ with

1. Q a set of rules $l \leftarrow\alpha\prec k \succ\beta\rightarrow r$ in $\mathbf{C}{\downarrow}T$, and
2. $s\colon S \to T$ a morphism, i.e. $s \in \mathbf{C}{\downarrow}T$.

Petri net $N = (P, T, {}^\bullet(\_), (\_)^\bullet)$
with
initial marking $m \in P^\oplus$

$$P \triangleq T$$
$$\{({}^\bullet t, t^\bullet) \mid t \in T\} \triangleq Q$$
$$m \triangleq s$$

Unfolding à la Winskel

$$\mathbf{SafeNets} \underset{\supseteq}{\overset{\mathcal{U}}{\rightleftarrows}} \mathbf{OccNets} \qquad (\top)$$

Petri net morphisms are $\begin{cases} \text{multirelations } (\approx \textbf{Set}\text{-spans}). \\ \text{monoid homomorphisms.} \end{cases}$

Unfolding à la Winskel

$$\mathbf{SafeNets} \underset{\supseteq}{\overset{\mathcal{U}}{\rightleftarrows}} \mathbf{OccNets} \quad (\top)$$

Petri net morphisms are
- multirelations (≈ **Set**-spans).
- monoid homomorphisms.

Grammar morphisms are
- **C**-span based,
- (monoidal) functors.

Unfolding à la Winskel

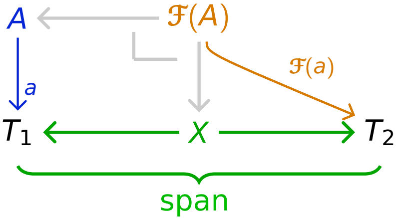
$$\mathbf{SafeNets} \underset{\supseteq}{\overset{\mathfrak{U}}{\rightleftarrows}} \mathbf{OccNets} \quad (\top)$$

Petri net morphisms are $\begin{cases} \text{multirelations } (\approx \textbf{Set}\text{-spans}). \\ \text{monoid homomorphisms.} \end{cases}$

Grammar morphisms are $\begin{cases} \textbf{C}\text{-span based,} \\ \quad \text{(monoidal) functors.} \end{cases}$

$\mathfrak{F}\colon \mathbf{C}{\downarrow}T_1 \to \mathbf{C}{\downarrow}T_2$

retyping functor



Unfolding à la Winskel

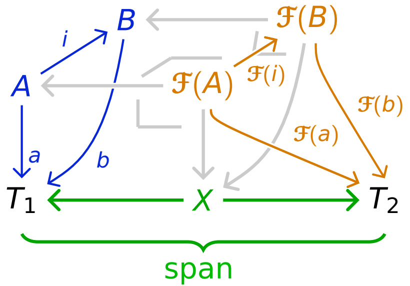
$$\mathbf{SafeNets} \underset{\supseteq}{\overset{\mathfrak{U}}{\rightleftarrows}} \mathbf{OccNets} \quad (\top)$$

Petri net morphisms are $\begin{cases} \text{multirelations } (\approx \textbf{Set}\text{-spans}). \\ \text{monoid homomorphisms.} \end{cases}$

Grammar morphisms are $\begin{cases} \textbf{C}\text{-span based,} \\ \quad \text{(monoidal) functors.} \end{cases}$

$\mathfrak{F} \colon \mathbf{C}{\downarrow}T_1 \to \mathbf{C}{\downarrow}T_2$

retyping functor



Category of grammars **Grm**

objects grammars $\langle Q, s\colon S \to T\rangle$ with $T \in \mathbf{C}$

objects grammars $\langle Q, s\colon S \to T\rangle$ with $T \in \mathbf{C}$

morphisms from $\langle Q_1, s_1\colon S_1 \to T_1\rangle$ to $\langle Q_2, s_2\colon S_2 \to T_2\rangle$:

retyping functors $\mathfrak{F}\colon \mathbf{C}{\downarrow}T_1 \to \mathbf{C}{\downarrow}T_2$

