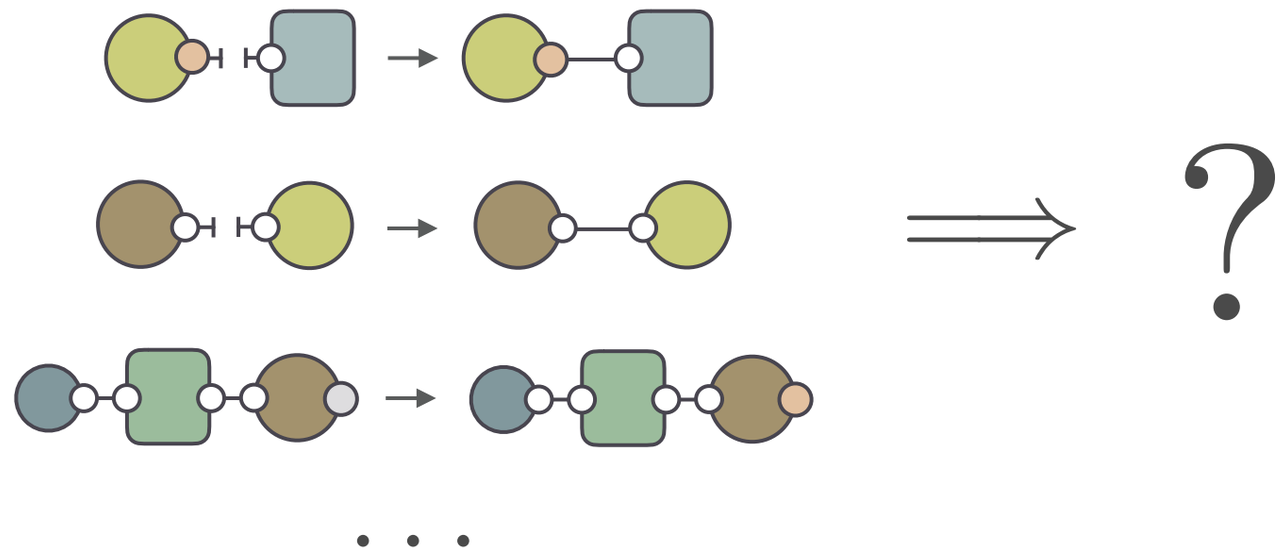
UNCOVERING PATHWAYS

in biomolecular interaction networks



Jonathan Laurent

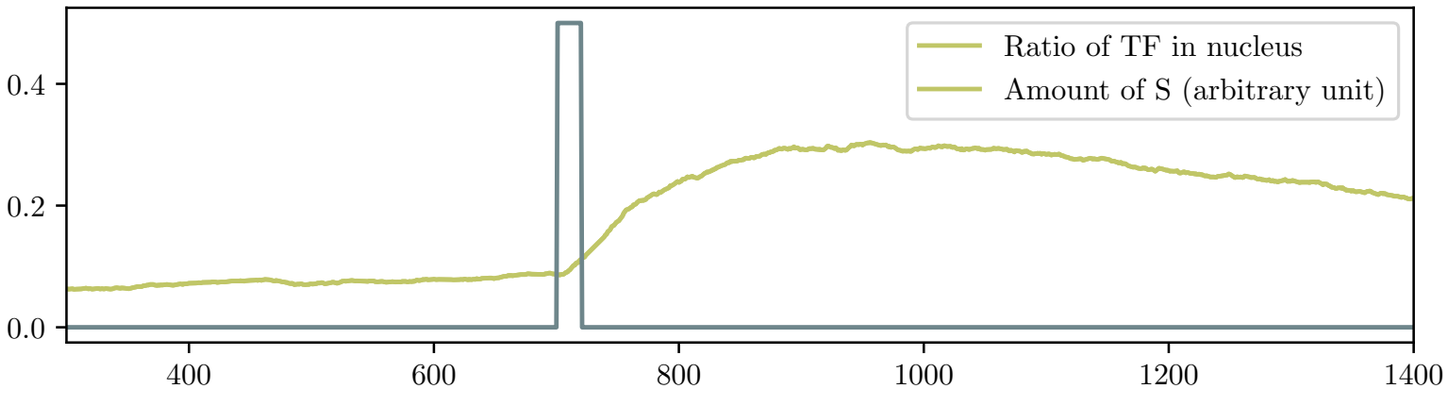
Joint work with Jean Yang, Walter Fontana and Jérôme Feret

How does a cell live ?

A SIMPLE USE CASE

Observation

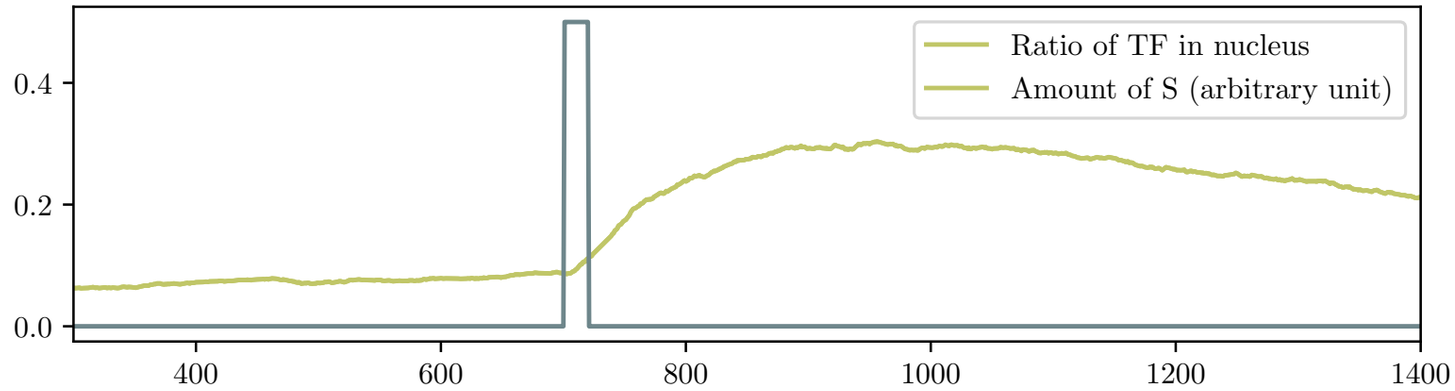
Injecting some amount of a signal molecule results in a temporary increase of the concentration of some transcription factor in the nucleus.

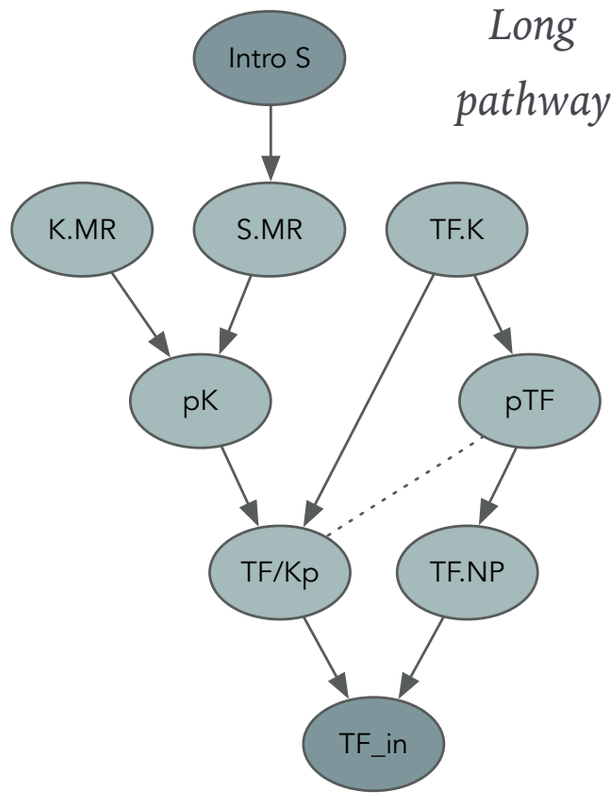


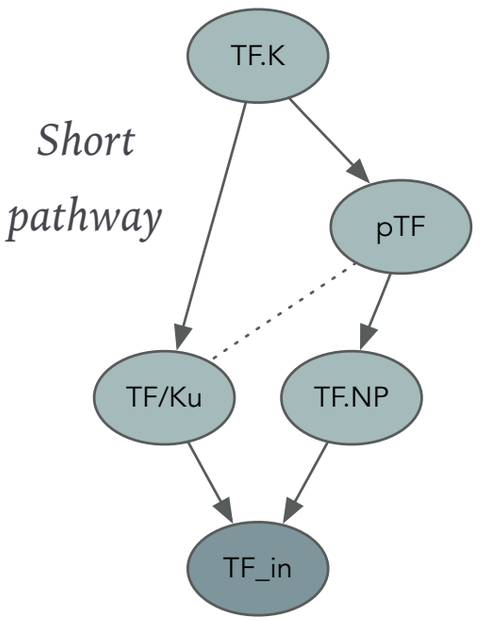
A SIMPLE USE CASE

Observation

Injecting some amount of a signal molecule results in a temporary increase of the concentration of some transcription factor in the nucleus.



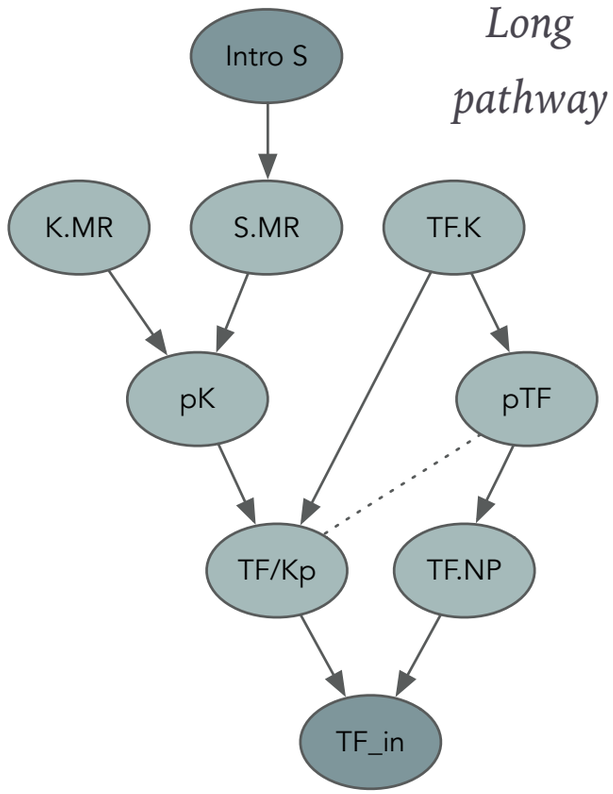


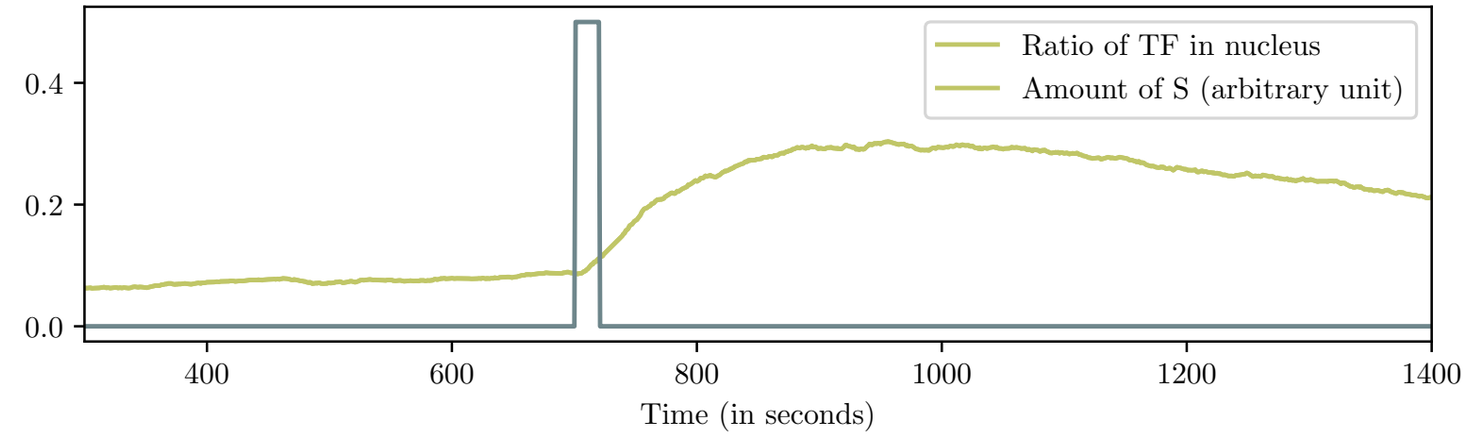


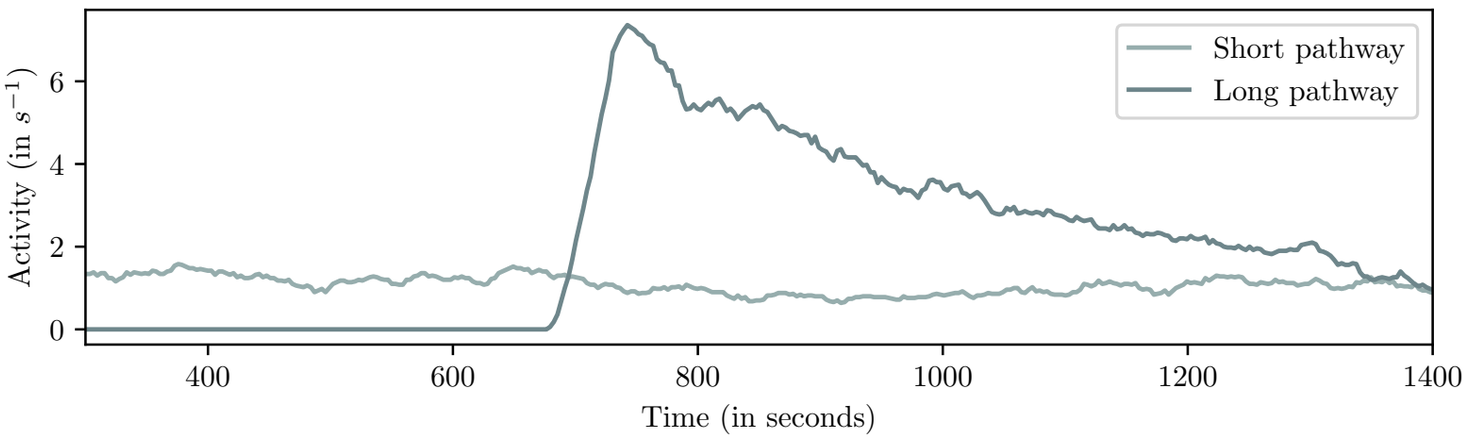
A SIMPLE USE CASE

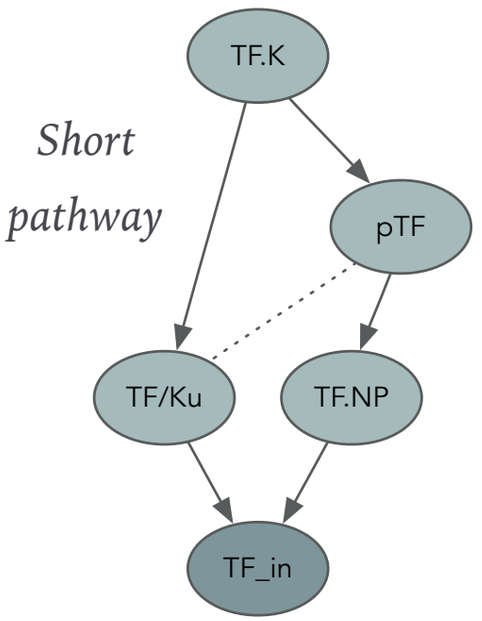
Observation

Injecting some amount of a signal molecule results in a temporary increase of the concentration of some transcription factor in the nucleus.



