Spatio-thematic data in a DL-based prototypical GIS

DoCoMo Euro Labs 16.2.2005

Michael Wessel

Software Systems Group (STS)
Hamburg University of Science and Technology (TUHH)
Hamburg-Harburg, Germany
mi.wessel@tuhh.de

Overview of Talk

- Vision of a “deductive GIS”

Overview of Talk

- Vision of a "deductive GIS"
  - Conventional GIS
  - Scenario: digital city maps ("DISK")
  - Desirable reasoning services
  - Ontology-based querying of the DISK

Overview of Talk

- Vision of a "deductive GIS"
- A Pragmatic Prototype (DLMAPS)

Overview of Talk

- Vision of a “deductive GIS”
- A Pragmatic Prototype (DLMAPS)
  - Bottom-up approach
  - Idea for architecture
  - Short Description Logic (DL) reminder
  - RacerPro: $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$
  - Representing and querying the DISK

Overview of Talk

- Vision of a “deductive GIS”
- A Pragmatic Prototype (DLMAPS)
- DLMAPS functionality integrated into RacerPro

Overview of Talk

- Vision of a “deductive GIS”
- A Pragmatic Prototype (DLMAPS)
- DLMAPS functionality integrated into RacerPro
  - DLMAPS query language $\longrightarrow$ nRQL
  - Description of nRQL
  - Hybrid representations and queries
  - The (experimental) RCC substrate

Overview of Talk

- Vision of a “deductive GIS”
- A Pragmatic Prototype (DLMAPS)
- DLMAPS functionality integrated into RacerPro
- Conclusion

Overview of Talk

- Vision of a “deductive GIS”
- A Pragmatic Prototype (DLMAPS)
- DLMAPS functionality integrated into RacerPro
- Conclusion

Conventional GIS

- "A GIS is a computer system capable of capturing, storing, analyzing, and displaying geographically referenced information; that is, data identified according to location."
- . . . a few dozen more
- 4 main aspects: capturing, representation, analysis, presentation of spatial data

Conventional GIS

- “A GIS is a computer system capable of capturing, storing, analyzing, and displaying geographically referenced information; that is, data identified according to location.”
- …a few dozen more
- 4 main aspects: capturing, representation, analysis, presentation of spatial data
- “Digital Maps” with enhanced functionality for analyzing geo referenced data (spatial data $\longrightarrow$ spatial information)

Conventional GIS

- “A GIS is a computer system capable of capturing, storing, analyzing, and displaying geographically referenced information; that is, data identified according to location.”
- … a few dozen more
- 4 main aspects: capturing, representation, analysis, presentation of spatial data
- “Digital Maps” with enhanced functionality for analyzing geo referenced data (spatial data → spatial information)
- Applications: decision support, logistics, urban planning, selection/matchmaking of location based services, …

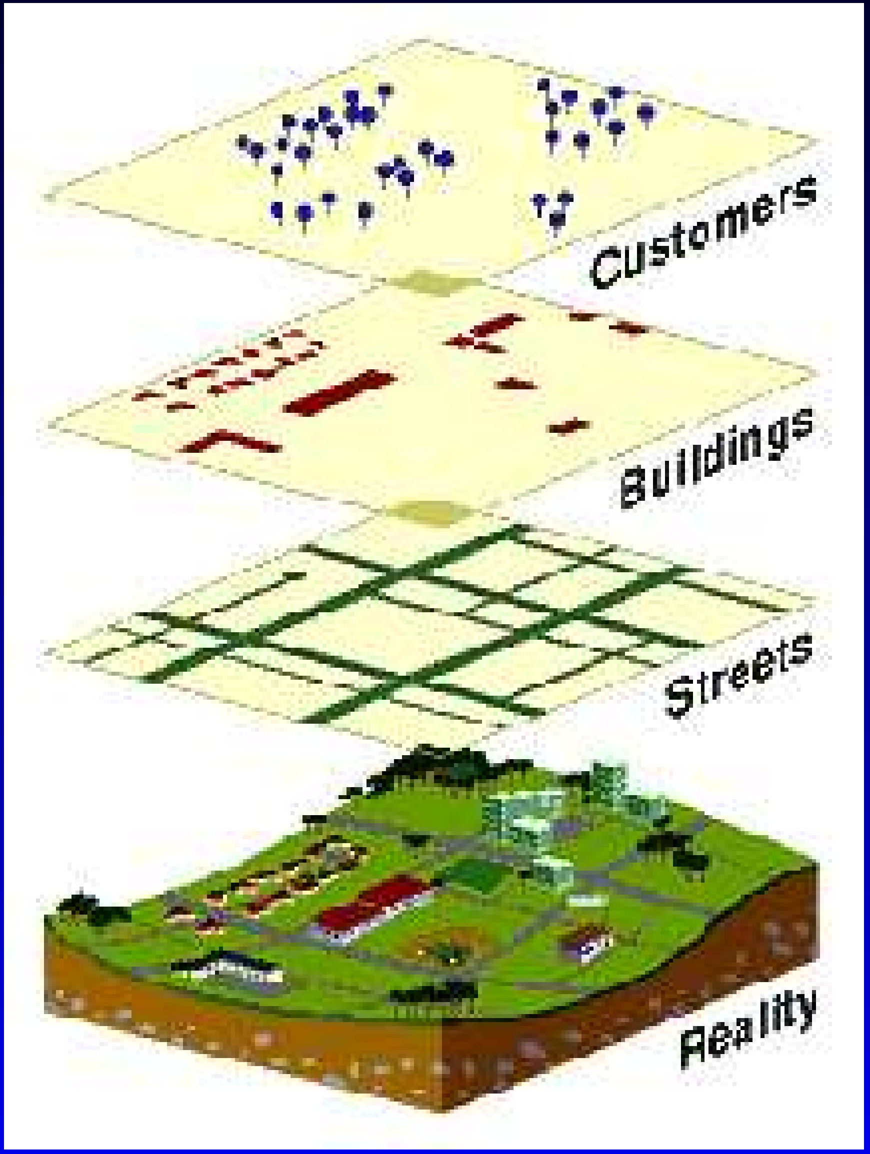
Conventional GIS

- "A GIS is a computer system capable of capturing, storing, analyzing, and displaying geographically referenced information; that is, data identified according to location."
- ...a few dozen more
- 4 main aspects: <u>capturing, representation, analysis, presentation of spatial data</u>
- "Digital Maps" with enhanced functionality for analyzing geo referenced data (spatial data $\longrightarrow$ spatial <u>information</u>)
- Applications: decision support, logistics, urban planning, selection/matchmaking of location based services, ...

Conventional GIS (2)

- Representation (storage): spatial database (offering spatial datatypes)
  - vector or raster data (or both)
  - often organized by subject in “layers”
  - spatial index structures (e.g., R^{+}-Trees)

Conventional GIS (2)



Conventional GIS (2)

- Representation (storage): spatial database (offering spatial datatypes)
  - vector or raster data (or both)
  - often organized by subject in “layers”
  - spatial index structures (e.g., R^+-Trees)
- Analysis: Spatial SQL and application programs: data $\longrightarrow$ information
  - spatial operators (`...WHERE`)
  - computation of spatial relationships
  - only efficiently implementable with spatial index structures

Conventional GIS (2)

- Representation (storage): spatial database (offering spatial datatypes)
    - vector or raster data (or both)
    - often organized by subject in “layers”
    - spatial index structures (e.g., R^+-Trees)
- Analysis: Spatial SQL and application programs: data $\longrightarrow$ information
    - spatial operators (`...WHERE`)
    - computation of spatial relationships
    - only efficiently implementable with spatial index structures
- Presentation: (preferably interactive) digital maps

Conventional GIS (2)

- Representation (storage): spatial database (offering spatial datatypes)
  - vector or raster data (or both)
  - often organized by subject in “layers”
  - spatial index structures (e.g., R^+-Trees)
- Analysis: Spatial SQL and application programs: data $\longrightarrow$ information
  - spatial operators (`...WHERE`)
  - computation of spatial relationships
  - only efficiently implementable with spatial index structures
- Presentation: (preferably interactive) digital maps

The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")

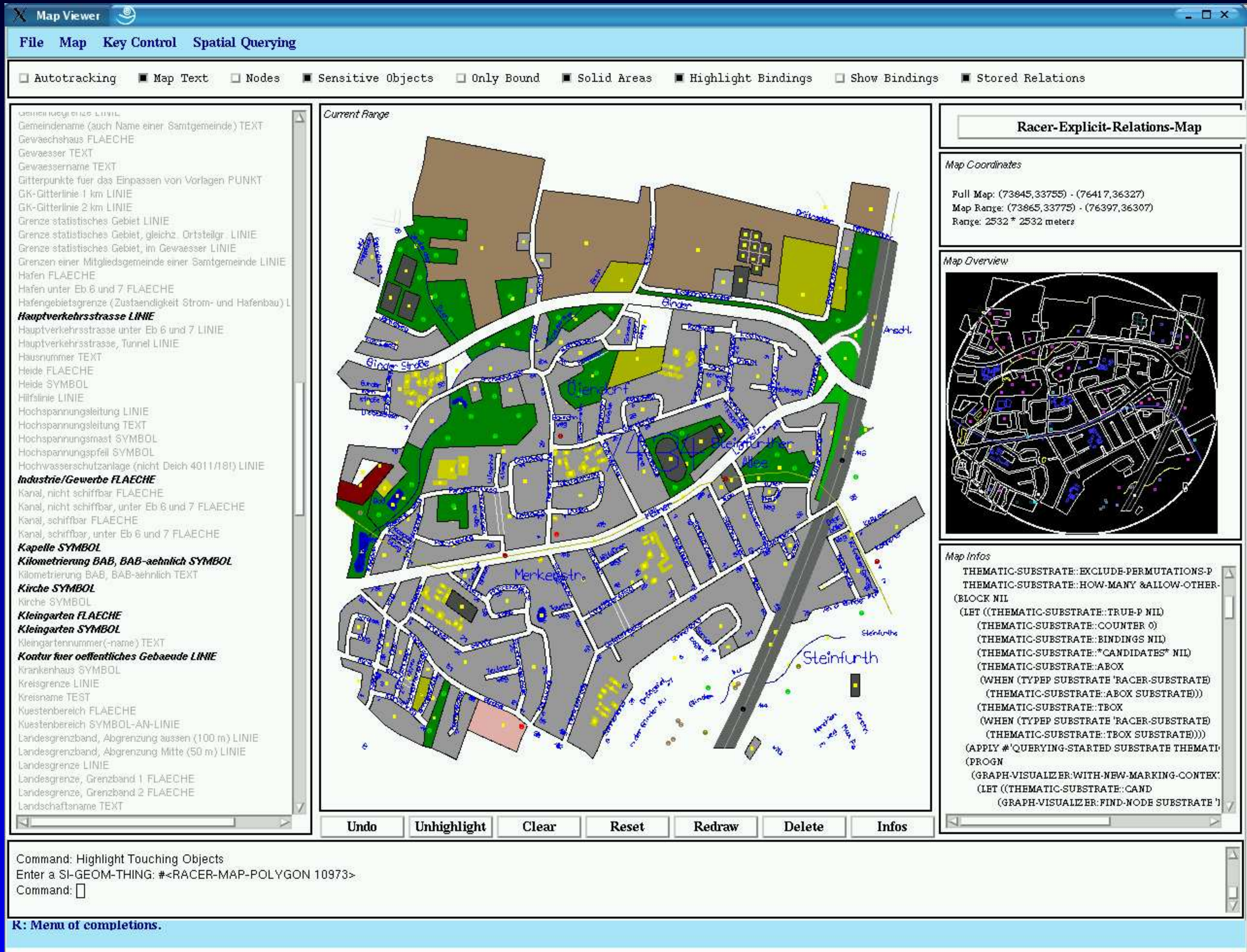
The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")
- Two digital © vector maps in SiCAD SQD format

The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")
- Two digital © vector maps in SiCAD SQD format
    - Map 1: 2694 geo-objects, 361 primary

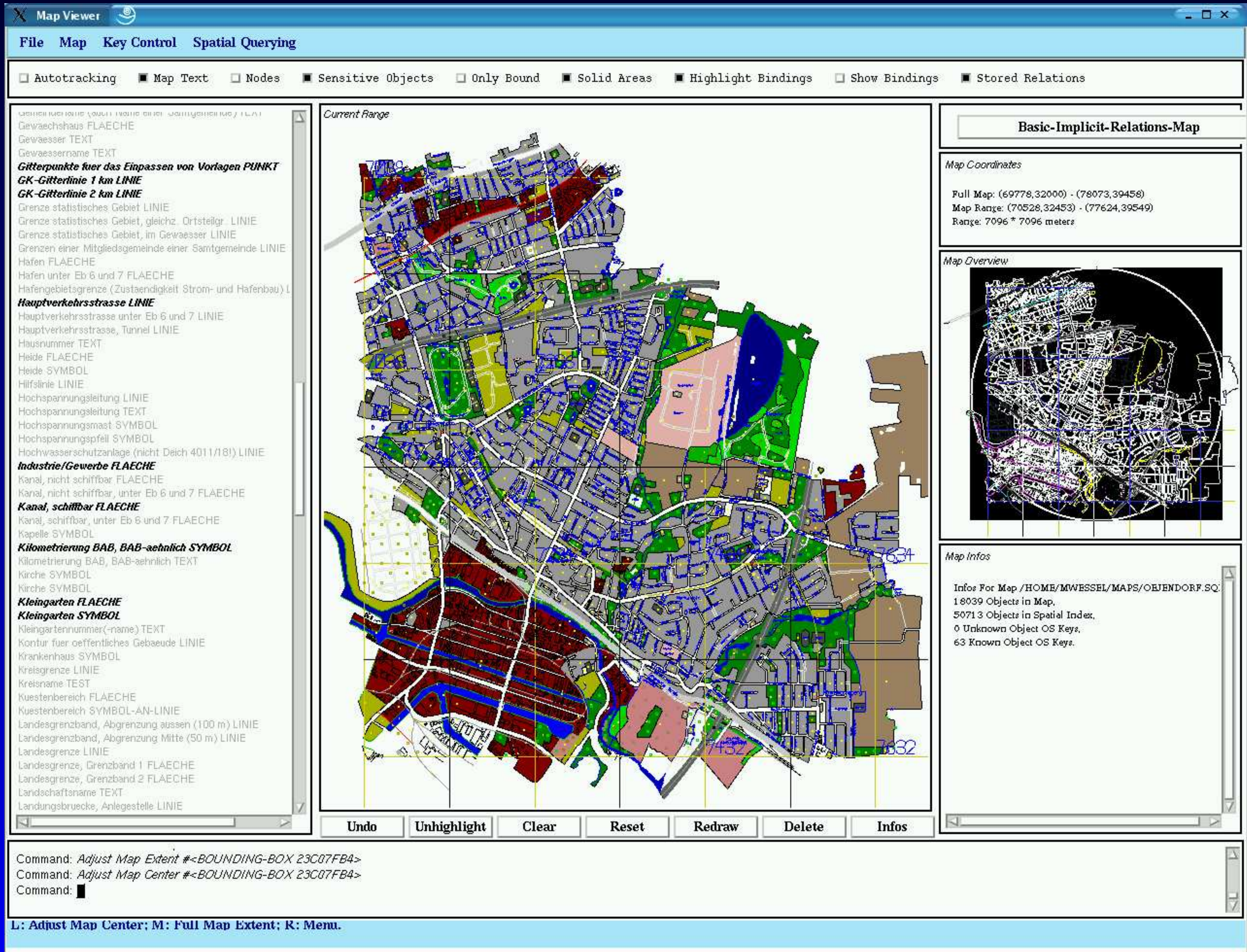
The DISK (Digital City Map)



The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")
- Two digital © vector maps in SiCAD SQD format
  - Map 2: 18.039 geo-objects, 5.418 primary

The DISK (Digital City Map)



The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")
- Two digital © vector maps in SiCAD SQD format
- Objects are "classified" according to a fixed list of categories ("Objektschlüssel-Katalog"):

The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")
- Two digital © vector maps in SiCAD SQD format
- Objects are "classified" according to a fixed list of categories ("Objektschlüssel-Katalog"):
  - 5164 ⇒ $\boxed{\text{lake, navigable}}$, 4128 ⇒ $\boxed{\text{meadow}}$, 2224 ⇒ $\boxed{\text{park}}$, 2119 ⇒ $\boxed{\text{living area,}}$ …

The DISK (Digital City Map)

- Data from authorities of Hamburg (“Amt für Geoinformation und Vermessung”)
- Two digital © vector maps in SiCAD SQD format
- Objects are “classified” according to a fixed list of categories (“Objektschlüssel-Katalog”):
  - 5164 $\Rightarrow$ $\boxed{\text{lake, navigable}}$, 4128 $\Rightarrow$ $\boxed{\text{meadow}}$, 2224 $\Rightarrow$ $\boxed{\text{park}}$, 2119 $\Rightarrow$ $\boxed{\text{living area,}}$ …
  - Taxonomic relationships (“is-a”) implicitly present, but not explicitly modeled $\Rightarrow$ needs remodeling

The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")
- Two digital © vector maps in SiCAD SQD format
- Objects are "classified" according to a fixed list of categories ("Objektschlüssel-Katalog"):
  - 5164 ⇒ lake, navigable, 4128 ⇒ meadow, 2224 ⇒ park, 2119 ⇒ living area, …
  - Taxonomic relationships ("is-a") implicitly present, but not explicitly modeled ⇒ needs remodeling
  - Very specific categories, not common sense!

The DISK (Digital City Map)

- Data from authorities of Hamburg ("Amt für Geoinformation und Vermessung")
- Two digital © vector maps in SiCAD SQD format
- Objects are "classified" according to a fixed list of categories ("Objektschlüssel-Katalog"):
  - 5164 ⇒ $\boxed{\text{lake, navigable}}$, 4128 ⇒ $\boxed{\text{meadow}}$, 2224 ⇒ $\boxed{\text{park}}$, 2119 ⇒ $\boxed{\text{living area,}}$ …
  - Taxonomic relationships ("is-a") implicitly present, but not explicitly modeled ⇒ needs remodeling
  - Very specific categories, not common sense!

Aspects of Geographic Data

- The geographic world is rich

Aspects of Geographic Data

- The geographic world is rich
  - Thematic Aspects: Geographic Category (Thematic Type, Subject, Semantics)

Aspects of Geographic Data

- The geographic world is rich
  - Thematic Aspects: Geographic Category (Thematic Type, Subject, Semantics)
  - Spatial Aspects: Length, Area, Shape, Spatial Relationships, . . .

Aspects of Geographic Data

- The geographic world is rich
  - Thematic Aspects: Geographic Category (Thematic Type, Subject, Semantics)
  - Spatial Aspects: Length, Area, Shape, Spatial Relationships, . . .
  - Feature: GIS-ready abstraction of a real-world geographic phenomenon

Aspects of Geographic Data

- The geographic world is rich
  - Thematic Aspects: Geographic Category (Thematic Type, Subject, Semantics)
  - Spatial Aspects: Length, Area, Shape, Spatial Relationships, . . .
  - Feature: GIS-ready abstraction of a real-world geographic phenomenon
- Data model for geo-data needed
- Various representations possible (purely qualitative, topological, geometric, network, dynamic, . . .)

Aspects of Geographic Data

- The geographic world is rich
  - Thematic Aspects: Geographic Category (Thematic Type, Subject, Semantics)
  - Spatial Aspects: Length, Area, Shape, Spatial Relationships, . . .
  - Feature: GIS-ready abstraction of a real-world geographic phenomenon
- Data model for geo-data needed
- Various representations possible (purely qualitative, topological, geometric, network, dynamic, . . .)
- Representation should be tailored for anticipated tasks

Aspects of Geographic Data

- The geographic world is rich
  - Thematic Aspects: Geographic Category (Thematic Type, Subject, Semantics)
  - Spatial Aspects: Length, Area, Shape, Spatial Relationships, . . .
  - Feature: GIS-ready abstraction of a real-world geographic phenomenon
- Data model for geo-data needed
- Various representations possible (purely qualitative, topological, geometric, network, dynamic, . . .)
- Representation should be tailored for anticipated tasks

Aspects of Geographic Data (2)

- Space has specific properties

Aspects of Geographic Data (2)

- Space has specific properties
- Instances of spatial data types (polygons etc.) automatically preserve many important spatial aspects of the represented geo object (for example, a polygon represents both shape and area) $\longrightarrow$ data is concrete

Aspects of Geographic Data (2)

- Space has specific properties
- Instances of spatial data types (polygons etc.) <u>automatically preserve</u> many important spatial aspects of the represented geo object (for example, a polygon represents both shape and area) $\longrightarrow$ data is concrete
- A map intrinsically represents (qualitative) spatial relationships $\longrightarrow$ rich representation

Aspects of Geographic Data (2)

- Space has specific properties
- Instances of spatial data types (polygons etc.) automatically preserve many important spatial aspects of the represented geo object (for example, a polygon represents both shape and area) → data is concrete
- A map intrinsically represents (qualitative) spatial relationships → rich representation
- However, symbolic descriptions are needed to describe spatial concepts in an ontology

Aspects of Geographic Data (2)

- Space has specific properties
- Instances of spatial data types (polygons etc.) automatically preserve many important spatial aspects of the represented geo object (for example, a polygon represents both shape and area) $\longrightarrow$ data is concrete
- A map intrinsically represents (qualitative) spatial relationships $\longrightarrow$ rich representation
- However, symbolic descriptions are needed to describe spatial concepts in an ontology
- Spatially underspecified descriptions: “A house at a lake”

Aspects of Geographic Data (2)

- Space has specific properties
- Instances of spatial data types (polygons etc.) automatically preserve many important spatial aspects of the represented geo object (for example, a polygon represents both shape and area) $\longrightarrow$ data is concrete
- A map intrinsically represents (qualitative) spatial relationships $\longrightarrow$ rich representation
- However, symbolic descriptions are needed to describe spatial concepts in an ontology
- Spatially underspecified descriptions: “A house at a lake”
- Reasoning with spatial concepts?

Spatial and Thematic Concepts

- Thematically underspecified information

Spatial and Thematic Concepts

- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park

Spatial and Thematic Concepts

- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic

Spatial and Thematic Concepts

- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic
- Spatially underspecified information

Spatial and Thematic Concepts

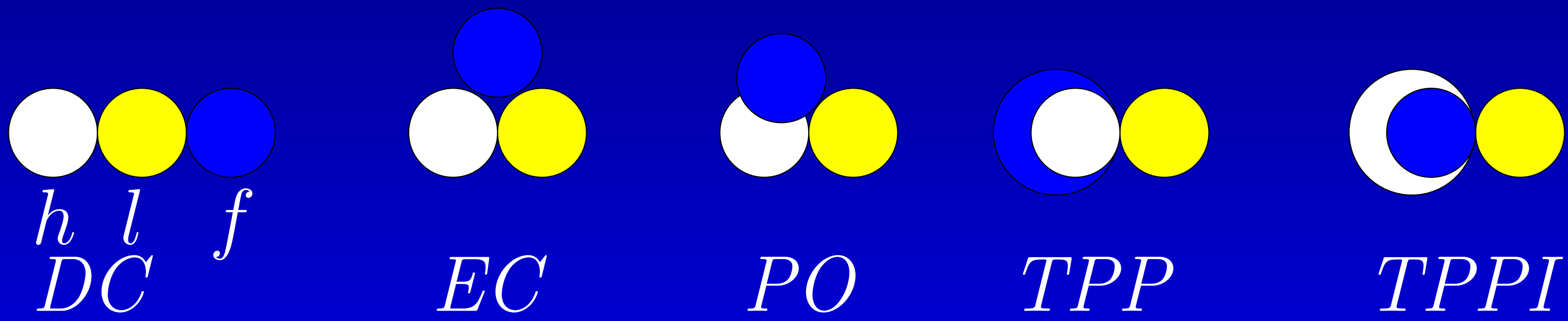
- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic
- Spatially underspecified information
  - There is ($\exists$) a house at a lake at a forest

Spatial and Thematic Concepts

- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic
- Spatially underspecified information
  - <u>There is (∃)</u> a house at a <u>lake at a forest</u>
  - What can you say about relationship between house and forest?

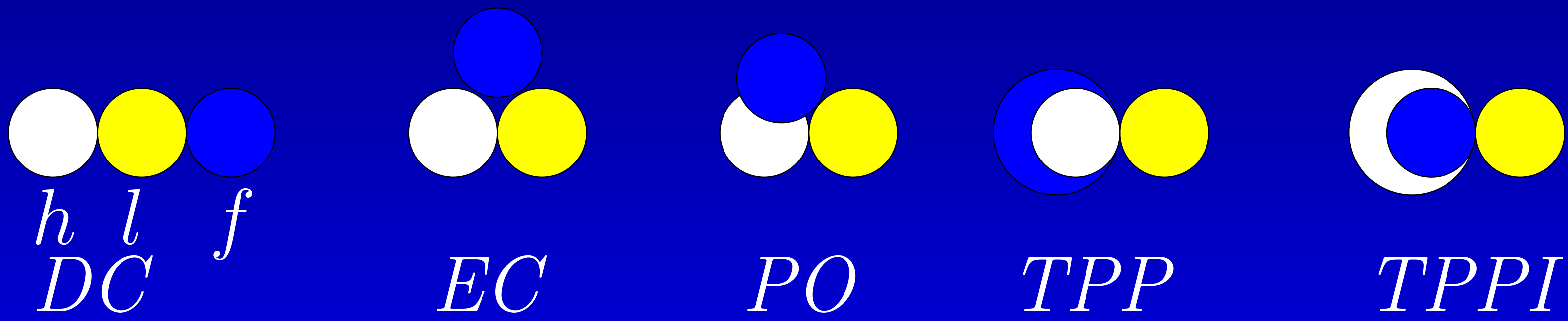
Spatial and Thematic Concepts

- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic
- Spatially underspecified information
  - There is ($\exists$) a house at a lake at a forest
  - What can you say about relationship between house and forest?



Spatial and Thematic Concepts

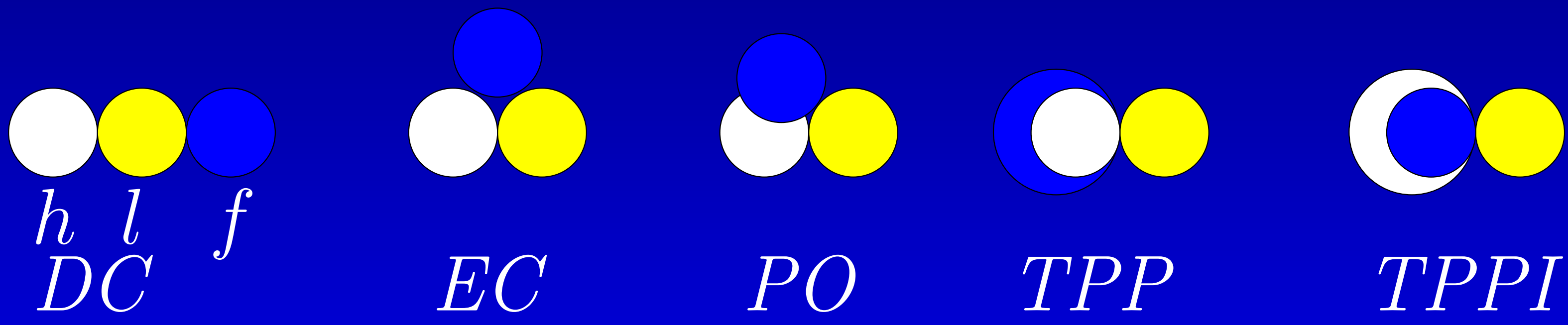
- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic
- Spatially underspecified information
  - There is (∃) a house at a lake at a forest
  - What can you say about relationship between house and forest?



- $(DC \vee EC \vee PO \vee TPP \vee TPPI)(h, f)$

Spatial and Thematic Concepts

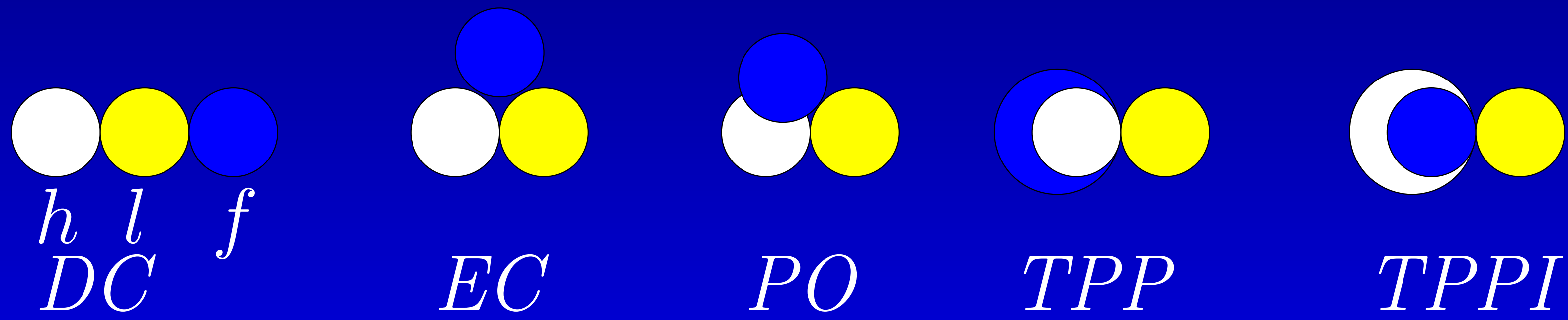
- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic
- Spatially underspecified information
  - There is ($\exists$) a house at a lake at a forest
  - What can you say about relationship between house and forest?



  - $(DC \vee EC \vee TPP \vee TPPI)(h, f)$ with background knowledge in ontology

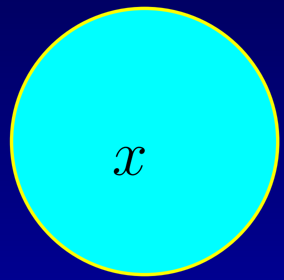
Spatial and Thematic Concepts

- Thematically underspecified information
  - $Polygon_{123}$ is a lake or a park
  - not a problem in logic
- Spatially underspecified information
  - There is ($\exists$) a house at a lake at a forest
  - What can you say about relationship between house and forest?