$$T_1 \longleftarrow X \longrightarrow T_2$$

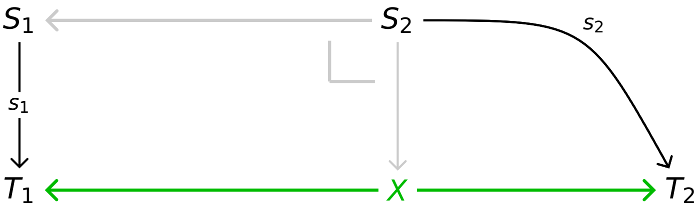
Category of grammars **Grm**

objects: grammars $\langle Q, s\colon S \to T\rangle$ with $T \in \mathbf{C}$

morphisms: from $\langle Q_1, s_1\colon S_1 \to T_1\rangle$ to $\langle Q_2, s_2\colon S_2 \to T_2\rangle$:

retyping functors $\mathfrak{F}\colon \mathbf{C}{\downarrow}T_1 \to \mathbf{C}{\downarrow}T_2$ such that

$$\mathfrak{F}(s_1) = s_2$$



Category of grammars **Grm**

objects grammars $\langle Q, s\colon S \to T\rangle$ with $T \in \mathbf{C}$

morphisms from $\langle Q_1, s_1\colon S_1 \to T_1\rangle$ to $\langle Q_2, s_2\colon S_2 \to T_2\rangle$:

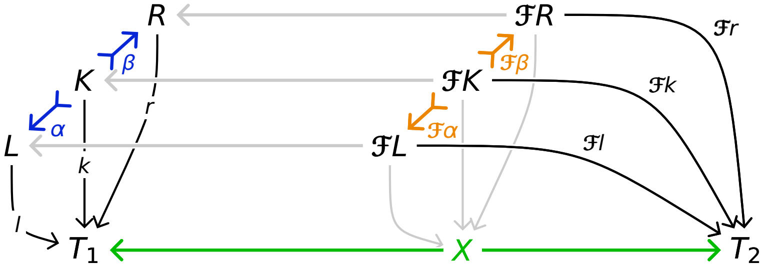
retyping functors $\mathfrak{F}\colon \mathbf{C}{\downarrow}T_1 \to \mathbf{C}{\downarrow}T_2$ such that

$$\mathfrak{F}(s_1) = s_2$$

and

$$\mathfrak{F}(q) \in Q_2$$

for all $q = (l \xleftarrow{\alpha} k \xrightarrow{\beta} r) \in Q_1$.



Category of grammars **Grm**

objects grammars $\langle Q, s\colon S \to T\rangle$ with $T \in \mathbf{C}$

morphisms from $\langle Q_1, s_1\colon S_1 \to T_1\rangle$ to $\langle Q_2, s_2\colon S_2 \to T_2\rangle$:

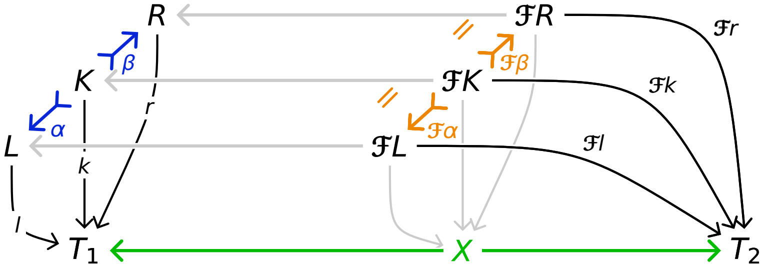
retyping functors $\mathfrak{F}\colon \mathbf{C}{\downarrow}T_1 \to \mathbf{C}{\downarrow}T_2$ such that

