conexus®

$$\Sigma \dashv \Delta \dashv \Pi$$

Symbolic Generative AI

Ryan Wisnesky | Conexus AI | ryan@conexus.com

Presented at Texas Data Day 2024

Outline

Claim 1

Data integration, properly (i.e., rigorously, mathematically) understood, is (deterministically and universally) generative.

Claim 2

Many expert systems (collections of logical rules) are data integration systems in disguise.

Therefore: Many expert systems are generative AIs.

Speculation: Generative symbolic AI could be more useful than generative stochastic AI

An aside on applied category theory

- Expert systems became popular in the 80s and data integration has been understood as logic since the 2010s, so why is “symbolic generativity” new?
- From a logic point of view, it is natural to “minimize generativity”.
- But from an algebraic view, it is natural to “maximize generativity”.
- In other words, realizing that data integration is “symbolically generative” requires a viewpoint change from one aspect of the “computational trinity” to another (logic to algebra).

https://en.wikipedia.org/wiki/Applied_category_theory

From wikipedia

Expert Systems

In **artificial intelligence**, an **expert system** is a computer system emulating the decision-making ability of a human expert.[1] Expert systems are designed to solve complex problems by **reasoning** through bodies of knowledge, represented mainly as if-then rules rather than through conventional **procedural code.**[2] The first expert systems were created in the 1970s and then proliferated in the 1980s.[3] Expert systems were among the first truly successful forms of **artificial intelligence (AI)** software.[4][5][6][7][8] An expert system is divided into two subsystems: the **inference engine** and the **knowledge base**. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities.

From wikipedia

Forward Chaining

There are mainly two modes for an inference engine: forward chaining and backward chaining. The different approaches are dictated by whether the inference engine is being driven by the antecedent (left hand side) or the consequent (right hand side) of the rule. In forward chaining an antecedent fires and asserts the consequent. For example, consider the following rule:

$$Rule1: Man(x) \rightarrow Mortal(x)$$

A simple example of forward chaining would be to assert Man (Socrates) to the system and then trigger the inference engine. It would match Rule1 and assert Mortal(Socrates) into the knowledge base.

Rule1: Man(x)→Mortal(x)

Backward chaining is a bit less straightforward. In backward chaining the system looks at possible conclusions and works backward to see if they might be true. So if the system was trying to determine if Mortal (Socrates) is true it would find Rule1 and query the knowledge base to see if Man(Socrates) is true. One of the early innovations of expert systems shells was to integrate inference engines with a user interface. This could be especially powerful with backward chaining. If the system needs to know a particular fact but does not, then it can simply generate an input screen and ask the user if the information is known. So in this example, it could use R1 to ask the user if Socrates was a Man and then use that new information accordingly.

From wikipedia

Backward Chaining

This is why RDF/OWL can perform poorly at data integration

Backward Chaining Dominant in Expert System Practice

Forward chaining only determines a unique model for certain logics. cf “why it is mathematically impossible to use RDF/OWL for data integration”

$$\forall x, \mathit{Actor}(x) \text{ and } \mathit{US\ Governor}\ (x) \rightarrow \mathit{Bodybuilder}(x) \text{ or } \mathit{not}(\mathit{Austrian}(x))$$

Such a formula lacks "repairs" because if you have an actor and US governor who is neither a bodybuilder nor non-Australian, there is no canonical choice of which model the repair should be - do you make them a bodybuilder, or do you make them non-Austrian, or both? Logics with disjunction at best admit "multi-repairs", i.e., databases can be repaired into unique sets of databases that individually satisfy the given theory. It is for this reason that relational data integration technology has traditionally favored the logic of existential Horn clauses over other, more expressive logic: RL posses "certain answers", tuples that must occur in all solutions, but DL does not. The existence of certain answers allows us to meaningfully query the result of a repaired/chased database without having to consider how it was repaired/chased.