  - $(DC \vee EC \vee TPP \vee TPPI)(h, f)$ with background knowledge in ontology

Spatial Incompleteness (Lemon)



$$circle_{r=10}(x) \wedge circle_{r=10}(y_1) \wedge \ldots \wedge circle_{r=10}(y_7) \wedge$$

$$EC(x, y_1) \wedge \ldots \wedge EC(x, y_7) \wedge$$

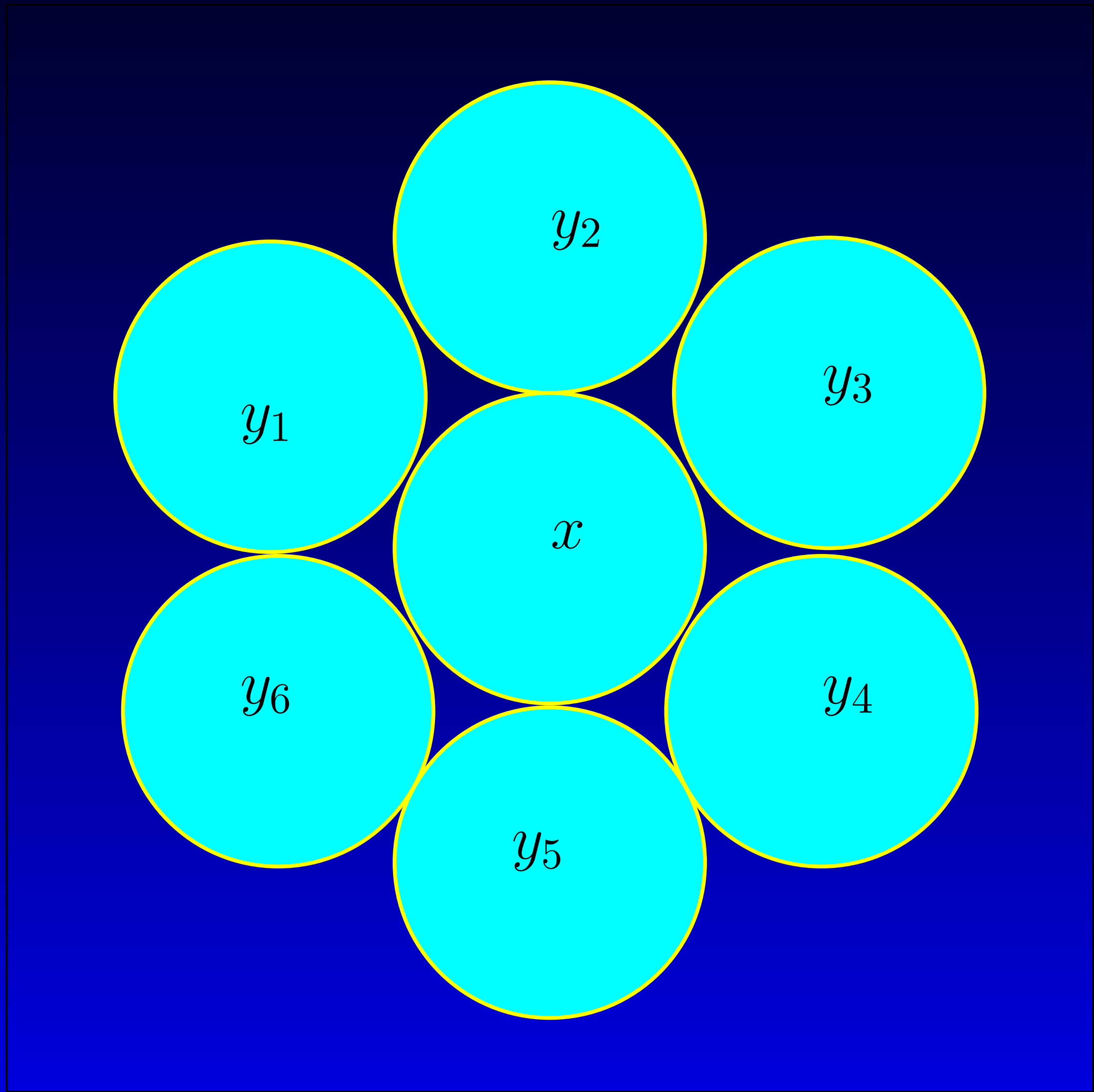
$$(EC(y_1, y_2) \wedge EC(y_2, y_3)) \wedge$$

$$(EC(y_2, y_3) \wedge EC(y_3, y_4)) \wedge \ldots \wedge$$

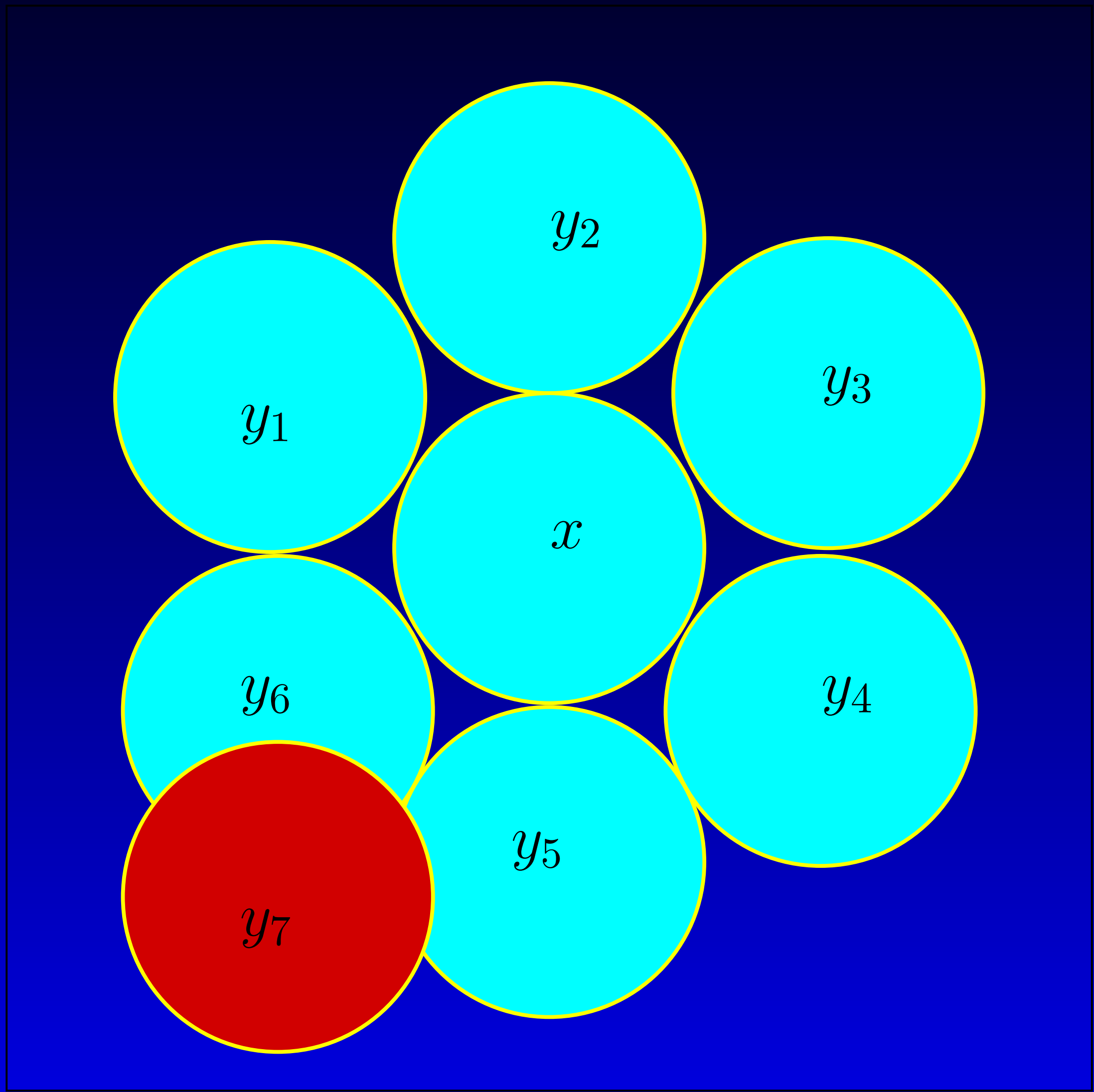
$$(EC(y_6, y_7) \wedge EC(y_7, y_1))$$

Satisfi able?

Spatial Incompleteness (Lemon)



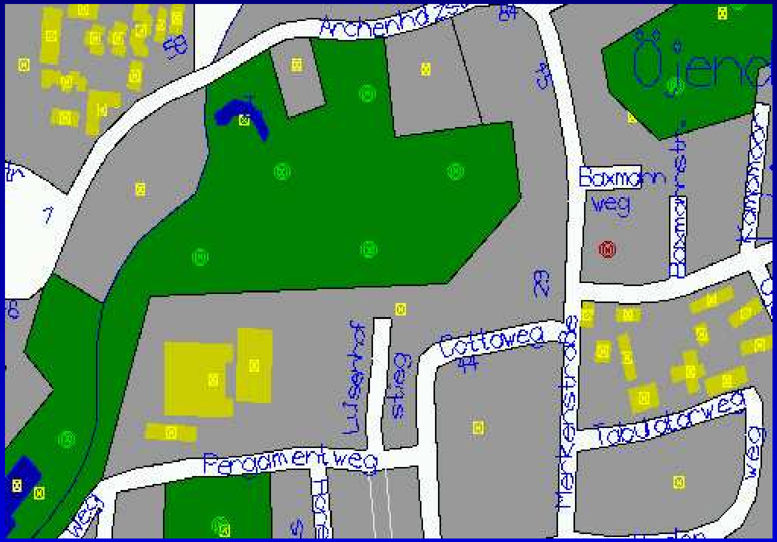
Spatial Incompleteness (Lemon)



Unsatisfiable!

Spatial Incompleteness (Lemon)

The Vision of a Deductive GIS



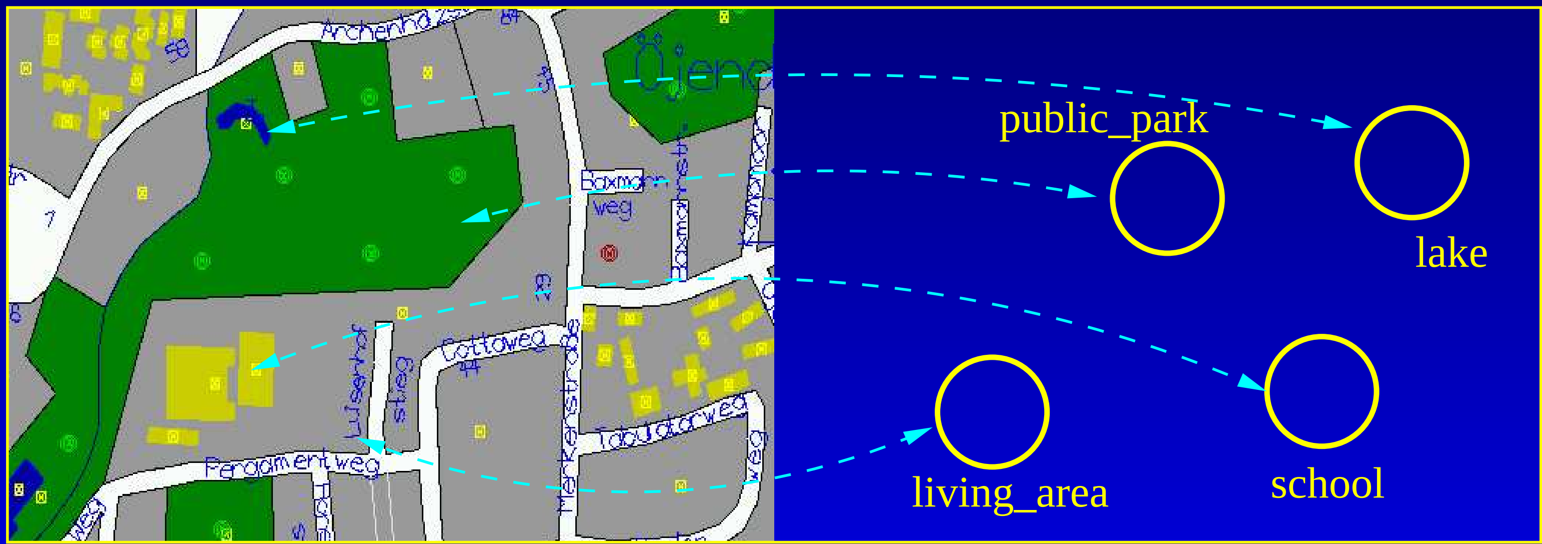
Concrete Geometry

Starting point: a digital vector map

TUHH Technische Universität Hamburg-Harburg

The Vision of a Deductive GIS

Extensional Component

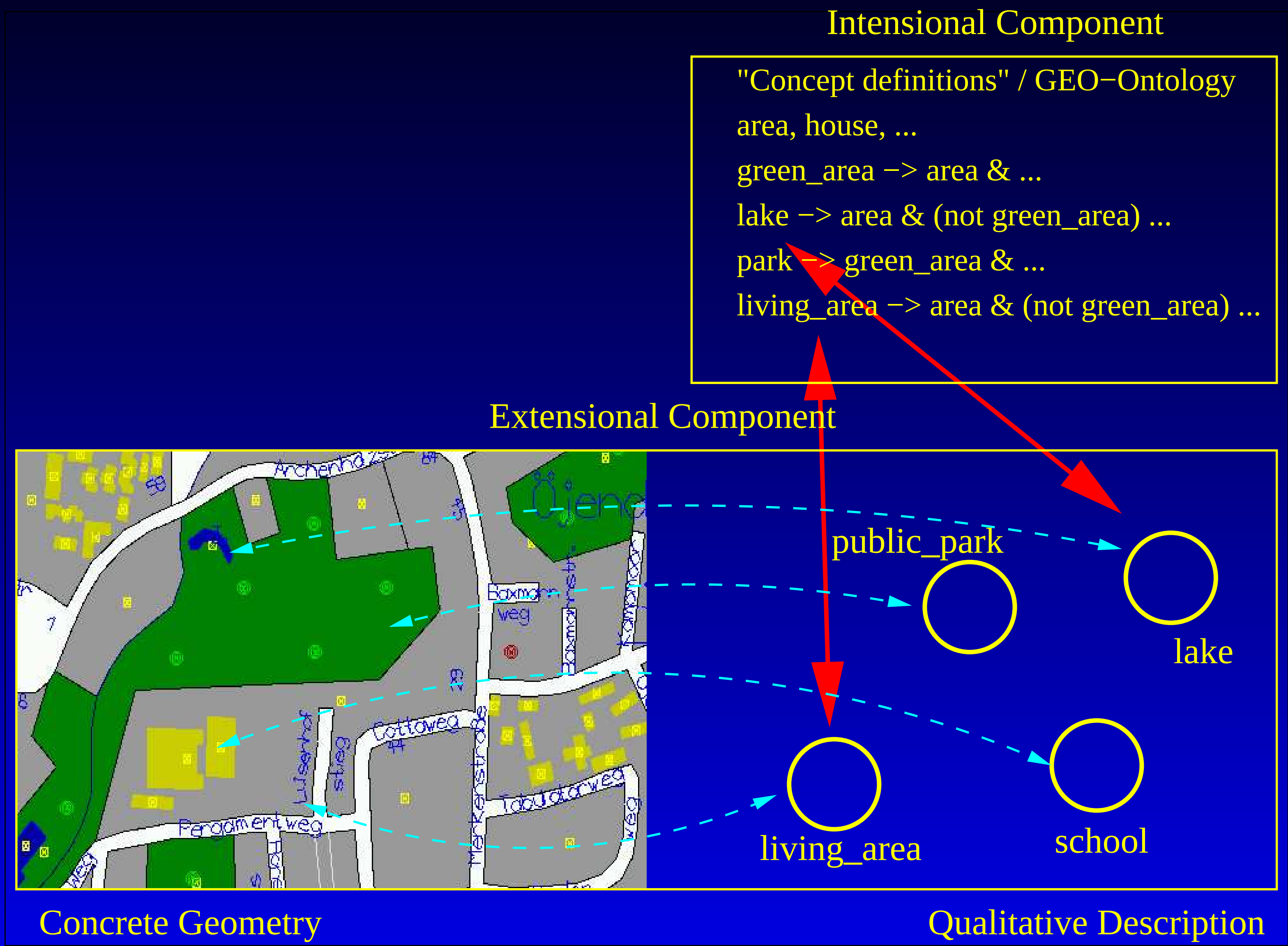


Concrete Geometry

Qualitative Description

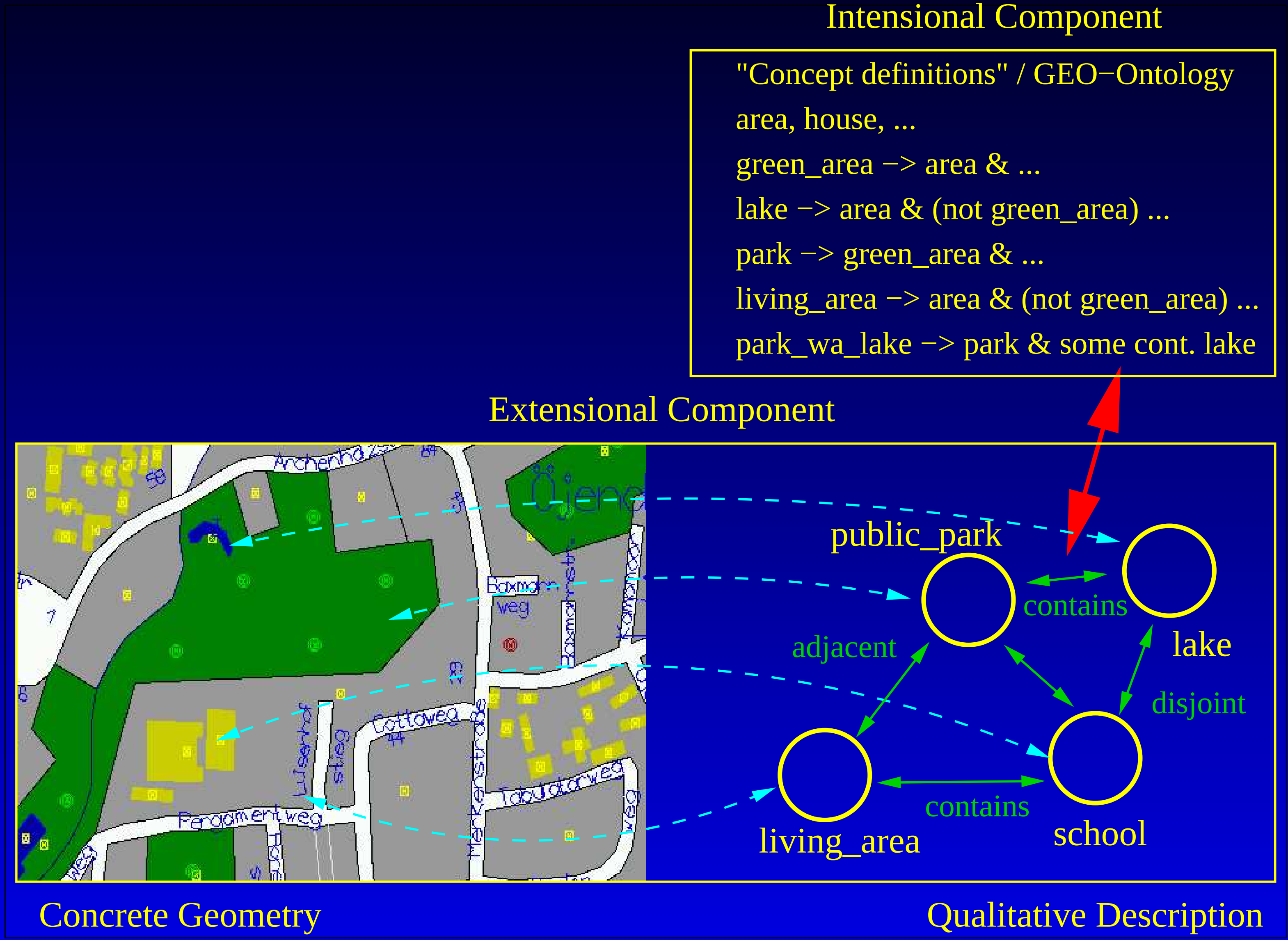
Thematic information in a map

The Vision of a Deductive GIS



Modeling of thematic concepts

The Vision of a Deductive GIS



Some concepts are really “spatio-thematic”

The Vision of a Deductive GIS

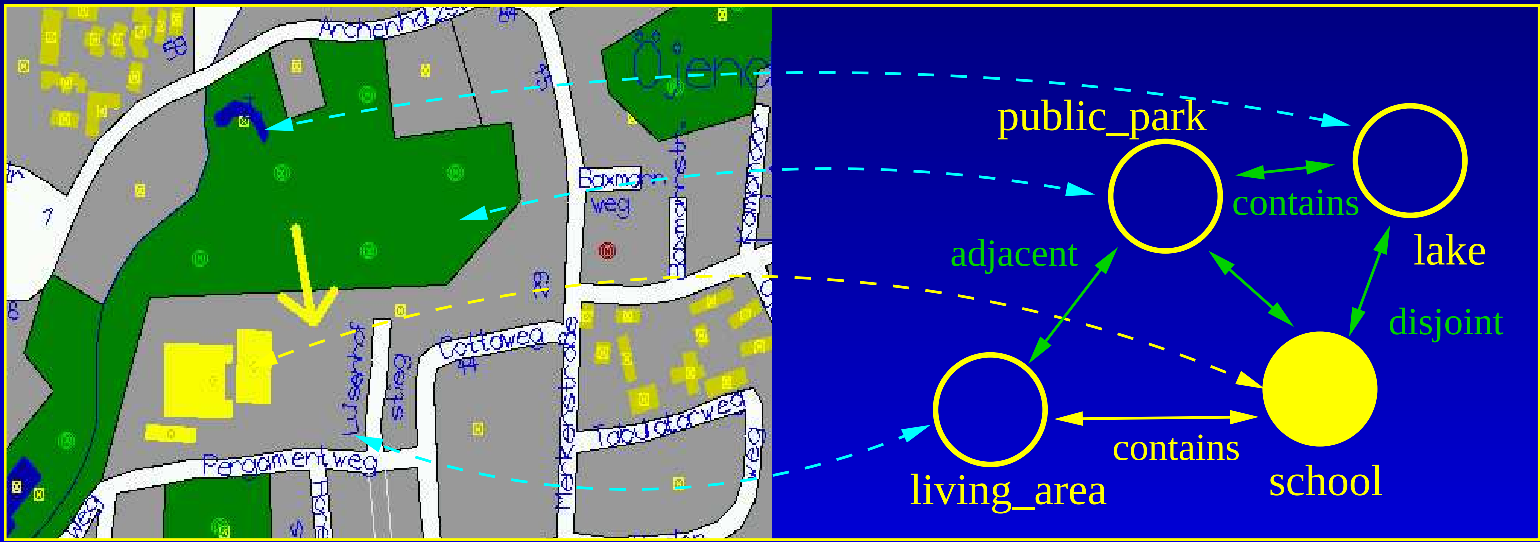
Query Component

Simple Spatial Queries:

Retrieve all areas contained within this area

Intensional Component

Extensional Component



Concrete Geometry

Qualitative Description

Purely spatial queries

The Vision of a Deductive GIS

Query Component

Simple Spatial Queries:

Retrieve all areas contained within this area

Thematic Queries:

retrieve_concept_instances(lake)

Intensional Component

"Concept definitions" / GEO−Ontology

area, house, ...

green_area −> area & ...

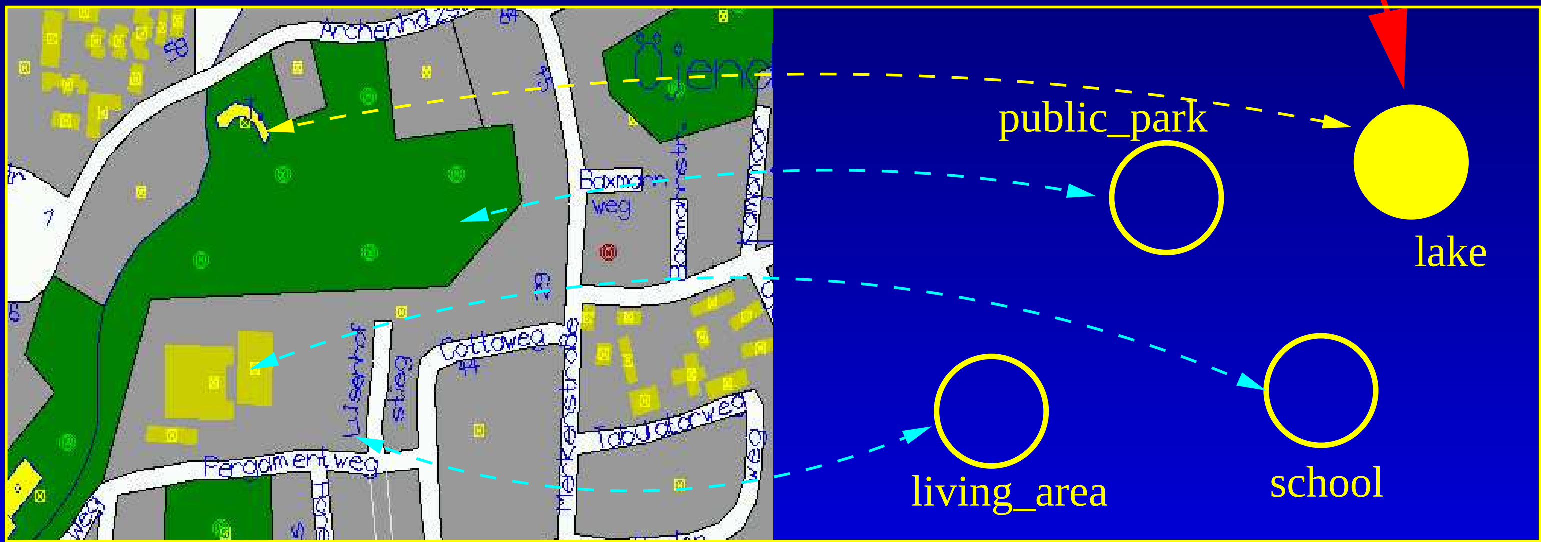
lake −> area & (not green_area) ...

park −> green_area & ...

living_area −> area & (not green_area) ...

park_wa_lake −> park & some cont. lake

Extensional Component



Concrete Geometry

Qualitative Description

Purely thematic queries

The Vision of a Deductive GIS

Query Component

Simple Spatial Queries:

- Retrieve all areas contained within this area

Thematic Queries:

- retrieve_concept_instances(lake)

Spatio−thematic Queries:

- Retrieve all parks that contain a lake

Intensional Component

"Concept definitions" / GEO−Ontology

area, house, ...

green_area −> area & ...

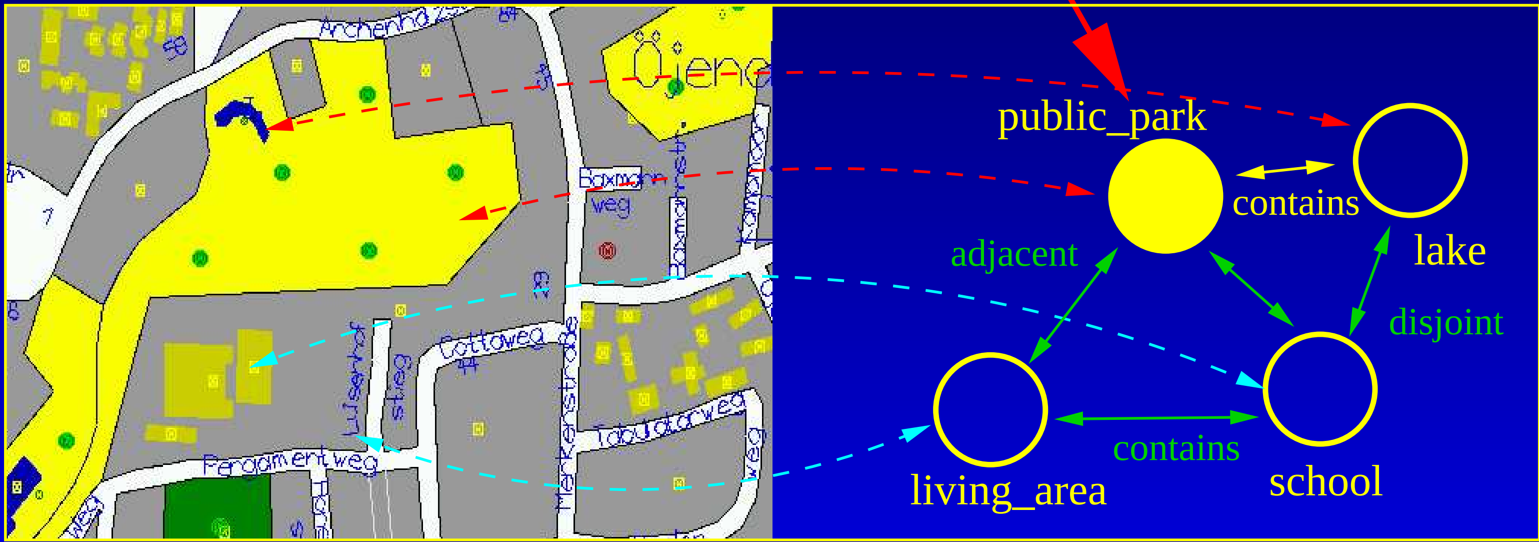
lake −> area & (not green_area) ...

park −> green_area & ...

living_area −> area & (not green_area) ...

park_wa_lake −> park & some cont. lake

Extensional Component



Concrete Geometry — Qualitative Description

“Spatio-thematic” queries

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
  - Representation of certain selected spatio-thematic aspects of a concrete map ("geographic world")
  - ? Which spatial and thematic aspects?
  - ? Underspecifi ed / indeterminate data?
  - ? Unifi ed or hybrid representation of spatial and/or thematic aspects (different "sources")?