$$\mathfrak{F}(s_1) = s_2$$

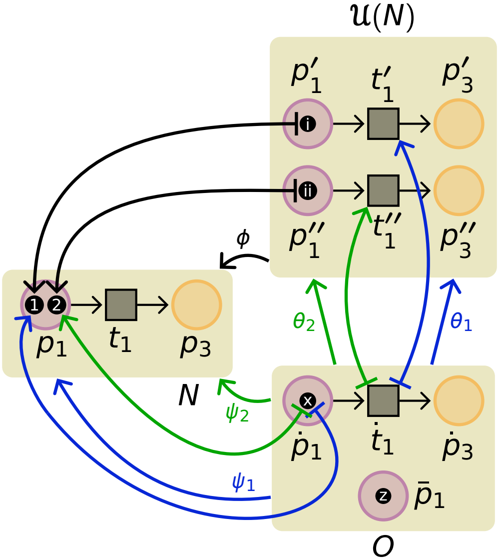
and

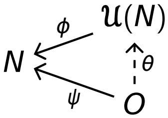
$$\mathfrak{F}(q) \in Q_2 \cup \{\mathfrak{F}(k) \leftarrow \mathfrak{F}(k) \rightarrow \mathfrak{F}(k)\}$$

for all $q = (l \leftarrow\alpha\prec k \succ\beta\rightarrow r) \in Q_1$.

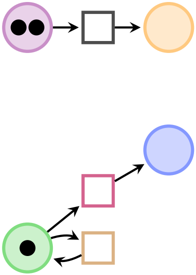


Coreflection in the example revisited

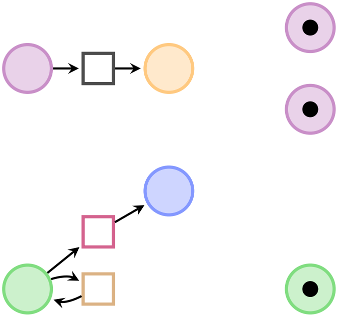




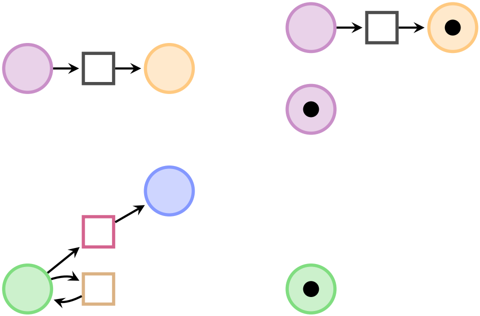
Unfolding . . .



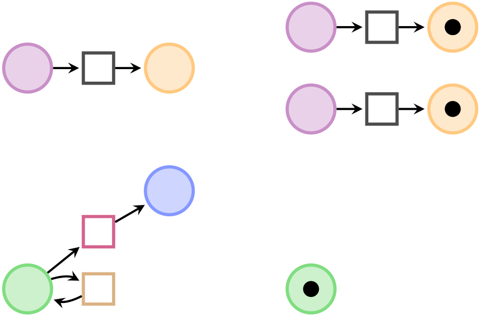
Unfolding . . .



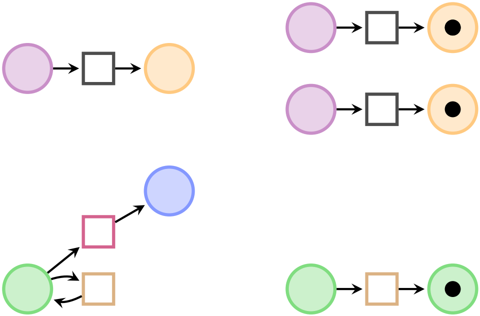
Unfolding . . .



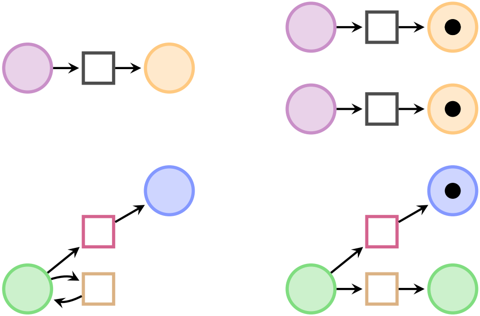
Unfolding . . .



