Datum Logic

A Formal Executable Semantics for Experimental Evidence

Carolyn Talcott
LMT&TCS
January 9, 2016

Joint work with
Vivek Nigam, Robin Donaldson, Merrill Knapp, Tim McCarthy

Executive Summary

- Executable models of signal transduction provide
  - insights into how cells work
  - explanations of observed outcomes
  - a means to understand and predict the effects of perturbations and mutations

- Developing such models from experimental findings is low throughput and requires substantial expertise. Automation can help.

- Logic to the rescue!

- Elements of automation:
  (1) formal representation of experimental findings
  (2) formal representation of biochemical reactions as elements of executable models,
  (3) extraction of formal representations of findings from papers
  (4) inference rules capturing the meaning of the findings
  (5) (meta) inference rules for assembly of executable models

Contributions

- A formal representation of experimental evidence called *datums*.
- A language of logical assertions that formalize the elements of a datum.
- A translation from datum syntax to logical assertions.
- A set of axioms that capture the semantics of datums interpreted as constraints on signal transduction rule patterns.
- Viewing the axioms and assertions as Answer Set Programs, minimal models can be inferred, and reaction rules extracted.

Plan

- Pathway Logic in a nutshell
- Intuitions for rule inference — what a datum tells you
- Formalizing rule Inference
- Hras case study
- Concluding