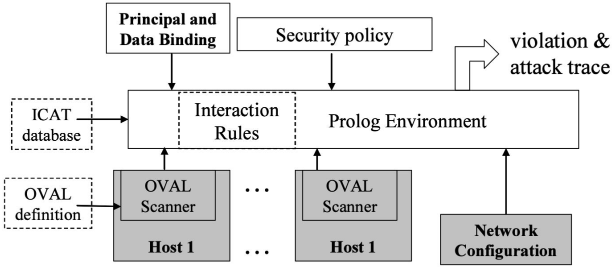


Figure 1: The MulVAL framework

```
execCode(Attacker, Host, Priv) :-
  vulExists(Host, VulID, Program),
  vulProperty(VulID, remoteExploit,
              privEscalation),
  networkService(Host, Program,
                 Protocol, Port, Priv),
  netAccess(Attacker, Host, Protocol, Port),
  malicious(Attacker).
```

From MulVAL: A Logic-based Network Security Analyzer

Many expert systems (collections of logical rules) are data integration systems in disguise.

“proof”: Many expert systems can be expressed in the language of ‘existential horn clauses’, the largest logic that generates unique forward chains. This is also the logic upon which modern data integration is based. In fact, modern data integration is based on this logic because it is the largest logic that generates unique forward chains.

Claim 2

Outline

Claim 1

Data integration, properly (i.e., rigorously, mathematically) understood, is (deterministically and universally) generative.

Claim 2

Many expert systems (collections of logical rules) are data integration systems in disguise.

Therefore: Many expert systems are generative AIs.

Speculation: generative symbolic AI could be more useful than generative stochastic AI

From "Foundations of Data Exchange"

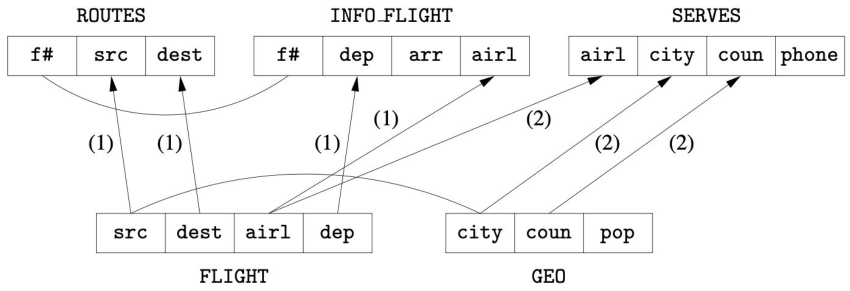


Figure 1.2 Schema mapping: a proper graphical representation

1.1 *A data exchange example*

(1) $\texttt{FLIGHT(src,dest,airl,dep)} \longrightarrow$
$\exists \texttt{f\#}\ \exists \texttt{arr}\ \big(\ \texttt{ROUTES(f\#,src,dest)}$
$\wedge\ \texttt{INFO\_FLIGHT(f\#,dep,arr,airl)}\big)$

(2) $\texttt{FLIGHT(city,dest,airl,dep)} \wedge \texttt{GEO(city,country,popul)}$
$\longrightarrow \exists \texttt{phone}\ \texttt{SERVES(airl,city,country,phone)}$

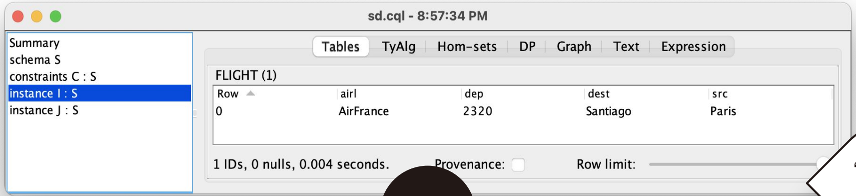
(3) $\texttt{FLIGHT(src,city,airl,dep)} \wedge \texttt{GEO(city,country,popul)}$
$\longrightarrow \exists \texttt{phone}\ \texttt{SERVES(airl,city,country,phone)}$

Figure 1.3 A schema mapping

Claim 1

Data integration, properly (i.e., rigorously, mathematically) understood, is (deterministically and universally) generative.