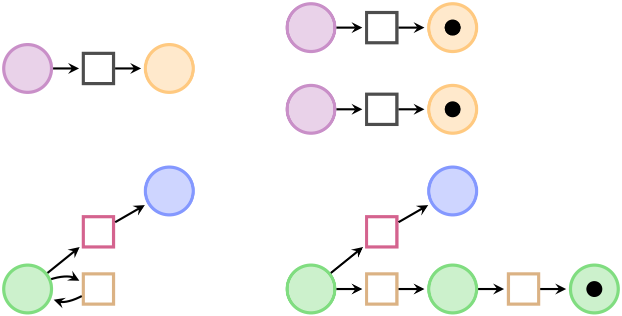
Unfolding . . .



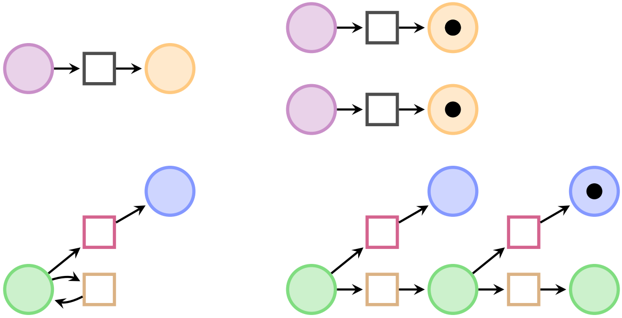
Unfolding . . .



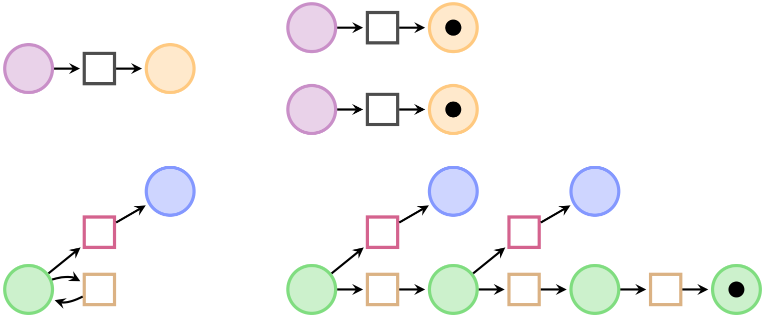
Unfolding . . .



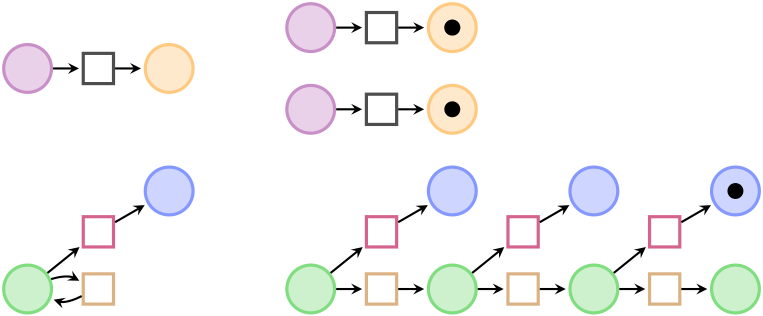
Unfolding . . .



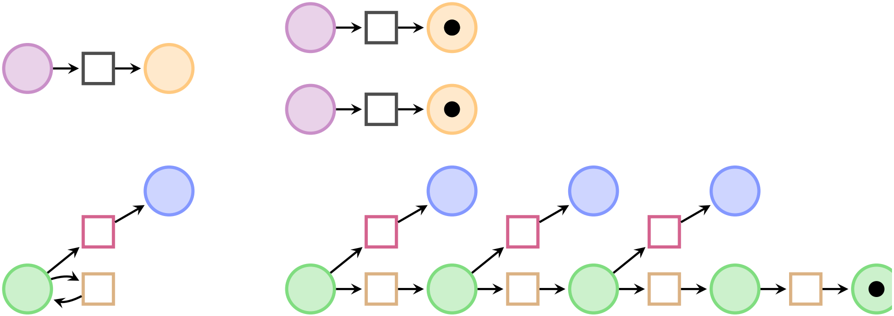
Unfolding . . .