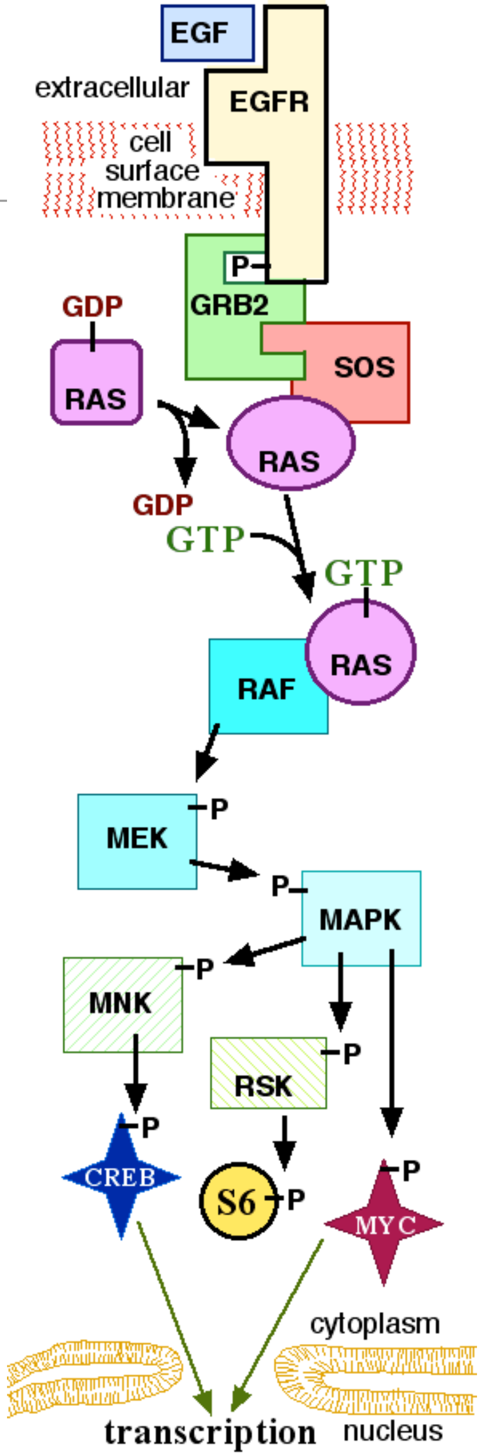
Pathway Logic

Executable models of cellular processes

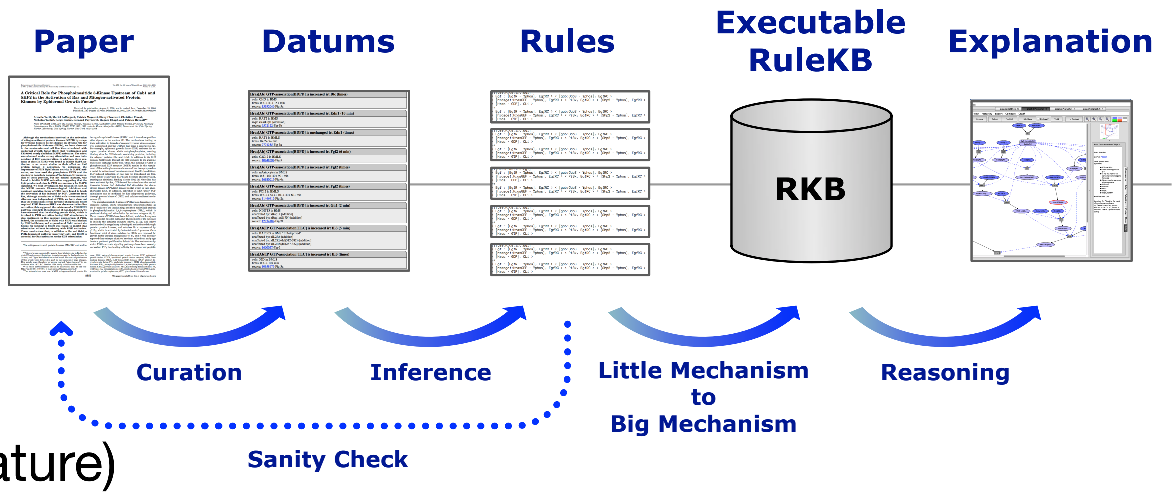
http://pl.csl.sri.com

Pathway Logic (PL) Goals

- Understanding how cells work
- Formal models of biomolecular processes that
  - capture biologist intuitions
  - can be executed and answer questions
- Tools to
  - organize and analyze experimental findings
  - carry out gedanken experiments
  - discover/assemble execution pathways
- New insights into the inner workings of a cell.
- A new kind of review