The "existential horn clauses" shown at left define a unique way to generate missing information from known information.



"Training Data"

+

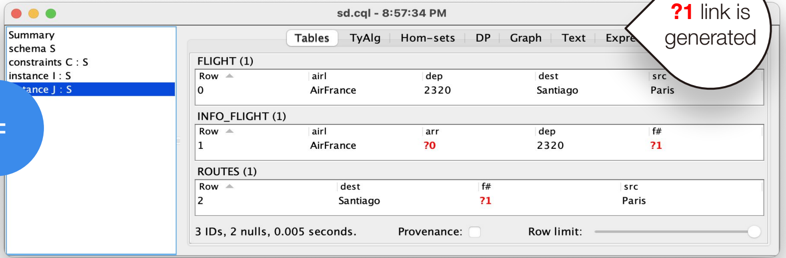
```
constraints C = literal : S {
    forall f:FLIGHT -> exists r:ROUTES i:INFO_FLIGHT
     where f.src=r.src f.dest=r.dest f.dep=i.dep f.airl=i.airl r."f#"=i."f#"

    forall f:FLIGHT g:GEO where f.src=g.city ->
    exists s:SERVES where s.airl=f.airl g.city=s.city g.country=s.country

    forall f:FLIGHT g:GEO where f.dest=g.city ->
    exists s:SERVES where s.airl=f.airl g.city=s.city g.country=s.country
}
```

"prompt"

categoricaldata.net

Claim 1: data integration, properly (i.e., rigorously, mathematically) understood, is (deterministically and universally) generative.

=



?1 link is generated

Example in CQL

Outline

Claim 1

Data integration, properly (i.e., rigorously, mathematically) understood, is (deterministically and universally) generative.

Claim 2

Many expert systems (collections of logical rules) are data integration systems in disguise.

Therefore: Many expert systems are generative AIs.

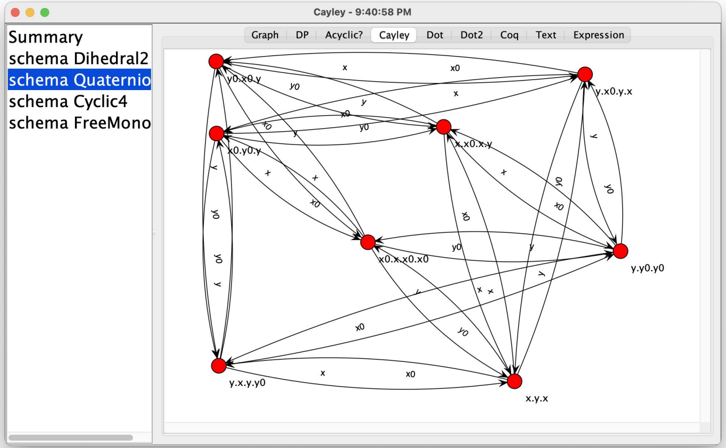
Speculation: generative symbolic AI could be more useful than generative stochastic AI

Conclusion: Ramifications of Generative Symbolic AI

- Generative symbolic AI is deterministic, but not predictable- arbitrarily complex behavior can be encoded using existential horn clauses
- The future is formal - expert systems can be made even more useful thanks to discoveries in categorical algebra
- AI systems will be composed of social-statistical-symbolic components, all generative in their own way
- Rest of talk: examples of symbolic generativity

Bonus claim: Knowledge graph merge and ontology merge are generative by definition

```
schema Quaternions = literal : empty {
	entities
		G
	foreign_keys
		x y x0 y0 : G -> G
	path_equations
		G.x.x0=G
		G.x0.x=G
		G.y.y0=G
		G.y0.y=G
		G.x.x.x.x = G
		G.x.x = G.y.y
		G.y0.x.y = G.x0
}
```



“Presentations by generators and relations”

Generativity

A universal mathematical phenomenon.