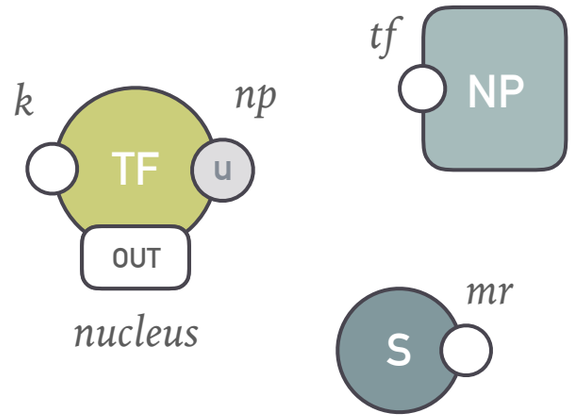




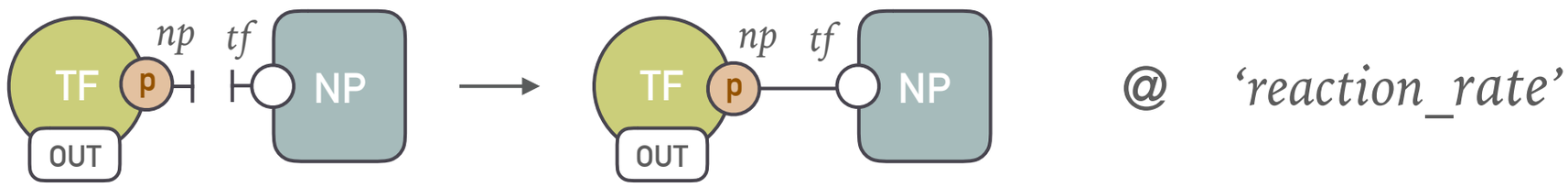


THE KAPPA LANGUAGE

Proteins are modeled by **agents** with binding **sites.** Besides, sites can carry an **internal state.**

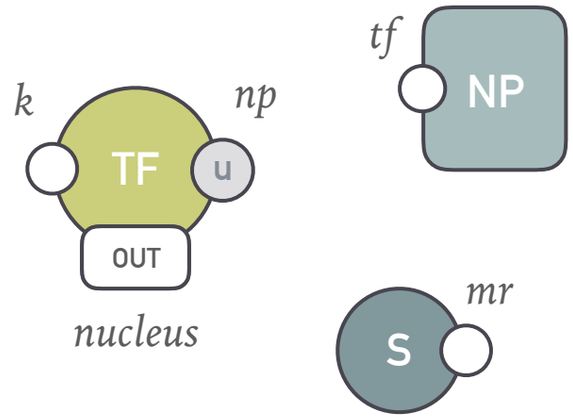


A **rule** is given by a graph-rewriting operation along with a reaction rate:

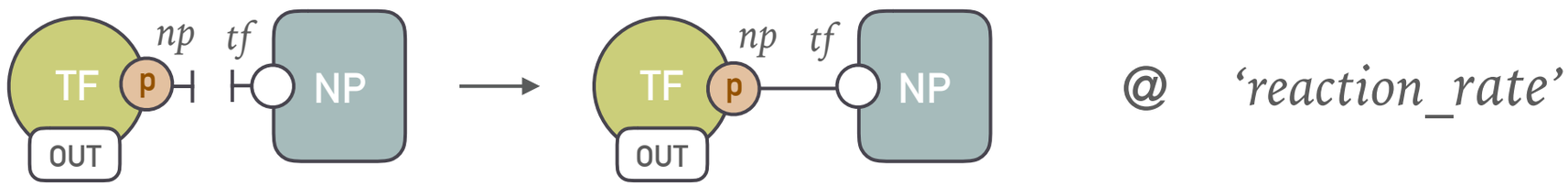


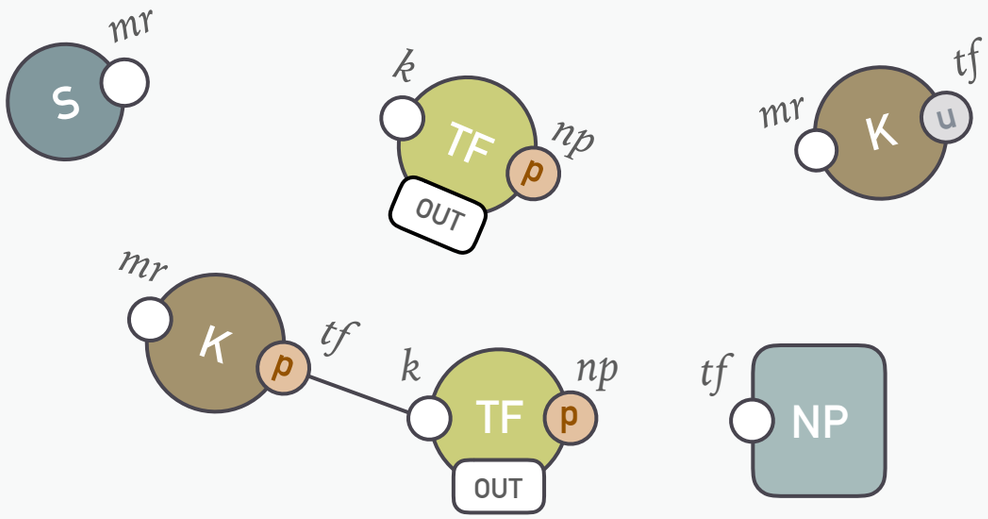
THE KAPPA LANGUAGE

Proteins are modeled by **agents** with binding **sites.** Besides, sites can carry an **internal state.**



A **rule** is given by a graph-rewriting operation along with a reaction rate:

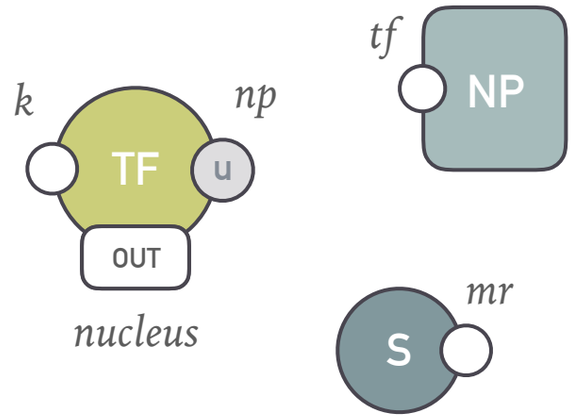




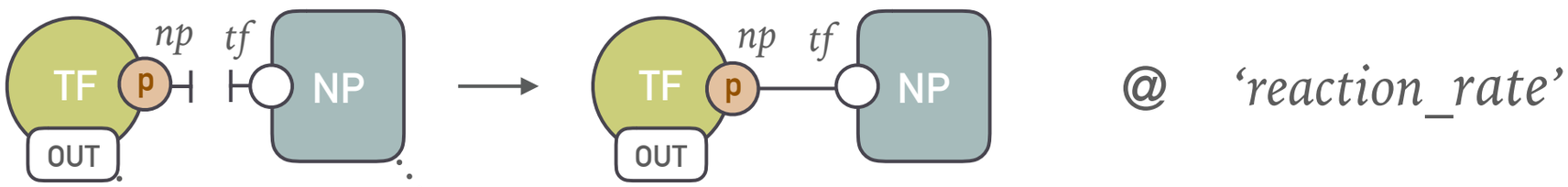
An **event** is given by a rule along with an injection of its LHS in the **state mixture**.

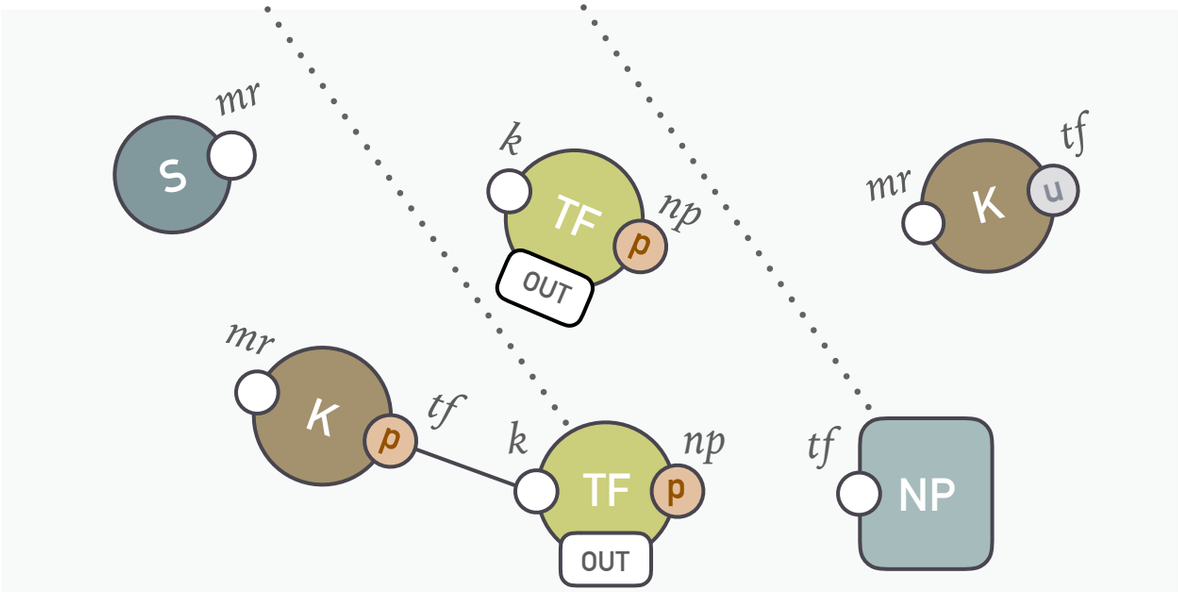
THE KAPPA LANGUAGE

Proteins are modeled by **agents** with binding **sites.** Besides, sites can carry an **internal state.**



A **rule** is given by a graph-rewriting operation along with a reaction rate:

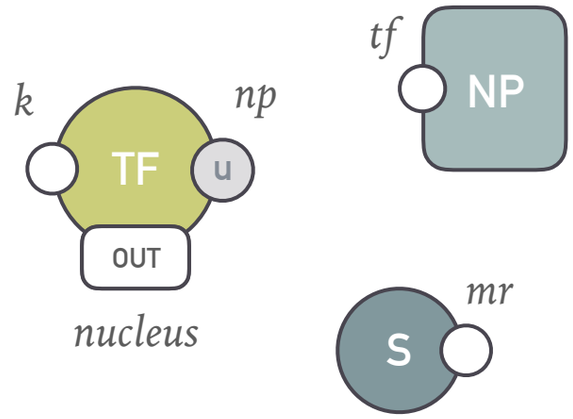




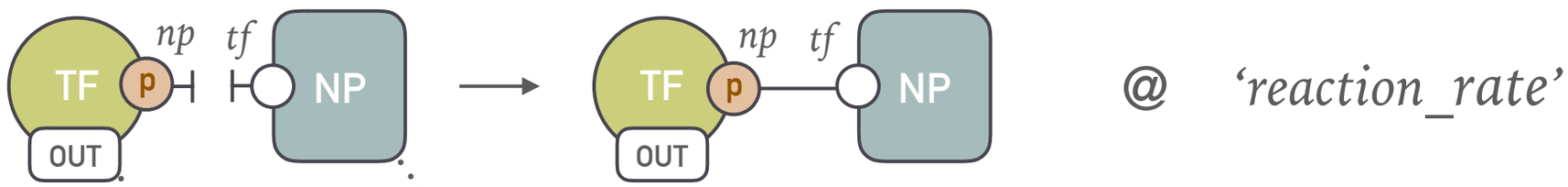
An **event** is given by a rule along with an injection of its LHS in the **state mixture**.

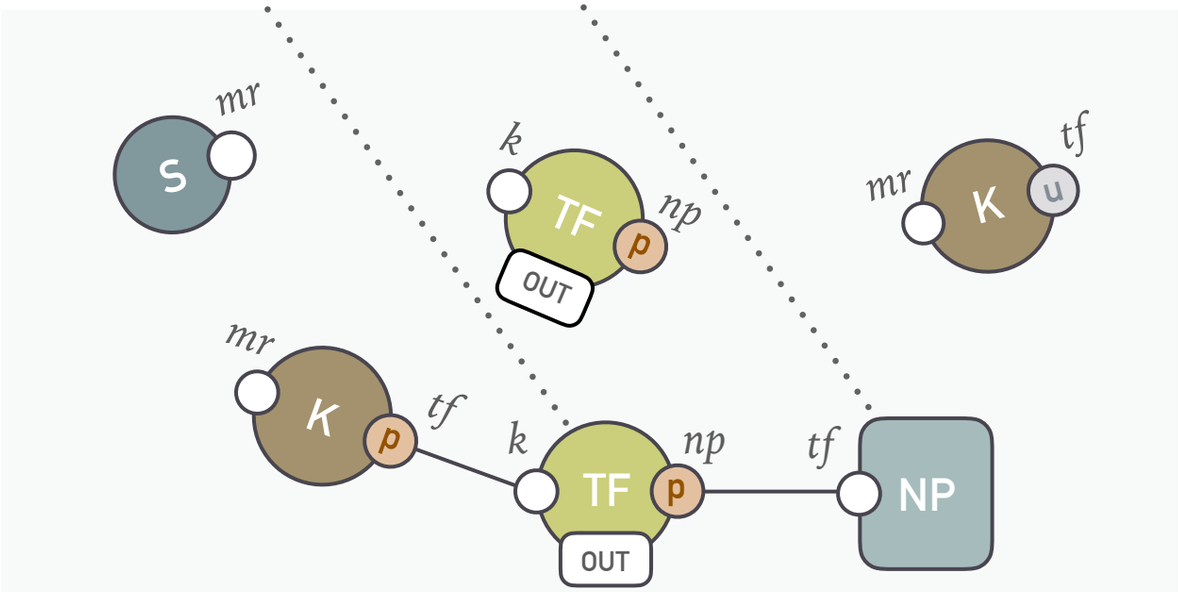
THE KAPPA LANGUAGE

Proteins are modeled by **agents** with binding **sites.** Besides, sites can carry an **internal state.**



A **rule** is given by a graph-rewriting operation along with a reaction rate:

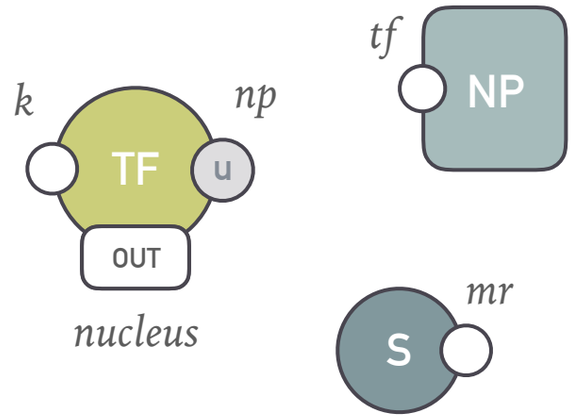




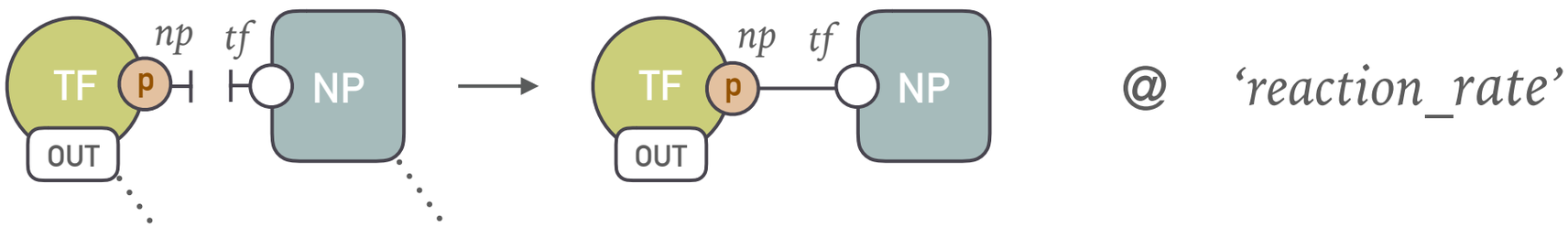
An **event** is given by a rule along with an injection of its LHS in the **state mixture**.

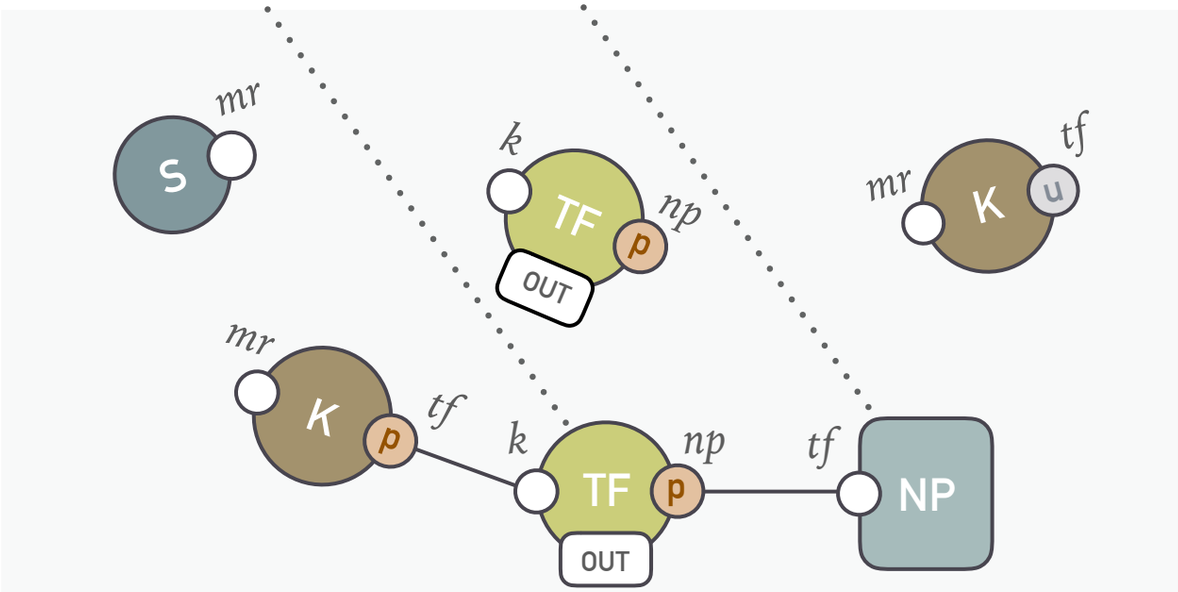
THE KAPPA LANGUAGE

Proteins are modeled by **agents** with binding **sites.** Besides, sites can carry an **internal state.**



A **rule** is given by a graph-rewriting operation along with a reaction rate:





An **event** is given by a rule along with an injection of its LHS in the **state mixture**.

A **trajectory** is a sequence of events produced by a stochastic simulation of the model.

PROBLEM AND CONTRIBUTIONS

General problem

How do we extract pathways from networks of biological interactions and monitor their activity ?

PROBLEM AND CONTRIBUTIONS

General problem

How do we extract pathways from networks of biological interactions and monitor their activity ?

Solution outline

- Pathways are just **abstract representations of equivalent trajectories** leading to the triggering of a rule of interest from initial conditions.

- In the context of a specific trajectory, an event instantiating the rule of interest can be **mapped to a unique pathway** through a slicing procedure that isolates a subset of events that is sufficient to replicate it and that is minimal in some sense.

PROBLEM AND CONTRIBUTIONS

General problem

How do we extract pathways from networks of biological interactions and monitor their activity ?

Solution outline

- Pathways are just **abstract representations of equivalent trajectories** leading to the triggering of a rule of interest from initial conditions.

- In the context of a specific trajectory, an event instantiating the rule of interest can be **mapped to a unique pathway** through a slicing procedure that isolates a subset of events that is sufficient to replicate it and that is minimal in some sense.

Contributions

- A characterization of **concurrency** in Kappa along with a notion of causal influence between events
- A principled **slicing procedure** to map events to their producing pathways that offers:
  - **Uniqueness** of the optimal slice
  - **Strong guarantees** regarding the faithfulness of slices to the original trajectory

CONCURRENCY IN KAPPA

Two events are **concurrent in some particular state** if the two of them can be triggered in any order from this state, both sequences leading to the same new state.

Two events are **concurrent** if:

- They are concurrent in every state where they are both triggerable
- They are both triggerable in at least one state

CONCURRENCY IN KAPPA

Two events are **concurrent in some particular state** if the two of them can be triggered in any order from this state, both sequences leading to the same new state.

Two events are **concurrent** if:

- They are concurrent in every state where they are both triggerable
- They are both triggerable in at least one state

Theorem

Two events are concurrent if and only if:

- None of them is modifying a logical site which is tested by the other
- They do not test a same logical site for different values

Besides, two events are concurrent if they are in at least one state.

CONCURRENCY IN KAPPA

Two events are **concurrent in some particular state** if the two of them can be triggered in any order from this state, both sequences leading to the same new state.

Two events are **concurrent** if:

- They are concurrent in every state where they are both triggerable
- They are both triggerable in at least one state

Theorem

Two events are concurrent if and only if:

- None of them is modifying a logical site which is tested by the other
- They do not test a same logical site for different values

Besides, two events are concurrent if they are in at least one state.