PL from 1k feet



Paper

Datums

Rules

Executable RuleKB

Explanation

RKB

Curation

Inference

Little Mechanism to Big Mechanism

Reasoning

Sanity Check

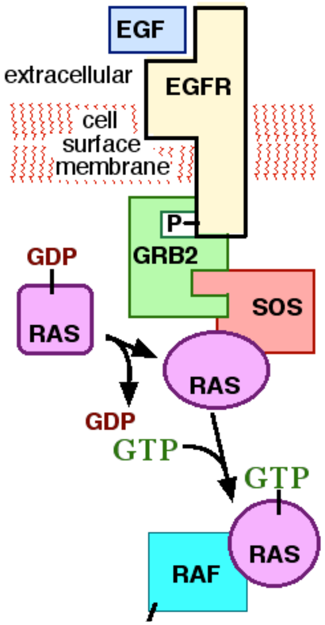
Key components

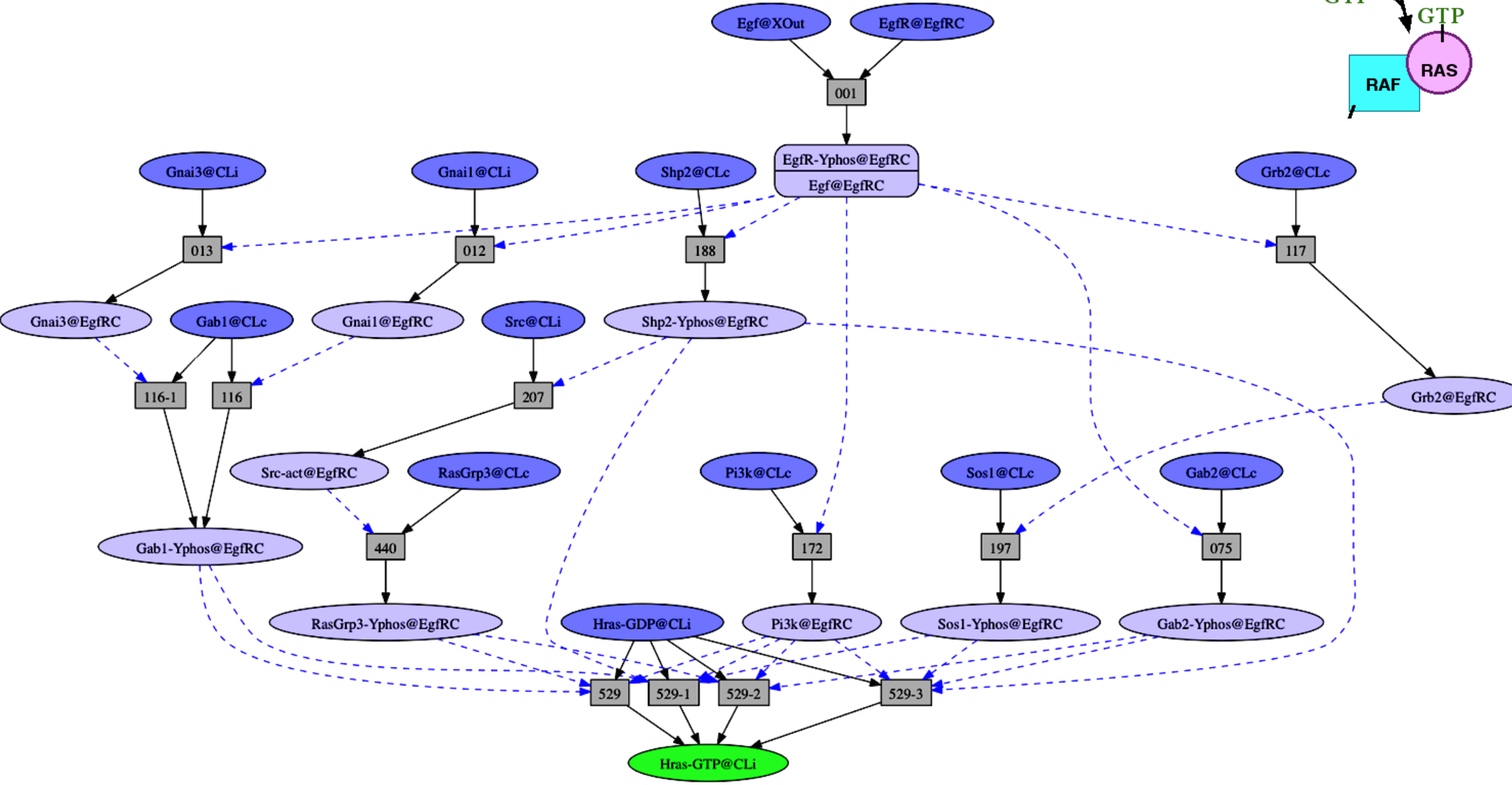
- Representation system
  - controlled vocabulary (signature)
  - datums (formalized experimental results)
  - rewrite rules describing local change/interactions
- Curated datum knowledge base (DKB) and search tool
- Evidence based rule knowledge bases (RKB)
  - STM, Protease, Mycolate, GlycoSTM …
- Executable models
  - generated by specifying initial conditions and constraints
  - queried using formal reasoning techniques
- PLA to visualize and browse and query models and submodels

Example: Hras `activation’

The subnet of the Egf model for activating Hras.

(Curated Gold Standard.)