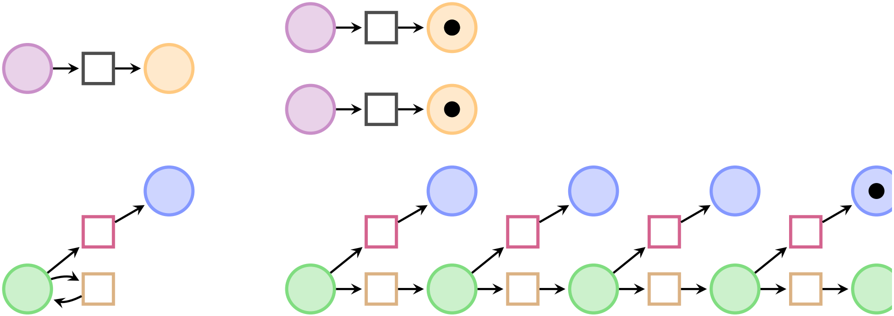
Unfolding . . .



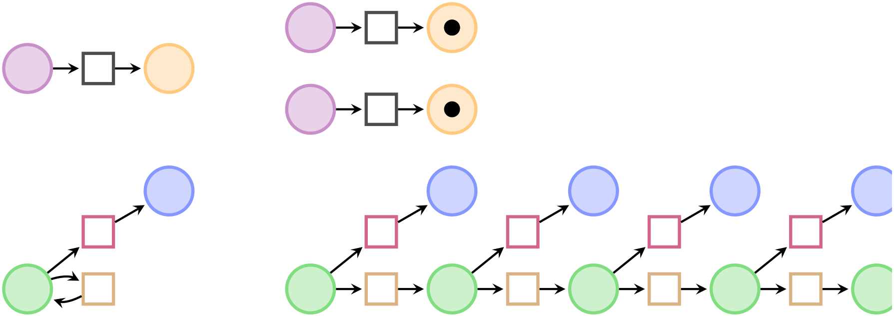
Unfolding . . .



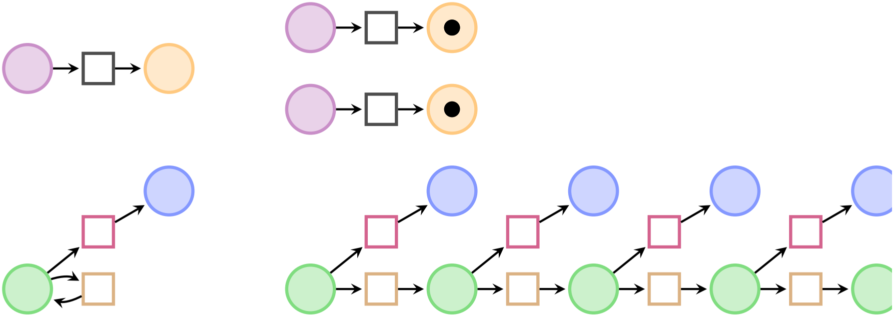
Unfolding . . .



Unfolding . . .