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
  - Modeling of geo ontologies with (spatial?) geo concepts in a description language (not necessary DL)
  - ? Which spatial and thematic aspects?
  - ? Thematic, spatial, spatio-thematic concepts?
  - ? Combined or separated description languages for different aspects?
  - ? Spatio-thematic interaction?

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
  - Retrieval of interesting objects / constellations; "map analysis / reasoning"
  - ? Kind of queries
  - ? With spatial and thematic aspects are addressable?
  - ? Usage of concepts from the ontologies within queries
  - ? Evaluation of queries ("specialists" for sources)?

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks
  - $\mathcal{E}$: consistency checking of data
  - $\mathcal{I}$: consistency checking of ontology & concepts
  - $\mathcal{I}, \mathcal{Q}$: satisfi ability and entailment of queries / concepts
  - $\mathcal{I}, \mathcal{Q}$: computation of query / concept subsumption hierarchies ("taxonomies")
  - $\mathcal{E} \times \mathcal{I}$: instance "realization"

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

$\Rightarrow$ Multi-dimensional space of design-decisions

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

$\Rightarrow$ Multi-dimensional space of design-decisions

- DLMAPS framework = allows for flexible experiments

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

$\Rightarrow$ Multi-dimensional space of design-decisions

- DLMAPS framework = allows for flexible experiments
- One instantiation of this framework uses RacerPro

The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a description logic (DL) system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

$\Rightarrow$ Multi-dimensional space of design-decisions

- DLMAPS framework = allows for flexible experiments
- One instantiation of this framework uses RacerPro

Two Approaches

- Theoretician's / Logician's Approach:
  - Top Down
  - Design suitable spatio-thematic logics
  - Very hard, problems with decidability
  - Long way to go until we get a working system

Two Approaches

- Theoretician's / Logician's Approach:
  - Top Down
  - Design <u>suitable</u> spatio-thematic logics
  - Very hard, problems with decidability
  - Long way to go until we get a working system
- Practitioner's / Engineer's Approach:
  - Bottom Up, use and exploit existing components
  - Use an existing DL system: RacerPro
  - Add expressive query language suitable for <u>ontology-based spatio-thematic querying</u>
  - More limited but one gets a working system soon

Ontology-based DISK Querying

- Ontology: Explicit, formal specification of a conceptualization (Gruber)

Ontology-based DISK Querying

- Ontology: Explicit, formal specification of a conceptualization (Gruber)
- Here: ontology provides “common sense” (mainly thematic) vocabulary

Ontology-based DISK Querying

- Ontology: Explicit, formal specification of a conceptualization (Gruber)
- Here: ontology provides “common sense” (mainly thematic) vocabulary
- “Global as View”: the ontology can be seen as defining views over the information sources

Ontology-based DISK Querying

- Ontology: Explicit, formal specification of a conceptualization (Gruber)
- Here: ontology provides “common sense” (mainly thematic) vocabulary
- “Global as View”: the ontology can be seen as defining views over the information sources
- Setting 1: No geometry, purely symbolic representation in a RacerPro ABox!

Ontology-based DISK Querying

- Ontology: Explicit, formal specification of a conceptualization (Gruber)
- Here: ontology provides "common sense" (mainly thematic) vocabulary
- "Global as View": the ontology can be seen as defining views over the information sources
- Setting 1: No geometry, purely symbolic representation in a RacerPro ABox!
    - Needed: expressive ABox query language

Ontology-based DISK Querying

- Ontology: Explicit, formal specification of a conceptualization (Gruber)
- Here: ontology provides “common sense” (mainly thematic) vocabulary
- “Global as View”: the ontology can be seen as defining views over the information sources
- Setting 1: No geometry, purely symbolic representation in a RacerPro ABox!
- Setting 2:

Ontology-based DISK Querying

- Ontology: Explicit, formal specification of a conceptualization (Gruber)
- Here: ontology provides “common sense” (mainly thematic) vocabulary
- “Global as View”: the ontology can be seen as defining views over the information sources
- Setting 1: No geometry, purely symbolic representation in a RacerPro ABox!
- Setting 2:
  - Source 1: Thematic annotations $\longrightarrow$ DL ABox
  - Source 2: Map geometry $\longrightarrow$ spatial box
  - Expressive <u>hybrid</u> query language needed
  - Query expansion

Ontology-based Querying (2)

- Benefits
  - Users can abstract from sources and schemas of the sources
  - Ontology is formal $\longrightarrow$ queries can be reasoned about (checking consistency, entailment, equivalence)
  - Ontologies can capture taxonomic ("is-a") relationships
  - The mapping expressed by the ontology is easily adjustable

$\Rightarrow$ declarative programming, the "ontology view" on the data can be changed easily

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete
- $\mathcal{ALC}$ concepts, defined inductively:
  - Each $C \in N_C$ is a *(atomic) concept*

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete
- $\mathcal{ALC}$ concepts, defined inductively:
  - Each $C \in N_C$ is a *(atomic) concept*
  - If C and D are concepts, R is a role $R \in N_R$, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$ are *(compound) concepts* as well.

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete
- $\mathcal{ALC}$ concepts, defined inductively:
  - Each $C \in N_C$ is a *(atomic) concept*
  - If C and D are concepts, R is a role $R \in N_R$, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$ are *(compound) concepts* as well.
- Interpretation $I = (\Delta^I, \cdot^I)$, Domain (Universe) Δ^I, Interpretation function

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete
- $\mathcal{ALC}$ concepts, defined inductively:
  - Each $C \in N_C$ is a *(atomic) concept*
  - If C and D are concepts, R is a role $R \in N_R$, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$ are *(compound) concepts* as well.
- Interpretation $I = (\Delta^I, \cdot^I)$, Domain (Universe) Δ^I, Interpretation function
  - $\cdot^I(C) \subseteq 2^{\Delta^I}, \cdot^I(R) \subseteq 2^{\Delta^I} \times 2^{\Delta^I}$

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete
- $\mathcal{ALC}$ concepts, defined inductively:
  - Each $C \in N_C$ is a *(atomic) concept*
  - If C and D are concepts, R is a role $R \in N_R$, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$ are *(compound) concepts* as well.
- Interpretation $I = (\Delta^I, \cdot^I)$, Domain (Universe) Δ^I, Interpretation function
  - $\cdot^I(C) \subseteq 2^{\Delta^I}, \cdot^I(R) \subseteq 2^{\Delta^I} \times 2^{\Delta^I}$
  - $(C \sqcap D)^I = C^I \cap D^I, (C \sqcup D)^I = C^I \cup D^I, \neg C^I = \Delta^I \setminus C^I,$

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete
- $\mathcal{ALC}$ concepts, defined inductively:
  - Each $C \in N_C$ is a *(atomic) concept*
  - If C and D are concepts, R is a role $R \in N_R$, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$ are *(compound) concepts* as well.
- Interpretation $I = (\Delta^I, \cdot^I)$, Domain (Universe) Δ^I, Interpretation function
  - $\cdot^I(C) \subseteq 2^{\Delta^I}, \cdot^I(R) \subseteq 2^{\Delta^I} \times 2^{\Delta^I}$
  - $(C \sqcap D)^I = C^I \cap D^I, (C \sqcup D)^I = C^I \cup D^I, \neg C^I = \Delta^I \setminus C^I$,
  - $(\exists R.C)^I = \{d \in \Delta^I \mid \exists e \in \Delta^I ((d,e) \in R^I, e \in C^I)\}$,

Some DL-Basics: $\mathcal{ALC}$

- Simple DL, propositionally complete
- $\mathcal{ALC}$ concepts, defined inductively:
  - Each $C \in N_C$ is a *(atomic) concept*
  - If C and D are concepts, R is a role $R \in N_R$, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$ are *(compound) concepts* as well.
- Interpretation $I = (\Delta^I, \cdot^I)$, Domain (Universe) Δ^I, Interpretation function
  - $\cdot^I(C) \subseteq 2^{\Delta^I}, \cdot^I(R) \subseteq 2^{\Delta^I} \times 2^{\Delta^I}$
  - $(C \sqcap D)^I = C^I \cap D^I, (C \sqcup D)^I = C^I \cup D^I, \neg C^I = \Delta^I \setminus C^I$,
  - $(\exists R.C)^I = \{d \in \Delta^I \mid \exists e \in \Delta^I ((d,e) \in R^I, e \in C^I)\}$,
  - $(\forall R.C)^I = \{d \in \Delta^I \mid \forall e \in \Delta^I ((d,e) \in R^I \Rightarrow e \in C^I)\}$

$\mathcal{ALC}$ (2)

- Satisfiability of concepts
  - C is satisfiable iff there exists interpretation I such that $C^I \neq \emptyset$
  - I is called model of C

$\mathcal{ALC}$ (2)

- Satisfiability of concepts
  - C is satisfiable iff there exists interpretation I such that $C^I \neq \emptyset$
  - I is called model of C
- Subsumption of concepts
  - $C \sqsubseteq D$ iff $C^I \subseteq D^I$ for all I iff
  - $C \sqcap \neg D$ unsatisfiable

$\mathcal{ALC}$ (2)

- Satisfiability of concepts
  - C is satisfiable iff there exists interpretation I such that $C^I \neq \emptyset$
  - I is called model of C
- Subsumption of concepts
  - $C \sqsubseteq D$ iff $C^I \subseteq D^I$ for all I iff
  - $C \sqcap \neg D$ unsatisfiable
- Simple example
  - $human \sqcap male \sqsubseteq human$
  - $human \sqcap male \sqcap \neg human$ unsatisfiable
  - $human \sqcap female \sqsubseteq female$
  - $human \sqcap female \sqsubseteq human \sqcup female$

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships
- $\forall contains.\neg lake \sqcap \exists contains.\exists contains.lake$

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships
- $\forall contains.\neg lake \sqcap \exists contains.\exists contains.lake$
- Satisfiable in $\mathcal{ALC}$ $\leadsto$ ↯

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships
- $\forall contains.\neg lake \sqcap \exists contains.\exists contains.lake$
- Satisfiable in $\mathcal{ALC}$ $\rightsquigarrow$ ↯
- Specific properties of spatial relationships not captured

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships
- $\forall contains.\neg lake \sqcap \exists contains.\exists contains.lake$
- Satisfiable in $\mathcal{ALC} \rightsquigarrow \lightning$
- Specific properties of spatial relationships not captured
- $contains$ should be transitive (required: $\mathcal{ALC}_{\mathcal{R}^+}$)

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships
- $\forall contains.\neg lake \sqcap \exists contains.\exists contains.lake$
- Satisfiable in $\mathcal{ALC} \rightsquigarrow \lightning$
- Specific properties of spatial relationships not captured
- $contains$ should be transitive (required: $\mathcal{ALC}_{\mathcal{R}^+}$)
- How to handle $disjoint \circ contains \longrightarrow disjoint$?

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships
- $\forall contains.\neg lake \sqcap \exists contains.\exists contains.lake$
- Satisfiable in $\mathcal{ALC}$ $\rightsquigarrow$ ↯
- Specific properties of spatial relationships not captured
- $contains$ should be transitive (required: $\mathcal{ALC}_{\mathcal{R}^+}$)
- How to handle $disjoint \circ contains \longrightarrow disjoint$?

$\Rightarrow$ special DLs required

Spatial Concept with $\mathcal{ALC}$?

- Idea: roles for qualitative spatial relationships
- $\forall contains.\neg lake \sqcap \exists contains.\exists contains.lake$
- Satisfiable in $\mathcal{ALC} \rightsquigarrow \lightning$
- Specific properties of spatial relationships not captured
- $contains$ should be transitive (required: $\mathcal{ALC}_{\mathcal{R}^+}$)
- How to handle $disjoint \circ contains \longrightarrow disjoint$?

$\Rightarrow$ special DLs required

- Not considered in the following (“Practitioner’s approach”)

The TBox

- Terminology, provides vocabulary (ontology)

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D$, $C \doteq D$

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D$, $C \doteq D$
- Simple TBoxes: no cycles and $C \in CN$

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D$, $C \dot{\equiv} D$
- Simple TBoxes: no cycles and $C \in CN$
- General TBoxes (RacerPro): C, D arbitrary

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D$, $C \dot{\equiv} D$
- Simple TBoxes: no cycles and $C \in CN$
- General TBoxes (RacerPro): C, D arbitrary
- Descriptive Semantics: $\mathcal{I} \models \mathcal{T}$ iff $\forall C \sqsubseteq D \in \mathcal{T}.\mathcal{I} \models C \sqsubseteq D, C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

Technische Universität Hamburg-Harburg

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D$, $C \dot{=} D$
- Simple TBoxes: no cycles and $C \in CN$
- General TBoxes (RacerPro): C, D arbitrary
- Descriptive Semantics: $\mathcal{I} \models \mathcal{T}$ iff $\forall C \sqsubseteq D \in \mathcal{T}.\mathcal{I} \models C \sqsubseteq D, C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- TBoxes affect concept satisfiability

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D, C \doteq D$
- Simple TBoxes: no cycles and $C \in CN$
- General TBoxes (RacerPro): C, D arbitrary
- Descriptive Semantics: $\mathcal{I} \models \mathcal{T}$ iff $\forall C \sqsubseteq D \in \mathcal{T}. \mathcal{I} \models C \sqsubseteq D, C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- TBoxes affect concept satisfiability
  - $mother \sqcap \neg female$ is satisfiable
  - $mother \sqcap \neg female$ is unsatisfiable wrt. TBox $\{mother \doteq human \sqcap female \sqcap \ldots\}$

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D$, $C \dot{\equiv} D$
- Simple TBoxes: no cycles and $C \in CN$
- General TBoxes (RacerPro): C, D arbitrary
- Descriptive Semantics: $\mathcal{I} \models \mathcal{T}$ iff $\forall C \sqsubseteq D \in \mathcal{T} . \mathcal{I} \models C \sqsubseteq D$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- TBoxes affect concept satisfiability
- Taxonomy: shows all direct subsumption relationships between concept names from $\mathcal{T}$
  - "Deduced is-a links"

The TBox

- Terminology, provides vocabulary (ontology)
- TBox $\mathcal{T}$ = finite set of axioms (definitions)
  - $C \dot{\sqsubseteq} D$, $C \doteq D$
- Simple TBoxes: no cycles and $C \in CN$
- General TBoxes (RacerPro): C, D arbitrary
- Descriptive Semantics: $\mathcal{I} \models \mathcal{T}$ iff $\forall C \sqsubseteq D \in \mathcal{T} . \mathcal{I} \models C \sqsubseteq D, C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- TBoxes affect concept satisfiability
- Taxonomy: shows all direct subsumption relationships between concept names from $\mathcal{T}$
  - "Deduced is-a links"

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships
- ABox $\mathcal{A}$ = finite set of assertions
  - Concept assertion: $betty : woman$
    $\mathcal{I} \models betty : woman$ iff $betty^{\mathcal{I}} \in woman^{\mathcal{I}}$

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships
- ABox $\mathcal{A}$ = finite set of assertions
  - Concept assertion: $betty : woman$
    $\mathcal{I} \models betty : woman$ iff $betty^{\mathcal{I}} \in woman^{\mathcal{I}}$
  - Role assertion: $(betty, doris) : has\_child$
    $\mathcal{I} \models (betty, doris) : has\_child$ iff
    $(betty^{\mathcal{I}}, doris^{\mathcal{I}}) \in has\_child^{\mathcal{I}}$

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships
- ABox $\mathcal{A}$ = finite set of assertions
  - Concept assertion: $betty : woman$
    $\mathcal{I} \models betty : woman$ iff $betty^{\mathcal{I}} \in woman^{\mathcal{I}}$
  - Role assertion: $(betty, doris) : has\_child$
    $\mathcal{I} \models (betty, doris) : has\_child$ iff $(betty^{\mathcal{I}}, doris^{\mathcal{I}}) \in has\_child^{\mathcal{I}}$
- Important services

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships
- ABox $\mathcal{A}$ = finite set of assertions
  - Concept assertion: $betty : woman$
    $\mathcal{I} \models betty : woman$ iff $betty^{\mathcal{I}} \in woman^{\mathcal{I}}$
  - Role assertion: $(betty, doris) : has\_child$
    $\mathcal{I} \models (betty, doris) : has\_child$ iff $(betty^{\mathcal{I}}, doris^{\mathcal{I}}) \in has\_child^{\mathcal{I}}$
- Important services
  - ABox satisfiability

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships
- ABox $\mathcal{A}$ = finite set of assertions
  - Concept assertion: $betty : woman$
    $\mathcal{I} \models betty : woman$ iff $betty^{\mathcal{I}} \in woman^{\mathcal{I}}$
  - Role assertion: $(betty, doris) : has\_child$
    $\mathcal{I} \models (betty, doris) : has\_child$ iff $(betty^{\mathcal{I}}, doris^{\mathcal{I}}) \in has\_child^{\mathcal{I}}$
- Important services
  - ABox satisfiability
  - Instance checking: $(\mathcal{A}, \mathcal{T}) \models a : C$?

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships
- ABox $\mathcal{A}$ = finite set of assertions
  - Concept assertion: $betty : woman$
    $\mathcal{I} \models betty : woman$ iff $betty^{\mathcal{I}} \in woman^{\mathcal{I}}$
  - Role assertion: $(betty, doris) : has\_child$
    $\mathcal{I} \models (betty, doris) : has\_child$ iff $(betty^{\mathcal{I}}, doris^{\mathcal{I}}) \in has\_child^{\mathcal{I}}$
- Important services
  - ABox satisfiability
  - Instance checking: $(\mathcal{A}, \mathcal{T}) \models a : C$?
  - Instance retrieval:
    $\{ a \in \mathsf{individuals}(\mathcal{A}) \mid (\mathcal{A}, \mathcal{T}) \models a : C \}$

The ABox

- Modeling of a concrete state of the world in terms of individuals and interrelationships
- ABox $\mathcal{A}$ = finite set of assertions
  - Concept assertion: $betty : woman$
    $\mathcal{I} \models betty : woman$ iff $betty^{\mathcal{I}} \in woman^{\mathcal{I}}$
  - Role assertion: $(betty, doris) : has\_child$
    $\mathcal{I} \models (betty, doris) : has\_child$ iff $(betty^{\mathcal{I}}, doris^{\mathcal{I}}) \in has\_child^{\mathcal{I}}$
- Important services
  - ABox satisfiability
  - Instance checking: $(\mathcal{A}, \mathcal{T}) \models a : C$?
  - Instance retrieval:
    $\{\, a \in \mathsf{individuals}(\mathcal{A}) \mid (\mathcal{A}, \mathcal{T}) \models a : C \,\}$

Expressive DLs

- RacerPro: $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$ **(**$\mathcal{SHIQ}(D^{-1})$**)**

Expressive DLs

- RacerPro: $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$ ($\mathcal{SHIQ}(D^{-1})$)
- $\mathcal{Q}$: Qualified number restrictions, $\geq_2 \, has\_child.male$ ($\leq_2 \, has\_child.female$)

Expressive DLs

- RacerPro: $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$ $(\mathcal{SHIQ}(D^{-1}))$
- $\mathcal{Q}$: Qualified number restrictions, $\geq_2\ has\_child.male$ ($\leq_2\ has\_child.female$)
- $\mathcal{H}$: Role hierarchies, $has\_son \dot{\sqsubseteq} has\_child$
- $\mathcal{I}$: Inverse roles, $has\_child \doteq has\_parent^{-1}$
- $\mathcal{R}^+$: Transitively closed roles, $transitiv(has\_descendant)$

Expressive DLs

- RacerPro: $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$ ($\mathcal{SHIQ}(D^{-1})$)
- $\mathcal{Q}$: Qualified number restrictions, $\geq_2\ has\_child.male$ ($\leq_2\ has\_child.female$)
- $\mathcal{H}$: Role hierarchies, $has\_son \dot{\sqsubseteq} has\_child$
- $\mathcal{I}$: Inverse roles, $has\_child \doteq has\_parent^{-1}$
- $\mathcal{R}^+$: Transitively closed roles, $transitiv(has\_descendant)$
- $\mathcal{D}^-$: Restricted concrete domains
  - CD = Concrete universe ($\mathbb{N}$, $\mathbb{R}$, $\mathbb{S}$, $\ldots$) + predicates ($=, \leq, \geq, \ldots$)
  - Restricted predicate exists operator: $\exists(age, body\_weight).=$

Expressive DLs

- RacerPro: $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$ ($\mathcal{SHIQ}(D^{-1})$)
- $\mathcal{Q}$: Qualified number restrictions, $\geq_2$ *has_child.male* ($\leq_2$ *has_child.female*)
- $\mathcal{H}$: Role hierarchies, $has\_son \dot{\sqsubseteq} has\_child$
- $\mathcal{I}$: Inverse roles, $has\_child \doteq has\_parent^{-1}$
- $\mathcal{R}^+$: Transitively closed roles, $transitiv(has\_descendant)$
- $\mathcal{D}^-$: Restricted concrete domains
  - CD = Concrete universe ($\mathbb{N}$, $\mathbb{R}$, $\mathbb{S}$, ...) + predicates ($=, \leq, \geq, \ldots$)
  - Restricted predicate exists operator: $\exists(age, body\_weight).=$

Representing DISK: Setting 1

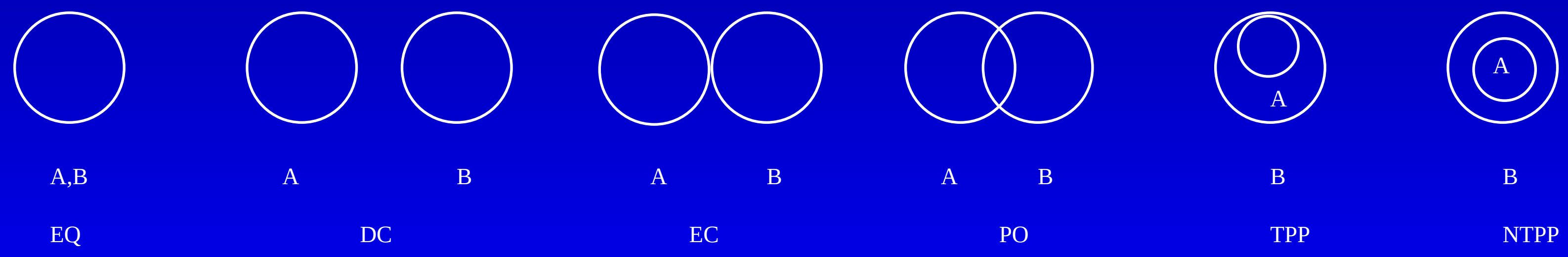
- Remodeling of thematic categories in a TBox
  - concept_for_key$(2224) =_{def} park$
  - Additional concepts ("ontology")
    $park\_with\_lake \doteq park \sqcap \exists contains.lake$

Representing DISK: Setting 1

- Remodeling of thematic categories in a TBox
  - concept_for_key$(2224) =_{def} park$
  - Additional concepts (“ontology”) $park\_with\_lake \doteq park \sqcap \exists contains.lake$
- Representation of the map as an ABox $\mathcal{A}$
  - For each map object i with key n, add $i :$ concept_for_key(n) to $\mathcal{A}$

Representing DISK: Setting 1

- Remodeling of thematic categories in a TBox
  - concept_for_key$(2224) =_{def} park$
  - Additional concepts (“ontology”)
    $park\_with\_lake \doteq park \sqcap \exists contains.lake$
- Representation of the map as an ABox $\mathcal{A}$
  - For each map object i with key n, add
    i : concept_for_key(n) to $\mathcal{A}$
  - Compute qualitative RCC8 relationships:
    (i, j) : EC, (i, k) : TPPI, ...



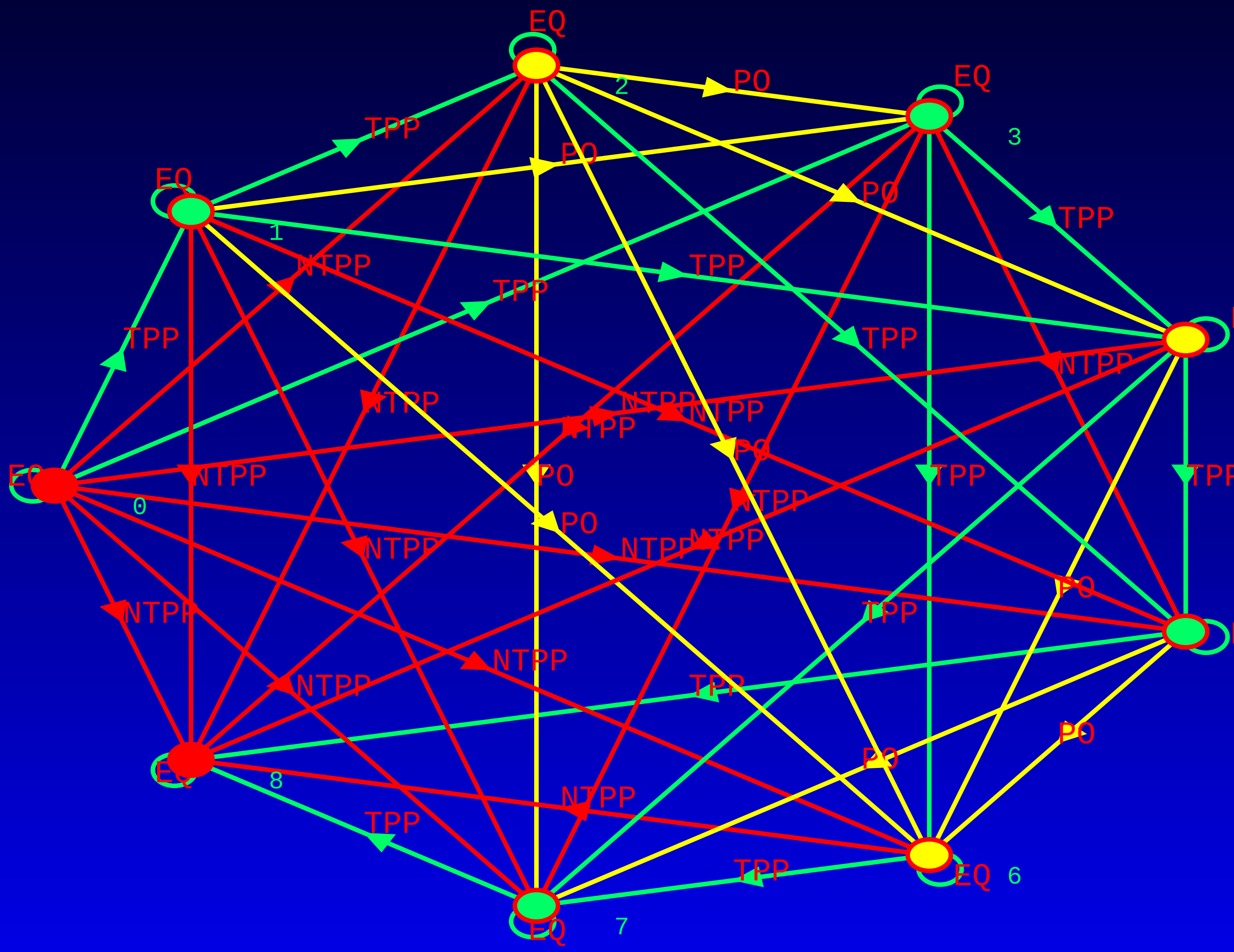
Representing DISK: Setting 1

- Remodeling of thematic categories in a TBox
  - $\mathsf{concept\_for\_key}(2224) =_{def} park$
  - Additional concepts ("ontology") $park\_with\_lake \doteq park \sqcap \exists contains.lake$
- Representation of the map as an ABox $\mathcal{A}$
  - For each map object i with key n, add $i : \mathsf{concept\_for\_key}(n)$ to $\mathcal{A}$
  - Compute qualitative RCC8 relationships: $(i, j) : \mathrm{EC}$, $(i, k) : \mathrm{TPPI}$, $\ldots$

$\Rightarrow$ RCC8 network in the ABox (network is always consistent)

TUHH Technische Universität Hamburg-Harburg

Illustration: DISK ABox



Querying the DISK (Setting 1)

- Ordinary instance retrieval queries:
  $\mathsf{retrieve}(park\_with\_lake) = \{i, \ldots\}$

Querying the DISK (Setting 1)

- Ordinary instance retrieval queries:
  $\mathsf{retrieve}(park\_with\_lake) = \{i, \ldots\}$
- Problems with the approach:
  - n^2 size of generated ABoxes (29 million role assertions with DC, 19.880 without DC)
  - Qualitative representation of “spatial data” in an ABox
  - Expressive queries wanted:
    $query(x, y) \leftarrow park(x) \wedge contains(x, y) \wedge lake(y)$
  - Universal quantification?
    $\mathsf{retrieve}(park \sqcap \forall contains.lake) = \{\}$ (due to OWA)
  - Query non-qualitative spatial aspects?

Querying the DISK (Setting 1)

- Ordinary instance retrieval queries: retrieve$(park_with_lake) = \{i, \ldots\}$
- Problems with the approach:
  - n^2 size of generated ABoxes (29 million role assertions with DC, 19.880 without DC)
  - Qualitative representation of "spatial data" in an ABox
  - Expressive queries wanted:

$$query(x, y) \leftarrow park(x) \wedge contains(x, y) \wedge lake(y)$$

  - Universal quantification? retrieve$(park \sqcap \forall contains.lake) = \{\}$ (due to OWA)
  - Query non-qualitative spatial aspects?