The Hras Rule formally

rl[529.Hras.irt.Egf]:

< Egf : [EgfR - Yphos], EgfRC > < [gab:GabS - Yphos], EgfRC >

< [hrasgef:HrasGEF - Yphos], EgfRC > < Pi3k, EgfRC > < [Shp2 - Yphos], EgfRC >

< [Hras - GDP], CLi >

=>

< Egf : [EgfR - Yphos], EgfRC > < [gab:GabS - Yphos], EgfRC >

< [hrasgef:HrasGEF - Yphos], EgfRC > < Pi3k, EgfRC > < [Shp2 - Yphos], EgfRC >

< [Hras - GTP], CLi >

*** ~/evidence/Egf-Evidence/Hras.irt.Egf.529.txt

Notation: occurrence : < thing,loc > (the state and location of a biomolecule)

thing : [biomolecule - modifications]. thing : thing

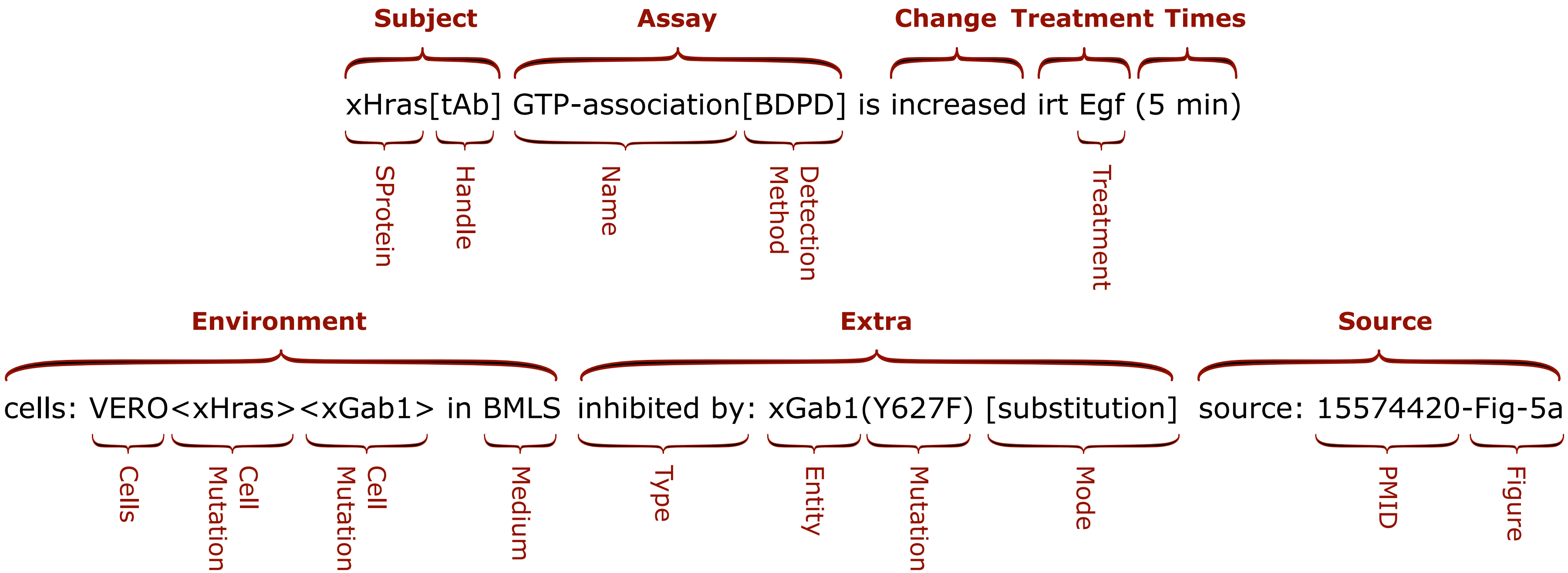
=> : `rewrites to' relation

The rule says that GDP will be exchanged for GTP if, in addition to the EgfR complex (Egf : [EgfR - Yphos]), there is tyrosine phosphorylated Gab1 or Gab2 ([gab:GabS - Yphos]), a tyrosine phosphorylated HrasGef ([hrasgef:HrasGEF - Yphos]), Pi3k, and tyrosine phosphorylated Shp2 all recruited to the EgfR complex (EgfRC).

Where do rules come from?