Taking concurrency into account

Let's call two trajectories **equivalent** if one can be obtained from the other by successively permuting contiguous concurrent events. In other words, we define $\sim$ as the smallest equivalence relation on trajectories such that:

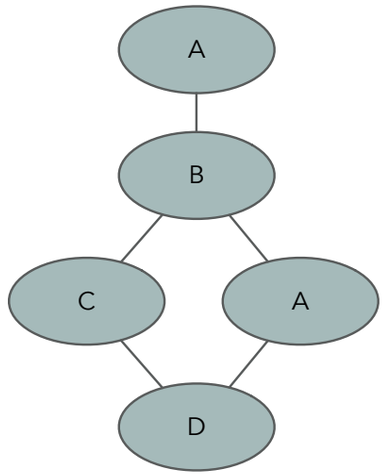
$$e_1 \diamond e_2 \implies (t \cdot e_1 \cdot e_2 \cdot t') \sim (t \cdot e_2 \cdot e_1 \cdot t')$$

An equivalence class of trajectories for this relation is called a (Mazurkiewicz) **trace**.

TRACES AS POMSETS OF EVENTS

Let's consider a trajectory ABCAD where only A and C are concurrent. The corresponding trace contains the two equivalent trajectories ABCAD and ABACD.

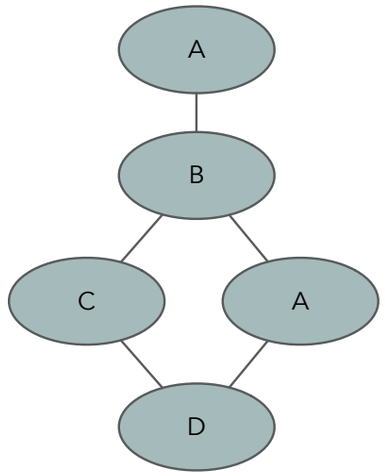
It can be concisely represented by the following **pomset** of events:



TRACES AS POMSETS OF EVENTS

Let's consider a trajectory ABCAD where only A and C are concurrent. The corresponding trace contains the two equivalent trajectories ABCAD and ABACD.

It can be concisely represented by the following **pomset** of events:



A **pomset** of events is given by:

- A set of **event identifiers** I
- A partial order relation $\preceq$ on I which we call **precedence**
- A **labelling function** that maps event identifiers to events

It is possible to abstract a trajectory into a pomset of events as follows:

$$e_1, \cdots, e_n \quad \overset{\alpha}{\longmapsto} \quad (\{1, \cdots, n\}, \preceq, j \mapsto e_j)$$

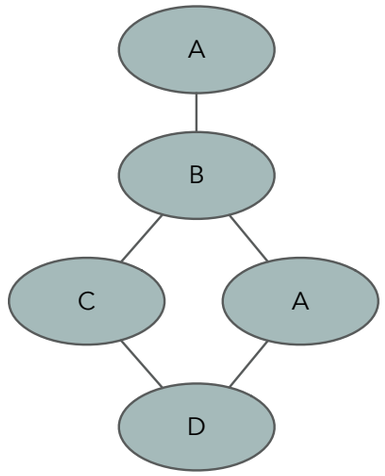
where $\preceq$ is the smallest partial order such that:

$$i < j \ \wedge \ \neg (e_i \diamond e_j) \implies i \preceq j$$

TRACES AS POMSETS OF EVENTS

Let's consider a trajectory ABCAD where only A and C are concurrent. The corresponding trace contains the two equivalent trajectories ABCAD and ABACD.

It can be concisely represented by the following **pomset** of events:



A **pomset** of events is given by:

- A set of **event identifiers** I
- A partial order relation $\preceq$ on I which we call **precedence**
- A **labelling function** that maps event identifiers to events

It is possible to abstract a trajectory into a pomset of events as follows:

$$e_1, \cdots, e_n \quad \overset{\alpha}{\longmapsto} \quad (\{1, \cdots, n\}\,,\ \preceq,\ j \mapsto e_j)$$

where $\preceq$ is the smallest partial order such that:

$$i < j \ \wedge\ \neg\,(e_i \diamond e_j) \quad \Longrightarrow \quad i \preceq j$$

Theorem

Two equivalent trajectories are abstracted to the same pomset (modulo renaming of identifiers).

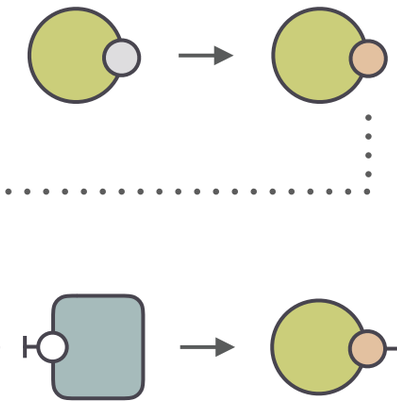
Besides, for any valid trajectory t, we have: $(\gamma \circ \alpha)(t) = \{t' \mid t' \sim t\}$

CAUSAL INFLUENCES

We say that event e_i has a **causal influence** on event e_j **through logical site** s in trajectory t if:

- e_j tests s to some value
- e_i is the last event before e_j in trajectory t that modifies s

We say that event e_i has a **causal influence** on event e_j in trajectory t if it has through some logical site.



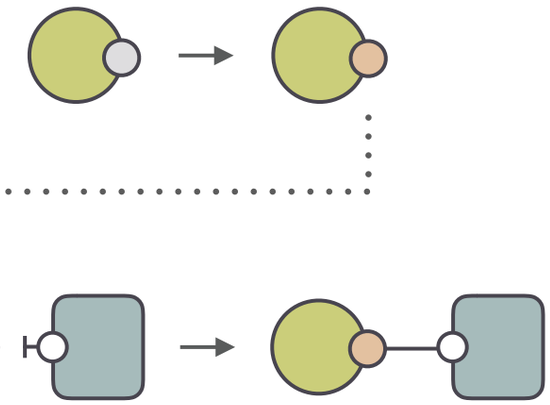
Causal influences are preserved by trajectory equivalence, and so they can be defined on traces.

CAUSAL INFLUENCES

We say that event e_i has a **causal influence** on event e_j **through logical site** s in trajectory t if:

- e_j tests s to some value
- e_i is the last event before e_j in trajectory t that modifies s

We say that event e_i has a **causal influence** on event e_j in trajectory t if it has through some logical site.



Causal influences are preserved by trajectory equivalence, and so they can be defined on traces.

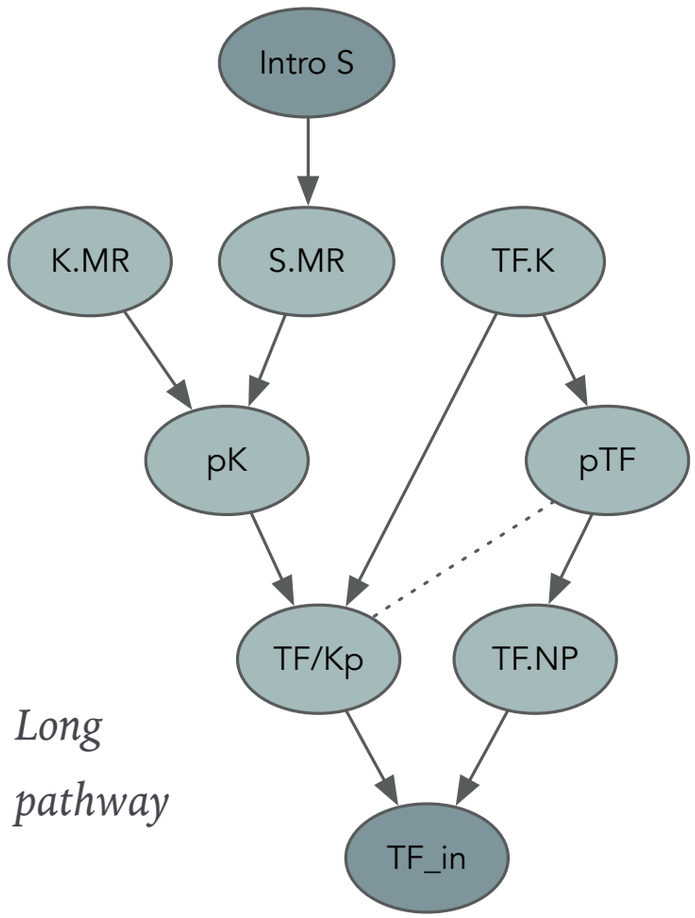
Causal past of an event

Given a trace t and an a specific event e in it, the set of events containing e along with all other events that have a transitive causal influence on it defines a valid sub-trace of t. We call it the **causal past** of e in t.

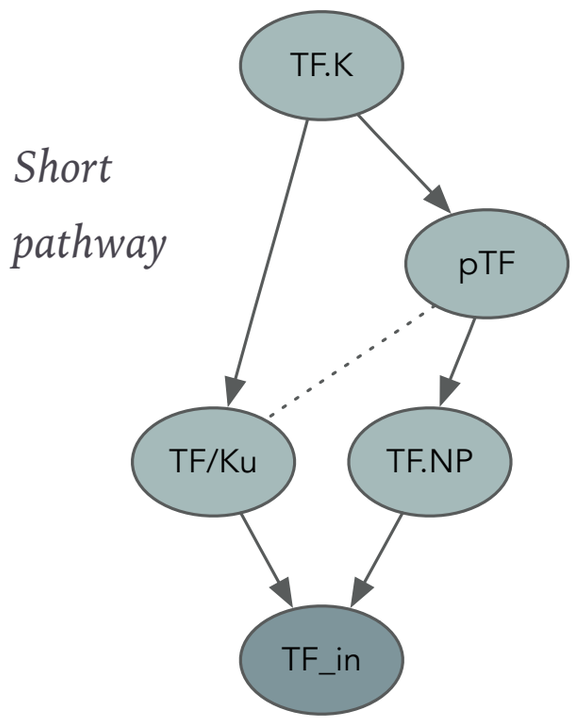
TRACE TYPES AND PATHWAYS

A **trace type** is a set of equivalent traces modulo alpha-renaming of agents.

Pathways are trace types featuring the **rule of interest**.