Representing DISK: Setting 2

- Map substrate: $(ABox, SBox, *)$
- ABox represents thematic aspects of map objects
- SBox (Spatial Box): geometry of the map

Representing DISK: Setting 2

- Map substrate: $(ABox, SBox, *)$
- ABox represents thematic aspects of map objects
- SBox (Spatial Box): geometry of the map

⊕ Non-symbolic spatial query atoms possible

⊕ On-demand computation and inspection of spatial aspects

⊕ Dedicated index structures

⊕ Closed world reasoning on maps

⊕ Simple model checking

Representing DISK: Setting 2

- Map substrate: $(ABox, SBox, *)$
- ABox represents thematic aspects of map objects
- SBox (Spatial Box): geometry of the map

⊕ Non-symbolic spatial query atoms possible

⊕ On-demand computation and inspection of spatial aspects

⊕ Dedicated index structures

⊕ Closed world reasoning on maps

⊕ Simple model checking

⊖ No more reasoning on spatial aspects, but was incomplete anyway

⊖ Works only for spatial “data”

Representing DISK: Setting 2

- Map substrate: $(ABox, SBox, *)$
- ABox represents thematic aspects of map objects
- SBox (Spatial Box): geometry of the map

⊕ Non-symbolic spatial query atoms possible

⊕ On-demand computation and inspection of spatial aspects

⊕ Dedicated index structures

⊕ Closed world reasoning on maps

⊕ Simple model checking

⊖ No more reasoning on spatial aspects, but was incomplete anyway

⊖ Works only for spatial “data”

Hybrid Queries (Setting 2)

- Two sorts of atoms

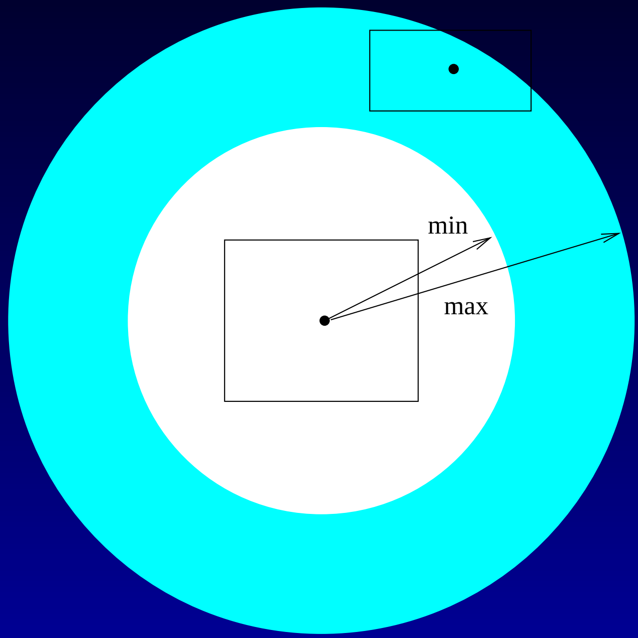
Hybrid Queries (Setting 2)

- Two sorts of atoms
- ABox atoms:
  - Variables range over ABox individuals
  - Atoms as in nRQL (see below)

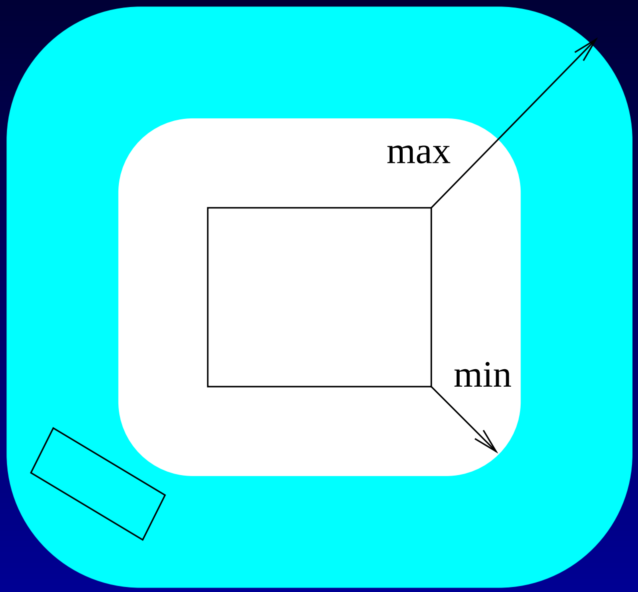
Hybrid Queries (Setting 2)

- Two sorts of atoms
- ABox atoms:
  - Variables range over ABox individuals
  - Atoms as in nRQL (see below)
- Spatial atoms:
  - Variables range over SBox individuals
  - RCC atoms
  - Geometric attributes: area, length, ...
  - Metric relationships: range queries, epsilon queries, ...

Hybrid Queries (Setting 2)



Range Query

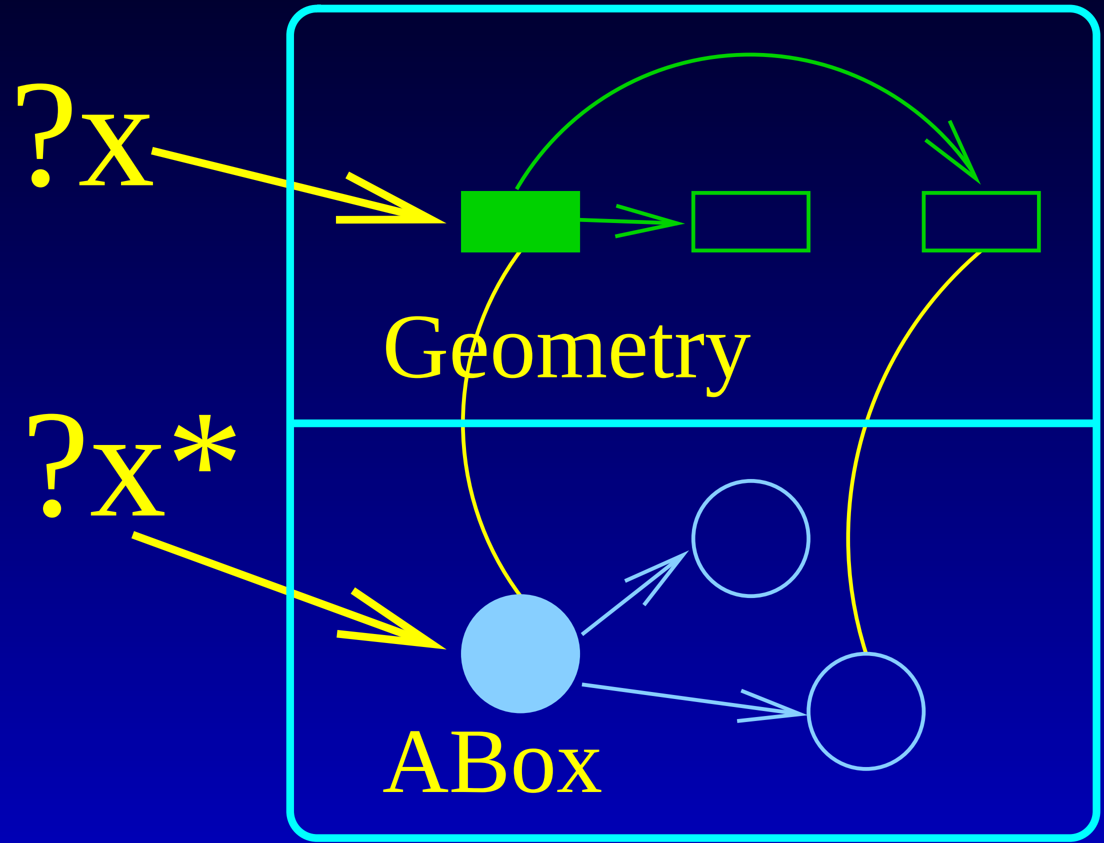


Epsilon Query

Hybrid Queries (Setting 2)

- Two sorts of atoms
- ABox atoms:
  - Variables range over ABox individuals
  - Atoms as in nRQL (see below)
- Spatial atoms:
  - Variables range over SBox individuals
  - RCC atoms
  - Geometric attributes: area, length, ...
  - Metric relationships: range queries, epsilon queries, ...
- Variables are bound in parallel, according to association

Hybrid Queries (Setting 2)



$?x^* / ?x$ are bound in parallel

Hybrid Queries (2)

- Hybridness can be made transparent

Hybrid Queries (2)

- Hybridness can be made transparent
- “Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant”
- $query(?x, ?y, ?z, ?f) \leftarrow$

$$industrial\_area(?x), creek(?y), lake\_or\_pond(?z),$$
$$contaminated(?z), chemical\_plant(?f), park(?u),$$
$$borders(?y, ?x), flows\_in(?y, ?z), contains(?u, ?z),$$
$$contains(?x, ?f)$$

Hybrid Queries (2)

- Hybridness can be made transparent
- "Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant"
- $query(?x, ?y, ?z, ?f) \leftarrow$

  $industrial\_area(?x), creek(?y), lake\_or\_pond(?z),$

  $contaminated(?z), chemical\_plant(?f), park(?u),$

  $borders(?y, ?x), flows\_in(?y, ?z), contains(?u, ?z),$

  $contains(?x, ?f)$
- Query rewriting / expansion

Hybrid Queries (2)

- Hybridness can be made transparent
- "Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant"
- $query(?x, ?y, ?z, ?f) \leftarrow$
  $industrial\_area(?x), creek(?y), lake\_or\_pond(?z),$
  $contaminated(?z), chemical\_plant(?f), park(?u),$
  $borders(?y, ?x), flows\_in(?y, ?z), contains(?u, ?z),$
  $contains(?x, ?f)$
- Query rewriting / expansion
  - $lake\_or\_pond(?z) \longrightarrow$
    $?z^* : (lake \sqcup pond)$

Hybrid Queries (2)

- Hybridness can be made transparent
- “Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant”
- $query(?x, ?y, ?z, ?f) \leftarrow$

  $industrial\_area(?x), creek(?y), lake\_or\_pond(?z),$

  $contaminated(?z), chemical\_plant(?f), park(?u),$

  $borders(?y, ?x), flows\_in(?y, ?z), contains(?u, ?z),$

  $contains(?x, ?f)$
- Query rewriting / expansion
  - $borders(?y, ?x), flows\_in(?y, ?x) \longrightarrow EC(?y, ?x)$

Hybrid Queries (2)

- Hybridness can be made transparent
- “Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant”
- $query(?x, ?y, ?z, ?f) \leftarrow$

  $industrial\_area(?x), creek(?y), lake\_or\_pond(?z),$

  $contaminated(?z), chemical\_plant(?f), park(?u),$

  $borders(?y, ?x), flows\_in(?y, ?z), contains(?u, ?z),$

  $contains(?x, ?f)$
- Query rewriting / expansion
  - $contaminated(?z) \rightarrow$
    $?z^* : \exists water\_quality.poisoned$

Hybrid Queries (2)

- Hybridness can be made transparent
- "Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant"
- $query(?x, ?y, ?z, ?f) \leftarrow$

  $industrial\_area(?x), creek(?y), lake\_or\_pond(?z),$

  $contaminated(?z), chemical\_plant(?f), park(?u),$

  $borders(?y, ?x), flows\_in(?y, ?z), contains(?u, ?z),$

  $contains(?x, ?f)$
- Query rewriting / expansion
- Result of the expansion/reformulation process: a hybrid query which we can process

Hybrid Queries (2)

- Hybridness can be made transparent
- “Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant”
- $query(?x, ?y, ?z, ?f) \leftarrow$

  $industrial\_area(?x), creek(?y), lake\_or\_pond(?z),$

  $contaminated(?z), chemical\_plant(?f), park(?u),$

  $borders(?y, ?x), flows\_in(?y, ?z), contains(?u, ?z),$

  $contains(?x, ?f)$
- Query rewriting / expansion
- Result of the expansion/reformulation process: a hybrid query which we can process

From DLMAPS to nRQL

- …DLMAPS query engine became core of nRQL engine
- …hybrid reprsentations added to RacerPro to faciliate "data representation"
- …the SBox of DLMAPS has not been incorporated, instead the RCC substrate has been added to support "RCC query atoms" (model checking no longer suffi cent!)

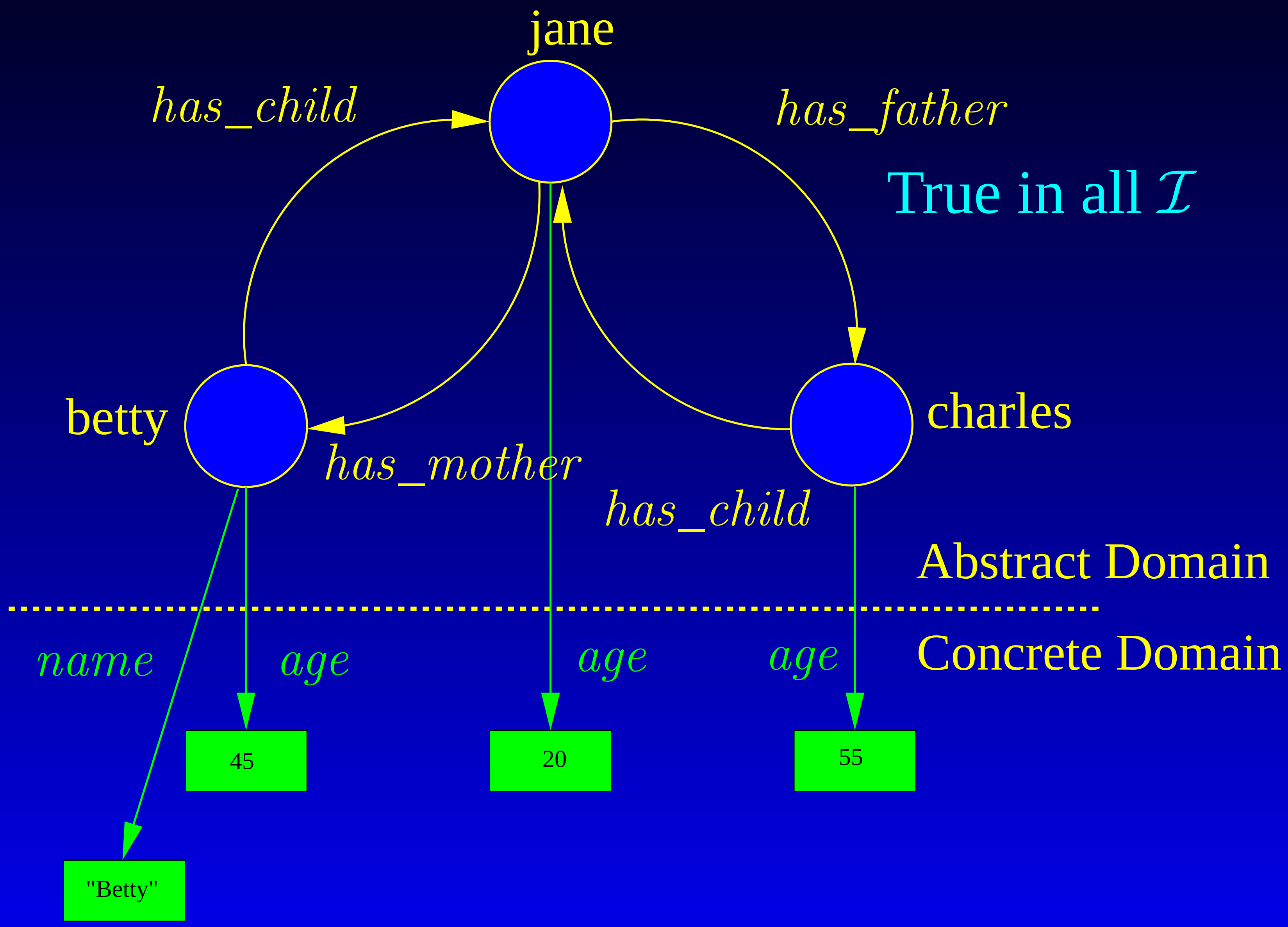
```
(retrieve (?x (told-value (age ?x)))
          (and (?x (and woman (an age)))
               (?x ?y has-child)
               (?y ?y (constraint
                           (has-father age)
                           (has-mother age)
                           (> age-1 (+ age-2 8))))))
```

From DLMAPS to nRQL

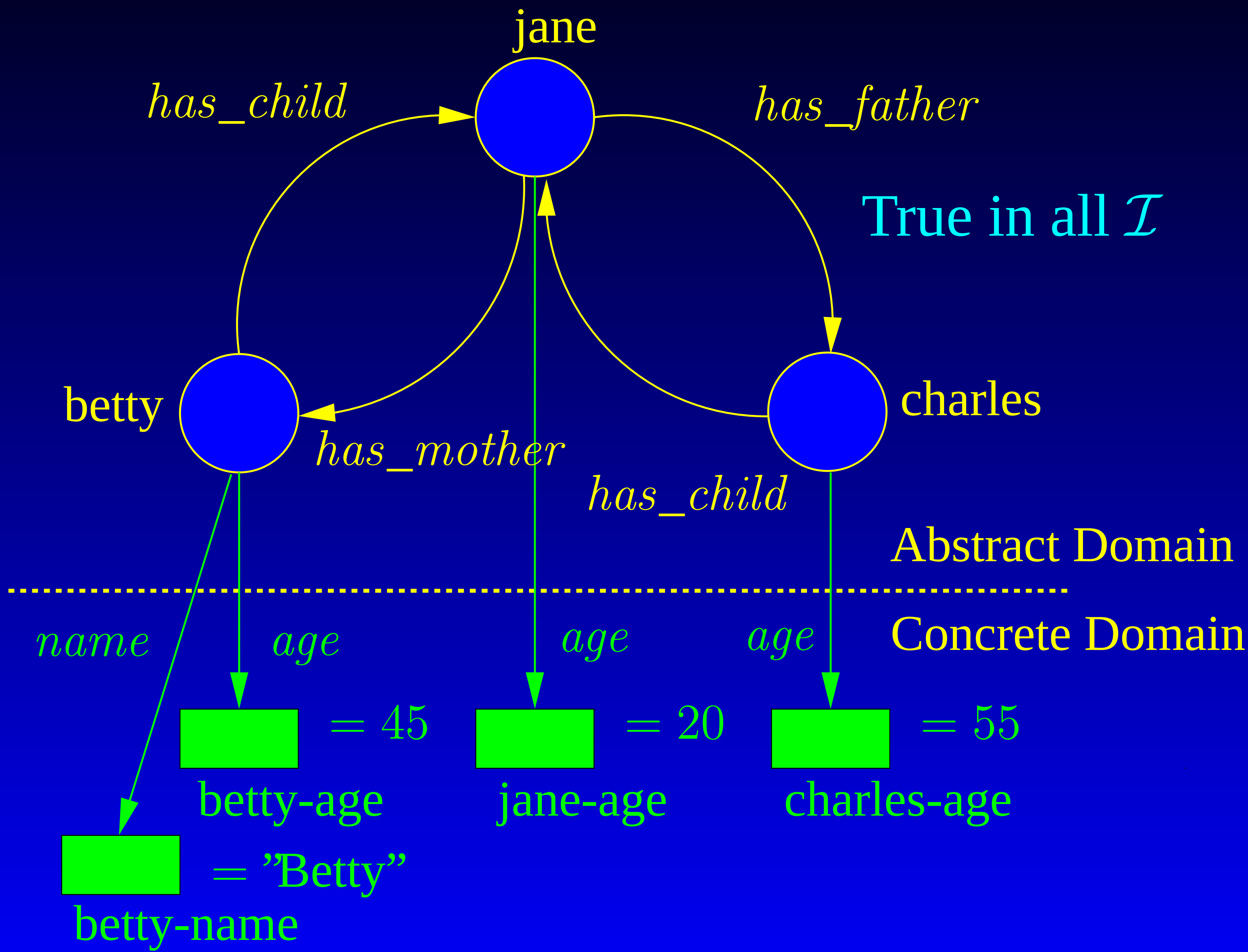
- … DLMAPS query engine became core of nRQL engine
- … hybrid reprsentations added to RacerPro to faciliate "data representation"
- … the SBox of DLMAPS has not been incorporated, instead the RCC substrate has been added to support "RCC query atoms" (model checking no longer suffi cent!)

```
(((?x betty) ((told-value (age ?x)) 45))
 ((?x diana) ((told-value (age ?x)) 55))
          ⋮
           )
```