- They are inferred from experimental findings, curated into a datum KB.
  - datums are available in text (readable) or json (computable)
- The datum below says that the amount of GTP (GTP-association) bound to Hras is increased 5 minutes after addition of Egf (Epidermal Growth Factor) to VERO cells. The extra implies a requirement for Gab1.

The Elements of a Datum



Subject Assay Change Treatment Times

xHras[tAb] GTP-association[BDPD] is increased irt Egf (5 min)

SProtein Handle Name Detection Method Treatment

Environment Extra Source

cells: VERO<xHras><xGab1> in BMLS inhibited by: xGab1(Y627F) [substitution] source: 15574420-Fig-5a

Cells Cell Mutation Cell Mutation Medium Type Entity Mutation Mode PMID Figure

Inferring the Hras rule: the basic pattern

The `first line’ of the previous Hras datum:

xHras[tAb] GTP-association[BDPD] is increased irt Egf (5 min)

can be represented by a rule pattern:

```
EgfTC  C  < [G - gmods act ], Lg > < [Hras - GDP pmods], CLi >
=>
EgfTC  C < [G - gmods  act ], Lg > < [Hras - GTP pmods ], CLi >
```

- <u>EgfTC</u> is the treatment complex formed when Egf binds to the Egf Receptor
- G is a variable ranging over Hras GEFs, representing the general knowledge that exchange of GDP for GTP requires a GEF (Guanine exchange factor).
- gmods, pmods are variables indicating that we don't know the exact state of G or Hras.
- C is a variable standing for possible additional requirements

Inferring that Sos1 is a candidate GEF

The datum

rHras GDP-dissociation[3H-GDP] is increased by xSos1[tAb]IP
cells: none
IPfrom: HEK293 in BMS
source: 15039778-Fig-2c

reports direct GEF action of Sos1 in a test tube,

while the datum

xHras[tAb]IP GTP-association[TLC] is increased itpo xSos1
cells: HEK293 in BMS
source: 10896938-Fig-1c

reports interaction in a live cell.

The combination tells us that Sos1 is a candidate GEF for Hras,
i.e., Sos1 is a possible value for the variable G.

Inferring the requirement for a Gab

The datum

Hras[Ab] GTP-association[BDPD] is increased irt Egf (times)
cells: mEFs in BMLS
times: 0 1++ 2++ 5+ min
partially reqs: Gab1 [KO]
source: 12629518(D)

tells us that Gab1 plays a role.
"Partially” indicates that Gab1 is not the only player of that role.

The extra from the previous Hras datum

inhibited by: xGab1(Y627F) [substitution]

says that some function of Gab1 that relies on Y627 is required.

A plausible conjecture is that phosphorylation on Y627 is required.

Formalizing Datum Logic

Answer Set Programing (ASP) in one slide

An ASP is a collection of clauses of one of three forms:

(1) D.   (2) D :- b1,...,bn.   (3) :- b1,...,bn.

D is a ground fact or a disjunction (the D in DLV)
b is a ground fact or negated ground fact

The meaning of an ASP is a collection of Answer Sets.

Each answer set is a set of ground facts that are minimal subject to making all clauses in the program true.

We use the DLV (DataLog with Disjunction V) engine for finding answer sets by constraint solving.

Datum assertions — predicates for observations

- datum(Dt) -- declares Dt as a datum identifier
- subject(S,Dt)
    - S is the subject of the experiment recorded by datum Dt
- assay(Aname,Aux,Dt)
    - Aname is the assay name,
    - Aux collects assay parameters, possibly none.
    - Examples: modification site, hook, substrate
- treatment(T,Dt)
    - the treatment used in the experiment, if any
- increased(Dt), decreased(Dt), unchanged(Dt)
    - the change observed.
- irt(Dt), itpo(Dt), itao(Dt), by(Dt)
    - the treatment type—determines how the observation is interpreted
- reqs(Q,Dt)
    - entity Q is required for the experimental outcome — usually from extras

Mapping datums to assertions (in DLV)

- Equivalent datums are merged into one super datum.
  - The merged datums have the same subject, assay, treatment/treatment type, and change
  - Extras are joined
- The shared parts of merged datums map directly to assertions
- Extras require some reasoning.
- The mapping function also reports conflicts for examination by an expert.