Unfolding . . .



transition occurrences ⇝ rule occurrences

+ +

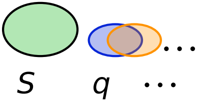
initial marking ⇝ start object

union + + ω-chain colimit

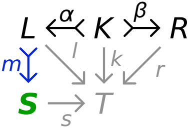
post-sets ⇝ right hand sides

Unfolding: “first” rule occurrences

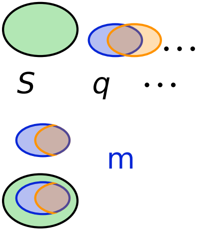
grammar $\langle Q, s \colon S \to T \rangle$,
rule $q = (l \xleftarrow{\alpha} k \xrightarrow{\beta} r) \in Q$



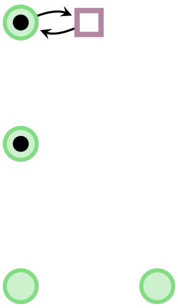
grammar $\langle Q, s\colon S \to T\rangle$,
rule $q = (l \leftarrow \alpha \prec k \succ \beta \rightarrow r) \in Q$



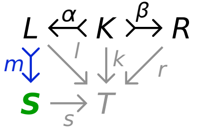
C$\downarrow T$ match m



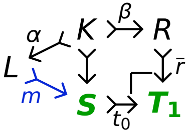
Unfolding: “first” rule occurrences



grammar $\langle Q, s\colon S \to T\rangle$,
rule $q = (l \leftarrow\alpha\prec k \succ\beta\rightarrow r) \in Q$

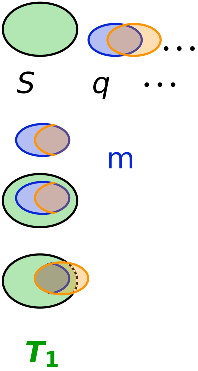


$\mathbf{C}{\downarrow}T$ match m



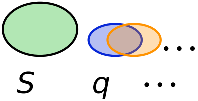
new
type object

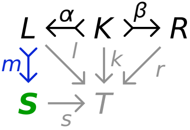
$T_1 = S +_K R$



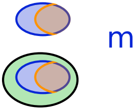
Unfolding: “first” rule occurrences

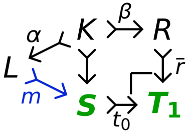
grammar $\langle Q, s\colon S \to T\rangle$,
rule $q = (l \xleftarrow{\alpha} k \xrightarrow{\beta} r) \in Q$



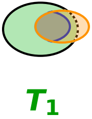


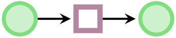
C$\downarrow T$ match m

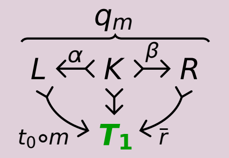




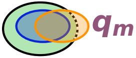
$T_1 =$ new type object $S +_K R$



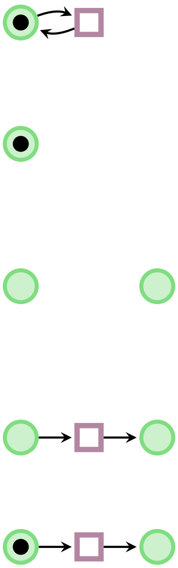




rule occurrence



Unfolding: “first” rule occurrences



grammar $\langle Q, s\colon S \to T\rangle$,
rule $q = (l \leftarrow\alpha\prec k \succ\beta\rightarrow r) \in Q$

$$\begin{array}{ccccc} L & \xleftarrow{\alpha} & K & \xrightarrow{\beta} & R \\ {\scriptstyle m}\downarrow & \searrow^{l} & \downarrow^{k} & \swarrow^{r} & \\ S & \xrightarrow{s} & T & & \end{array}$$

$\mathbf{C}{\downarrow}T$ match m

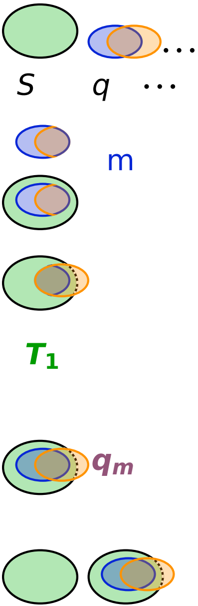
$$\begin{array}{ccccc} & & K & \xrightarrow{\beta} & R \\ L & \xleftarrow{\alpha} & \downarrow & & \downarrow \bar{r} \\ & \xrightarrow{m} & S & \xrightarrow{t_0} & T_1 \end{array} = S +_K R$$