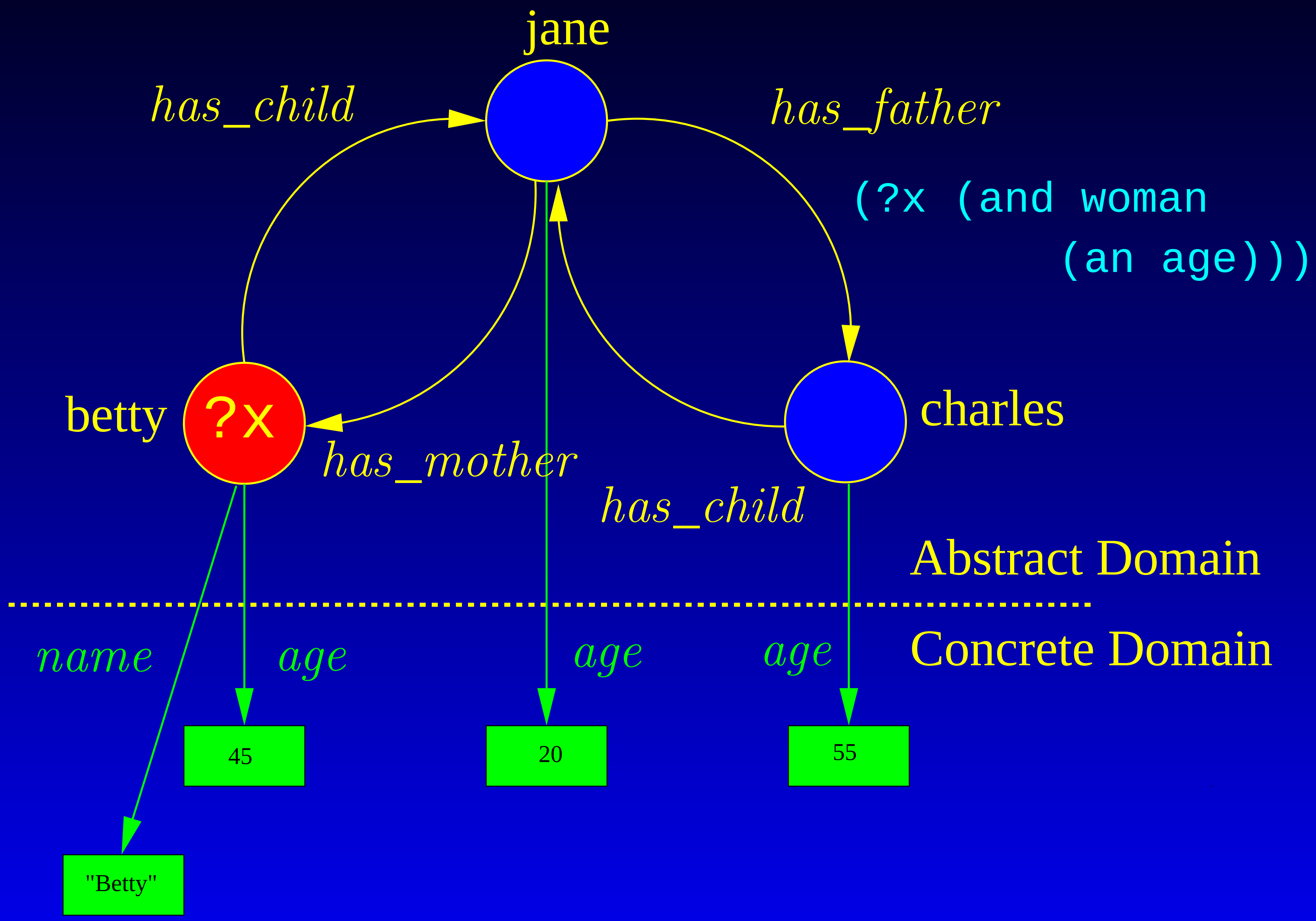
nRQL Language – Example



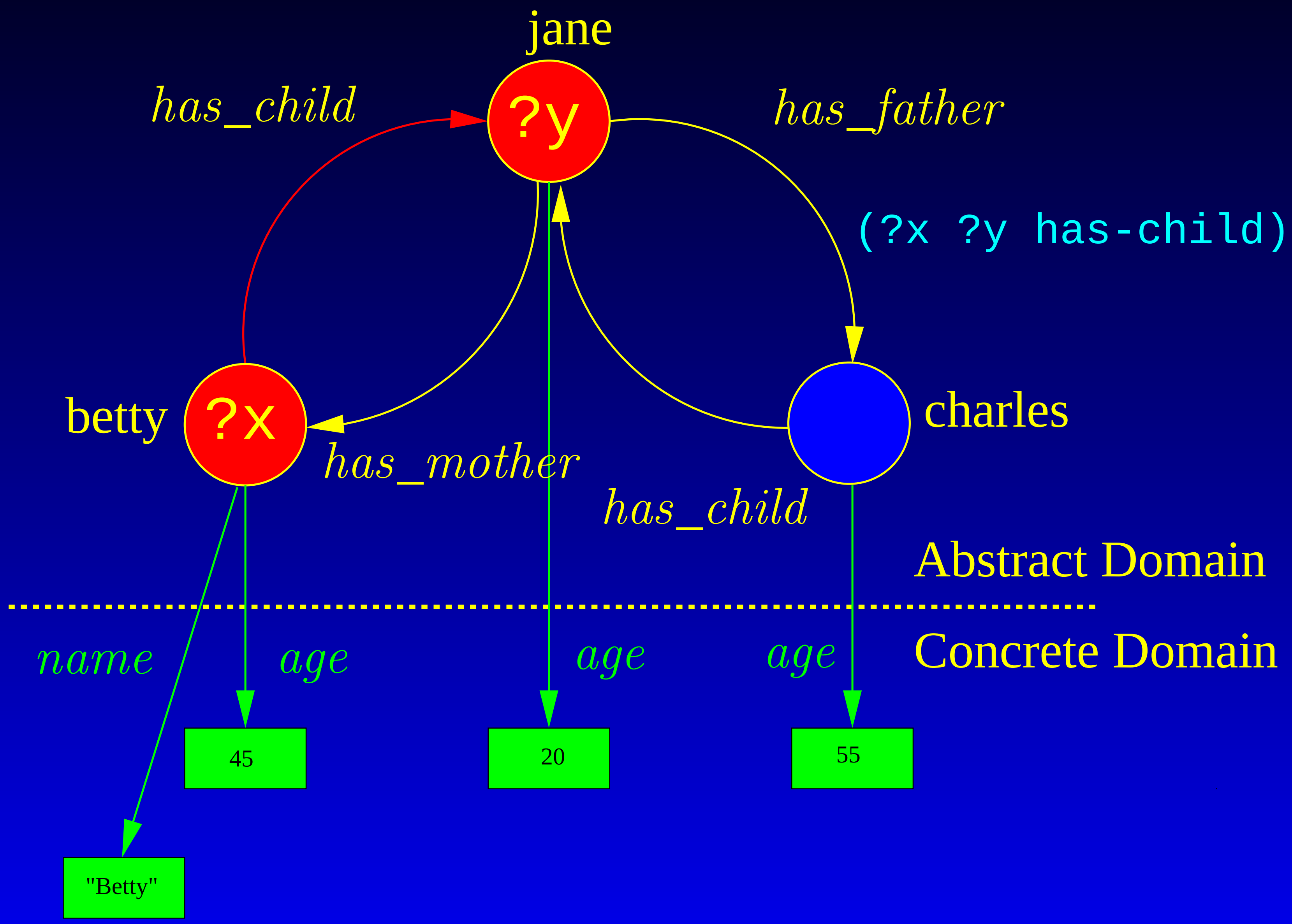
nRQL Language – Example



nRQL Language – Example

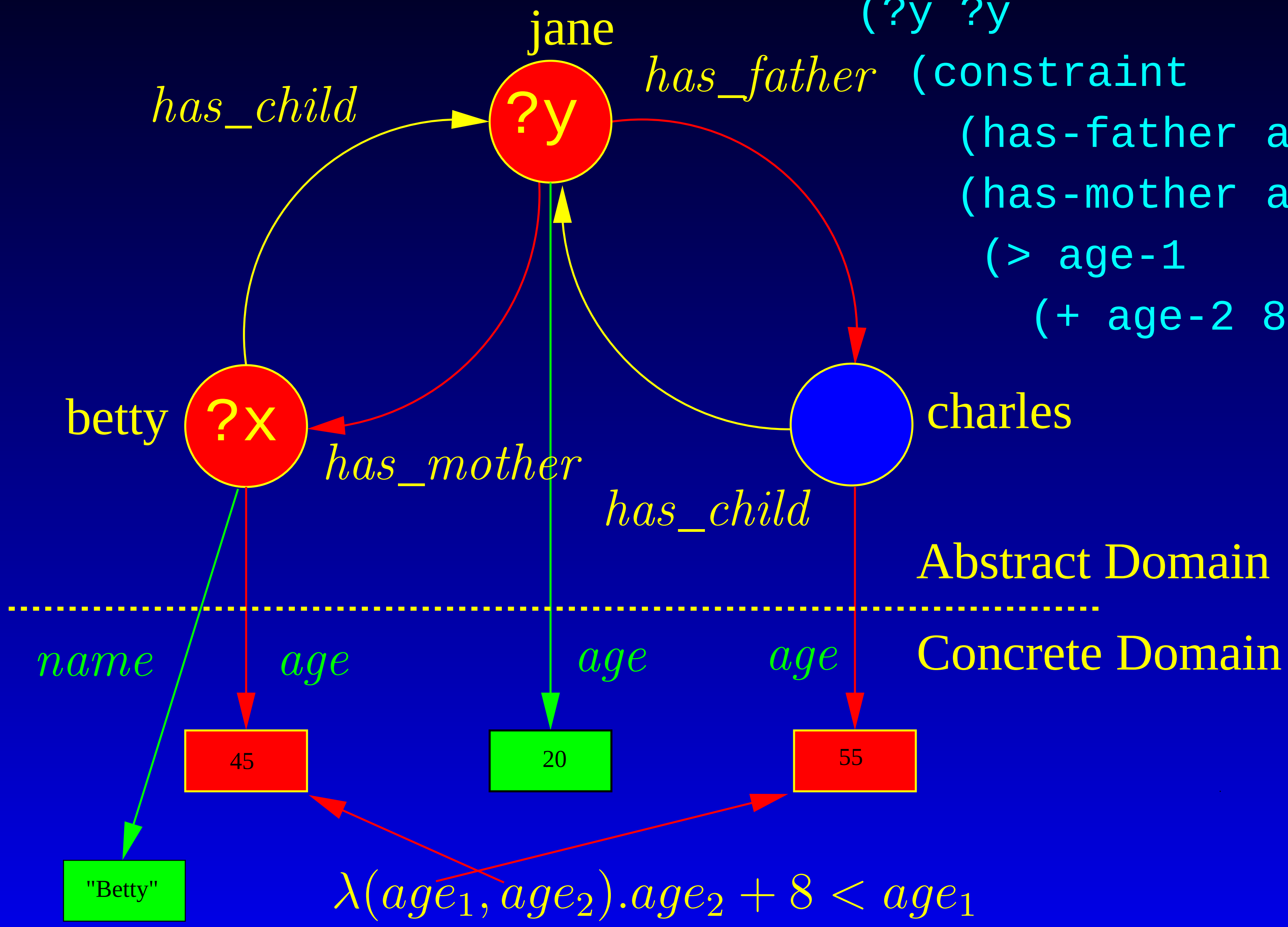


nRQL Language – Example



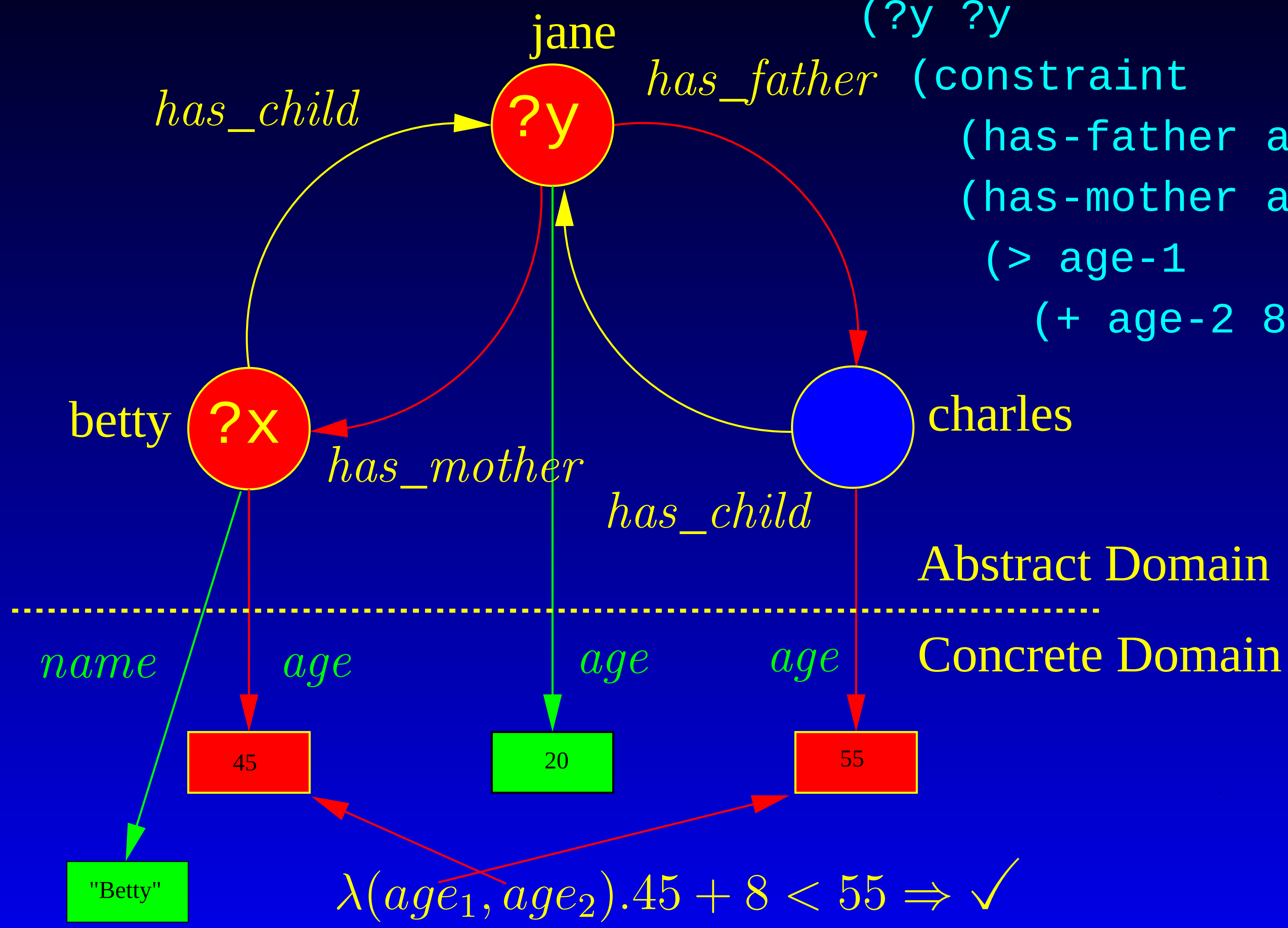
nRQL Language – Example

```
(?y ?y
 (constraint
   (has-father age)
   (has-mother age)
    (> age-1
      (+ age-2 8))))
```



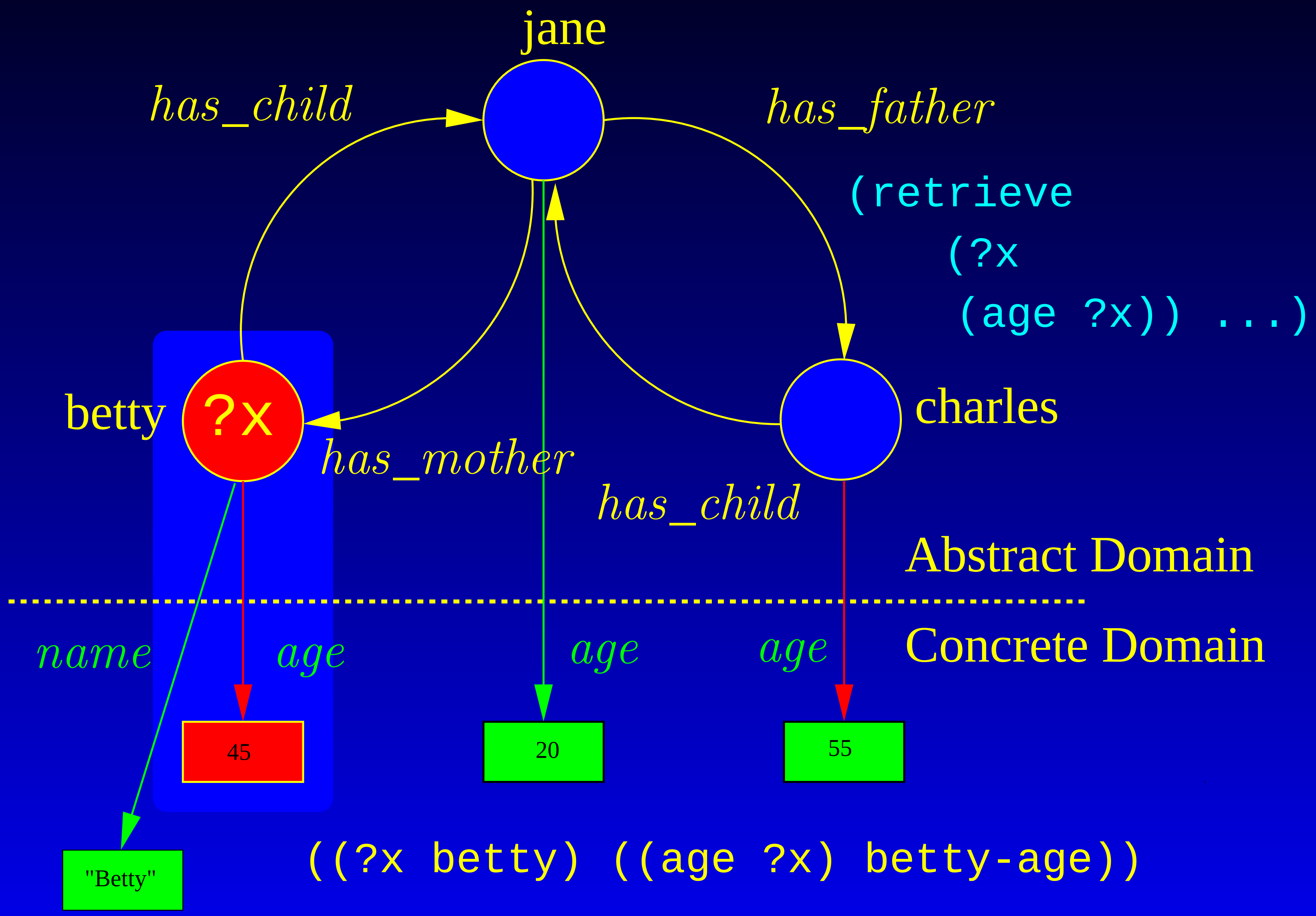
nRQL Language – Example

```
(?y ?y
 (constraint
   (has-father age)
   (has-mother age)
    (> age-1
      (+ age-2 8))))
```

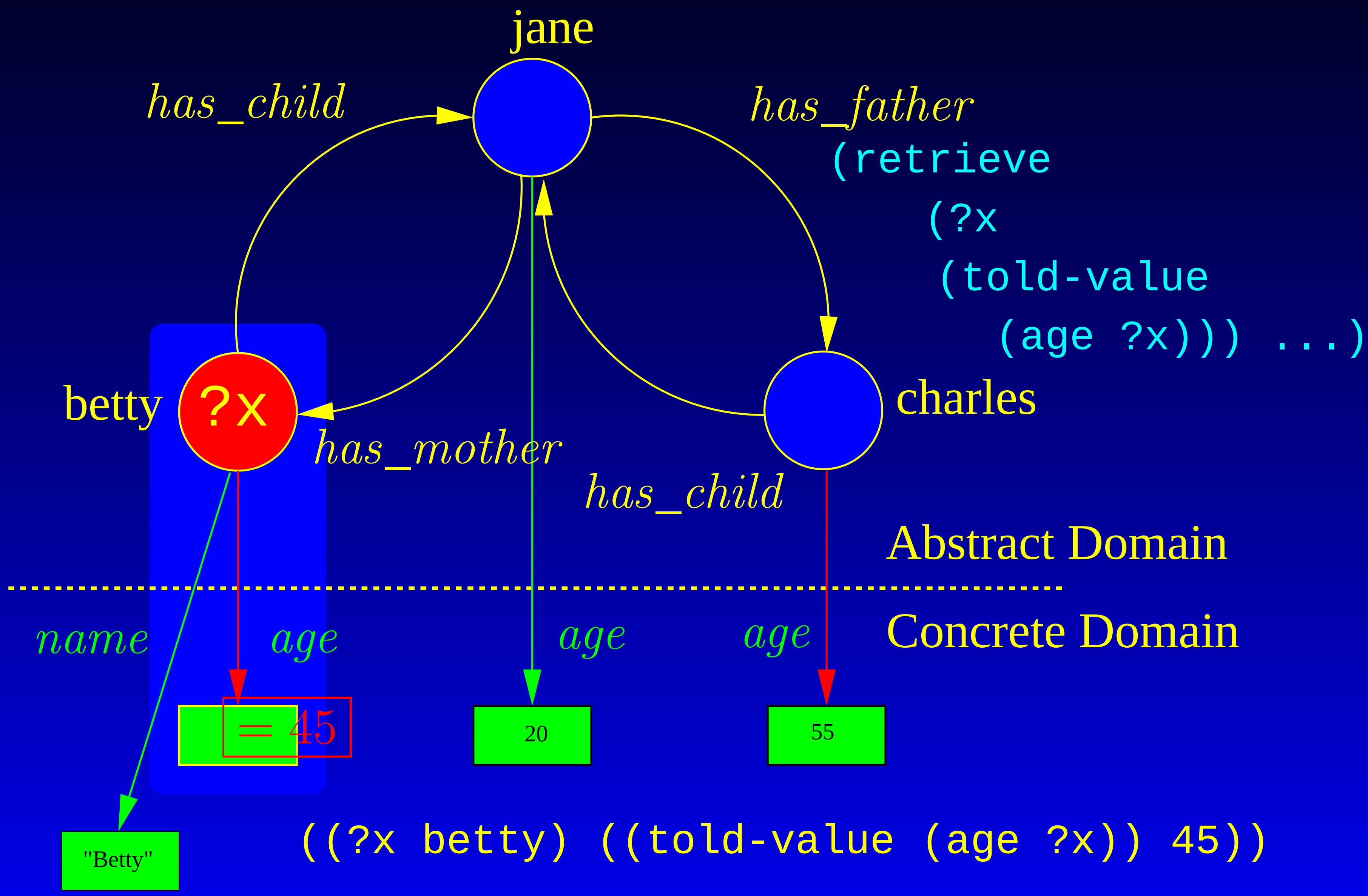


$\lambda(age_1, age_2).45 + 8 < 55 \Rightarrow \checkmark$

nRQL Language – Example



nRQL Language – Example



nRQL Language – Syntax

- Queries have a head and a body:
  `(retrieve <head> <body>)`
- Syntax for `<head>`

$$
\begin{array}{rcl}
head & := & (head\_entry^*) \\
head\_entry & := & object \mid head\_projection\_operator \\
object & := & variable \mid individual \\
variable & := & \text{a symbol beginning with “?”} \\
individual & := & \text{a symbol} \\
head\_projection\_operator & := &
\end{array}
$$

$$
\begin{array}{l}
(cd\_attribute\ object) \mid \\
(\texttt{told-value}\ (cd\_attribute\ object)) \mid \\
(\texttt{told-value}\ (datatype\_property\ object)) \mid \\
(\texttt{annotations}\ (annotation\_property\ object))
\end{array}
$$

nRQL Language – Syntax (2)

- Syntax for <body>

nRQL Language – Syntax (2)

- Syntax for <body>

$$\mathit{body} \quad := \quad \mathit{atom} \mid (\ \{\texttt{and} \mid \texttt{union}\}\ \mathit{body}^*) \mid (\ \{\ \texttt{neg} \mid \texttt{inv}\}\ \mathit{body}\ ) \mid$$
$$(\texttt{project-to}\ \ (\mathit{object}^*)\ \mathit{body})$$

nRQL Language – Syntax (2)

- Syntax for <body>

$$body \;:=\; atom \mid (\,\{\texttt{and} \mid \texttt{union}\}\; body^{*}) \mid (\,\{\,\texttt{neg} \mid \texttt{inv}\}\; body\,) \mid$$

$$(\texttt{project-to}\;\; (object^{*})\; body)$$

nRQL Language – Syntax (2)

- Syntax for `<body>`

$$body \;:=\; atom \mid (\{\texttt{and} \mid \texttt{union}\}\; body^*) \mid (\{\texttt{neg} \mid \texttt{inv}\}\; body) \mid$$
$$(\texttt{project-to}\; (object^*)\; body)$$

nRQL Language – Syntax (2)

- Syntax for <body>

$$body \;:=\; atom \mid (\,\{\texttt{and} \mid \texttt{union}\}\; body^*) \mid (\,\{\,\texttt{neg} \mid \texttt{inv}\}\; body\,) \mid$$
$$(\texttt{project-to}\;\; (object^*)\; body)$$

nRQL Language – Syntax (2)

- Syntax for `<body>`

$body \;:=\; atom \mid (\{\texttt{and} \mid \texttt{union}\}\; body^*) \mid (\{\texttt{neg} \mid \texttt{inv}\}\; body) \mid$
$\quad (\texttt{project-to}\; (object^*)\; body)$

$atom \;:=\; (object\; concept\_expr) \mid (object\; object\; role\_expr) \mid$
$\quad (object\; object\; (\texttt{constraint}\; chain\; chain\; constraint\_expr)) \mid$
$\quad (\texttt{same-as}\; variable\; individual)$

$chain \;:=\; (role\_expr^*\; cd\_attribute)$

Example concept query atoms

- `(?x woman)`
- `(betty woman)`

nRQL Language – Syntax (2)

- Syntax for <body>

$$body \;:=\; atom \mid (\{\texttt{and} \mid \texttt{union}\}\ body^*) \mid (\{\texttt{neg} \mid \texttt{inv}\}\ body) \mid$$
$$(\texttt{project-to}\ (object^*)\ body)$$

$$atom \;:=\; (object\ concept\_expr) \mid (object\ object\ role\_expr) \mid$$
$$(object\ object\ (\texttt{constraint}\ \ chain\ chain\ constraint\_expr)) \mid$$
$$(\texttt{same-as}\ \ variable\ individual)$$
$$chain \;:=\; (role\_expr^*\ cd\_attribute)$$

Example role query atoms

- `(?x ?y has-child)`
- `(betty ?child-of-betty has-child)`
- `(?x ?y (inv has-child))`
- `(?x ?y (not has-father))`

nRQL Language – Syntax (2)

- Syntax for <body>

$$body \;:=\; atom \mid (\{\texttt{and} \mid \texttt{union}\}\ body^*) \mid (\{\texttt{neg} \mid \texttt{inv}\}\ body) \mid$$
$$(\texttt{project-to}\ (object^*)\ body)$$

$$atom \;:=\; (object\ concept\_expr) \mid (object\ object\ role\_expr) \mid$$
$$(object\ object\ (\texttt{constraint}\ \ chain\ chain\ constraint\_expr)) \mid$$
$$(\texttt{same-as}\ \ variable\ individual)$$
$$chain \;:=\; (role\_expr^*\ cd\_attribute)$$

Example constraint query atoms

- ```
  (?x ?y (constraint (has-mother age)
                     (has-father age) <))
  ```
- ```
  (?x ?y (constraint (has-brother age)
                     (age)
                     (= age-1 (+ 8 age-2))))
  ```

nRQL Language – Syntax (2)

- Syntax for `<body>`

$$body \quad := \quad atom \mid (\ \{\texttt{and} \mid \texttt{union}\}\ body^*) \mid (\ \{\ \texttt{neg} \mid \texttt{inv}\}\ body\ ) \mid$$
$$(\texttt{project-to}\ \ (object^*)\ body)$$

$$atom \quad := \quad (object\ concept\_expr) \mid (object\ object\ role\_expr) \mid$$
$$(object\ object\ (\texttt{constraint}\ \ chain\ chain\ constraint\_expr)) \mid$$
$$(\texttt{same-as}\ \ variable\ individual)$$
$$chain \quad := \quad (role\_expr^*\ cd\_attribute)$$

Example `same-as` query atoms

- `(same-as ?x betty)`

nRQL Language – Syntax (2)

- Syntax for `<body>`

$$
\begin{aligned}
\mathit{body} &:= \mathit{atom} \mid (\{\texttt{and} \mid \texttt{union}\}\ \mathit{body}^*) \mid (\{\texttt{neg} \mid \texttt{inv}\}\ \mathit{body}) \mid \\
&\quad (\texttt{project-to}\ (\mathit{object}^*)\ \mathit{body})
\end{aligned}
$$

$$
\begin{aligned}
\mathit{atom} &:= (\mathit{object}\ \mathit{concept\_expr}) \mid (\mathit{object}\ \mathit{object}\ \mathit{role\_expr}) \mid \\
&\quad (\mathit{object}\ \mathit{object}\ (\texttt{constraint}\ \mathit{chain}\ \mathit{chain}\ \mathit{constraint\_expr})) \mid \\
&\quad (\texttt{same-as}\ \mathit{variable}\ \mathit{individual}) \\
\mathit{chain} &:= (\mathit{role\_expr}^*\ \mathit{cd\_attribute})
\end{aligned}
$$

---

concept_expr	:=	a RacerPro concept, with some extensions for OWL
role_expr	:=	a RacerPro role | (`inv` *role_expr*) | (`not` *role_e*
constraint_expr	:=	a RacerPro concrete predicate
cd_attribute	:=	a RacerPro concrete domain attribute
datatype_property	:=	a RacerPro role used as OWL datatype property
annotation_property	:=	a RacerPro role used as OWL annotation property

nRQL Variables

- Variables can only be bound to ABox individuals, not to concrete domain objects or even concrete domain values
- nRQL offers two kinds of variables: `?x`, `$?x`
  - `?x` prohibits binding to individuals which are already bound by other variables, e.g. `?y` (mapping must be injective)
  - "UNA" for variables

```
? (retrieve (?x ?y) (and (?x man) (?y man)))
> NIL
? (retrieve ($?x $?y) (and ($?x man)
                           ($?y man)))
> ((($?X CHARLES) ($?Y CHARLES)))
```

Complex nRQL Queries

- Compound nRQL queries are defined inductively
  - Every query atom `ai` is a body.
  - If `a1 ...an` are query bodies, then the following expressions are also bodies
    - `(neg ai)`
    - `(inv ai)`
    - `(and a1 ...an)`
    - `(union a1 ...an)`
    - `(project-to (objects-in-ai) ai)`
- Each variable creates a new axis in an n-dimensional tuple space
- A projection (specified by `<head>`) is made before that set is returned.

Illustration of Semantics

```
(retrieve (?x)
          (and (?x woman)
               (?y man)
               (?x ?y has-child)))
```

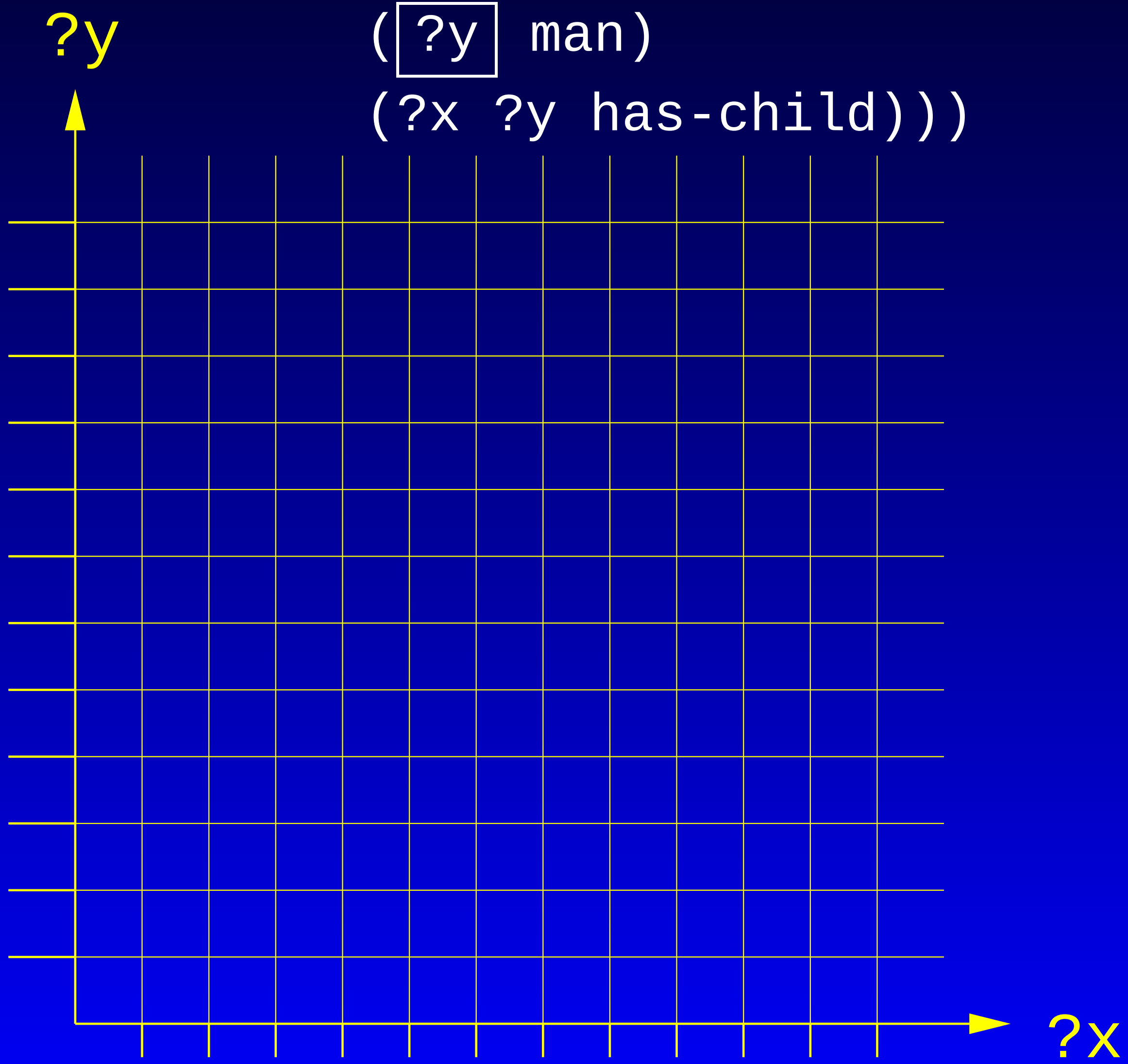


Illustration of Semantics

```
(retrieve (?x)
          (and (?x woman)
               (?y man)
               (?x ?y has-child)))
```

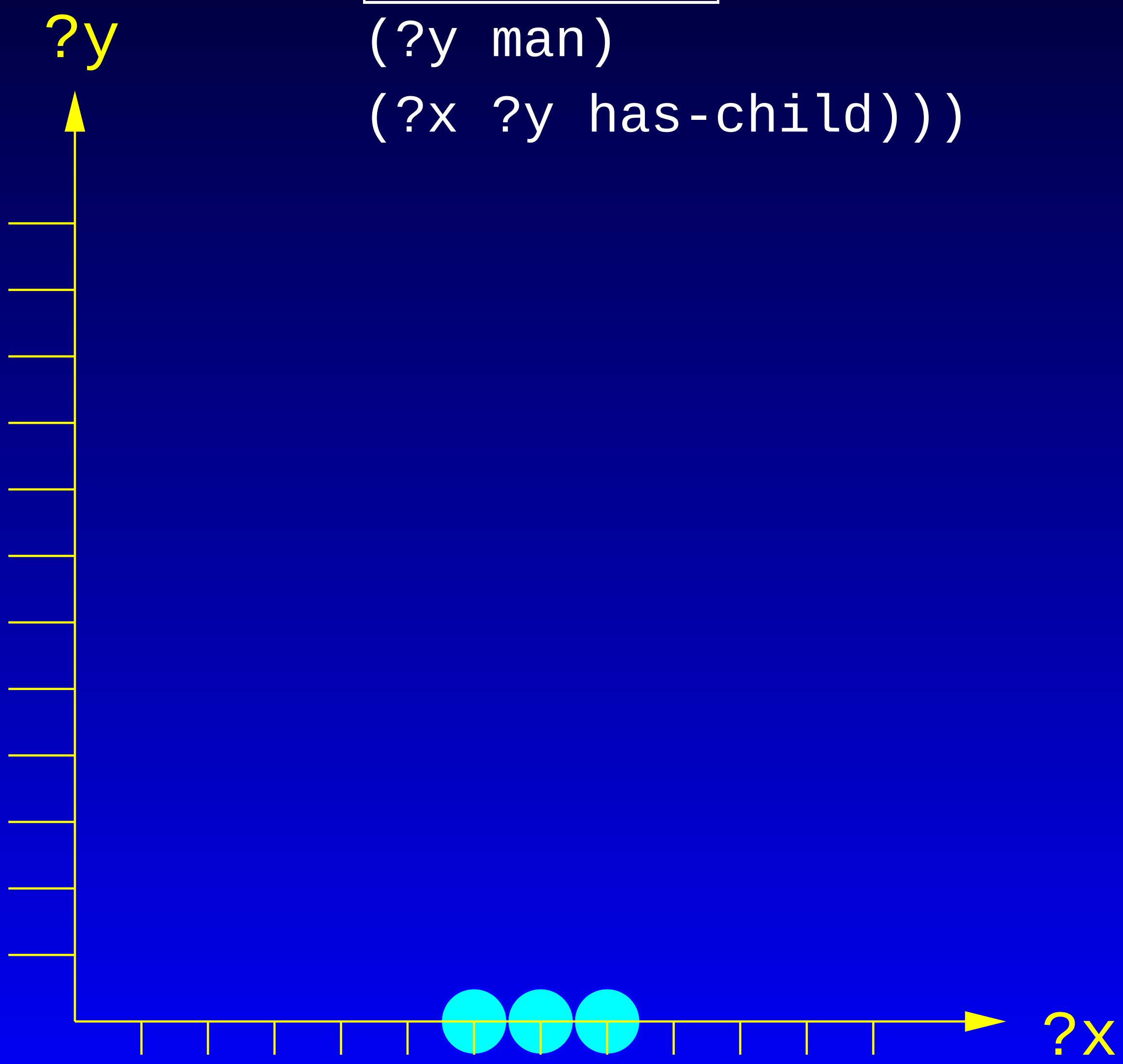


Illustration of Semantics

```
(retrieve (?x)
          (and (and (?x woman) (?y top))
               (?y man)
               (?x ?y has-child)))
```