patient				
ID	first_name	last_name	birthdate	create_date
937189	john	doe	340465020	1187438212
937190	amrit	kumar	246222505	1187444008
937191	alexandra	grant	121408849	1187445155

visit		
ID	patient_id	visit_date
12378727	937189	1187438212
12378728	937190	1187444008
12378729	937191	1187445155

observation				
ID	visit_id	clinician_id	obs_type	observation
487298329	12378727	562	HR	114
487298330	12378727	562	WT	180
487298331	12378727	562	BP	130/82

+

patient			
ID	fname	lname	dob
25234	alexandra	grant	121408849
25235	vincent	von hoff	409235232
25236	brian	tsai	380665171

prescription			
ID	patient_id	date	details
675345	25234	1639676732	Enalapril Ma...
675346	25234	1639696544	chlorthalid...
675347	25235	1639704522	Lisinopril 5...

observation				
ID	patient_id	clinician_id	obs_type	observation
154298449	25234	132	HR	116
154298450	25234	132	WT	220
154298451	25234	132	BP	132/82

= ?

Generativity in data warehousing

patient

ID	first_name	last_name	birthdate	create_date
937189	john	doe	340465020	1187438212
937190	amrit	kumar	246222505	1187444008
937191	alexandra	grant	121408849	1187445155

visit

ID	patient_id	visit_date
12378727	937189	1187438212
12378728	937190	1187444008
12378729	937191	1187445155

observation

ID	visit_id	clinician_id	obs_type	observation
487298329	12378727	562	HR	114
487298330	12378727	562	WT	180
487298331	12378727	562	BP	130/82

+

patient

ID	fname	lname	dob
25234	alexandra	grant	121408849
25235	vincent	von hoff	409235232
25236	brian	tsai	380665171

prescription

ID	patient_id	date	details
675345	25234	1639676732	Enalapril Ma...
675346	25234	1639696544	chlorthalid...
675347	25235	1639704522	Lisinopril 5...

observation

ID	patient_id	clinician_id	obs_type	observation
154298449	25234	132	HR	116
154298450	25234	132	WT	220
154298451	25234	132	BP	132/82

=

patient

ID1	ID2	fname	lname	dob	create_date
937191	25234	alexandra	grant	121408849	1187445155
	25235	vincent	von hoff	409235232	
	25236	brian	tsai	380665171	
937189	?1	john	doe	340465020	1187438212
937190		amrit	kumar	380665171	1187444008

visit

ID	patient_id	visit_date
12378727	937189	1187438212
12378728	937190	1187444008
12378729	937191	1187445155
?0	937191	1639676732

prescription

ID	patient_id	date	details
675345	25234	1639676732	Enalapril Ma...
675346	25234	1639696544	chlorthalid...
675347	25235	1639704522	Lisinopril 5...

observation

ID	visit_id	clinician_id	obs_type	observation	patient_id
487298329	12378727	562	HR	114	?1
487298330	12378727	562	WT	180	?1
487298331	12378727	562	BP	132/82	?1
154298449	?0	132	HR	116	25234
154298450	?0	132	WT	220	25234
154298451	?0	132	BP	132/82	25234

Generativity in data warehousing

patient				
ID	first_name	last_name	birthdate	create_date
937189	john	doe	340465020	1187438212
937190	amrit	kumar	246222505	1187444008
937191	alexandra	grant	121408849	1187445155
?1	vincent	von hoff	409235232	
?2	brian	tsai	246222505	

visit		
ID	patient_id	visit_date
12378727	937189	1187438212
12378728	937190	1187444008
12378729	937191	1187445155
?0	937191	1639676732