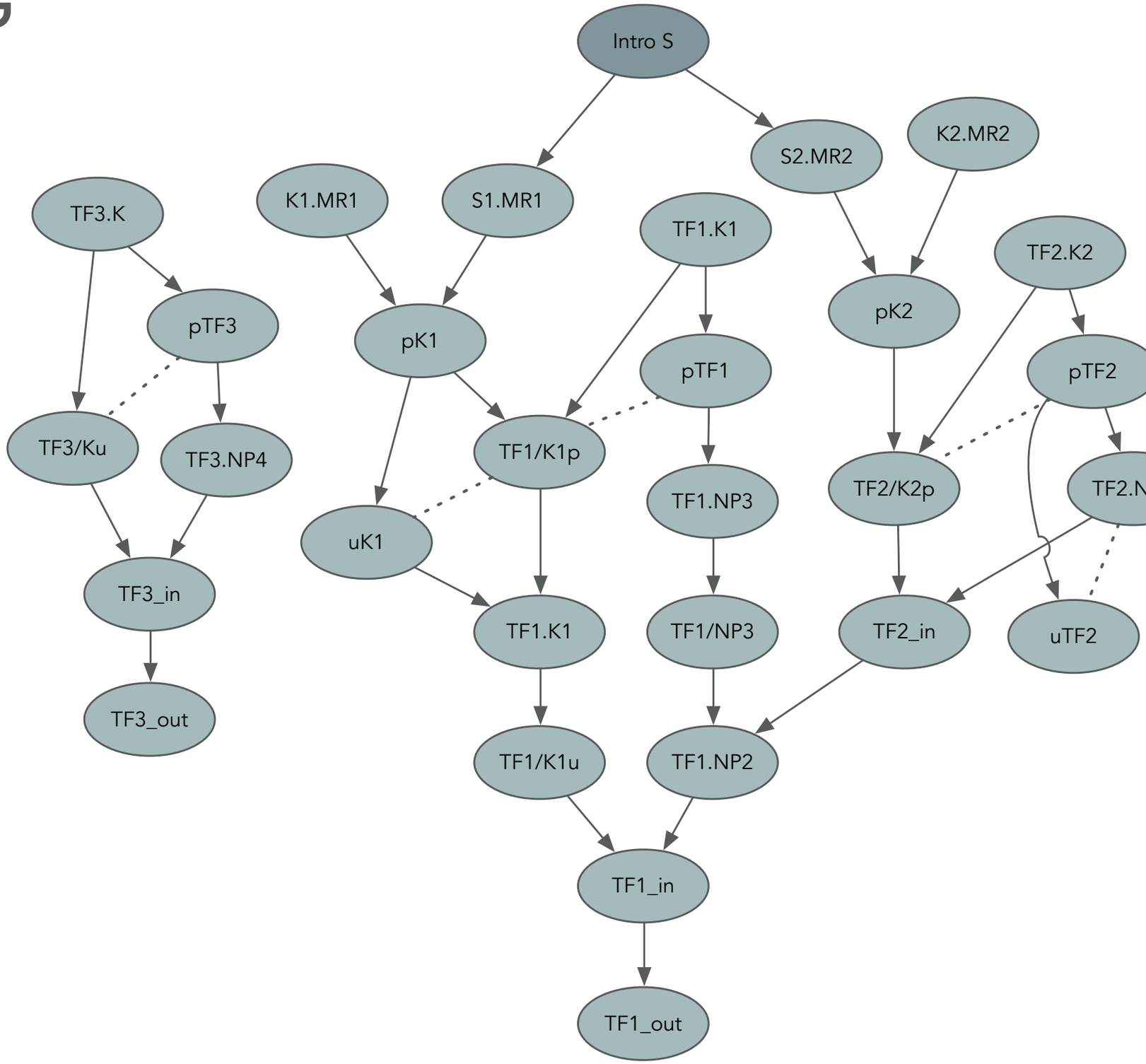


Long pathway

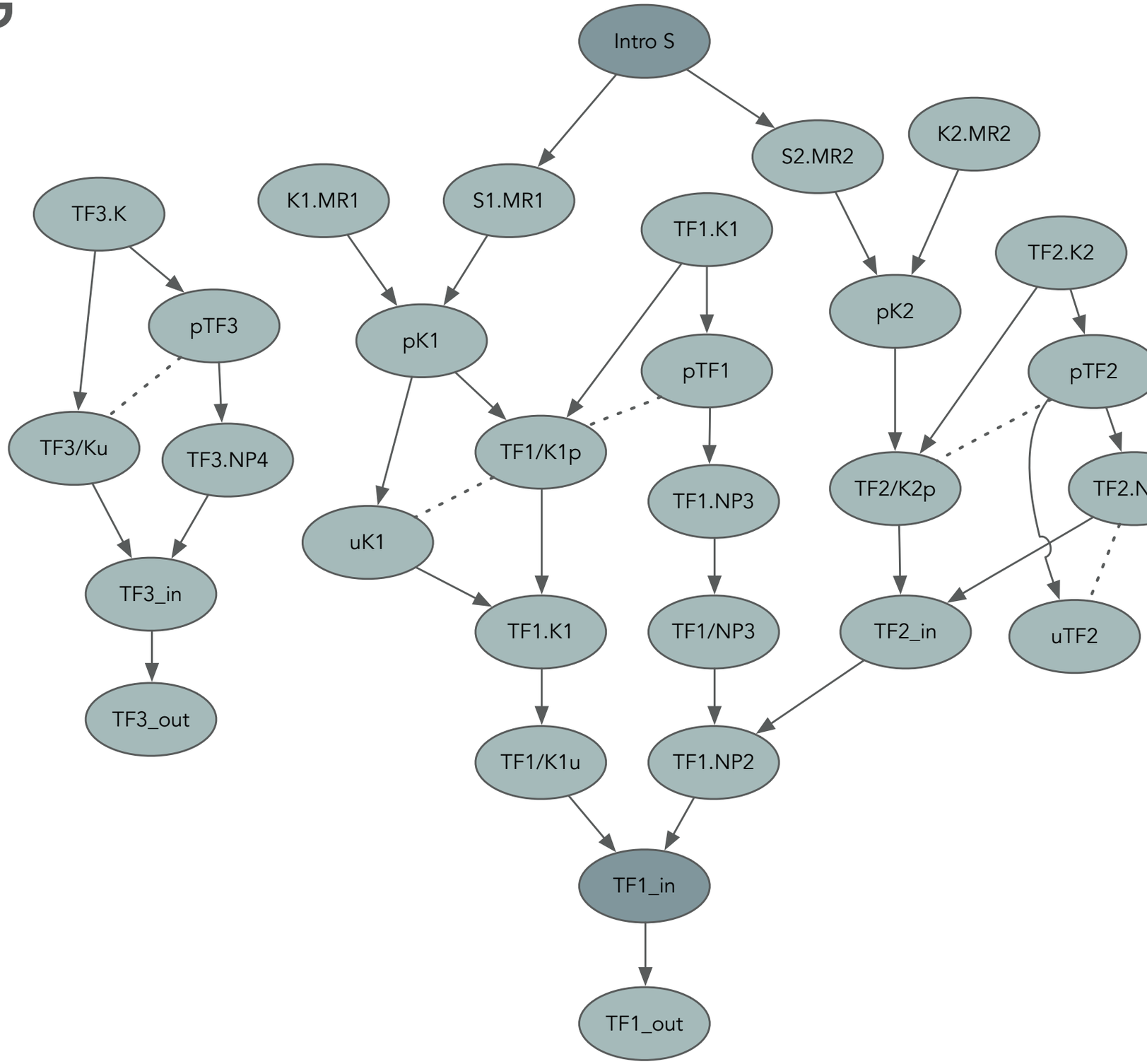


Short pathway

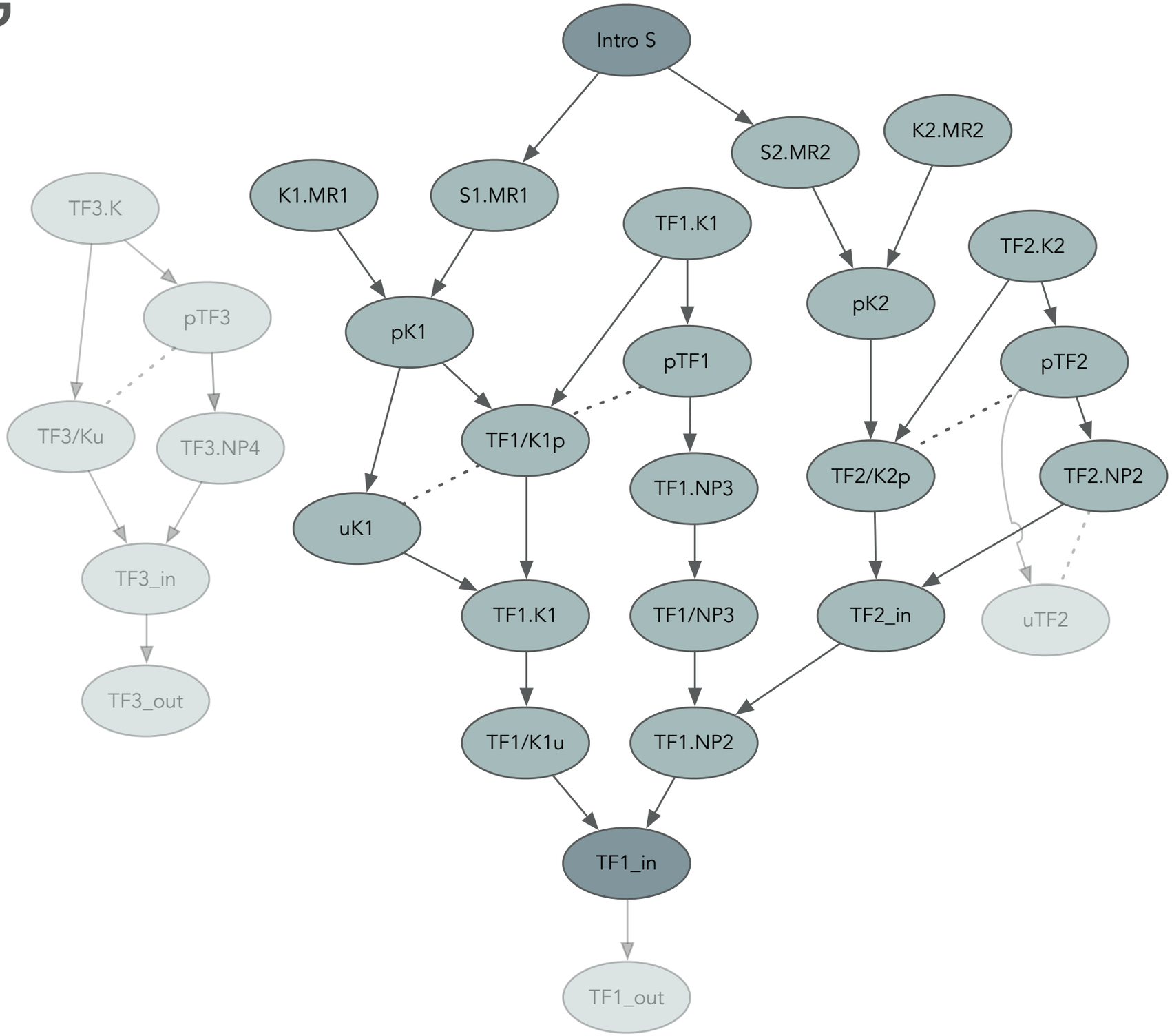
SLICING


Intro S
S1.MR1
S2.MR2
K1.MR1
K2.MR2
TF3.K
TF1.K1
TF2.K2
pTF3
pK1
pK2
pTF1
pTF2
TF3/Ku
TF3.NP4
TF1/K1p
TF1.NP3
TF2/K2p
TF2.NP2
uK1
TF3_in
TF1.K1
TF1/NP3
TF2_in
uTF2
TF3_out
TF1/K1u
TF1.NP2
TF1_in
TF1_out

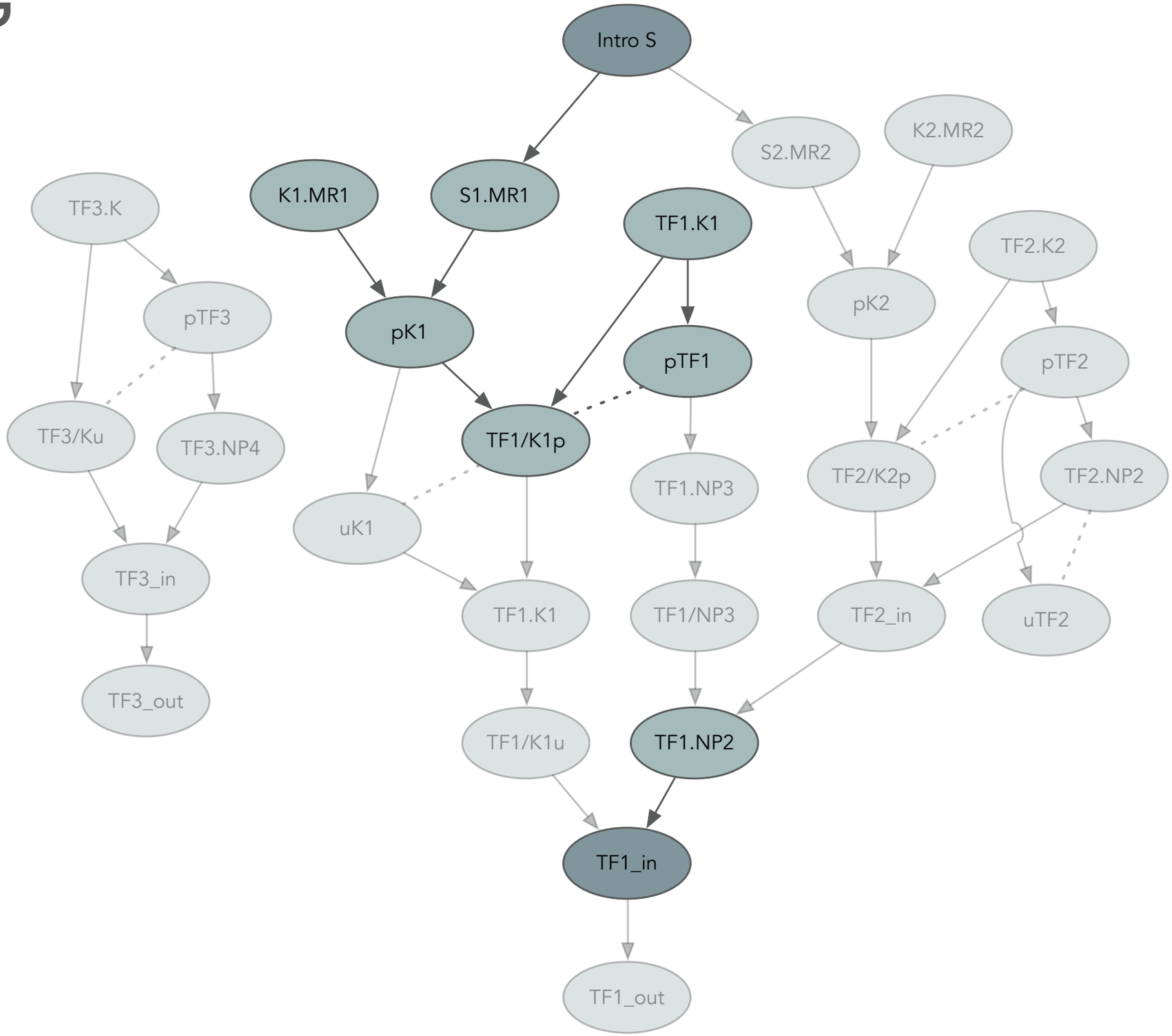
SLICING


Intro S
S1.MR1
S2.MR2
K1.MR1
K2.MR2
TF3.K
TF1.K1
TF2.K2
pTF3
pK1
pK2
pTF1
pTF2
TF3/Ku
TF3.NP4
TF1/K1p
TF1.NP3
TF2/K2p
TF2.NP2
uK1
TF3_in
TF1.K1
TF1/NP3
TF2_in
uTF2
TF3_out
TF1/K1u
TF1.NP2
TF1_in
TF1_out

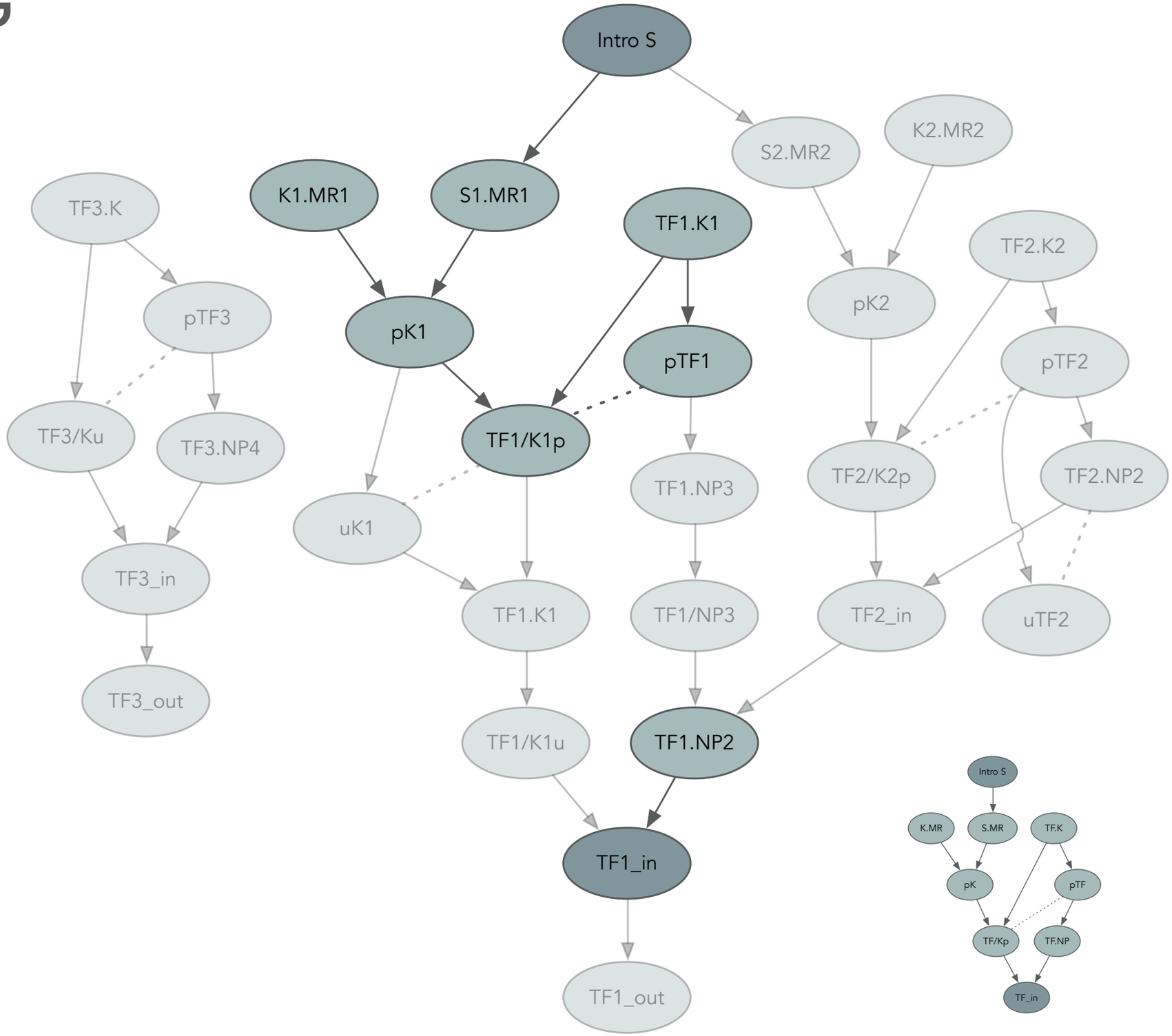
SLICING


Intro S
K1.MR1
S1.MR1
S2.MR2
K2.MR2
TF3.K
pTF3
TF1.K1
TF2.K2
pK1
pK2
pTF1
pTF2
TF3/Ku
TF3.NP4
TF1/K1p
TF1.NP3
TF2/K2p
TF2.NP2
uK1
TF3_in
TF1.K1
TF1/NP3
TF2_in
uTF2
TF3_out
TF1/K1u
TF1.NP2
TF1_in
TF1_out

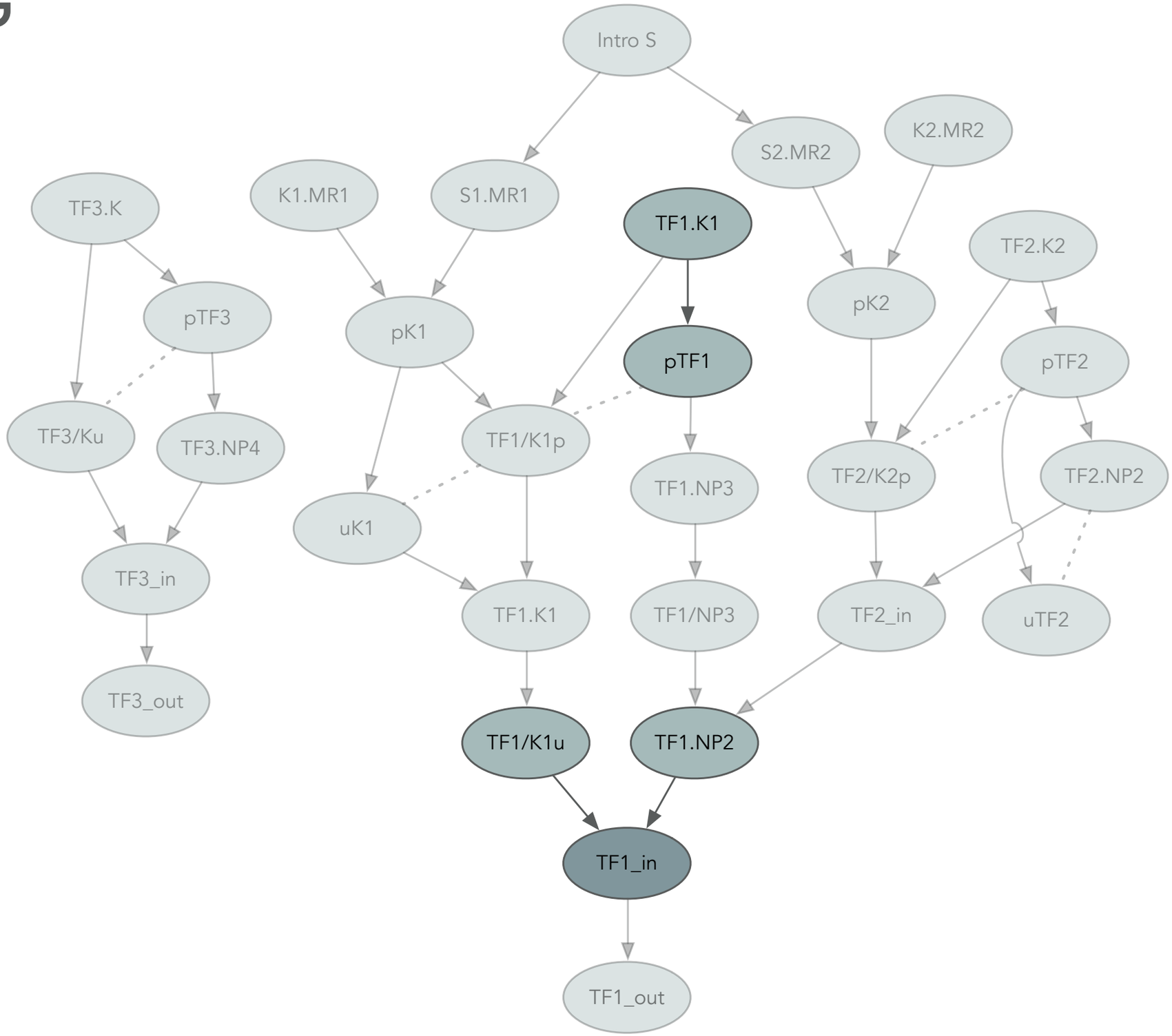
SLICING


Intro S
K1.MR1
S1.MR1
S2.MR2
K2.MR2
TF3.K
pTF3
TF1.K1
TF2.K2
pK1
pK2
pTF1
pTF2
TF3/Ku
TF3.NP4
TF1/K1p
TF1.NP3
TF2/K2p
TF2.NP2
uK1
TF3_in
TF1.K1
TF1/NP3
TF2_in
uTF2
TF3_out
TF1/K1u
TF1.NP2
TF1_in
TF1_out

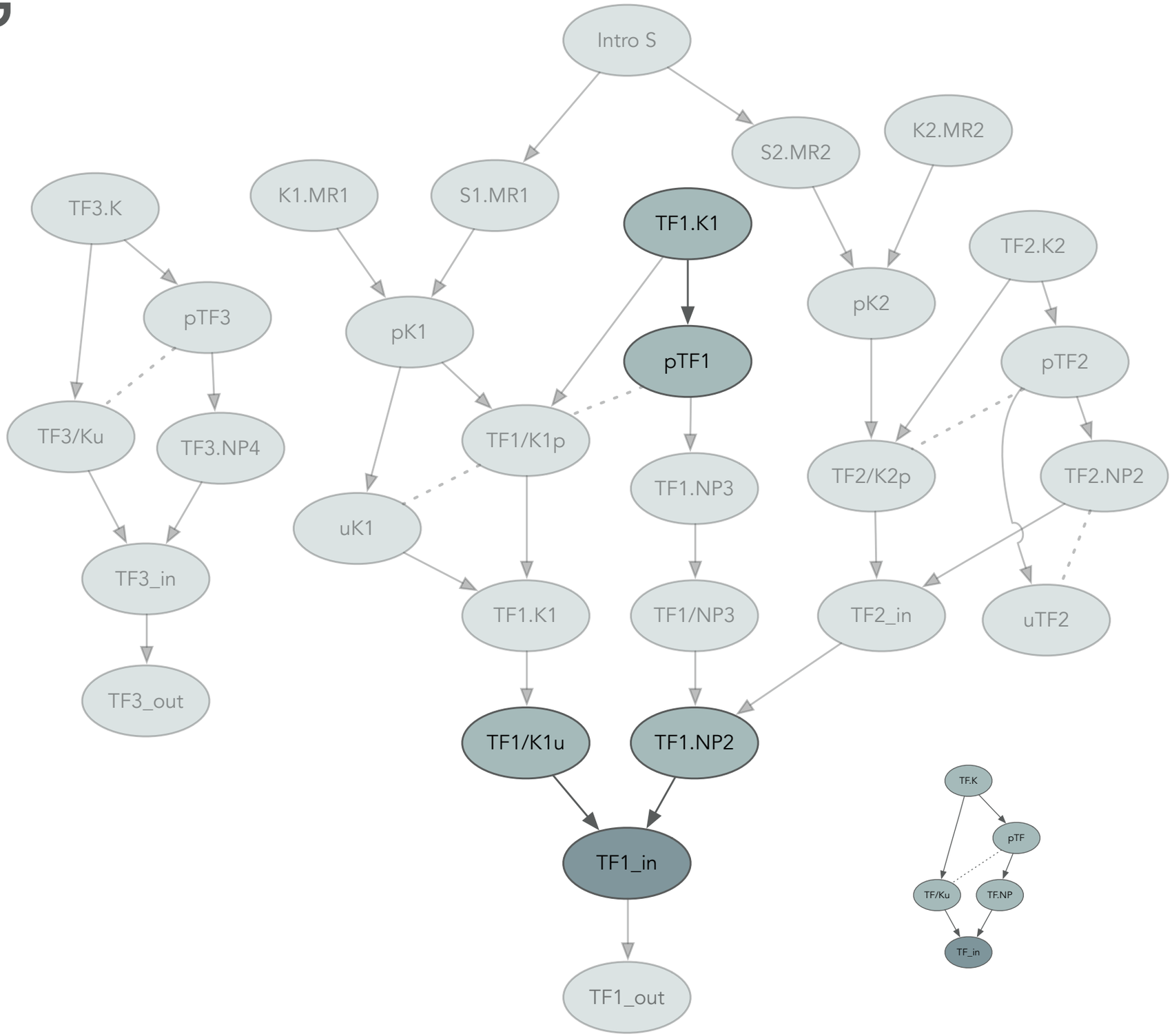
SLICING


Intro S
K1.MR1
S1.MR1
S2.MR2
K2.MR2
TF3.K
TF1.K1
TF2.K2
pTF3
pK1
pK2
pTF1
pTF2
TF3/Ku
TF3.NP4
TF1/K1p
TF1.NP3
TF2/K2p
TF2.NP2
uK1
TF3_in
TF1.K1
TF1/NP3
TF2_in
uTF2
TF3_out
TF1/K1u
TF1.NP2
TF1_in
TF1_out
Intro S
K.MR
S.MR
TF.K
pK
pTF
TF/Kp
TF.NP
TF_in