- Mapping the two Hras datums produces:

```
datum(“Dt1-Dt2").
subject("Hras", "Dt1-Dt2").
assay("GTP-association", none, "Dt1-Dt2").
increased("Dt1-Dt2").
irt("Dt1-Dt2").
treatment("Egf", "Dt1-Dt2").
reqs("Gab1", "Dt1-Dt2").
```

- The actual merged datum for GTP-association assays combines **51** datums from the input datum collection.

Formalizing rule templates

- The template C < X,L > => C < X',L' > is represented by predicates:
  - occBf(X, L) formalizes < X,L >
  - occAf(X', L') formalizes < X',L' >
  - occ(Y, L) specifies that < Y,L > is in C

- The subject and location (X, L) are reasoned about separately, reflecting the kind of information experiments give you.
  - inC(X), inTC(X) capture the 'thing' part of occ(X,L)
  - inO(X), inOp capture the 'thing' part of occBf, occAf

- Information location L often needs separate experiments, common knowledge, or hypothesis by need.
  - location(X,L,Dt): Datum Dt provides evidence that X is at location L

- We restrict rules to represent change in the subject state (reactRule) or change in its location (moveRule) to simplify reasoning.
  - useM(Dt): Dt determines the template, subject, treatment, and change.
  - useA(Dt): Dt provides auxiliary information such as enzymes, or requirements

Sample Clauses: Basics

- The model is a reactRule if the datum describes a reaction and is the main datum

```
reactRule :- react(Dt), useM(Dt).
```

- Interpreting datum assertions

```
inBf(X - mods(X) - GDP) :- irt(Dt), increased(Dt),
        assay(GTP-association, none, Dt), subject(X, Dt), useM(Dt).
inAf(X, mods(X) - GTP) :- irt(Dt), increased(Dt),
        assay(GTP-association, none, Dt), subject(X, Dt), useM(Dt).
in(X) :- treatment(X, Dt), useM(Dt).
```

- Connecting assertions to template variables

```
occBf(X,L(X)) :- inBf(X), reactRule .
occAf(X,L(X)) :- inAf(X), reactRule .
occ(X, L(X))  :-  in(X), not hasLocation(X).   % use a variable if unknown
```

Sample Clauses: Enzyme requirements

- The requirement for a GEF in C

```
 occ(Q, "LGEF") :- not hasLocation(Q), isSelGEF(Q,X,T),reqGEF(X).
% use a variable for if location is unknown
```

- GTP association requires a GEF

```
 reqGEF(X) :- assay("GTP-association",none,Dt),
                   increased(Dt), subject(X,Dt), useM(Dt).
```

- Only one GEF is needed

```
  isSelGEF(Q,X,T) v notIsSelGEF(Q,X,T) :- isGEF(Q,X,T), reactRule.
```

- Connecting the GEF relation to datums

```
isGEF(modBy(Q, [mods(Q)]), X, "tt-itpo") :-
            ttGEF(Q,X,Dt1), itpoGEF(Q,X,Dt2), use(Dt1), use(Dt2).
 ttGEF(Q,X,Dt) :- assay("GTP-association", none, Dt),
             by(Dt), increased(Dt), subject(X, Dt), treatment(Q,Dt), use(Dt).
```

Example answer set and associated rule

```
Answer Set
     reactRule
     occBf(Hras - mods(Hras) - GDP,L(Hras)),
     occAf(Hras - mods(Hras) - GTP,L(Hras))
     occ(Egf:EgfR-Yphos,EgfRC)
     occ(Sos1 – act - mods(Sos1),L(Sos1))
     occ(Gab1 - mods(Gab1),L(Gab1))

Rule
 < [Hras - mods(Hras) GDP], L(Hras) > < Egf : [EgfR - Yphos], EgfRC >
 < [Sos1 - act mods(Sos1)], L(Sos1) > < [Gab1 - mods(Gab1)], L(Gab1) >
   =>
 < [Hras - mods(Hras)  GTP], L(Hras) > < Egf : [EgfR - Yphos], EgfRC >
 < [Sos1 - act mods(Sos1)], L(Sos1) > < [Gab1 - mods(Gab1)], L(Gab1) >
```