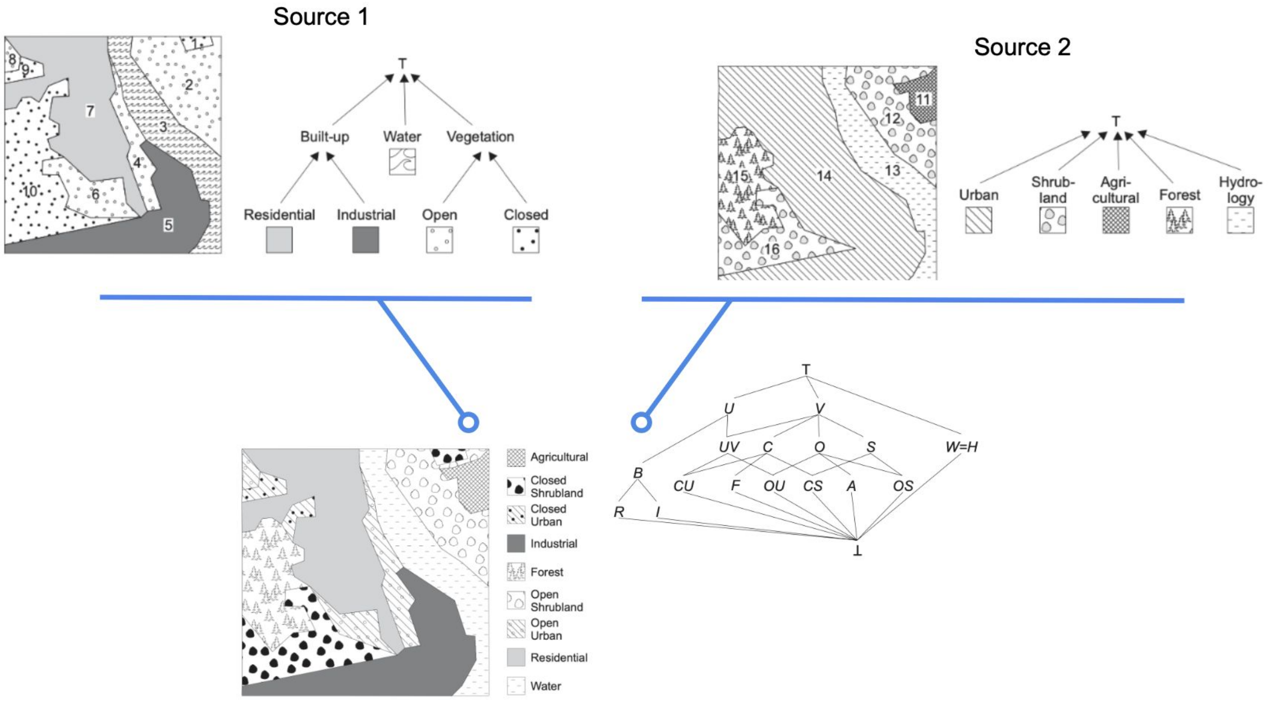
observation				
ID	visit_id	clinician_id	obs_type	observation
487298329	12378727	562	HR	114
487298330	12378727	562	WT	180
487298331	12378727	562	BP	130/82
154298451	?0	132	HR	116
154298450	?0	132	WT	220
154298449	?0	132	BP	132/82

patient			
ID	fname	lname	dob
25234	alexandra	grant	121408849
25235	vincent	von hoff	409235232
25236	brian	tsai	380665171
?1	john	doe	340465020
?2	amrit	kumar	246222505

prescription			
ID	patient_id	date	details
675345	25234	1639676732	Enalapril Ma…
675346	25234	1639696544	chlorthalid…
675347	25235	1639704522	Lisinopril 5…

observation				
ID	patient_id	clinician_id	obs_type	observation
154298449	25234	132	HR	116
154298450	25234	132	WT	220
154298451	25234	132	BP	132/82
487298329	?1	562	HR	114
487298330	?1	562	WT	180
487298331	?1	562	BP	130/82

Generative Warehousing is Bidirectional Exchange



Merged output

From “An algebraic approach to automated information fusion”

Satellite Image Analysis / Ontology merge

Thank you

Ryan Wisnesky
ryan@conexus.com
http://wisnesky.net