SLICING

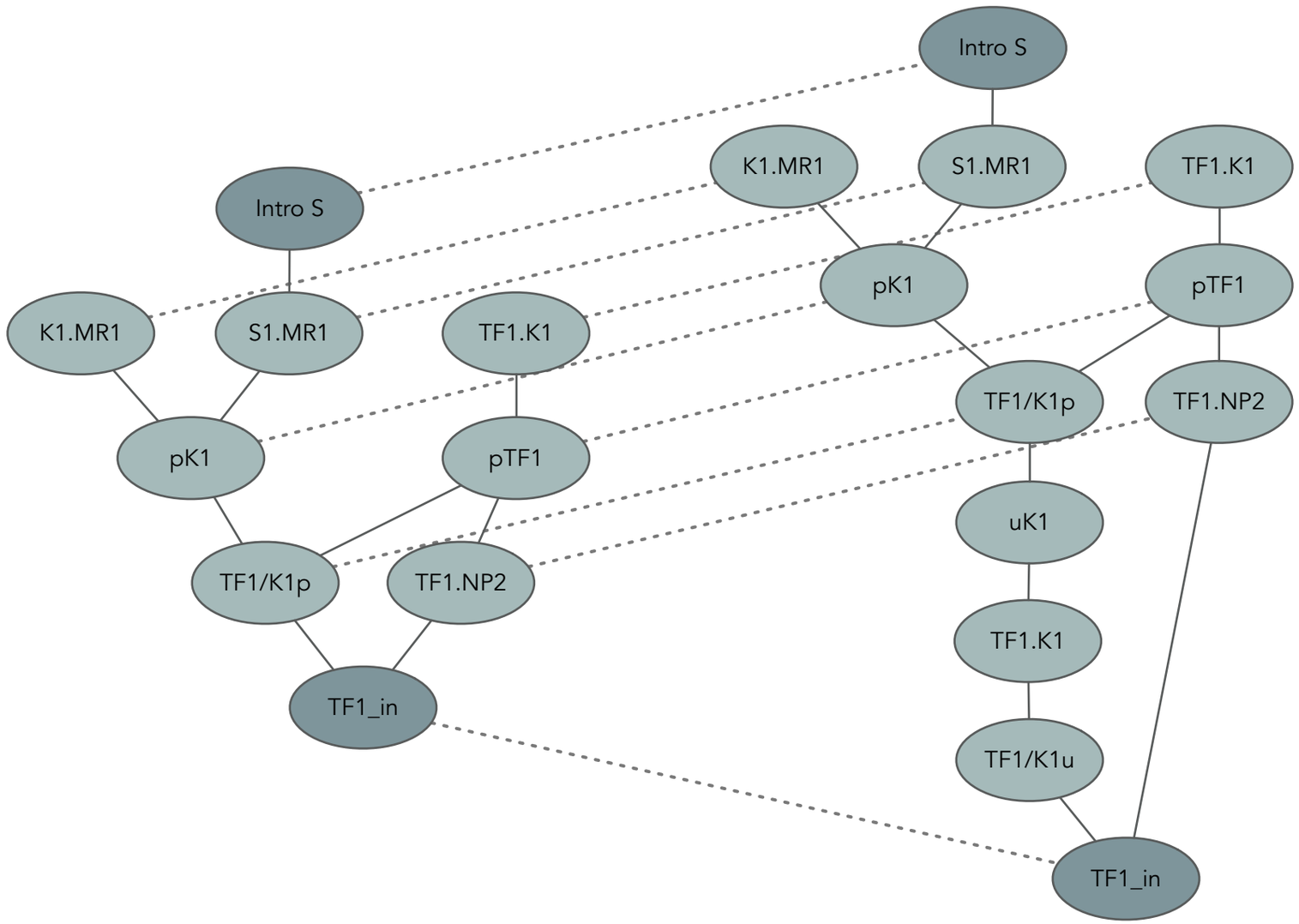

Intro S
S2.MR2
K2.MR2
TF3.K
K1.MR1
S1.MR1
TF1.K1
TF2.K2
pTF3
pK1
pK2
pTF1
pTF2
TF3/Ku
TF3.NP4
TF1/K1p
TF1.NP3
TF2/K2p
TF2.NP2
uK1
TF3_in
TF1.K1
TF1/NP3
TF2_in
uTF2
TF3_out
TF1/K1u
TF1.NP2
TF1_in
TF1_out

SLICING


Intro S
S2.MR2
K2.MR2
K1.MR1
S1.MR1
TF3.K
TF1.K1
TF2.K2
pTF3
pK1
pK2
pTF1
pTF2
TF3/Ku
TF3.NP4
TF1/K1p
TF1.NP3
TF2/K2p
TF2.NP2
uK1
TF3_in
TF1.K1
TF1/NP3
TF2_in
uTF2
TF3_out
TF1/K1u
TF1.NP2
TF1_in
TF1_out
TF.K
pTF
TF/Ku
TF.NP
TF_in

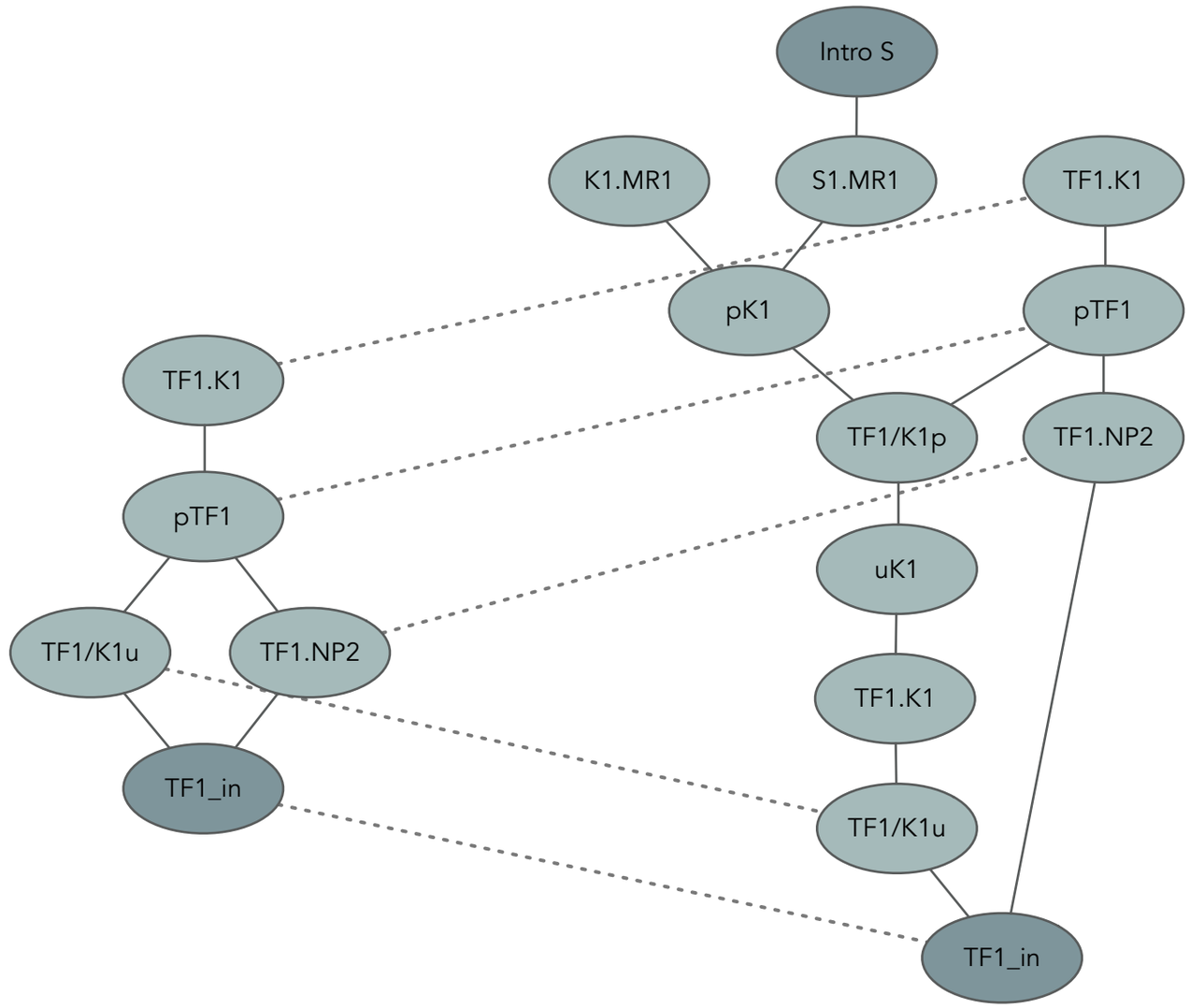
TRACE MATCHINGS

A trace **matching** is an injective pomset morphism in the sense that it is a one-to-one mapping from a trace to another one that preserves event labels and precedence.



TRACE MATCHINGS

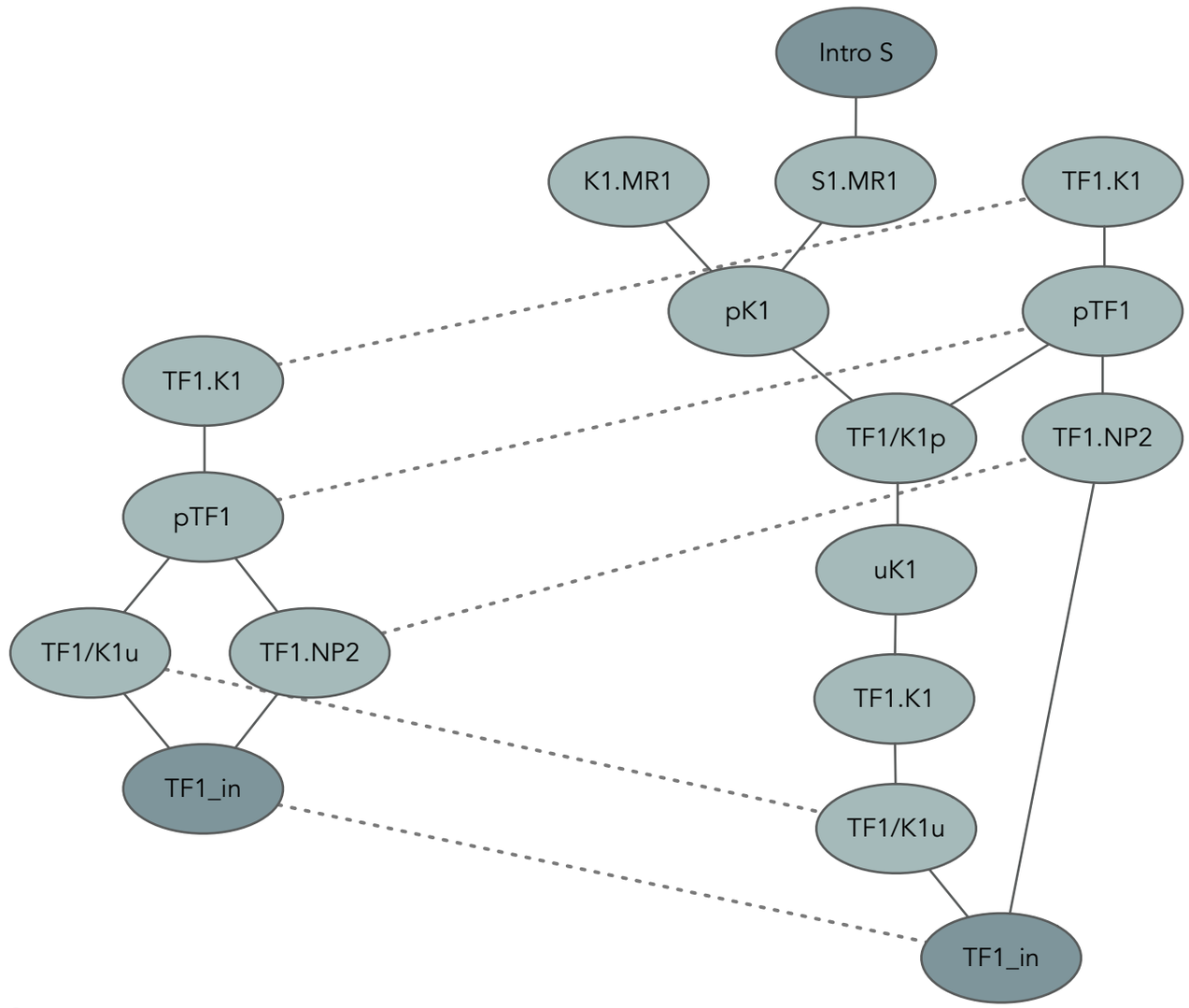
A trace **matching** is an injective pomset morphism in the sense that it is a one-to-one mapping from a trace to another one that preserves event labels and precedence.



TRACE MATCHINGS

A trace **matching** is an injective pomset morphism in the sense that it is a one-to-one mapping from a trace to another one that preserves event labels and precedence.

We want to define a notion of **faithful matching t**hat would testify that a trace is a faithful summary of another one.

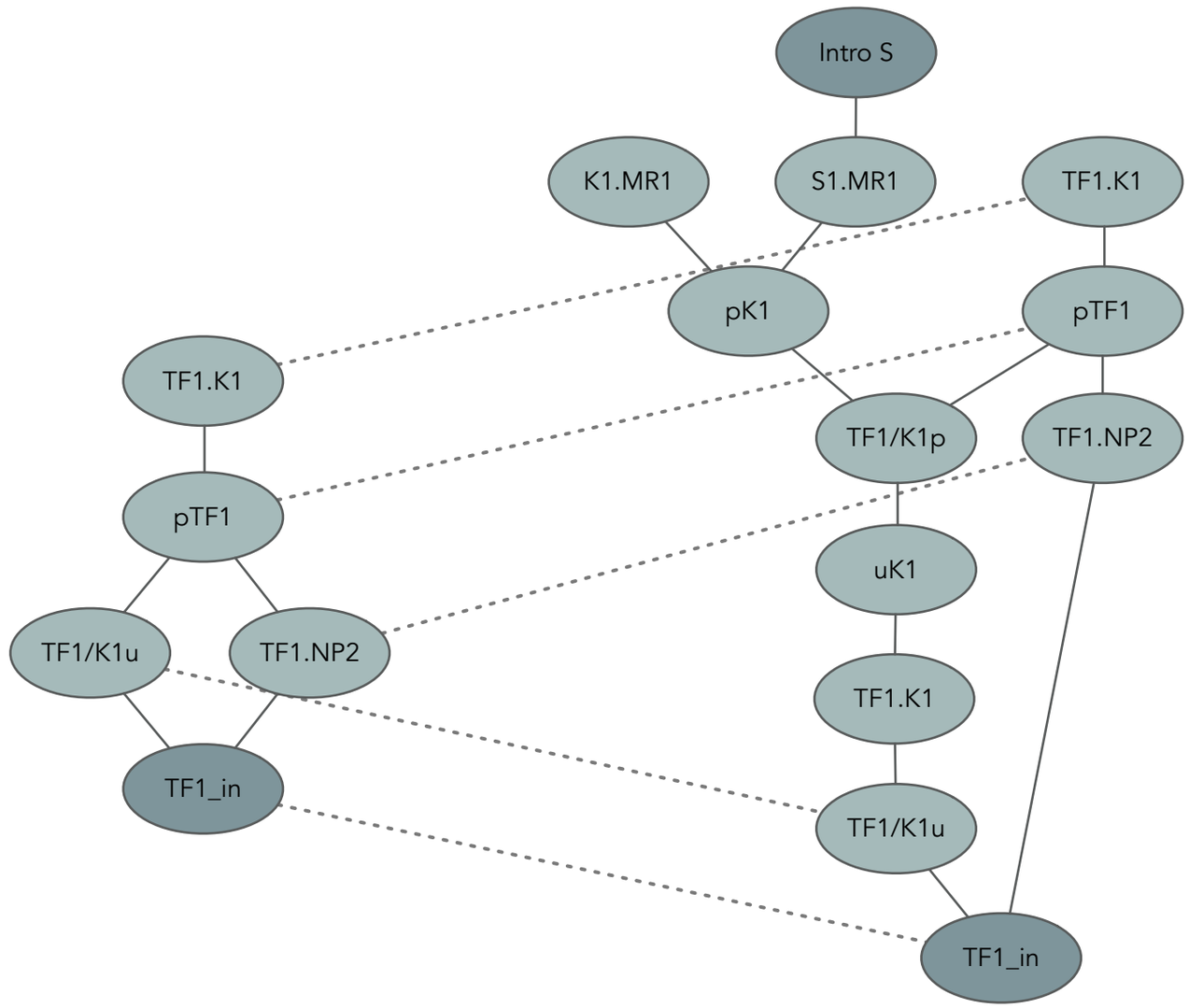


Question: what relations between events should be preserved by a faithful matching ?