Application to Hras network datums

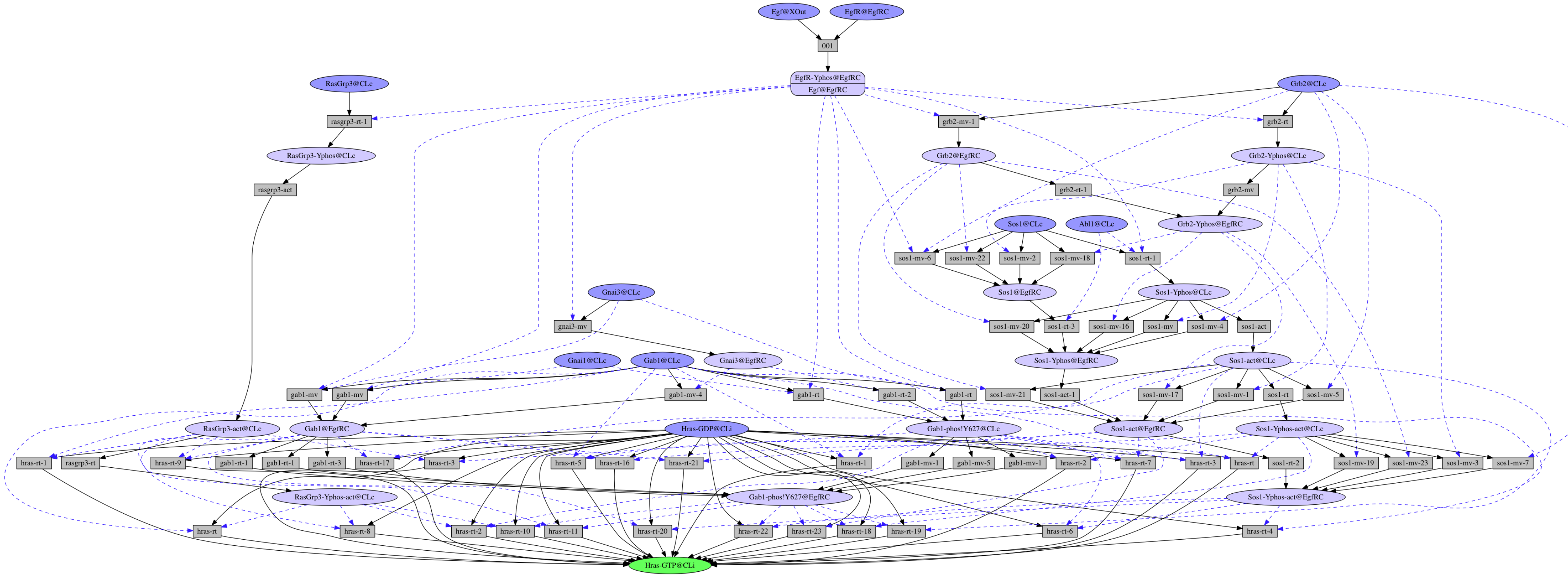
Inferring the Hras irt Egf model from Datums

1. Select datums: evidence files for the rules in our gold standard Hras subnet plus files containing evidence for Hras GEFs.
2. Map datums to DLV assertions
3. Run DLV with assertions + core clauses to get answer sets
4. Translate answer sets to PL rules
5. Assemble PL model – non trivial
   - normalize mismatches such as Yphos => act (the biologists never do exactly the experiments that fit together)
   - reduce combinatorial explosion due to modification variables and combinations of phos, Yphos, phos(Y 627), phos(Y 301), …
   - use PLA for derivation of concrete model from symbolic rules and a concrete initial state (dish)