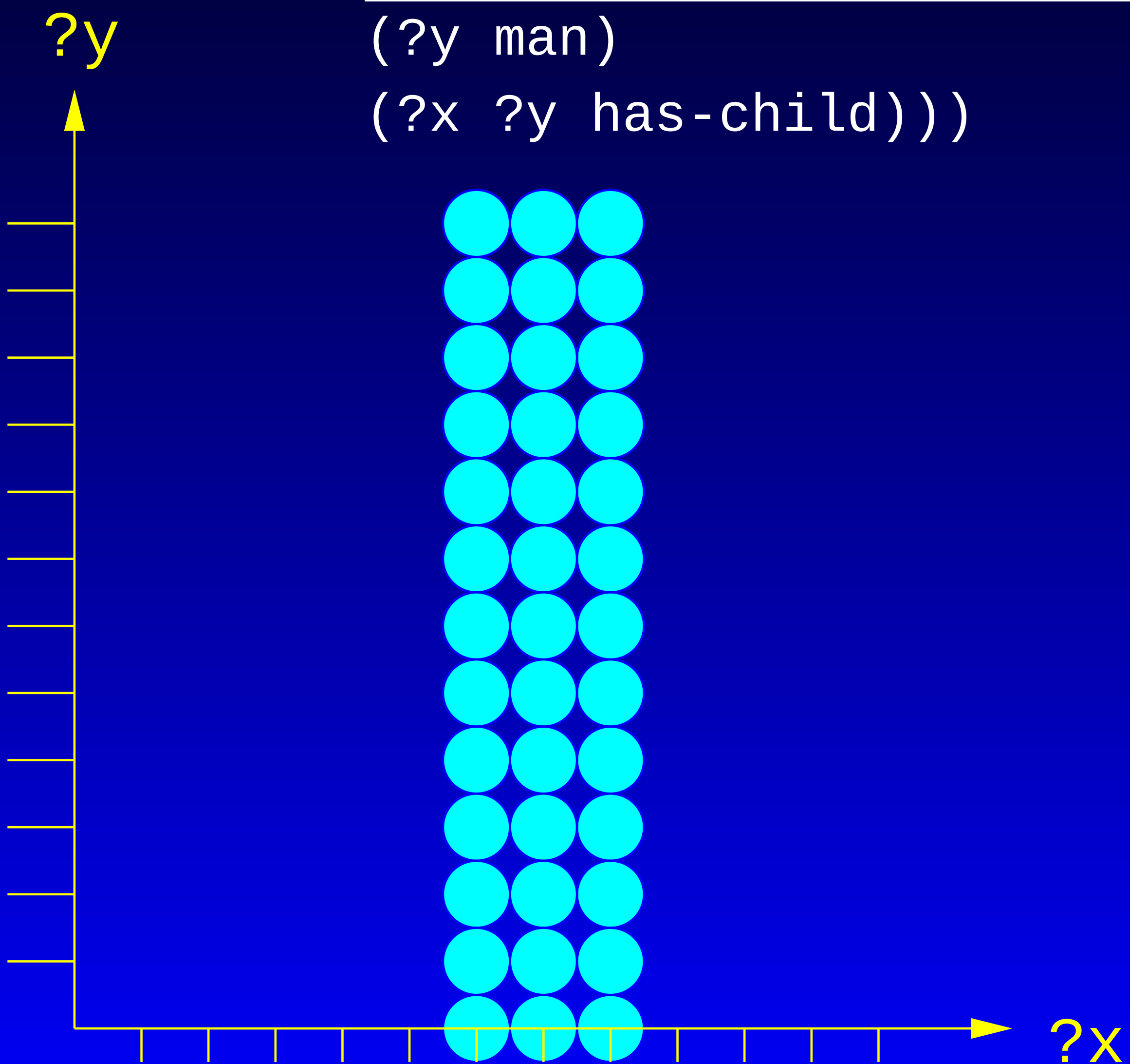


Illustration of Semantics

```
(retrieve (?x)
          (and (and (?x woman) (?y top))
               (?y man)
               (?x ?y has-child)))
```

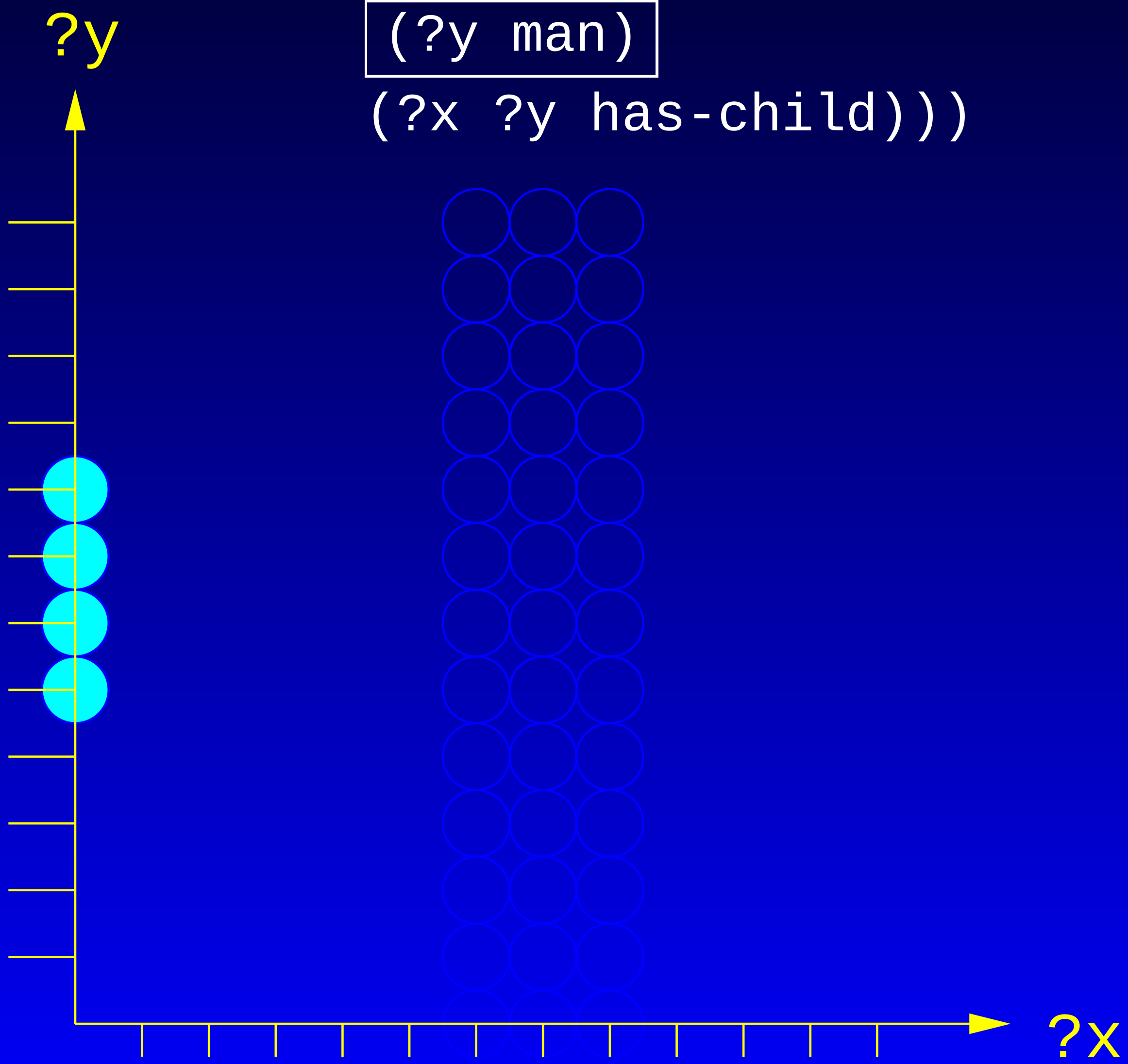


Illustration of Semantics

```
(retrieve (?x)
          (and (and (?x woman) (?y top))
               (and (?x top) (?y man))
               (?x ?y has-child)))
```

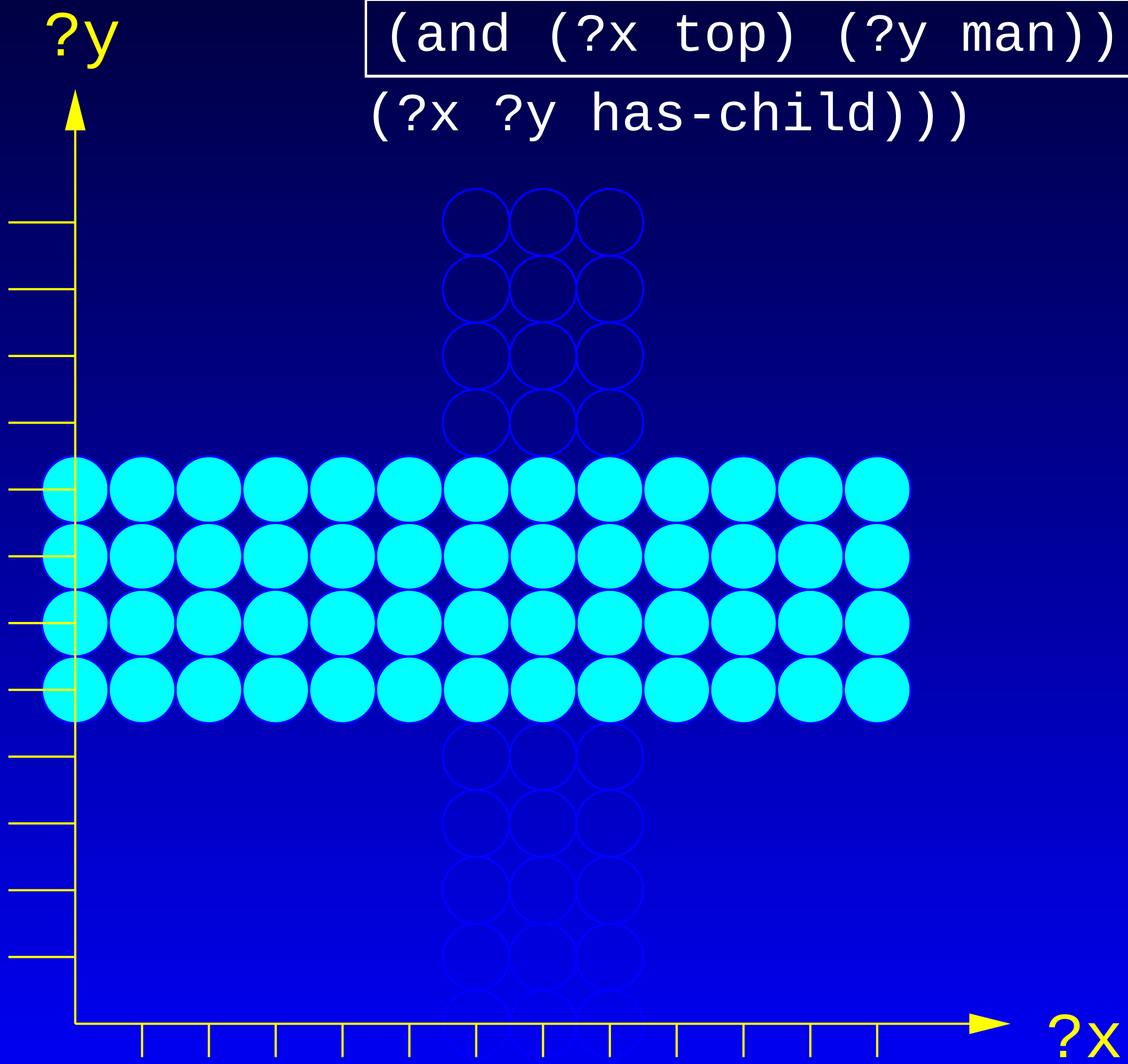


Illustration of Semantics

```
(retrieve (?x)
          ( and (and (?x woman) (?y top))
                (and (?x top) (?y man))
                (?x ?y has-child)))
```

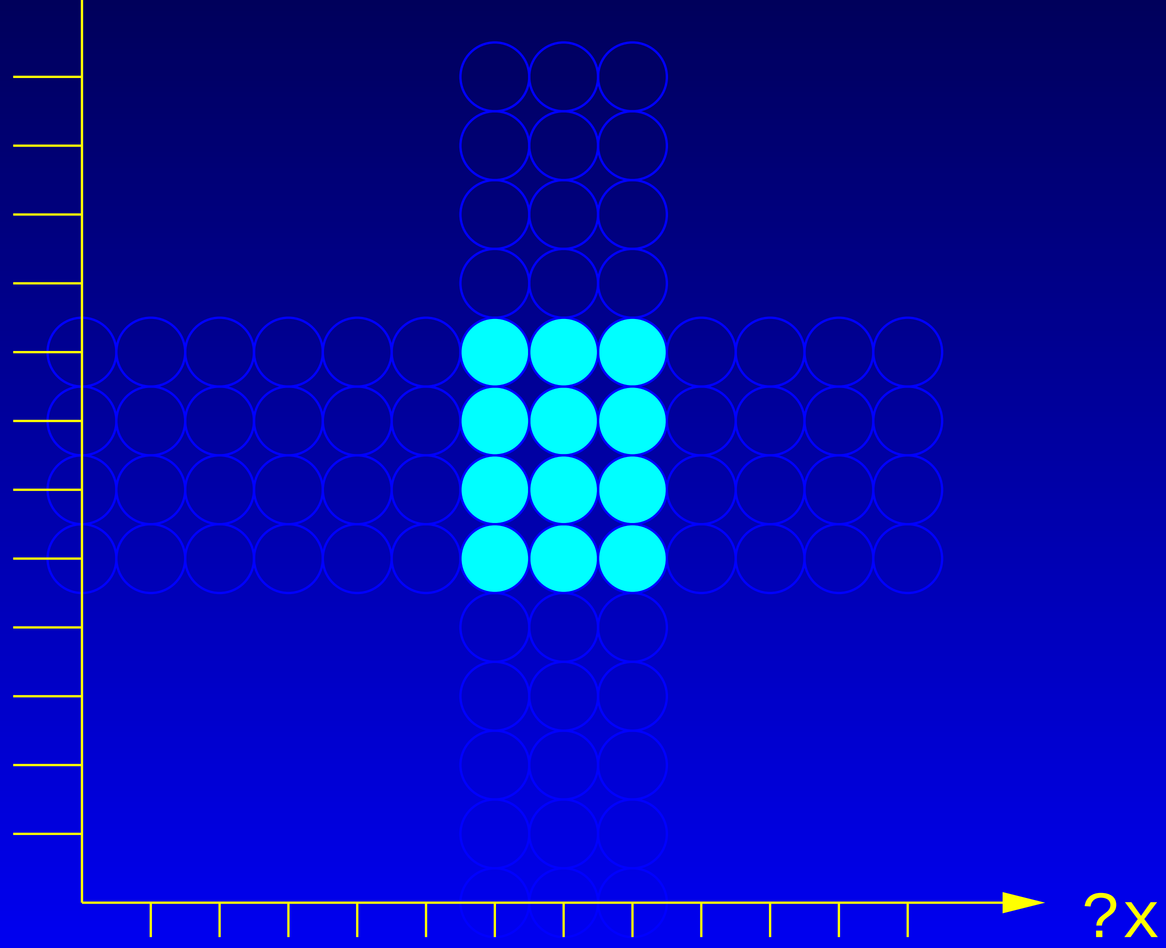


Illustration of Semantics

```
(retrieve (?x)
          (and (and (?x woman) (?y top))
               (and (?x top) (?y man))
               (?x ?y has-child)))
```

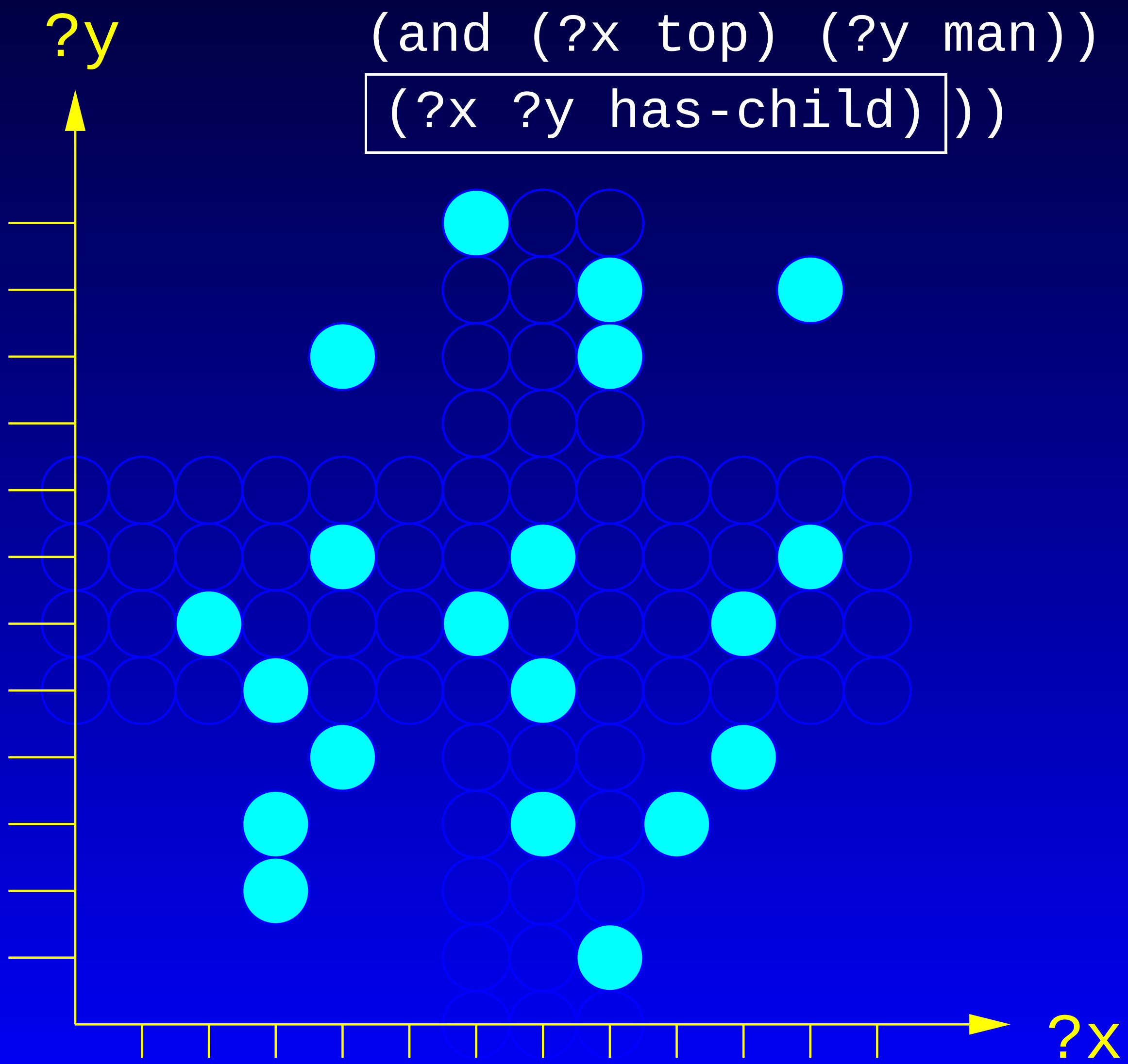


Illustration of Semantics

```
(retrieve (?x)
          ( and  (and (?x woman) (?y top))
                 (and (?x top) (?y man))
                 (?x ?y has-child)))
```

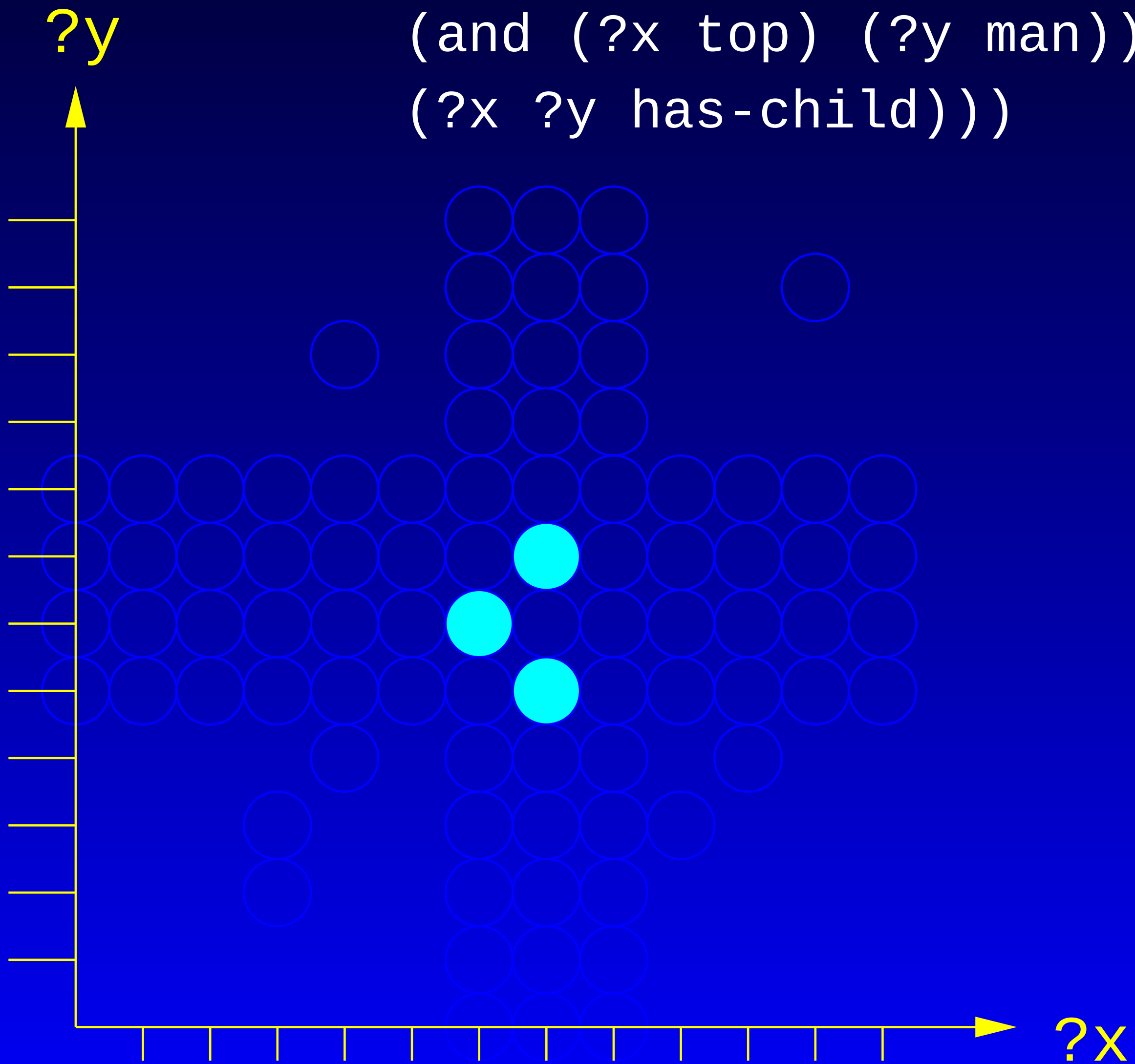
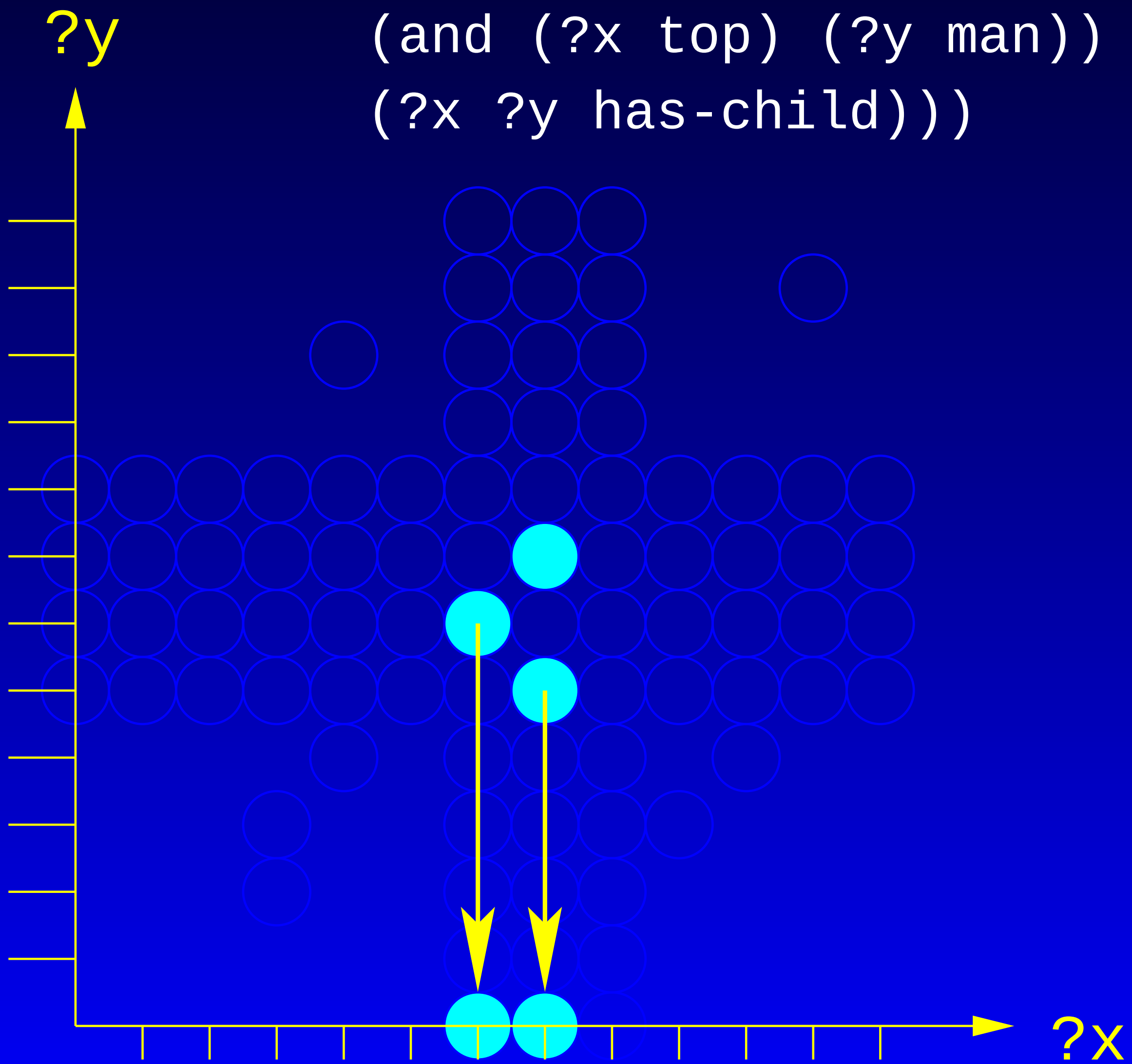


Illustration of Semantics

```
(retrieve (?x)
          (and (and (?x woman) (?y top))
               (and (?x top) (?y man))
               (?x ?y has-child)))
```



Negation in nRQL

- nRQL offers NAF with `neg`
- Semantics: simple set complement
  - well-defined for concept and role query atoms
  - well-defined for compound queries (DeMorgan etc.)
  - some “tricks” are needed for `same-as` and `constraint` query atoms
- Classical DL-like negation
  - obviously, in concept query atoms
  - but also in role query atoms `(?x ?y (not has-father))`

Illustration of neg

```
(retrieve (?x)
          (and (?x woman)
               (?y man)
               (?x ?y has-child)))
```

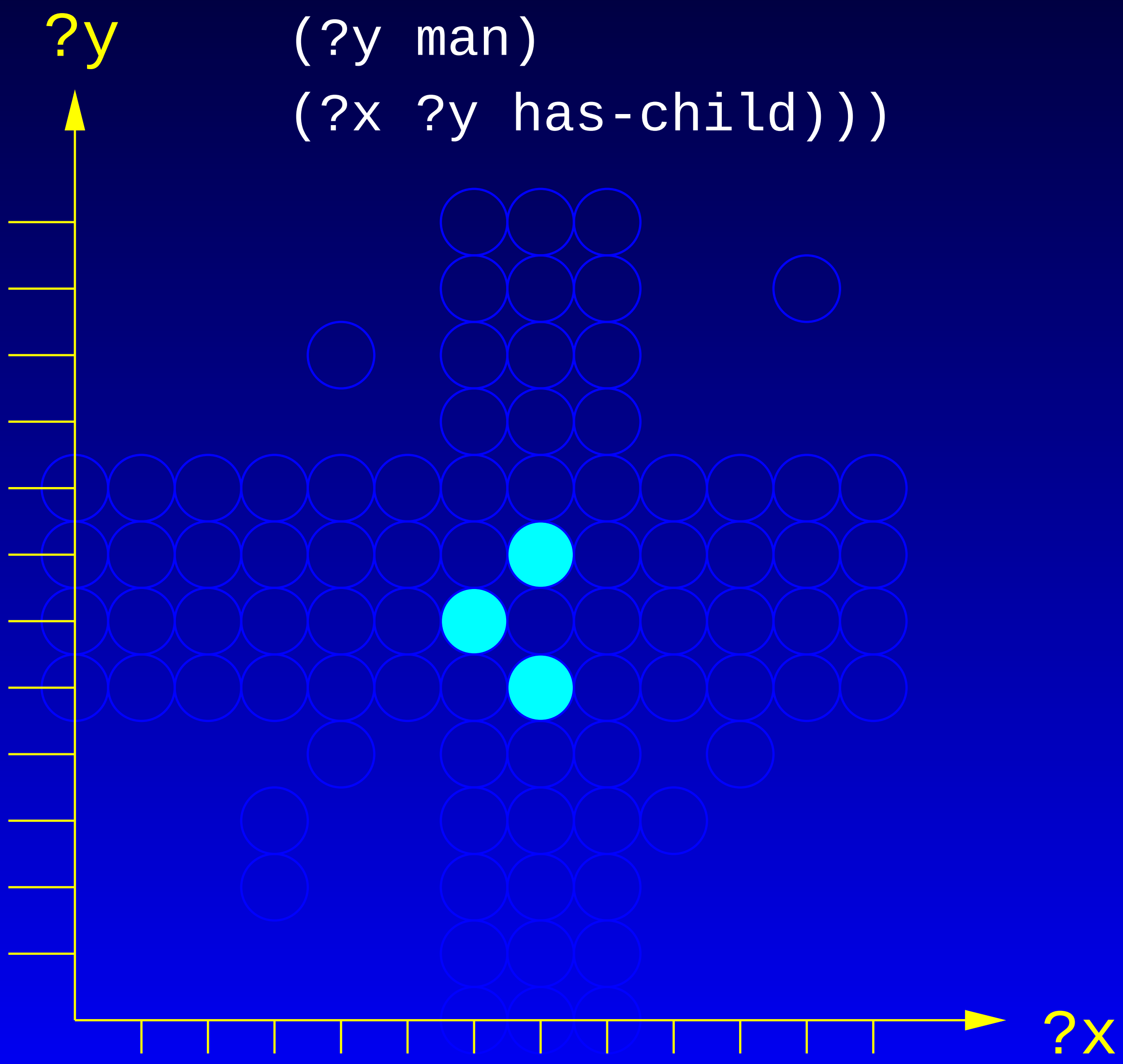


Illustration of neg

```
(retrieve (?x)
          (neg (and (?x woman)
                    (?y man)
                    (?x ?y has-child))))
```