A FIRST CRITERION

A **binding pair** is a pair of two events where the second one removes a bond that is created by the first one.

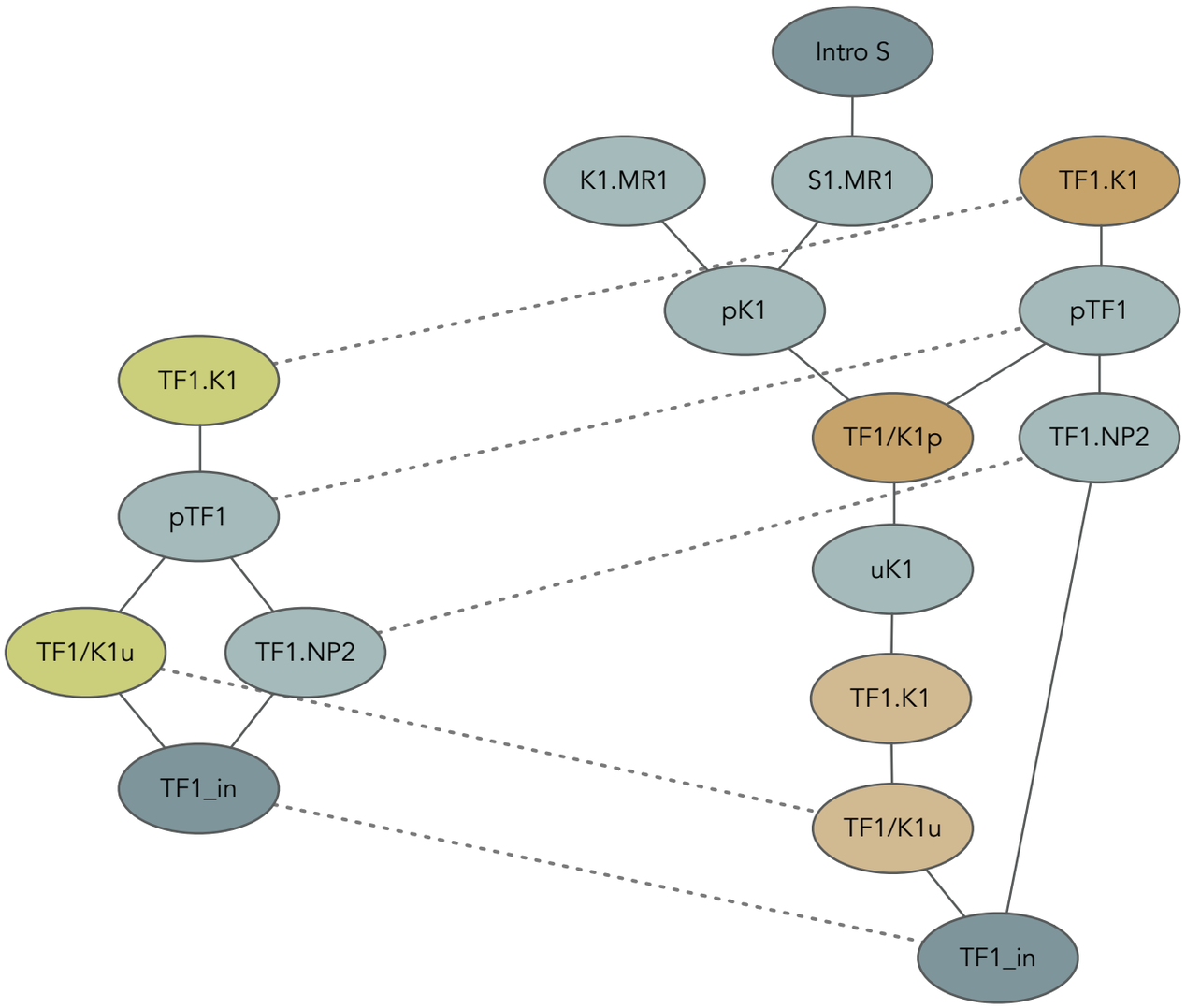
A faithful matching should **preserve binding pairs**.



A FIRST CRITERION

A **binding pair** is a pair of two events where the second one removes a bond that is created by the first one.

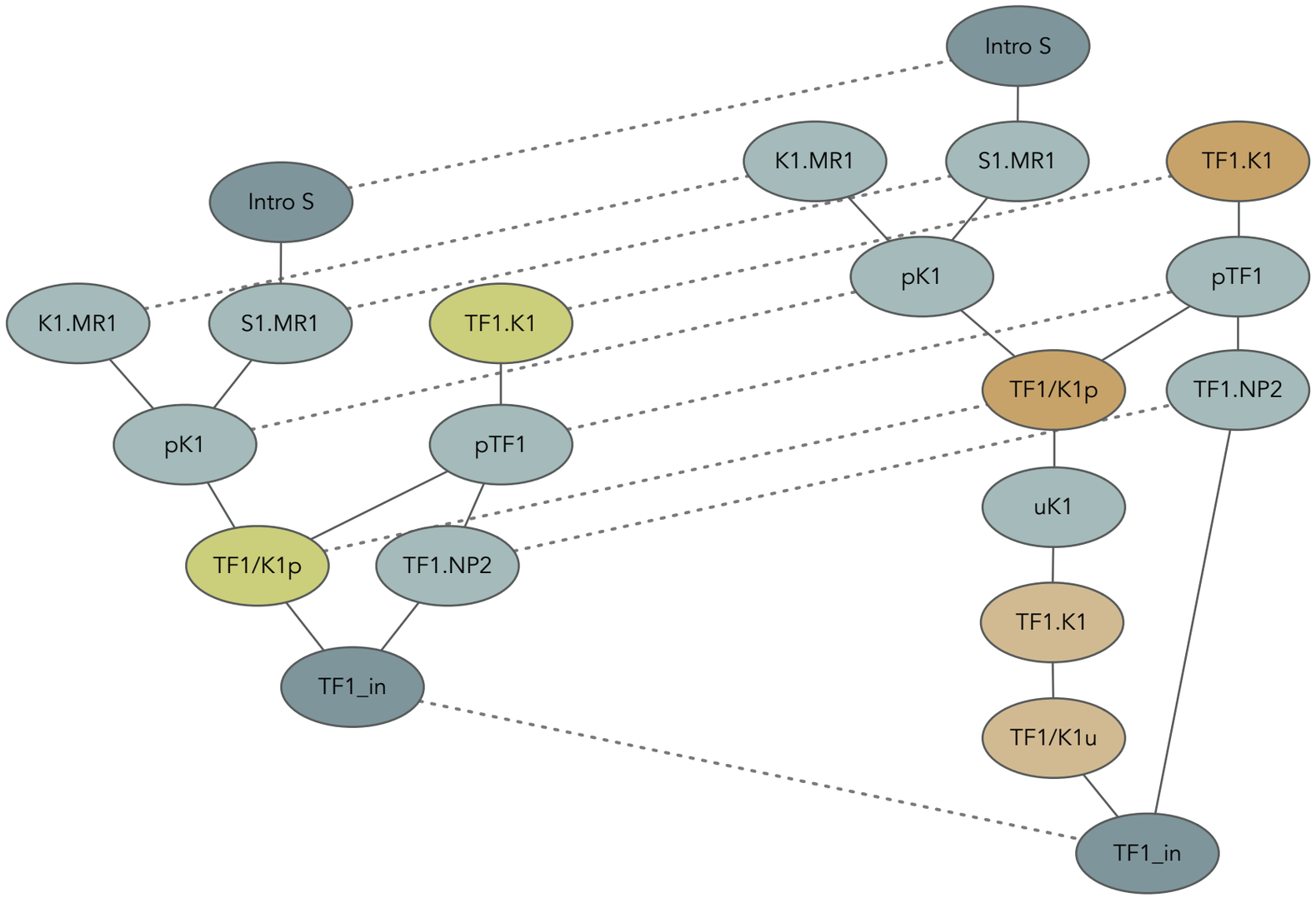
A faithful matching should **preserve binding pairs**.



A FIRST CRITERION

A **binding pair** is a pair of two events where the second one removes a bond that is created by the first one.

A faithful matching should **preserve binding pairs**.



GENERALIZATION

Default and altered states for logical sites

Each logical site is assigned at most one **default state**. Non-default states are said to be **altered**. Typically, sites are free and unphosphorylated by default. Other choices can be made, as long as the **alternating condition** is respected:

> *If a logical site has a default value, then it cannot transition from an altered value to an other altered value.*

GENERALIZATION

Default and altered states for logical sites

Each logical site is assigned at most one **default state**. Non-default states are said to be **altered**. Typically, sites are free and unphosphorylated by default. Other choices can be made, as long as the **alternating condition** is respected:

> *If a logical site has a default value, then it cannot transition from an altered value to an other altered value.*

Conjugate pairs

We say that two events e_1 and e_2 form a **conjugate pair** for logical site s if e_1 sets s to an altered value and e_2 is the first event after e_1 to set s back to its default value.

GENERALIZATION

Default and altered states for logical sites

Each logical site is assigned at most one **default state**. Non-default states are said to be **altered**. Typically, sites are free and unphosphorylated by default. Other choices can be made, as long as the **alternating condition** is respected:

> *If a logical site has a default value, then it cannot transition from an altered value to an other altered value.*

Conjugate pairs

We say that two events e_1 and e_2 form a **conjugate pair** for logical site s if e_1 sets s to an altered value and e_2 is the first event after e_1 to set s back to its default value.

Strong causal influences

Suppose an event e_1 has a causal influence on another event e_2. This influence is said to be **strong** if it goes through a logical site that is set to an altered value by e_1.

MAIN RESULT

A trace matching is said to be **faithful** if it preserves strong causal influences and conjugate pairs.

MAIN RESULT

A trace matching is said to be **faithful** if it preserves strong causal influences and conjugate pairs.

Theorem

Let t a trace and e an event in t. Then there exists a unique minimal valid sub-trace σ of t such that:

- σ contains contains e
- The identity matching from σ to t is faithful

The type of σ is called the **producing pathway** of e in t.

MAIN RESULT

A trace matching is said to be **faithful** if it preserves strong causal influences and conjugate pairs.

Theorem

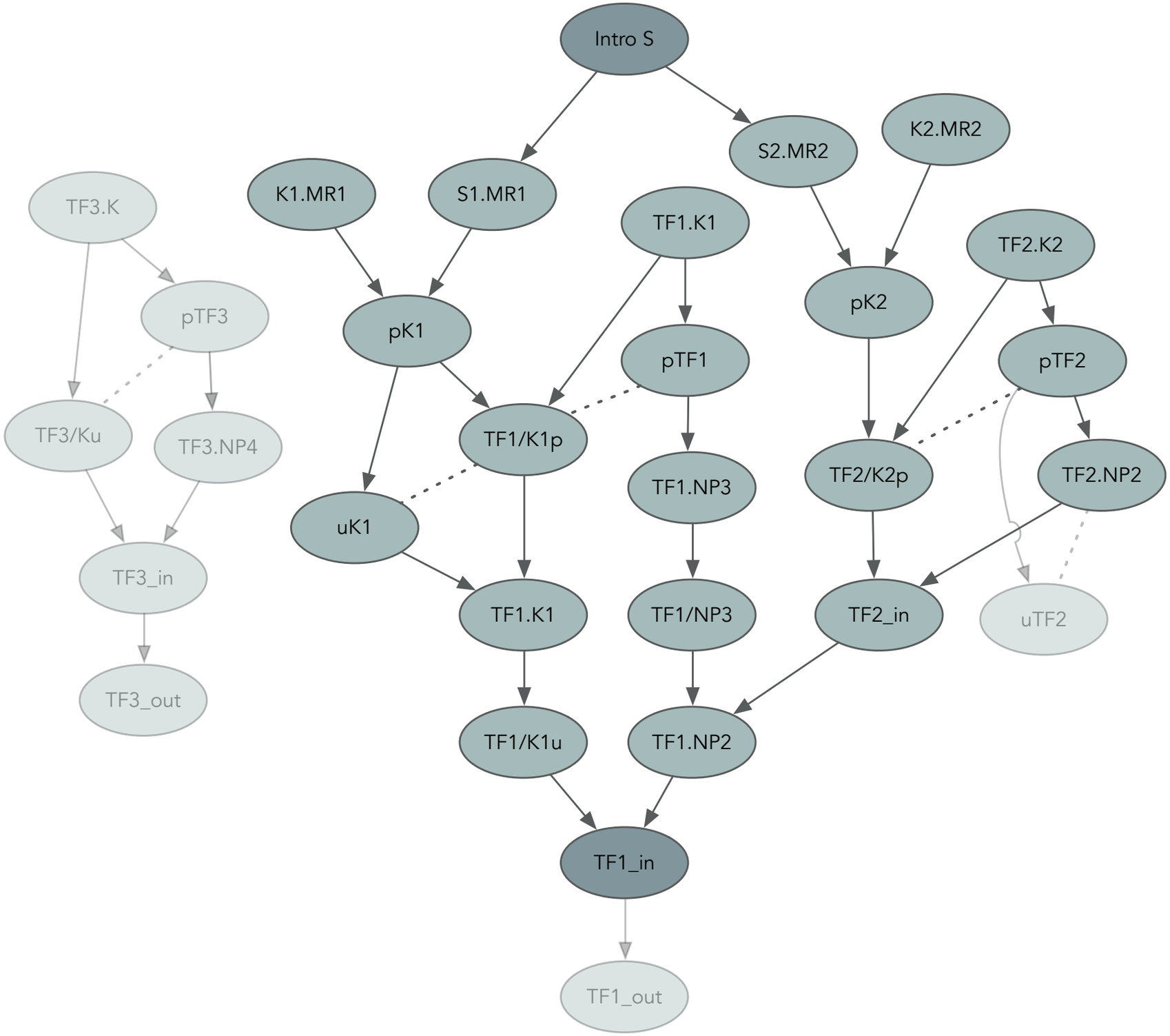
Let t a trace and e an event in t. Then there exists a unique minimal valid sub-trace σ of t such that:

- σ contains contains e
- The identity matching from σ to t is faithful

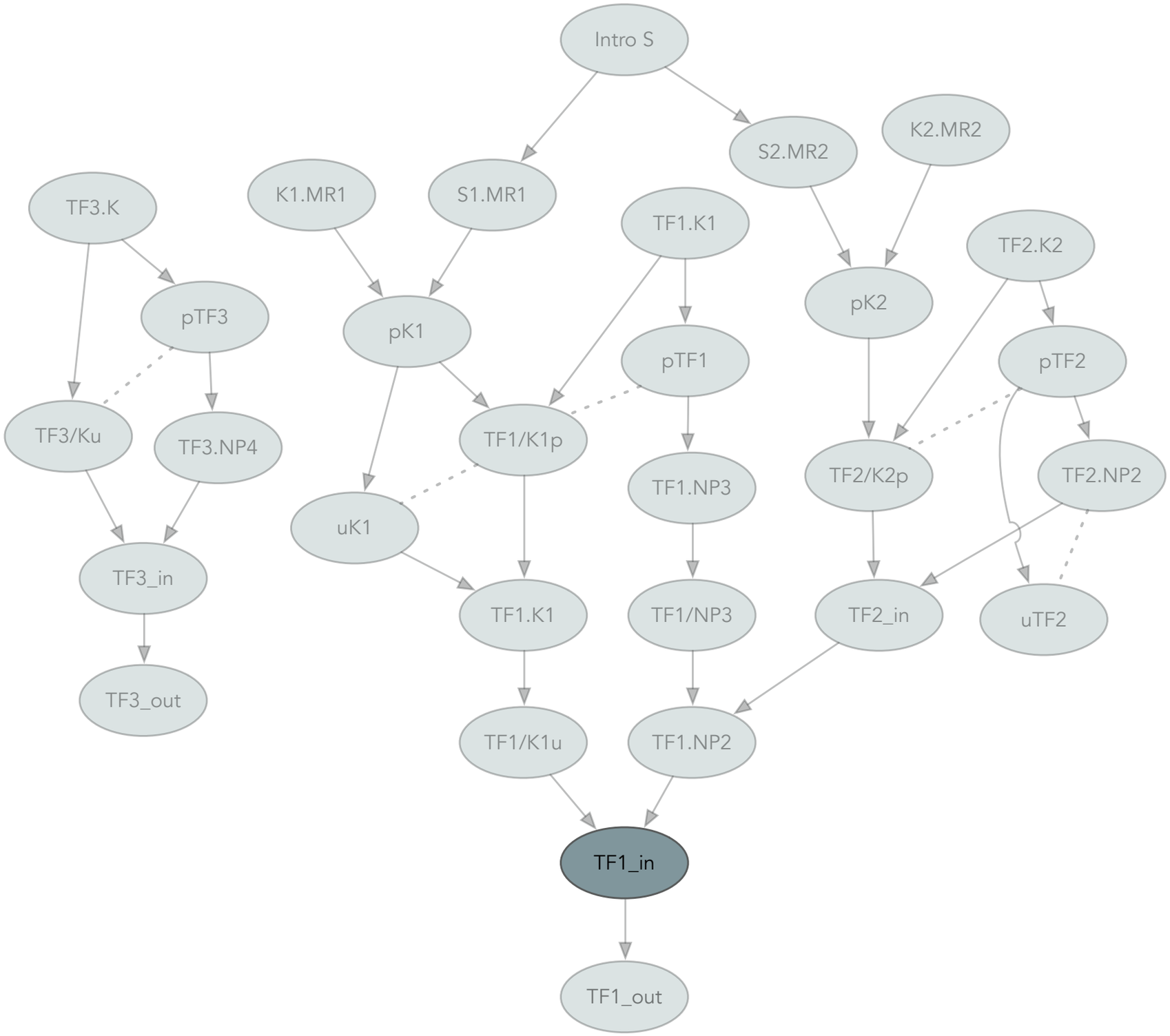
The type of σ is called the **producing pathway** of e in t.

Besides, there exists a greedy **quasi-linear algorithm** to compute the producing pathway of an event in a trace.