new type object

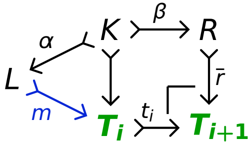
$$q_m = \overbrace{L \xleftarrow{\alpha} K \xrightarrow{\beta} R}, \quad t_0 \circ m\colon L \to T_1,\ K \to T_1,\ \bar{r}\colon R \to T_1$$

rule occurrence

$U_1 = \langle \{q_m\}, t_0\colon S \rightarrowtail T_1\rangle$

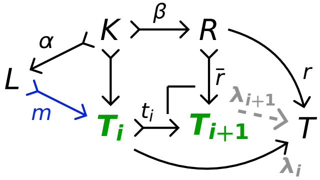


Unfolding: general unfolding step



$$U_i = \langle \{\ldots, \boldsymbol{q^i}\}, s_i \colon \boldsymbol{S} \rightarrowtail \boldsymbol{T_i} \rangle$$
$$\rightsquigarrow U_{i+1}$$

Unfolding: general unfolding step

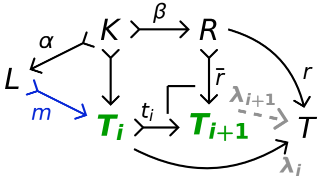


folding morphism $\boldsymbol{\lambda_i} \colon \boldsymbol{T_i} \to T$

$U_i = \langle \{ \ldots, \boldsymbol{q^i} \}, s_i \colon \boldsymbol{S} \rightarrowtail \boldsymbol{T_i} \rangle$

$\rightsquigarrow U_{i+1}$

Unfolding: general unfolding step



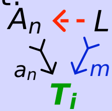
folding morphism $\lambda_i : T_i \to T$

$U_i = \langle \{ \dots, q^i \}, s_i : S \rightarrowtail T_i \rangle$

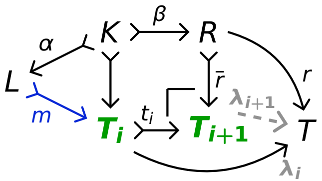
$\rightsquigarrow U_{i+1}$

match m: "coverable"/concurrent subobject!

$$s_i \overset{q_1^i}{\Longrightarrow} a_1 \cdots \overset{q_n^i}{\Longrightarrow} a_n$$



Unfolding: general unfolding step



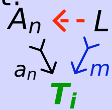
folding morphism $\lambda_i : T_i \to T$

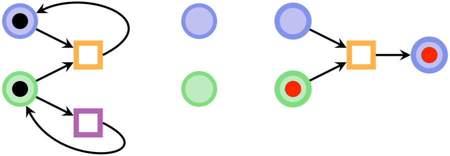
$U_i = \langle \{ \ldots, q^i \}, s_i : S \rightarrowtail T_i \rangle$

$\rightsquigarrow U_{i+1}$

match m: “coverable”/concurrent subobject!