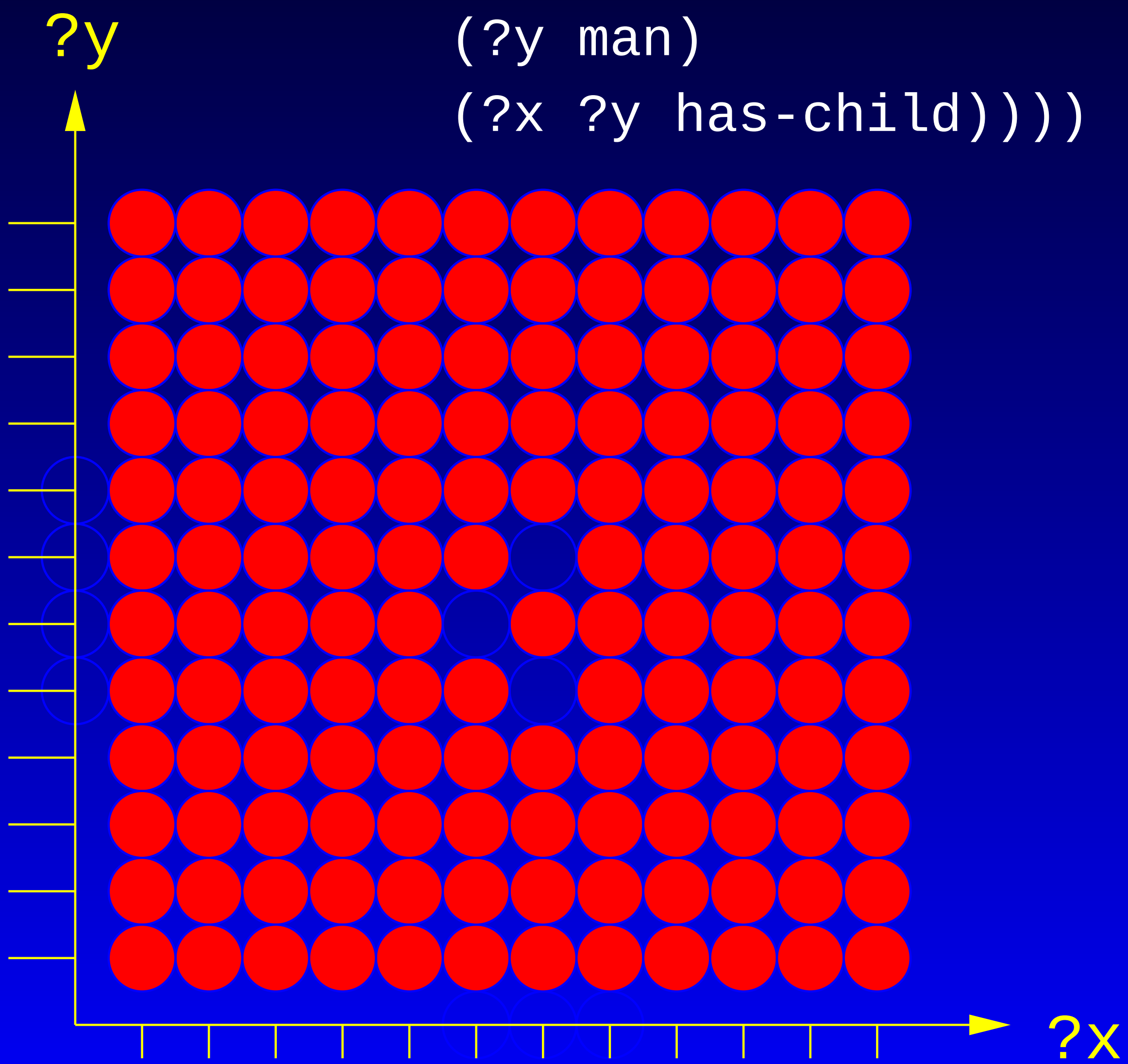


Illustration of `neg`

```
(retrieve (?x)
          (union (neg (?x woman))
                 (neg (?y man))
                 (neg (?x ?y has-child))))
```

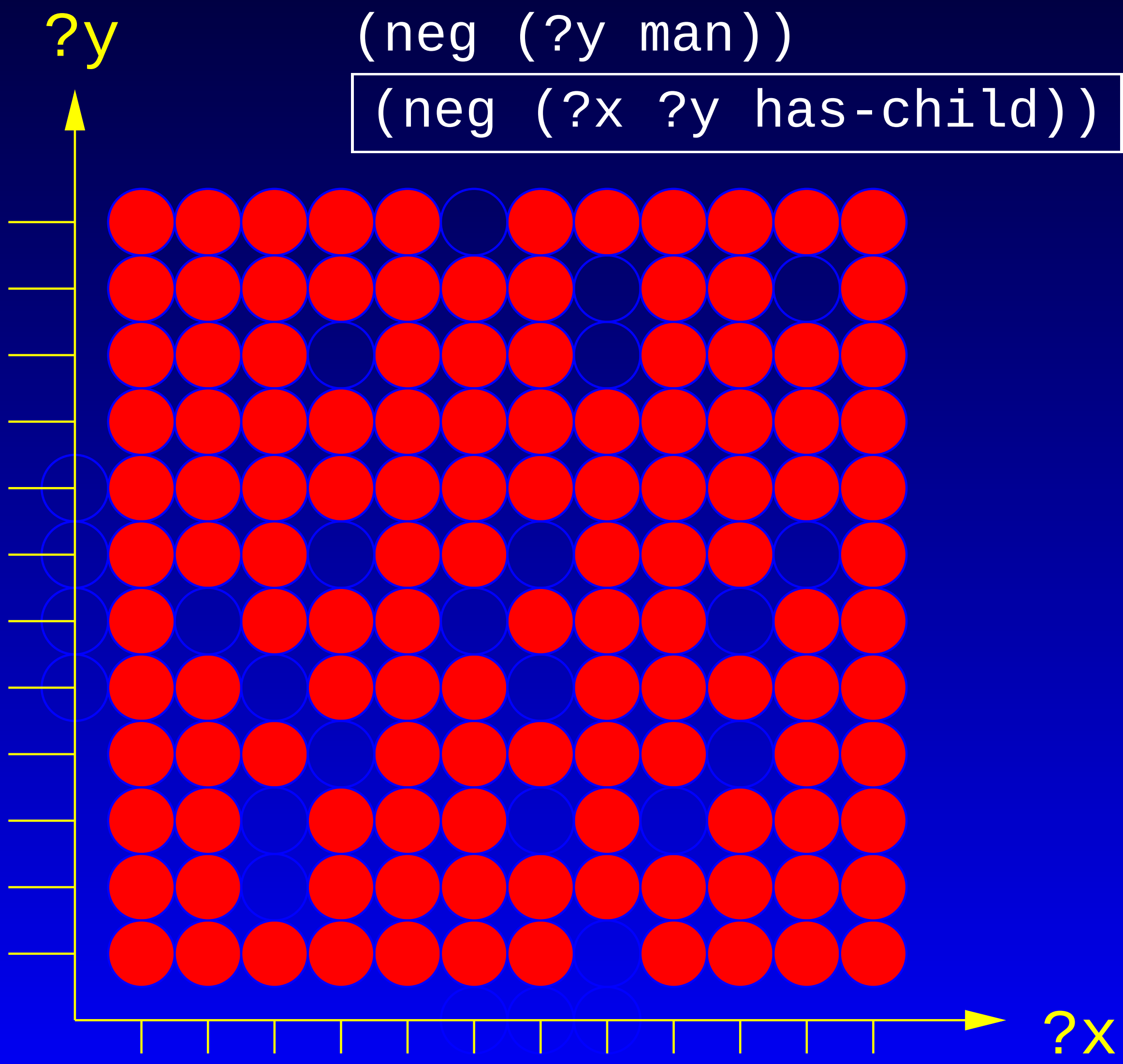


Illustration of neg

```
(retrieve (?x)
          (union (neg (?x woman))
                 (neg (?y man))
                 (neg (?x ?y has-child))))
```

?y

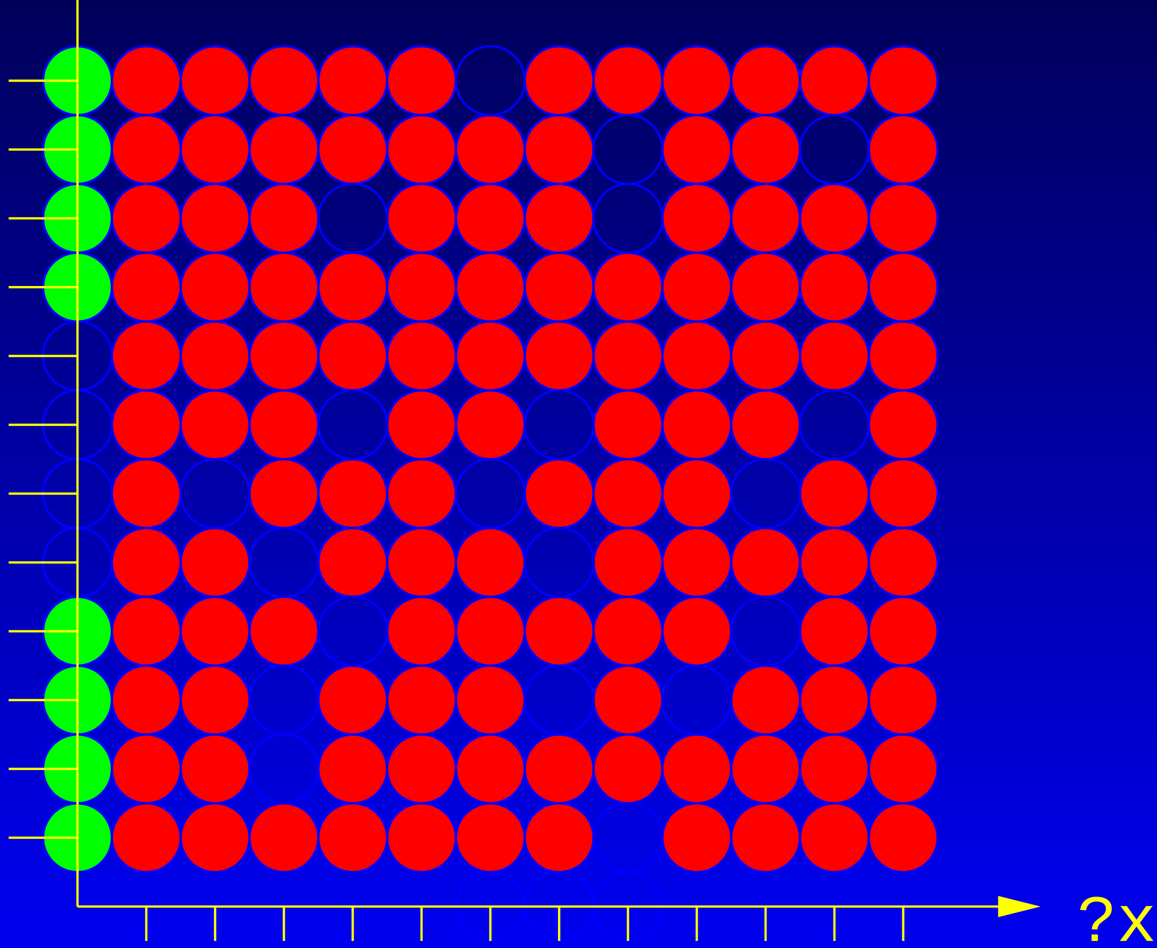


Illustration of `neg`

```
(retrieve (?x)
          (union (neg (?x woman))
                 (and (?x top) (neg (?y man)))
                 (neg (?x ?y has-child))))
```

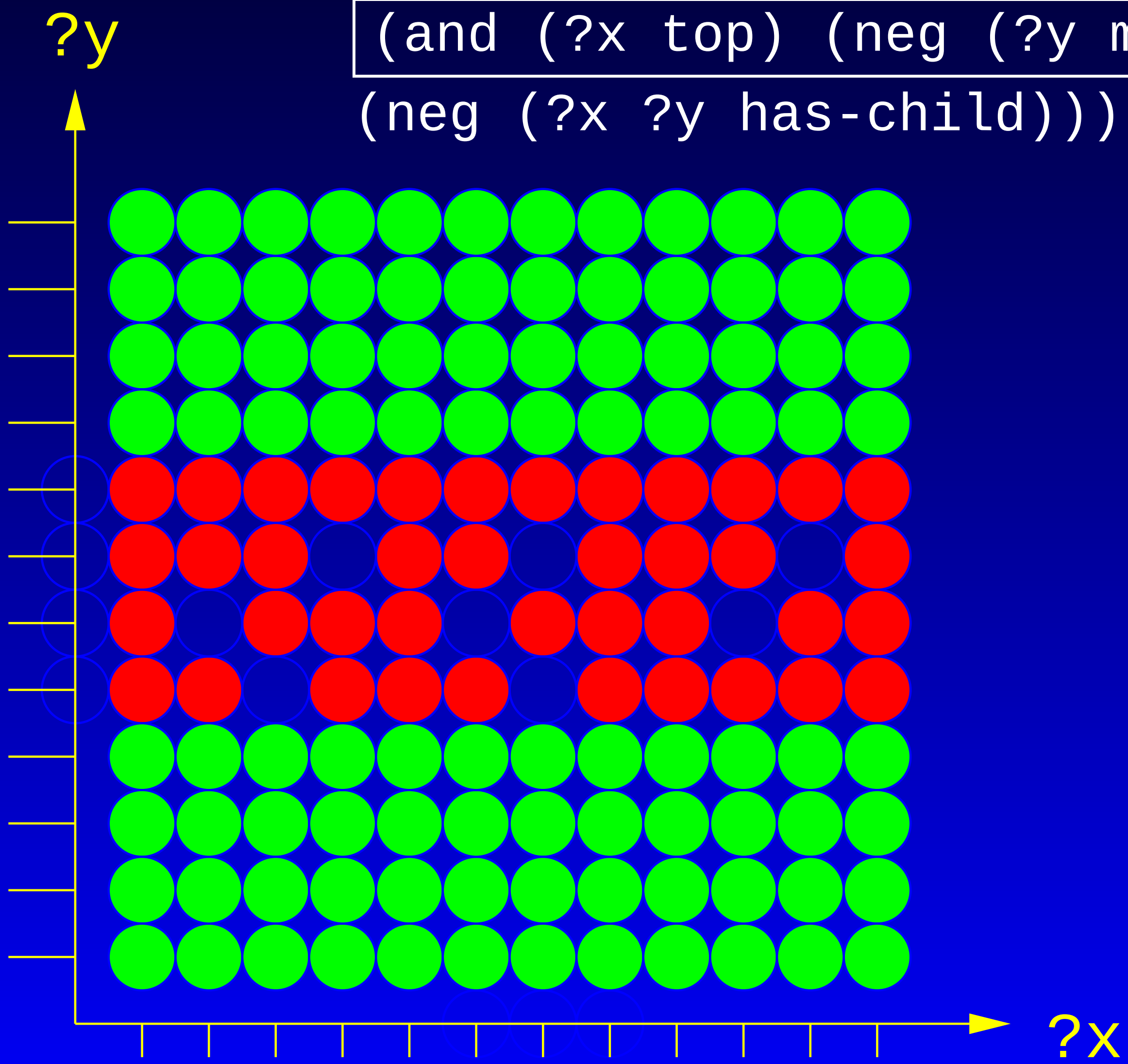


Illustration of neg

```
(retrieve (?x)
          (union (?x woman)
                 (and (?x top) (neg (?y man)))
                 (neg (?x ?y has-child))))
```

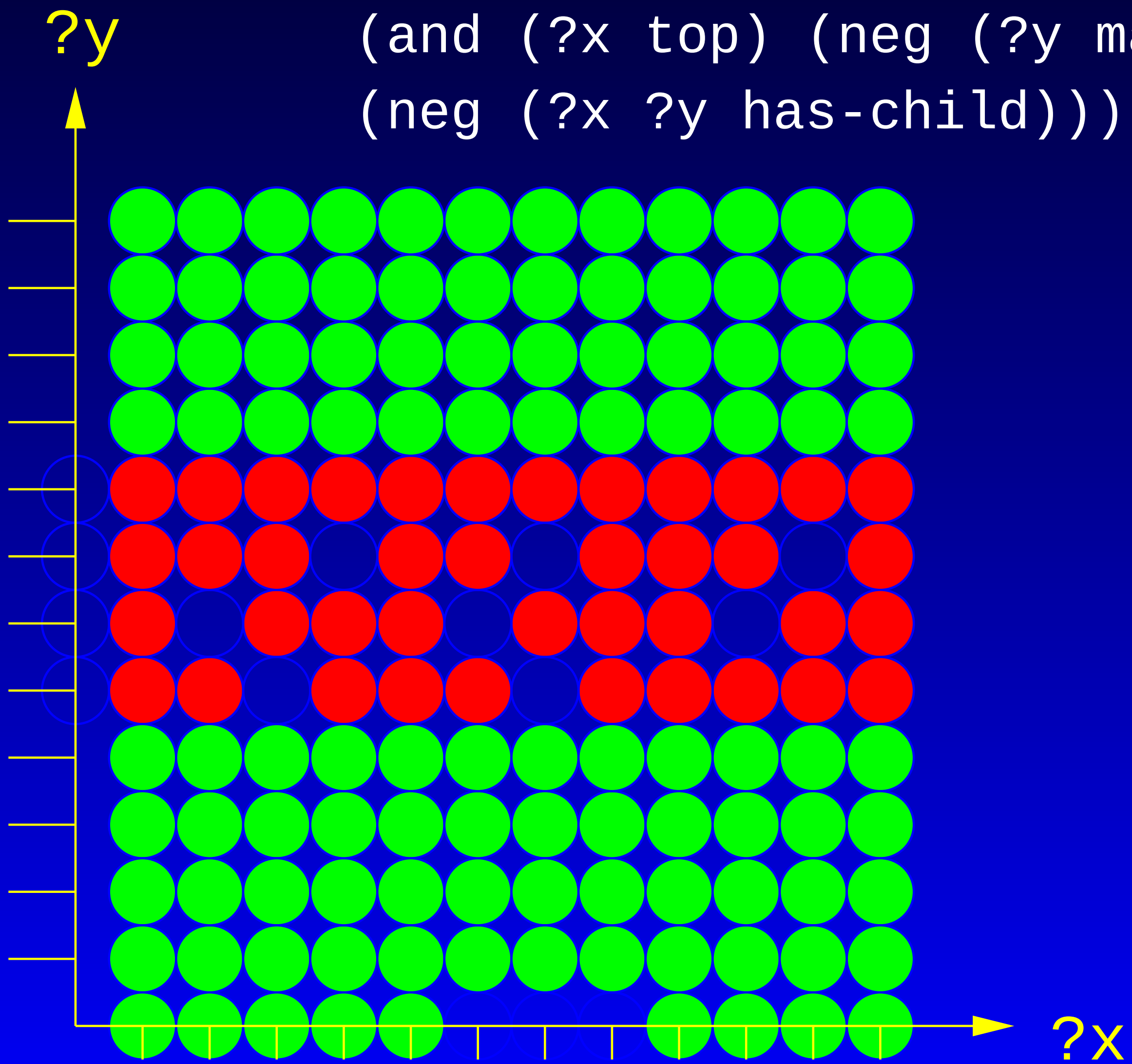


Illustration of `neg`

```
(retrieve (?x)
          (union (and (neg (?x woman) (?y top)))
                 (and (?x top) (neg (?y man)))
                 (neg (?x ?y has-child))))
```

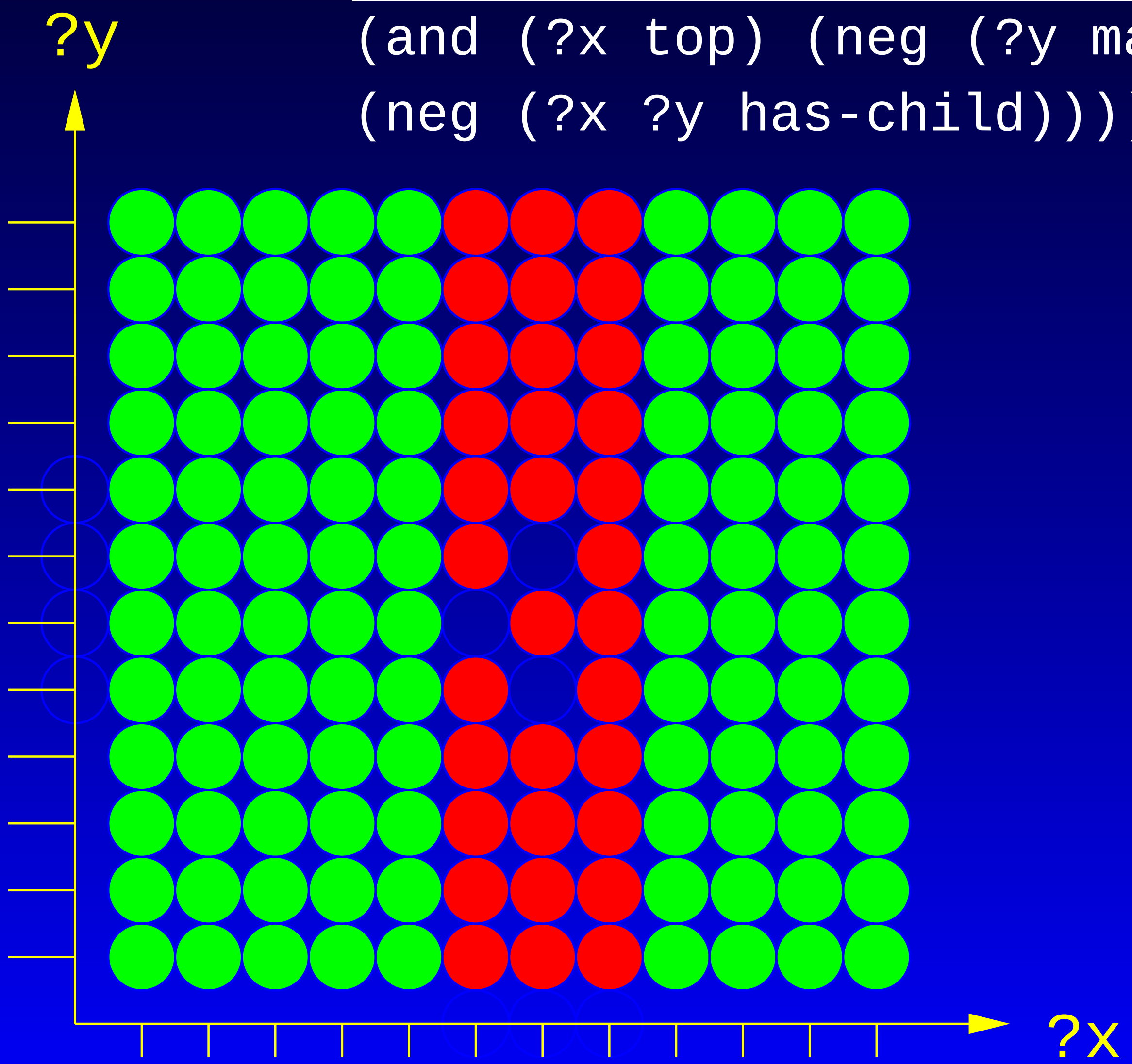


Illustration of neg

```
(retrieve (?x)
          ( union  (and (neg (?x woman) (?y top)))
                   (and (?x top) (neg (?y man)))
                   (neg (?x ?y has-child))))
```

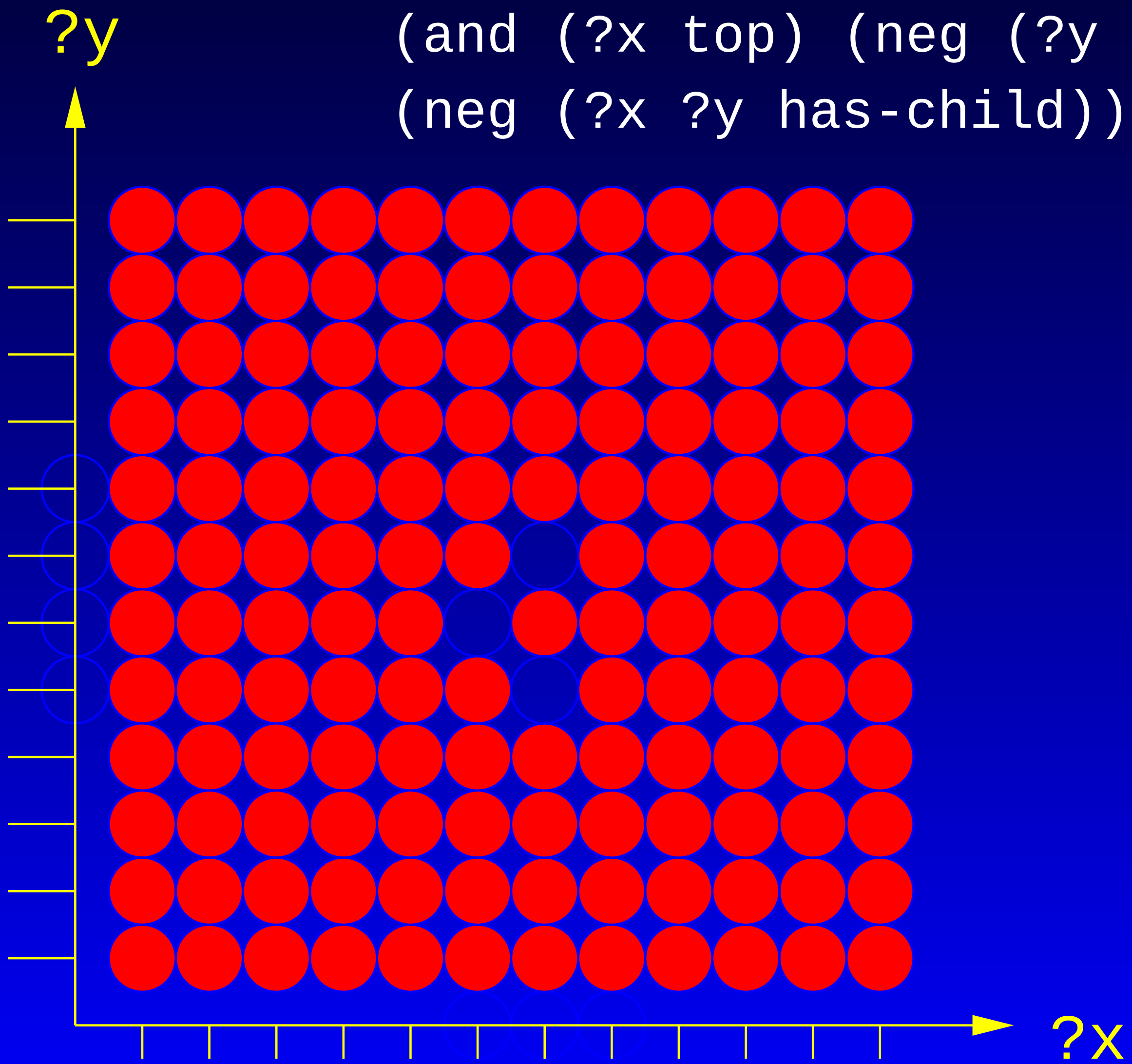


Illustration of neg

```
(retrieve (?x)
          (and (?x woman)
               (?y man)
               (?x ?y has-child)))
```