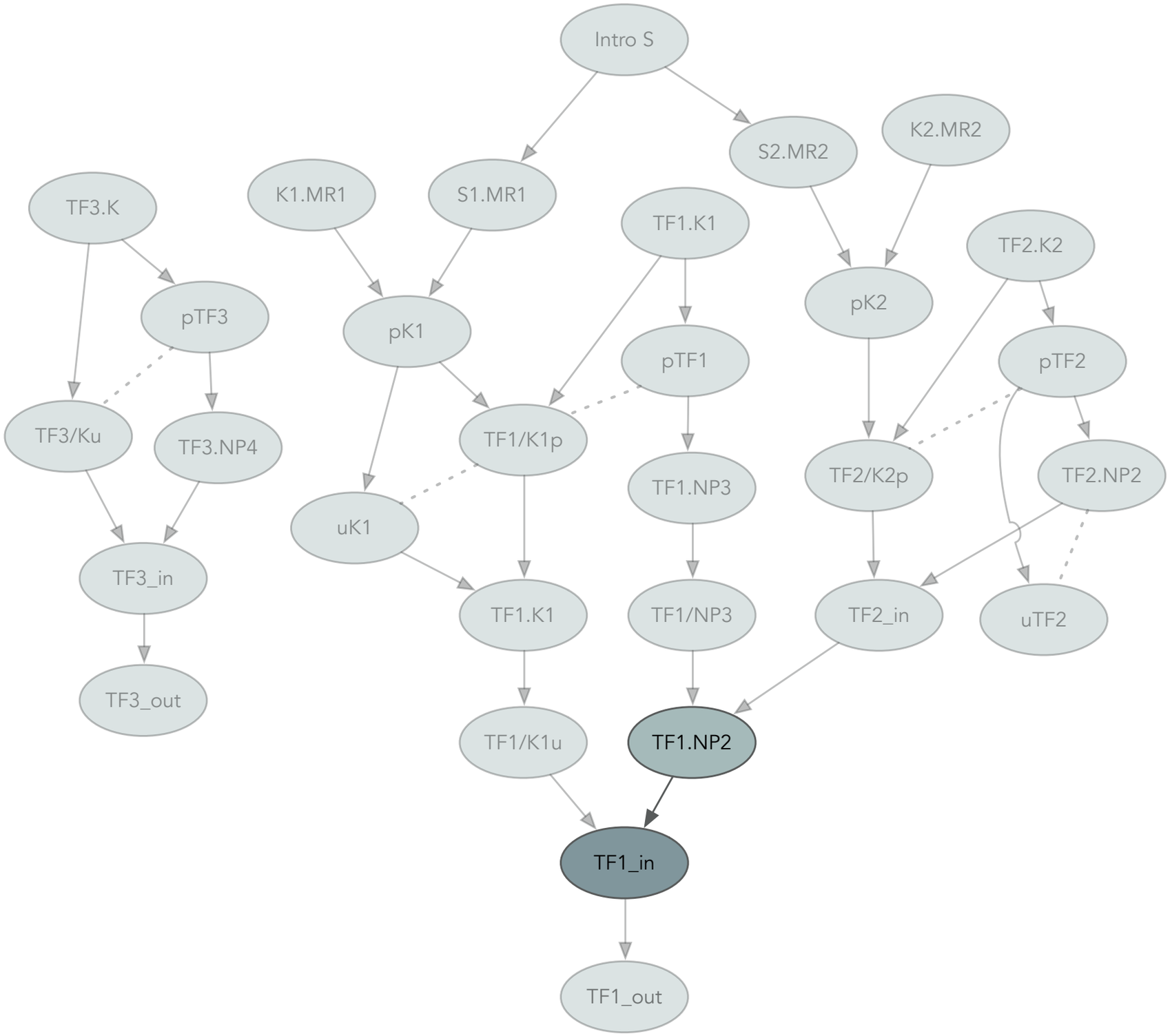
BACK TO OUR EXAMPLE



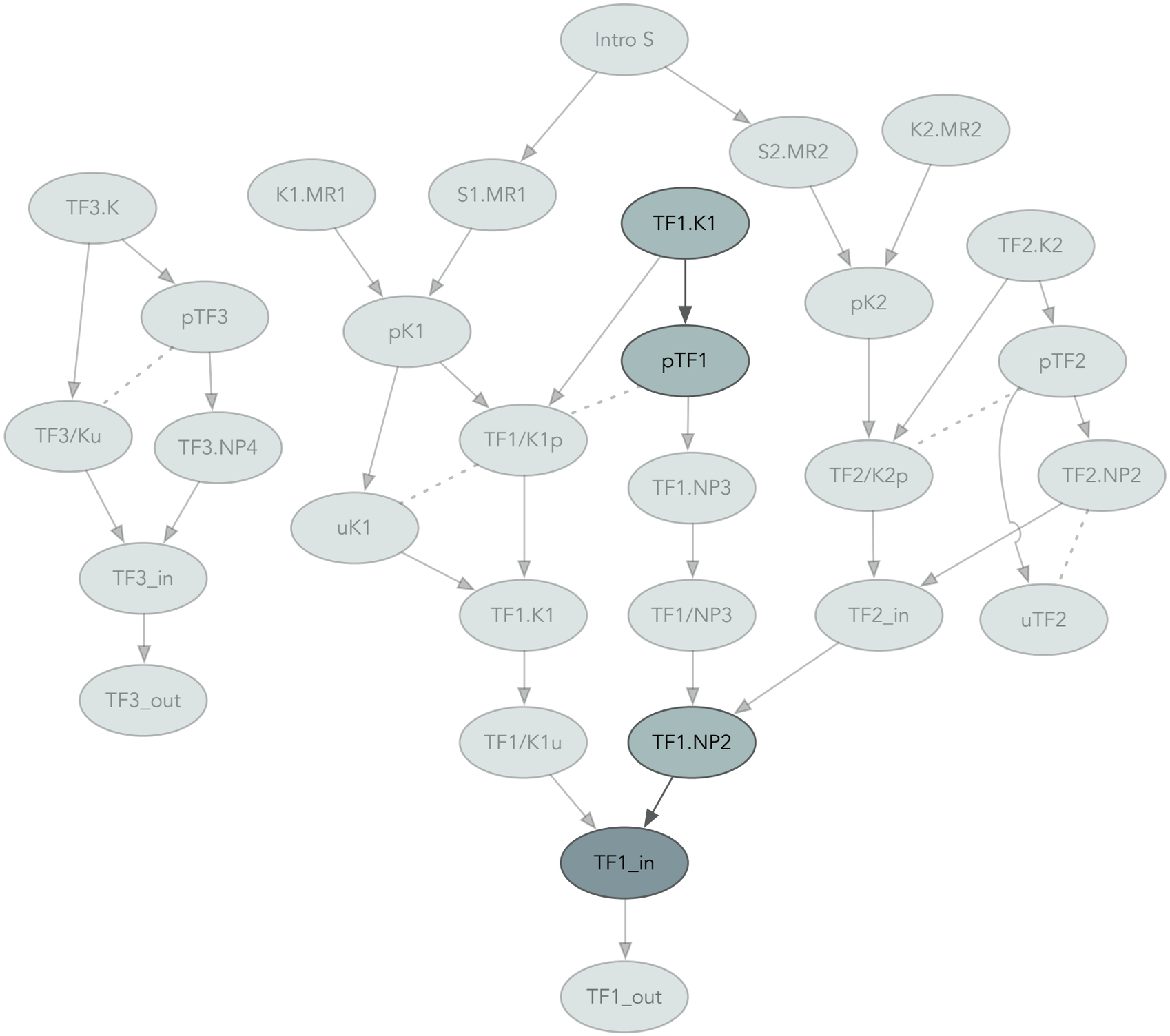
BACK TO OUR EXAMPLE



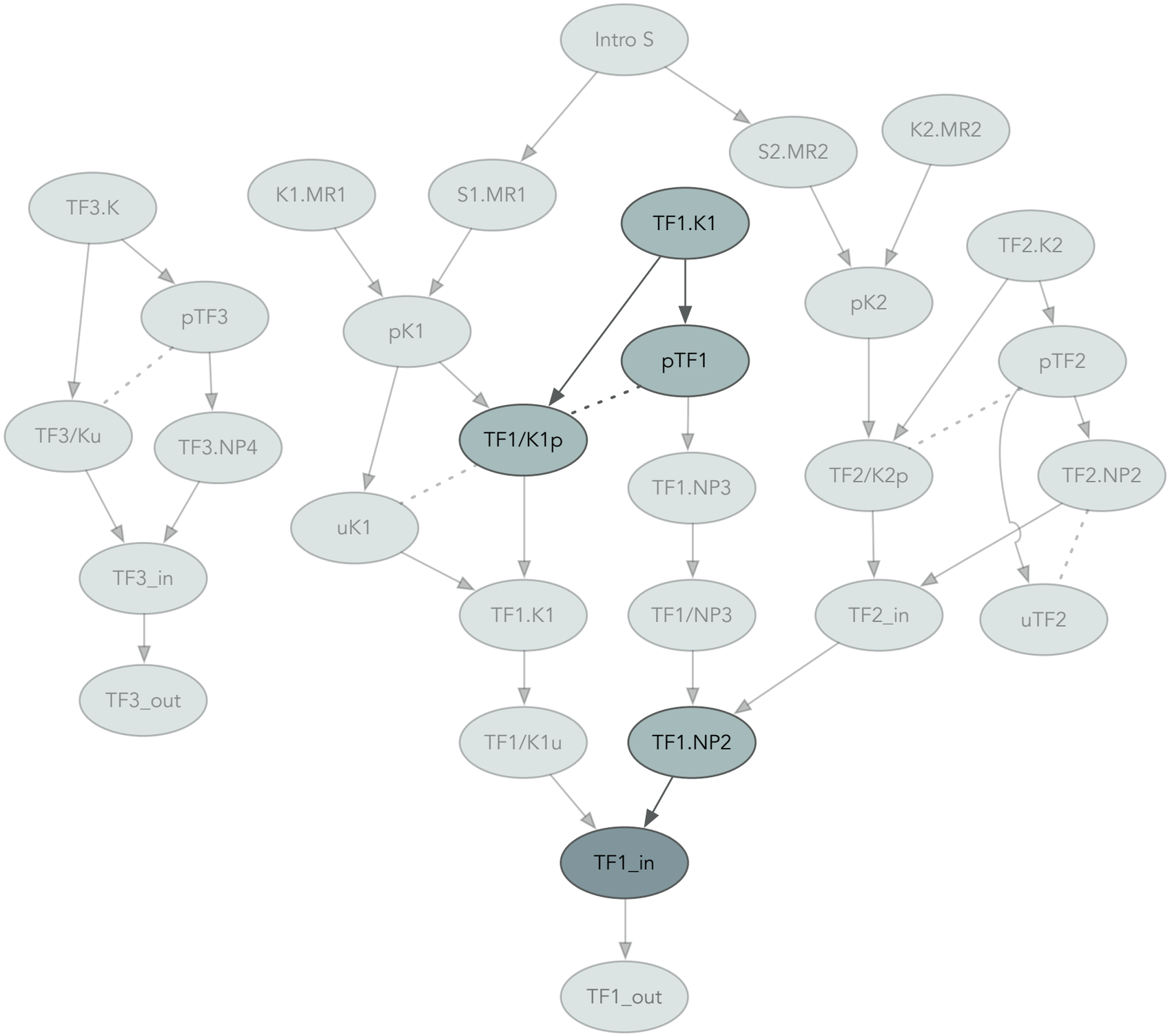
BACK TO OUR EXAMPLE



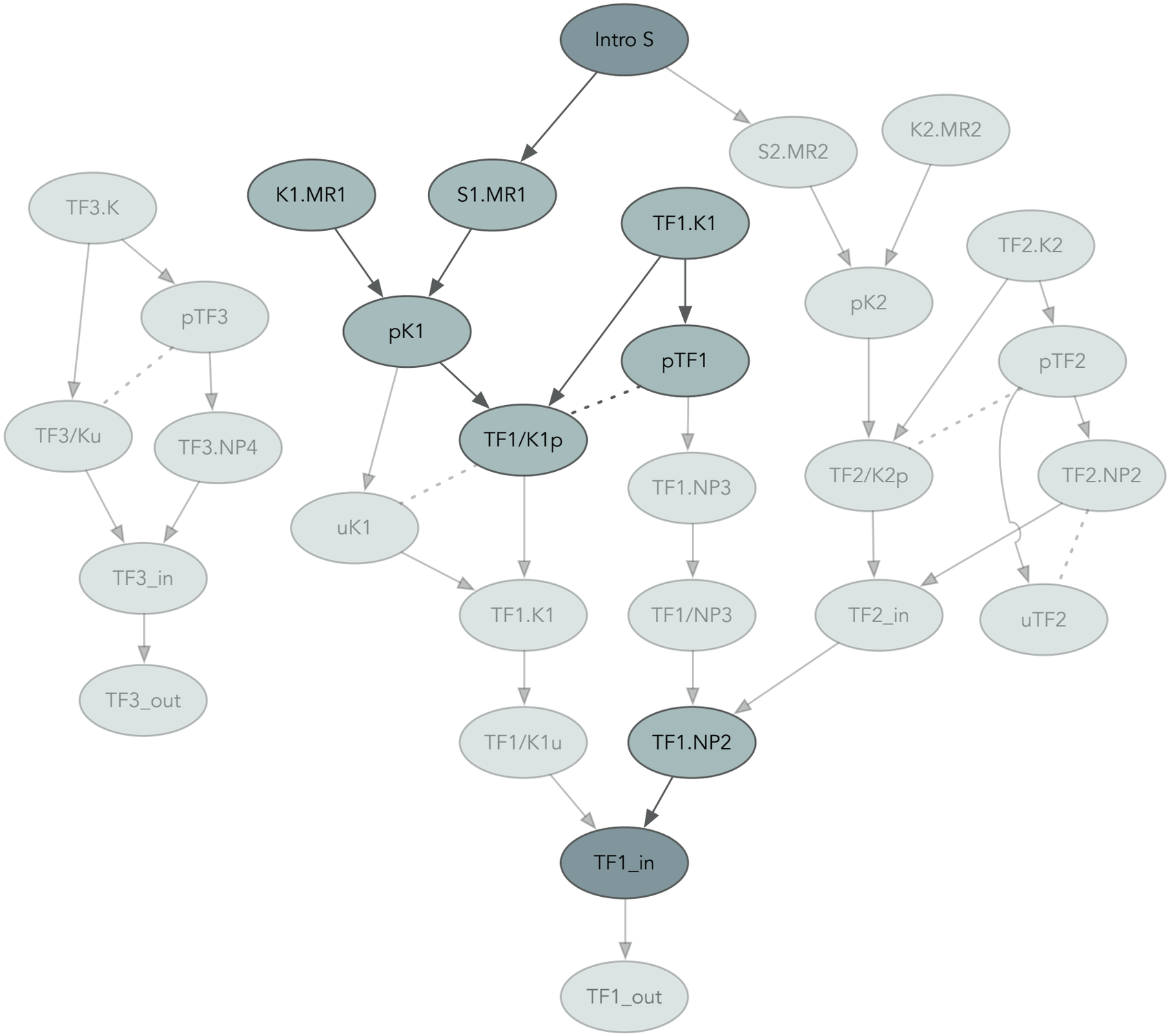
BACK TO OUR EXAMPLE



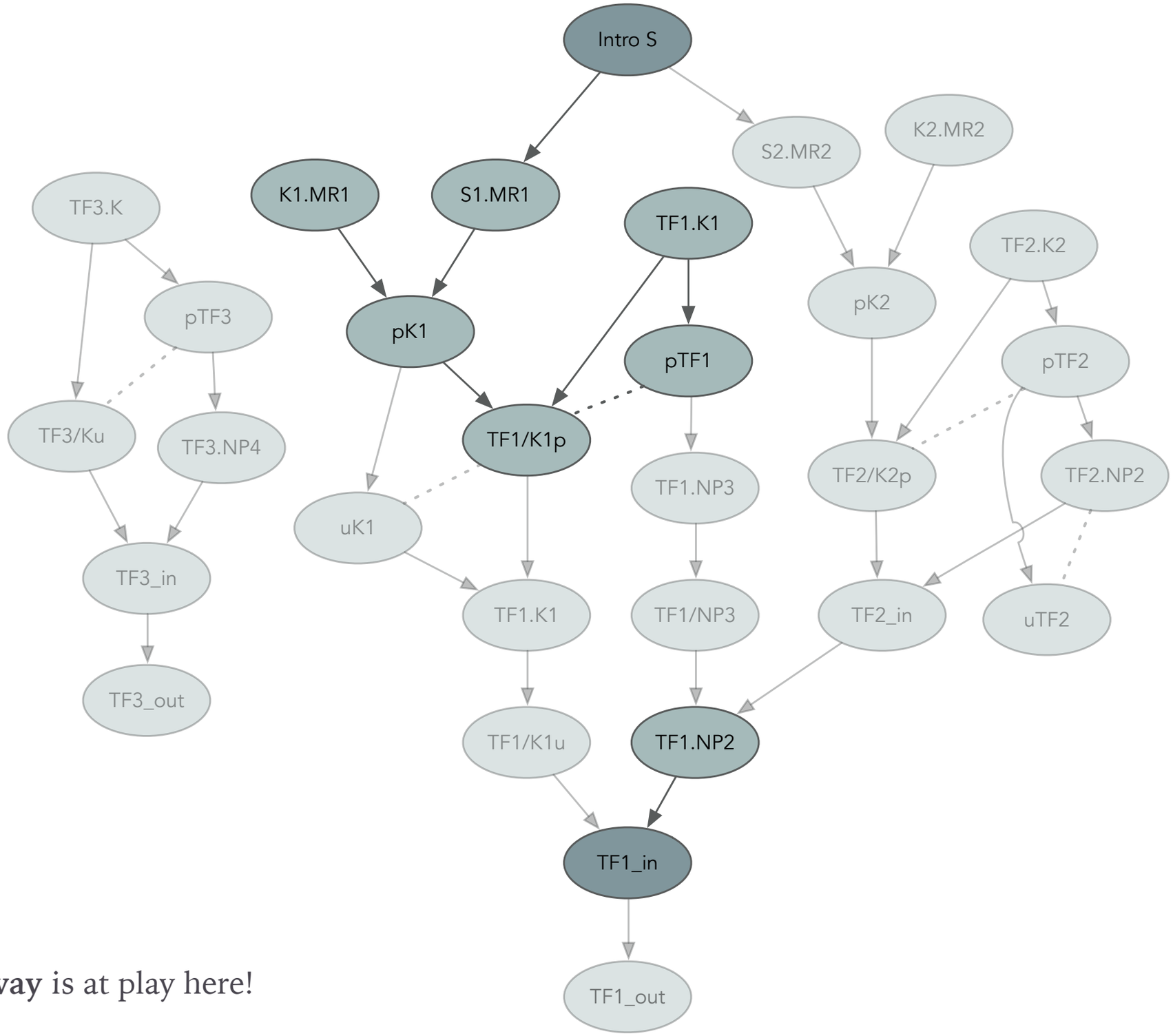
BACK TO OUR EXAMPLE



BACK TO OUR EXAMPLE

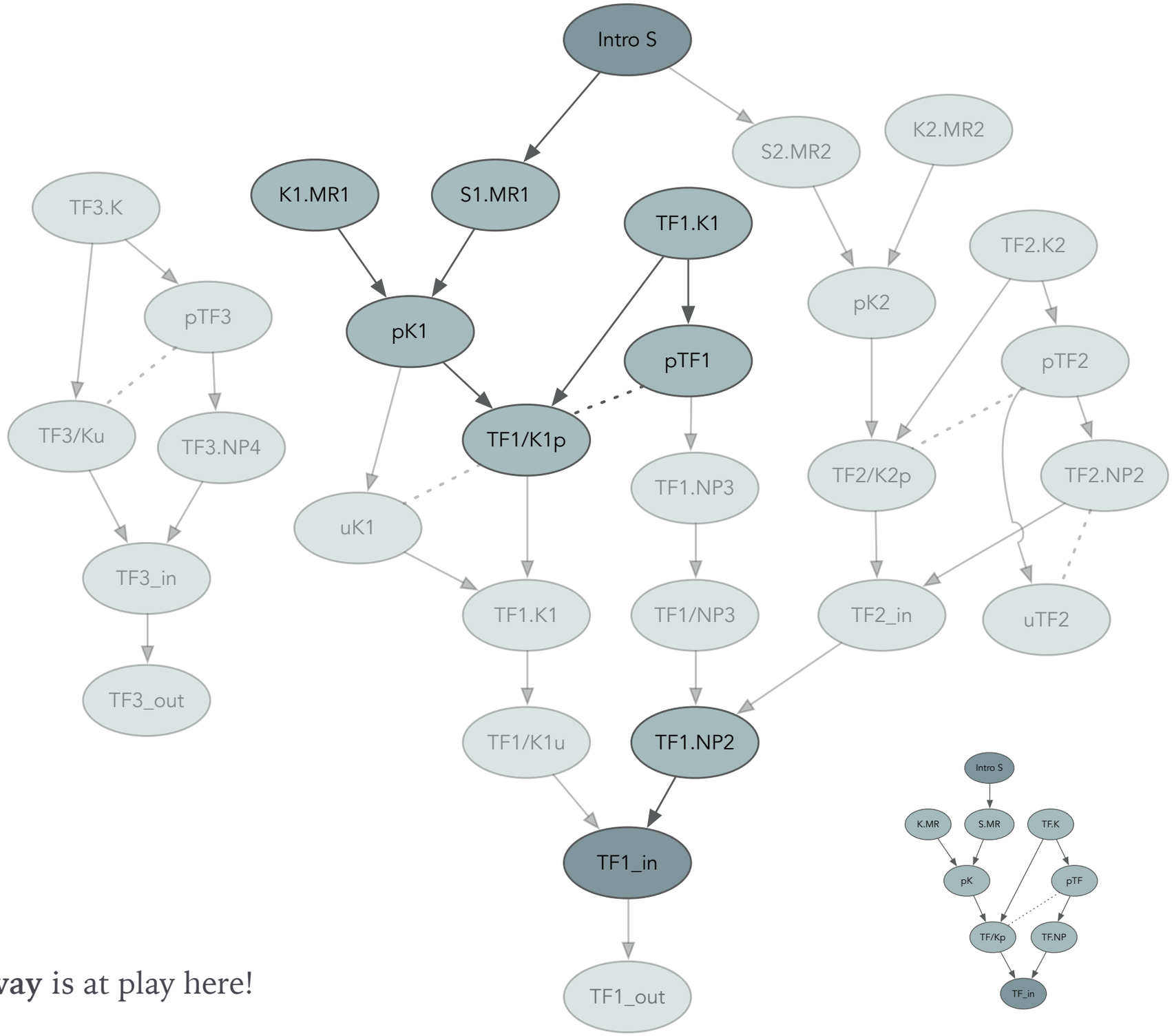


BACK TO OUR EXAMPLE



The **long pathway** is at play here!

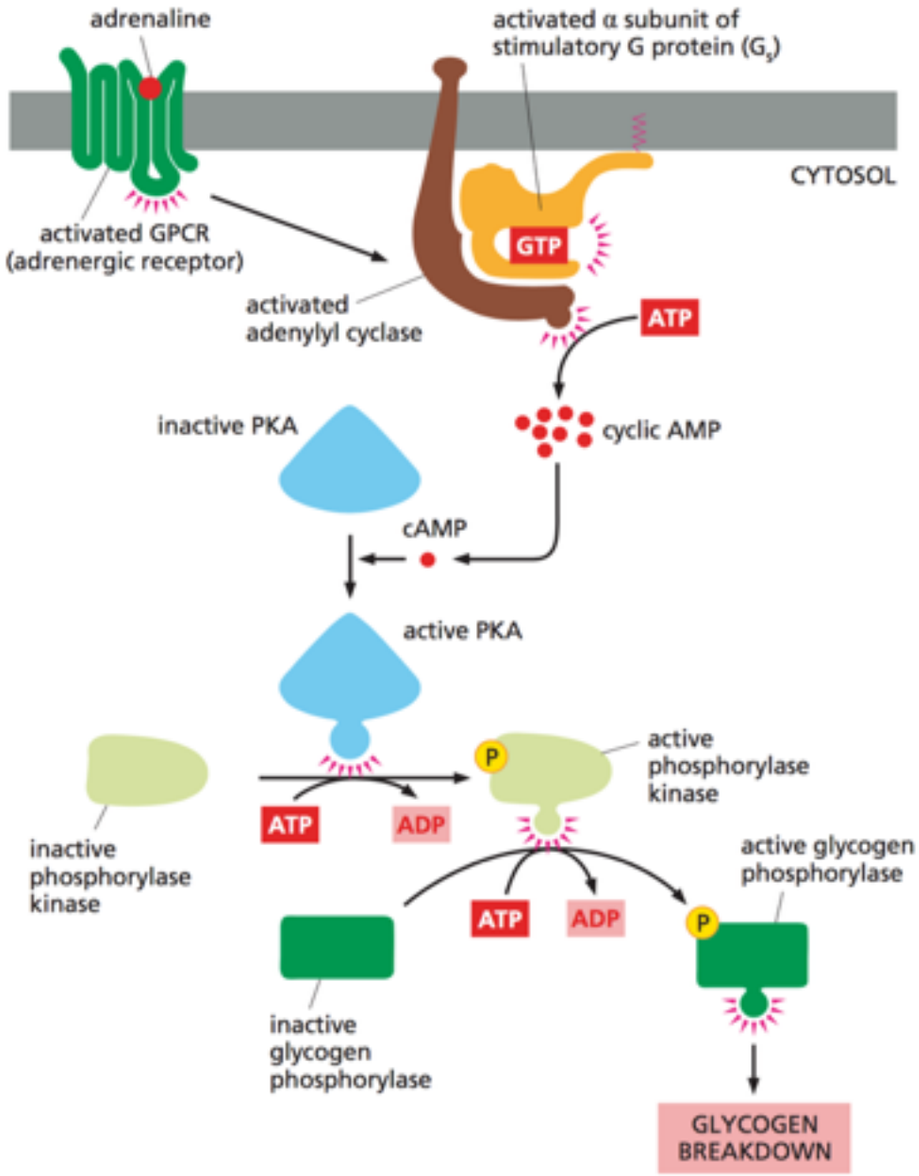
BACK TO OUR EXAMPLE



The **long pathway** is at play here!

CONCLUSION

CONCLUSION



Source: *Essential Cell Biology, 4th edition*

The notion of a pathway is **fundamental** in systems biology. So far, it has been a mostly informal concept though.

Besides, pathways being mostly **curated by hand**, they often reflect the focus and biases of their curators more than they reflect any objective reality.

For these reasons, we find the perspective of uncovering pathways from very large low-level networks of interactions and being able to monitor their activity very exciting!

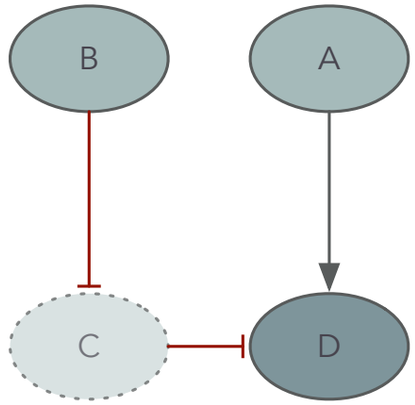
ONGOING AND FUTURE WORK

ONGOING AND FUTURE WORK

Adding inhibition to the picture

One important aspect of biological pathways that is missing in our framework is the notion of **inhibition**.

One difficulty in extracting pathways featuring inhibition is that they would involve events that, per definition, do **not** appear in the simulation trajectory.

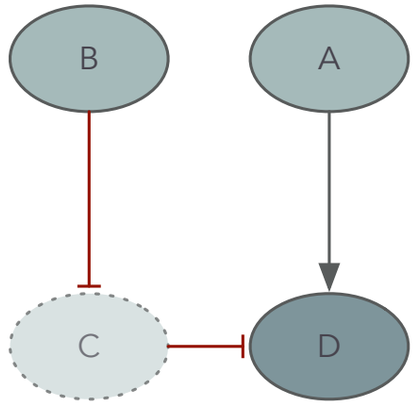


ONGOING AND FUTURE WORK

Adding inhibition to the picture

One important aspect of biological pathways that is missing in our framework is the notion of **inhibition**.

One difficulty in extracting pathways featuring inhibition is that they would involve events that, per definition, do **not** appear in the simulation trajectory.



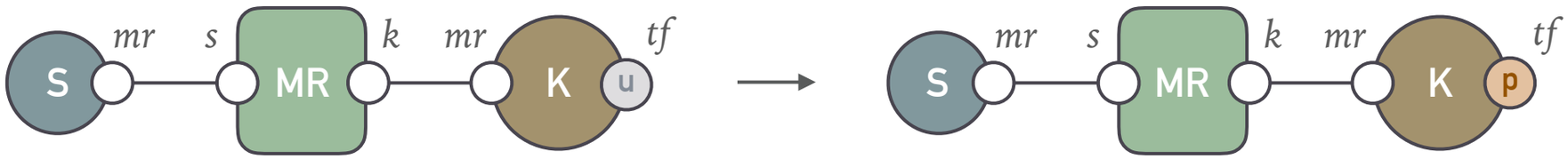
Towards a more solid foundation for causal reasoning

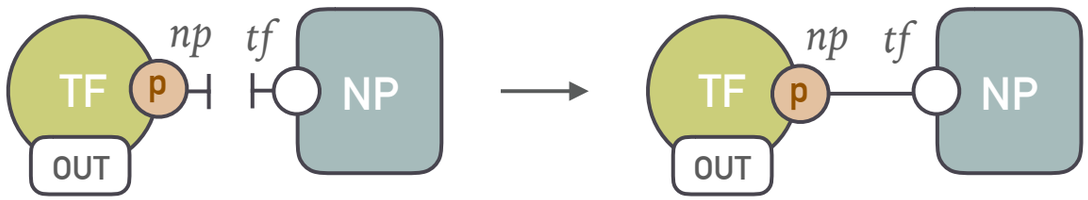
We would like to investigate a **more powerful notion of causality** to serve as a strong foundation for pathway extraction. In particular, B should be considered a cause of D in the example above.

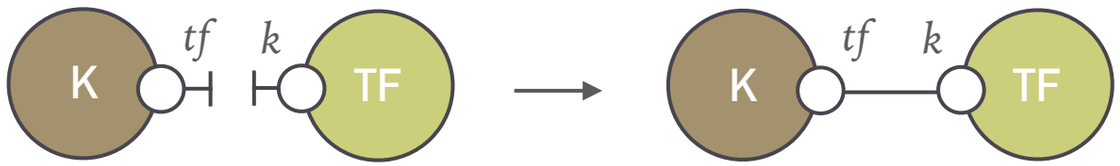