2-4 are automated (this paper)

1,5 done mostly by hand

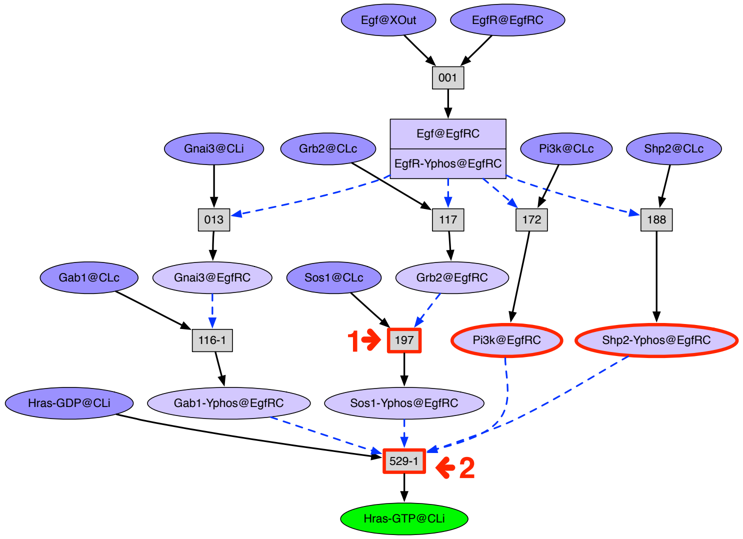
Impression of the inferred network



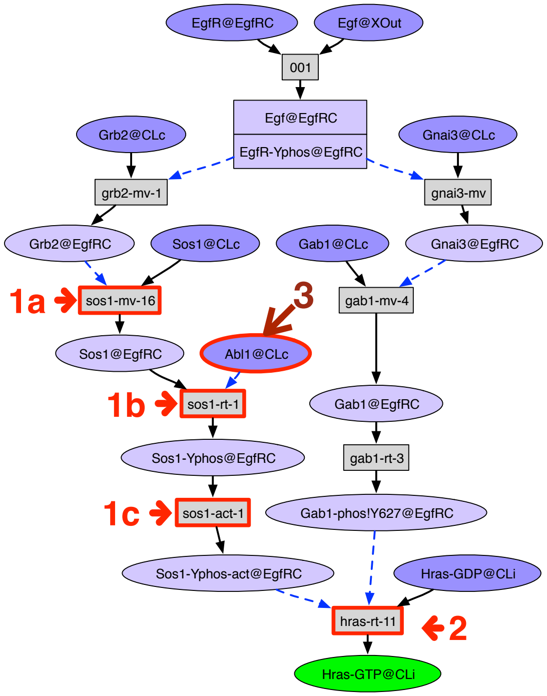
The inferred model recapitulates key properties:

- reachability
- multiple paths to the Hras-GTP goal
- (Sos1,RasGref3) as a double knockout

Pathway in gold standard model (GSM) compared to inferred net (INF)



GSM



INF

Some Differences

- Complexity due to separation of modifiation and move rules **[1]**
- Missing requirements - come from parts of datum not yet interpreted **[2]**

  inhibited by: xPik3r?(mnr)"DN" [addition]

  inhibited by: xShp2(mnr)"CIA" [addition]
- Requirement for Abl1 in inferred rule set – based on single datum **[3]**

Conclusion and Future Work

- We presented an inference system for deriving signal transduction rules from formally represented experimental findings (datums), illustrated by derivation of rules for a model of Hras activation.

- This is a step towards (partial) automation of the process of building models of cellular processes.

- There is much more todo!
  - Capture more from datums
    - reasoning about inhibition and mutation effects
    - what does decrease tell us
    - reasoning about protein/mRNA expression
  - Scaling to larger models developing queries to find relevant datums
  - Automation of datum collection
    - NLP - ongoing DARPA Big Mechanism project
  - Automation of model assembly (from Rule KBs)

Questions ???