$$s_i = q_1^i \Rightarrow a_1 \cdots = q_n^i \Rightarrow a_n$$





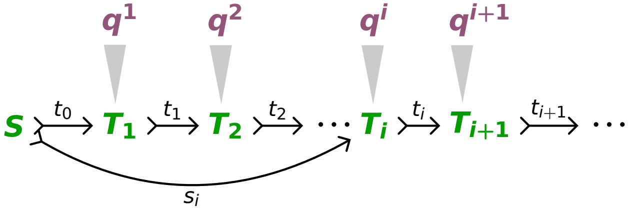
not coverable

⇒ no new occurrence

of □

Fair sequence of unfoldings

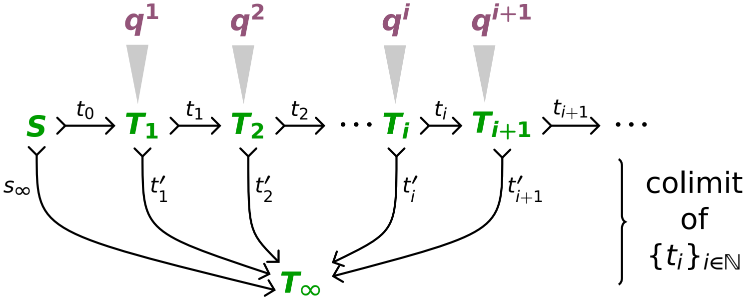
$$U_i = \left\langle \{\ldots, \boldsymbol{q^i}\},\ s_i \colon \boldsymbol{S} \rightarrowtail \boldsymbol{T_i} \right\rangle$$



The final construction step

Fair sequence of unfoldings

$$U_i = \left\langle \{\ldots, \boldsymbol{q^i}\},\ s_i : \boldsymbol{S} \rightarrowtail \boldsymbol{T_i} \right\rangle$$

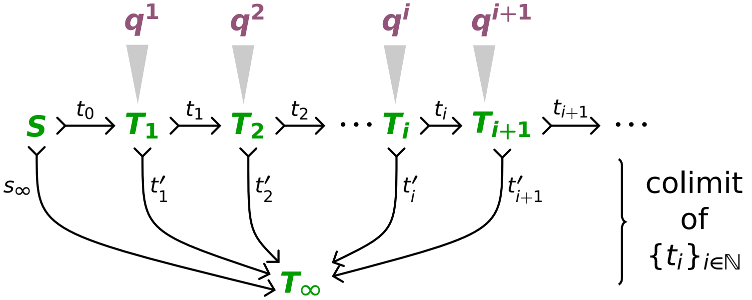


$$U = \left\langle \left\{ t_i' \circ \boldsymbol{q^i} \mid i \in \mathbb{N} \right\},\ s_\infty : \boldsymbol{S} \rightarrow \boldsymbol{T_\infty} \right\rangle$$

The final construction step

Fair sequence of unfoldings

$$U_i = \left\langle \{\ldots, q^i\},\ s_i : S \rightarrowtail T_i \right\rangle$$



$$U = \left\langle \left\{ t'_i \circ q^i \mid i \in \mathbb{N} \right\},\ s_\infty : S \to T_\infty \right\rangle$$

. . . folding morphism $\lambda : T_\infty \to T$

The coreflection result

Unfolding produces

"safe" grammars with acyclic causality.

$\rightsquigarrow$ *occurrence grammars*

$$\mathbf{Grm} \xrightarrow{\mathcal{U}} \mathbf{OccGrm}$$

The coreflection result

Unfolding produces

"safe" grammars with acyclic causality.

$\rightsquigarrow$ *occurrence grammars*

$$\mathbf{Grm} \underset{\supseteq}{\overset{\mathcal{U}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathbf{OccGrm}$$

Theorem

Unfolding of grammars is a coreflector.

The coreflection result

Unfolding produces

"safe" grammars with acyclic causality.

$\rightsquigarrow$ *occurrence grammars*

$$\mathbf{Grm} \underset{\supseteq}{\overset{\mathfrak{U}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathbf{OccGrm}$$

Theorem

Unfolding of grammars is a coreflector.

$\rightsquigarrow$ static notion of coverability

$\rightsquigarrow$ system composition:

$$\mathfrak{U}(G \times H) = \mathfrak{U}(G) \times \mathfrak{U}(H)$$

Conclusion and discussion

Conclusion

- *alternative approach to obtain coreflective unfolding semantics for PT nets*
- *grammar morphisms as span-based functors for unfolding as a coreflector*

(e.g. in any topos with countable sums)

$\rightsquigarrow$ Heindel (PhD) ✓

Conclusion and discussion

Conclusion

- *alternative approach to obtain coreflective unfolding semantics for PT nets*
- *grammar morphisms as span-based functors for unfolding as a coreflector*

  (e.g. in any topos with countable sums)

$\rightsquigarrow$ Heindel (PhD) ✓

Thanks !

Discussion

. . .