We found an interesting starting point in the work of **Pearl and Halpern**, who developed a theory of causality based on **counterfactual reasoning** that fits quite well the intuition of causality that is prevalent among experimental scientists.

QUESTIONS TIME

BONUS SLIDES


mr
s
k
mr
tf
S
MR
K
u
mr
s
k
mr
tf
S
MR
K
p

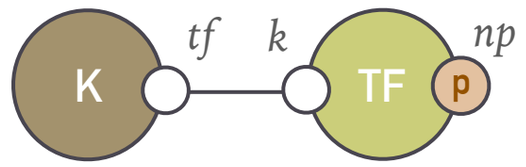

np
tf
TF
p
NP
OUT
np
tf
TF
p
NP
OUT

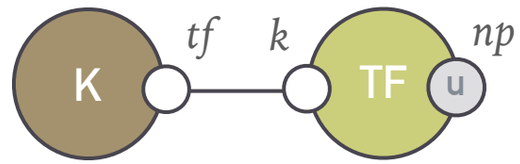

tf
k
K
TF
tf
k
K
TF

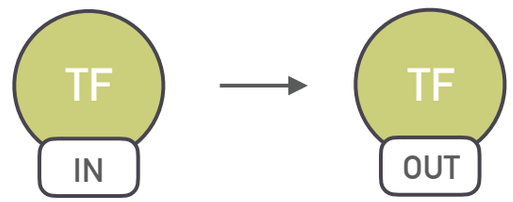

tf
k
K
p
TF
tf
k
K
p
TF


tf
k
K
u
TF
tf
k
K
u
TF


k
np
tf
TF
NP
OUT
k
np
tf
TF
NP
IN


tf
k
K
TF
p
np


tf
k
K
TF
u
np

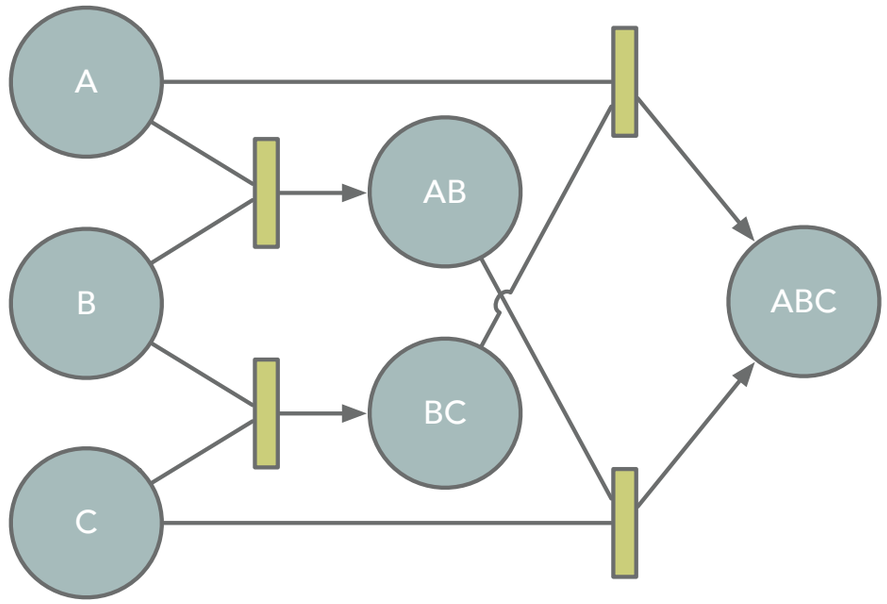

TF
IN
TF
OUT


p

PETRI NETS AND DIFFERENTIAL EQUATIONS

Petri nets have been used for a long time to model chemical systems.

$$
\begin{aligned}
A + B &\xrightarrow{k_A} AB \\
A + BC &\xrightarrow{k_A} ABC \\
B + C &\xrightarrow{k_C} BC \\
AB + C &\xrightarrow{k_C} ABC
\end{aligned}
$$



A petri net can be translated to a set of **differential equations**:

$$
\begin{aligned}
\frac{d[AB]}{dt} &= k_A[A][B] - k_C[AB][C] \\
\frac{d[BC]}{dt} &= k_C[B][C] - k_A[A][BC] \\
\frac{d[ABC]}{dt} &= k_A[BC] + k_C[AB]
\end{aligned}
\qquad
\begin{aligned}
\frac{d[A]}{dt} &= -k_A[B] - k_A[BC] \\
\frac{d[B]}{dt} &= -k_A[A] - k_C[C] \\
\frac{d[C]}{dt} &= -k_C[B] - k_C[AB]
\end{aligned}
$$

EVENT SYSTEMS

An **event system** is given by a set of states Q and a set E of events.

An event e is given by:

A label	A precondition	An effect
$\mathsf{label}(e)$	$\mathsf{pre}(e) \subseteq Q$	$\mathsf{eff}(e) : Q \to Q$

A **trace** is a sequence of events: $\mathsf{eff}(e_1, \cdots, e_n) = \mathsf{eff}(e_n) \circ \cdots \circ \mathsf{eff}(e_1)$

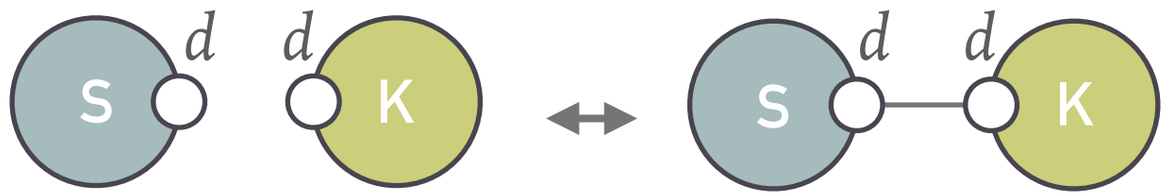
For a context $c \subseteq Q$, we define: $\mathsf{post}_c(t) = \mathsf{eff}(c)(t)$

A trace is **valid** in a context if its events can be triggered in order from any of its states:

$$c \vdash e_1, \cdots, e_n \quad \equiv \quad \forall i \in [1, n],\ \mathsf{post}(e_1, \cdots, e_{i-1}) \subseteq \mathsf{pre}(e_i)$$

A TRANSLATION TO CLF

Here is an example of how a rule translates to CLF




```
bind:
    has-type S s * has-type K k *
    free-site S d * free-site K d
    -o { bond S d K d *
         has-type S s * has-type K k }.
```

GLUCAGON SIGNALING PATHWAY