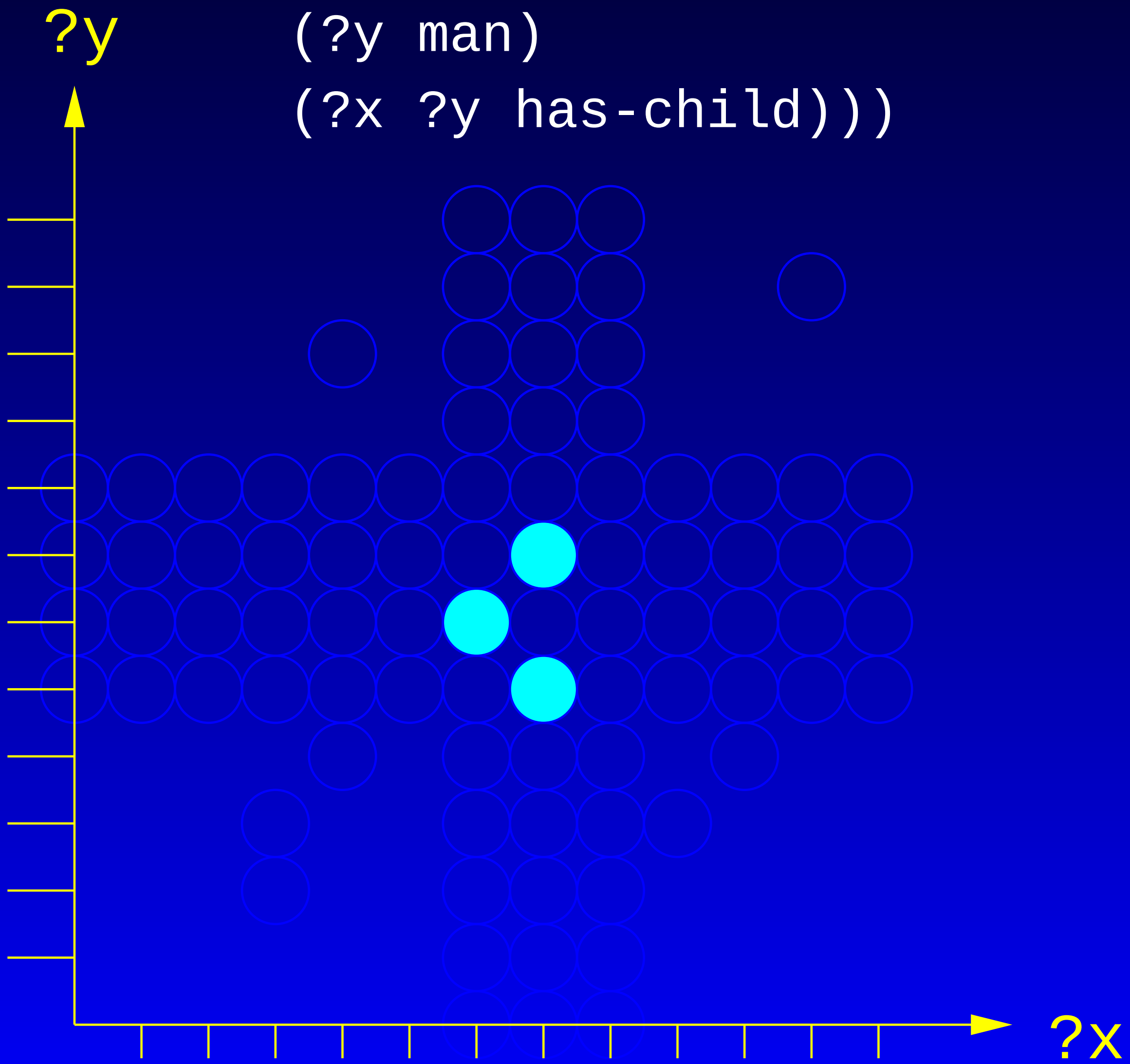
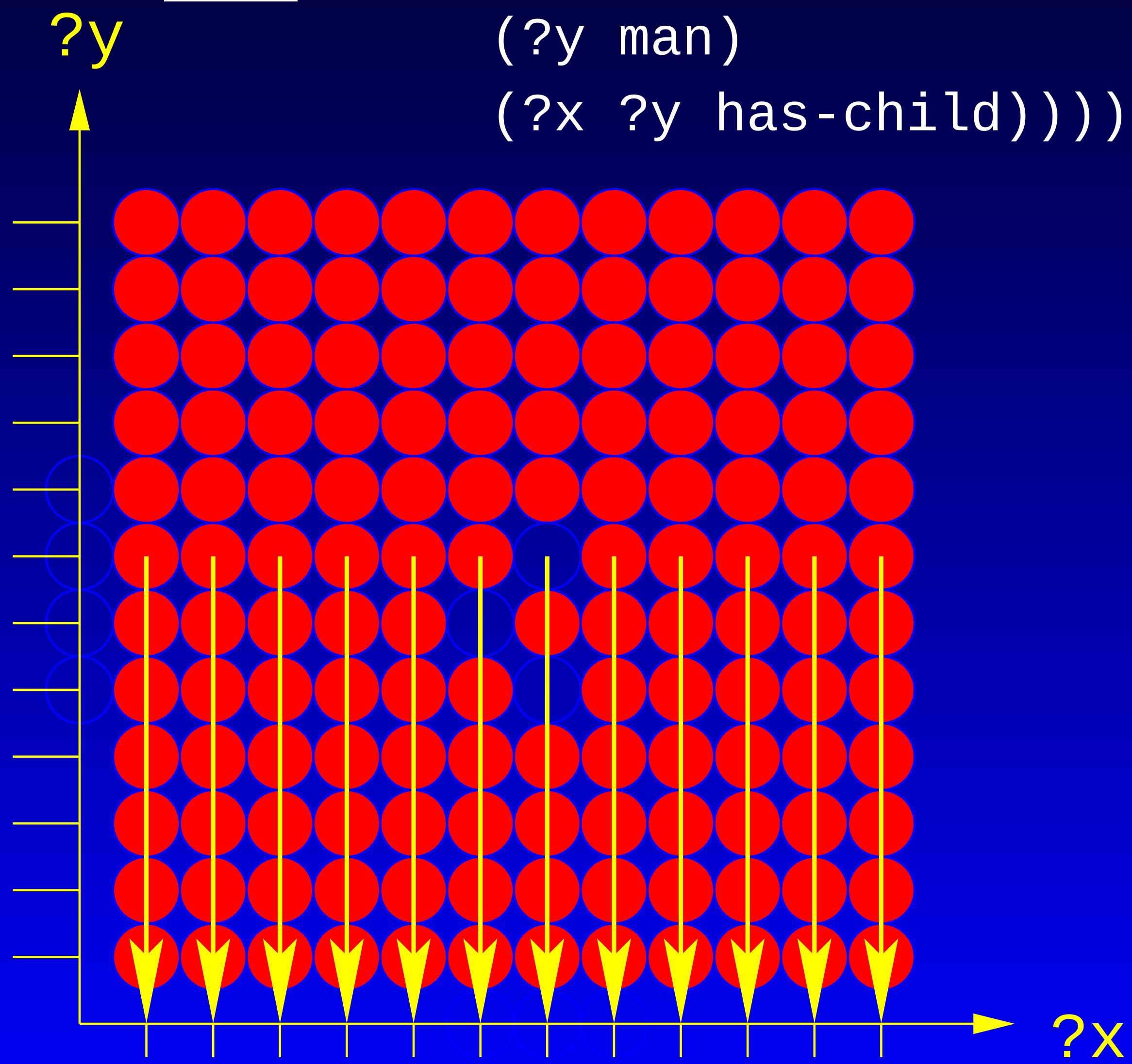


Illustration of neg

```
(retrieve (?x)
          ( neg  (and (?x woman)
                      (?y man)
                      (?x ?y has-child))))
```



TUHH Technische Universität Hamburg-Harburg

Illustration of `neg`

```
(retrieve (?x)
          (neg (and (?x woman)
                    (?y man)
                    (?x ?y has-child))))
```

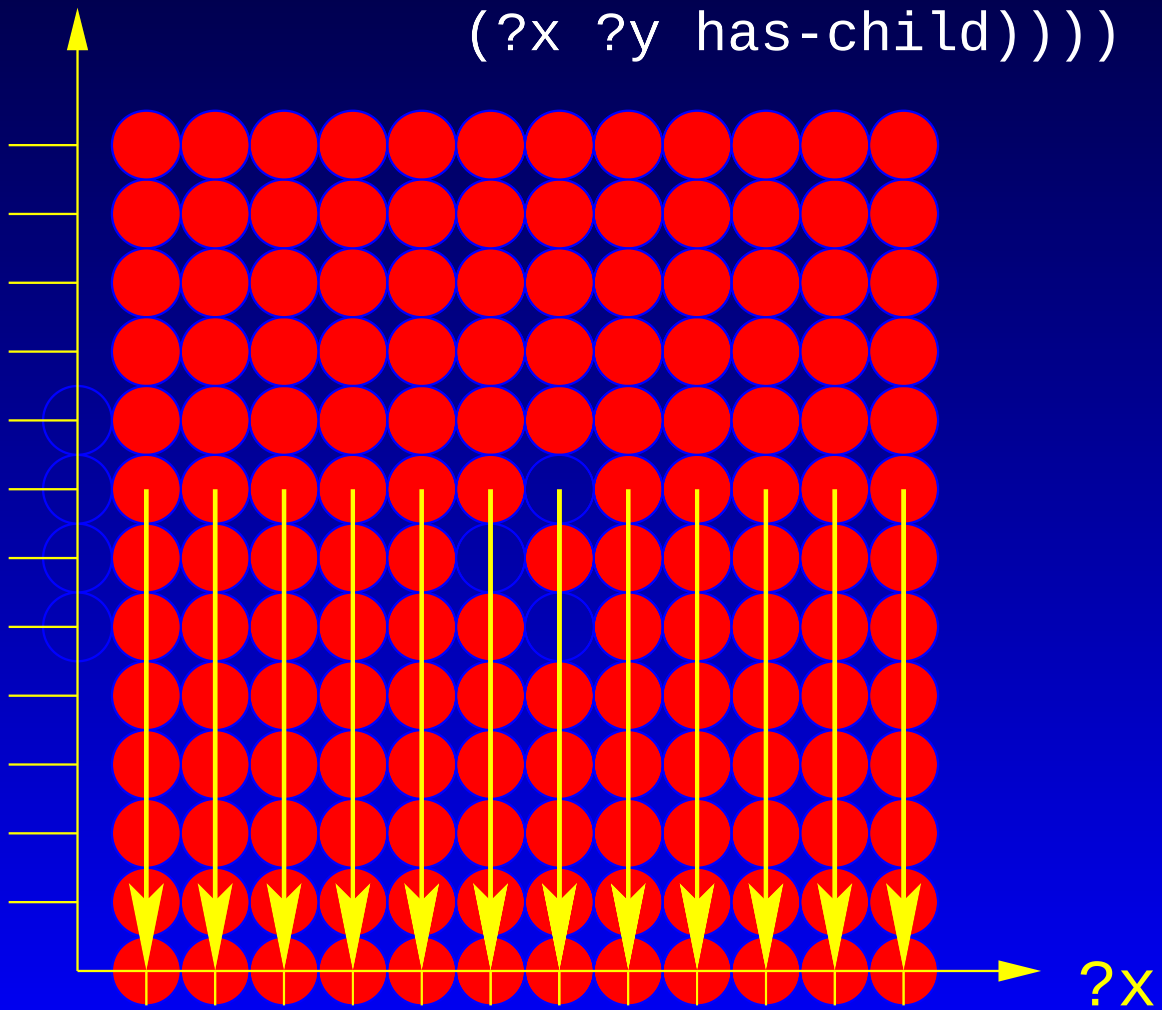


Illustration of neg

```
(retrieve (?x)
          (and (?x woman)
               (?y man)
               (?x ?y has-child)))
```

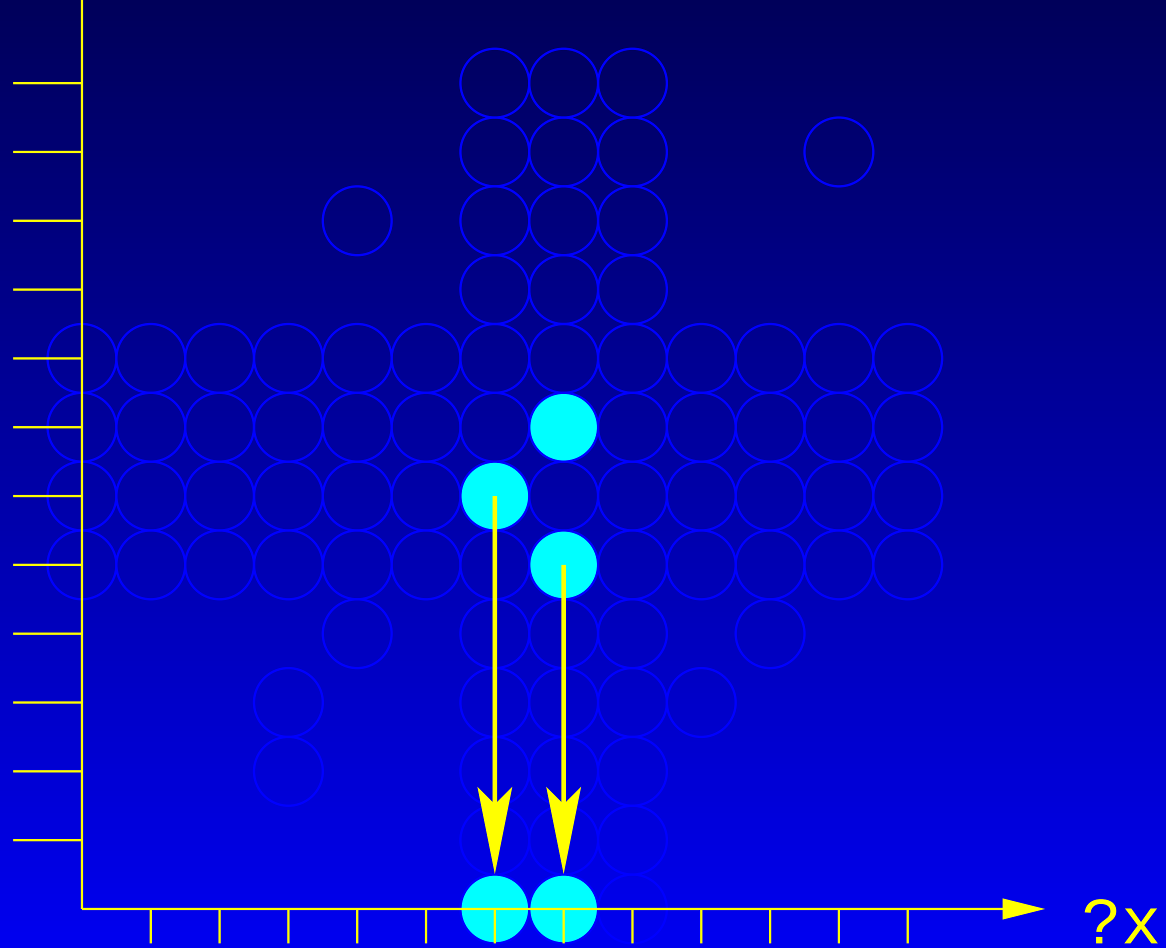


Illustration of neg

```
(NEG (retrieve (?x)
               (and (?x woman)
                    (?y man)
                    (?x ?y has-child))))
```

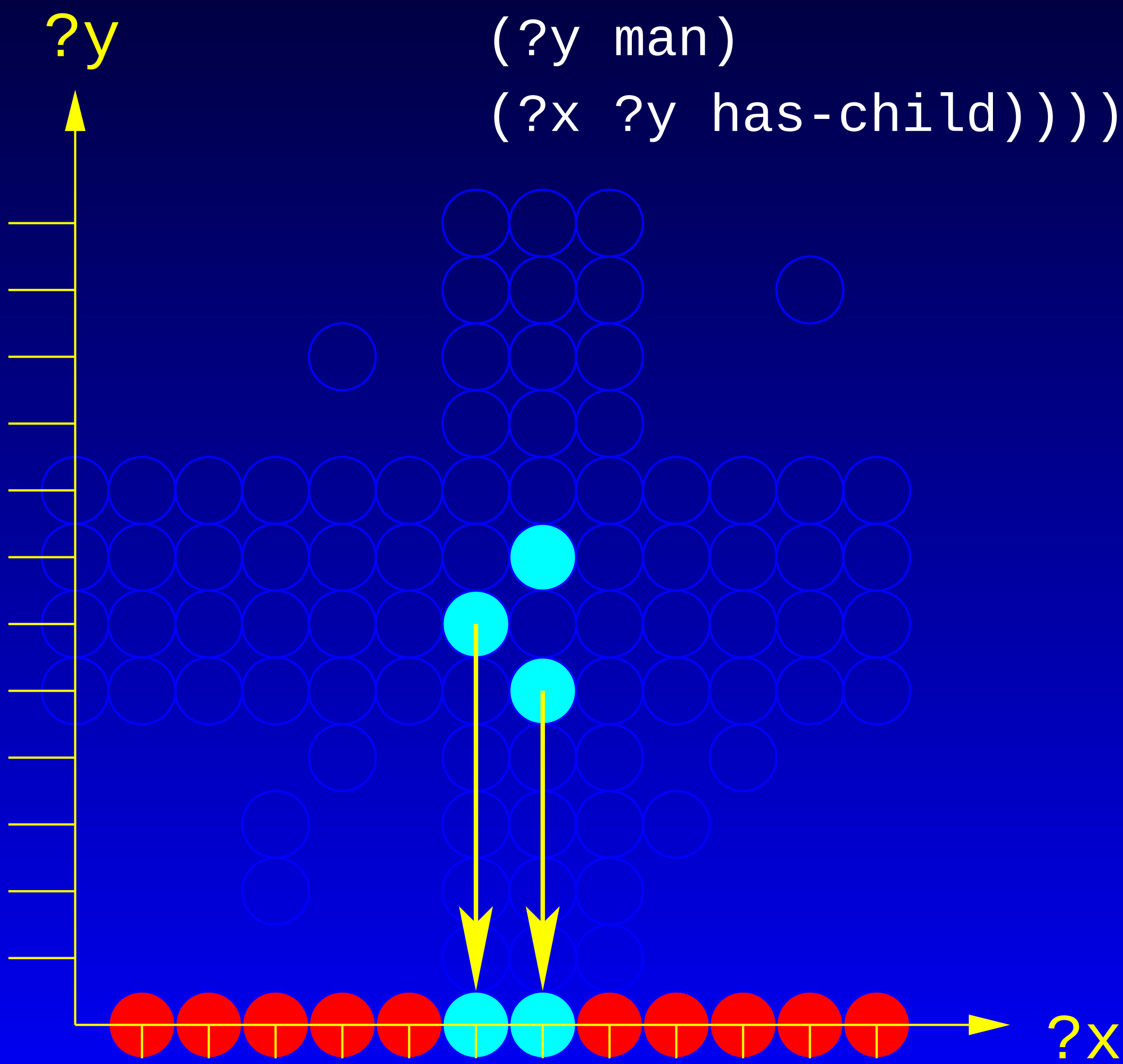
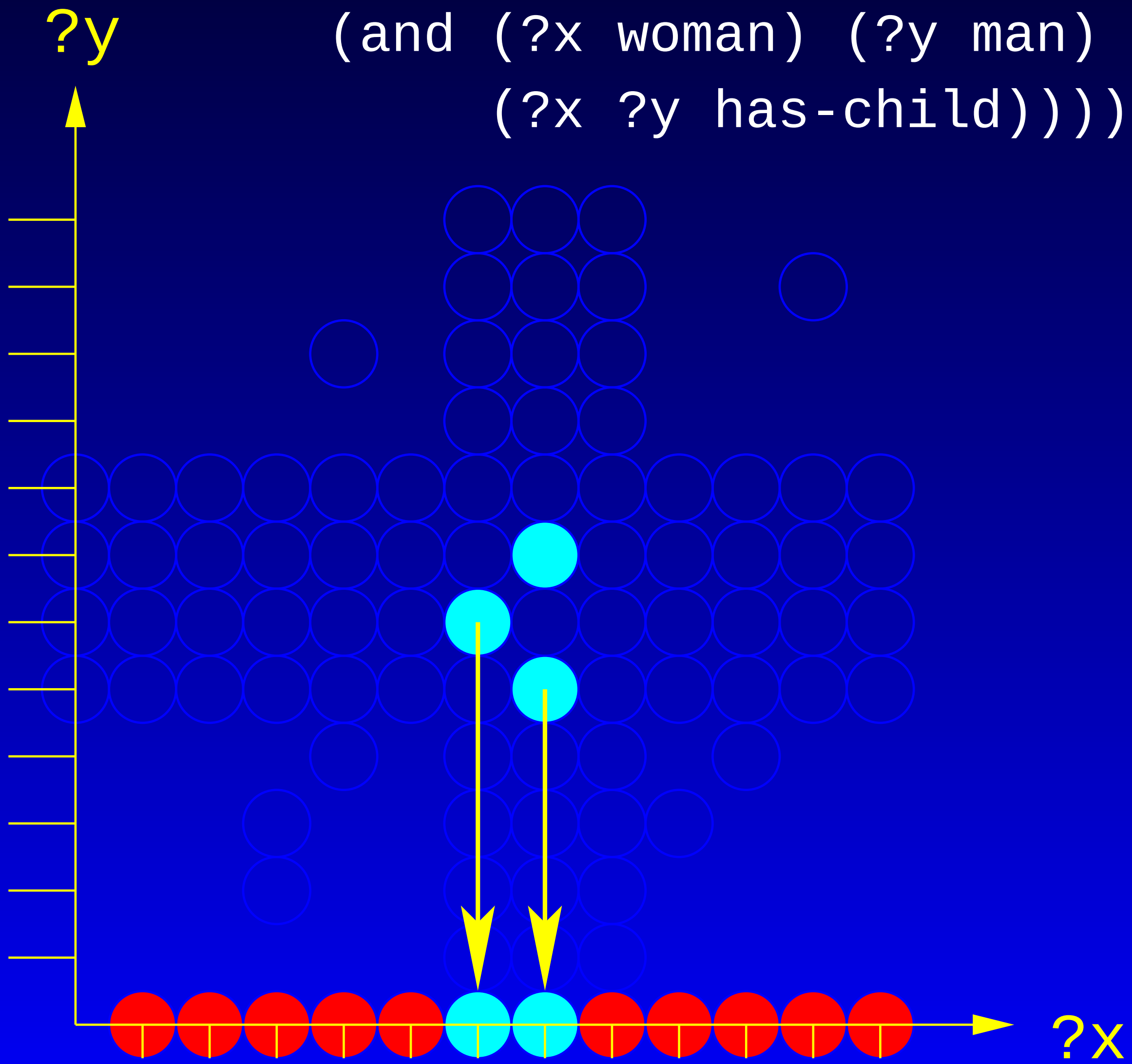


Illustration of neg

```
(retrieve (?x)
          (neg (project-to (?x)
                   (and (?x woman) (?y man)
                        (?x ?y has-child))))
```



Queries with Individuals

```
? (retrieve (betty)
            (betty woman))
```

```
> ((($?BETTY BETTY)))
```

- Explanation: query is rewritten into

```
(AND (SAME-AS $?BETTY BETTY)
     ($?BETTY WOMAN))
```

- `$?BETTY` is a variable that does not obey the unique name assumption for variables
- `(SAME-AS $?BETTY BETTY)` enforces binding of `$?BETTY` to `BETTY`

Semantic Consequences

- “NAF” for atoms with individuals can be tricky

```
(retrieve (betty)
          (neg (betty woman)))
=
(retrieve ($?betty)
    (neg (and ($?betty woman)
              (same-as $?betty betty))))
=
(retrieve ($?betty)
    (UNION (neg ($?betty woman))
           (neg (same-as $?betty betty))))
```

- one must define the semantics in such a way if the orthogonality of the language shall be preserved!

The Projection Operator

- We can retrieve all woman having children with

```
(retrieve (?x)
  (and (?x woman) (?x ?y has-child)))
```

- How can we retrieve woman which have <u>no</u> (known) children?

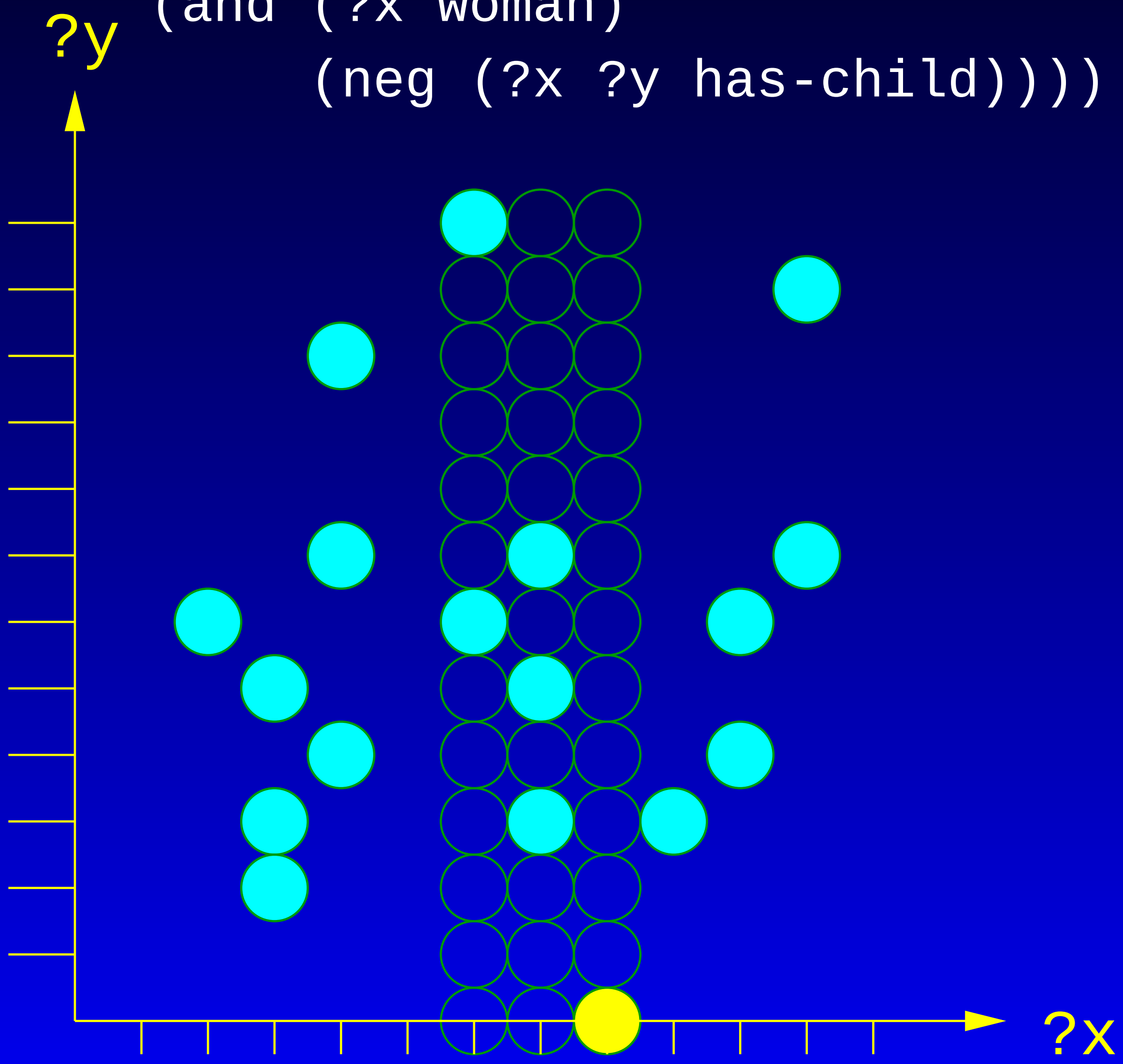
```
? (retrieve (?x)
     (?x (and woman (all has-child bottom))))

? (retrieve (?x)
     (and (?x woman)
          (neg (?x ?y has-child))))

? (retrieve (?x)
     (neg (and (?x woman) (?x ?y has-child))))
```

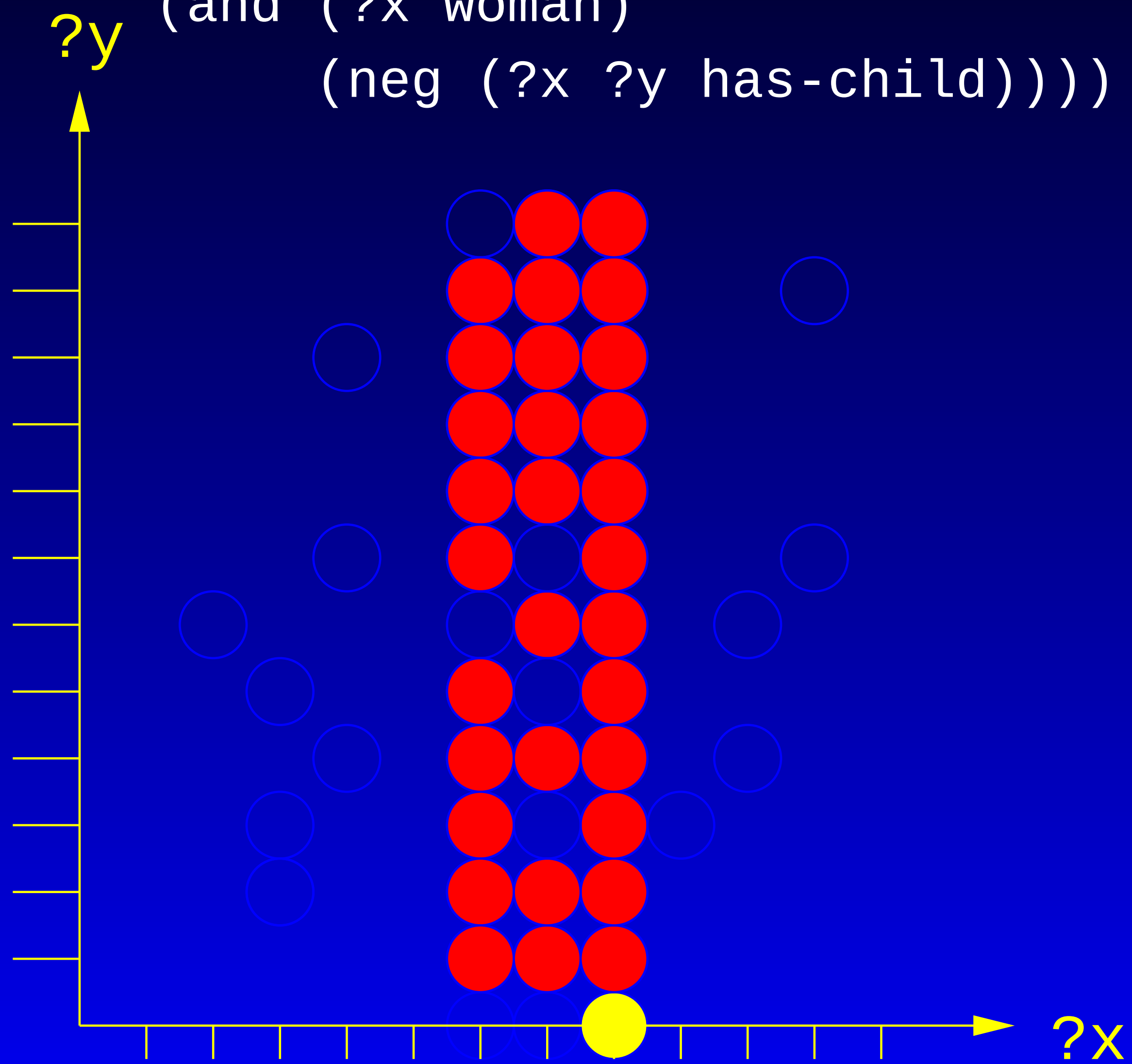
The Projection Operator (2)

```
Q1: (retrieve (?x)
              (and (?x woman)
                   (neg (?x ?y has-child))))
```



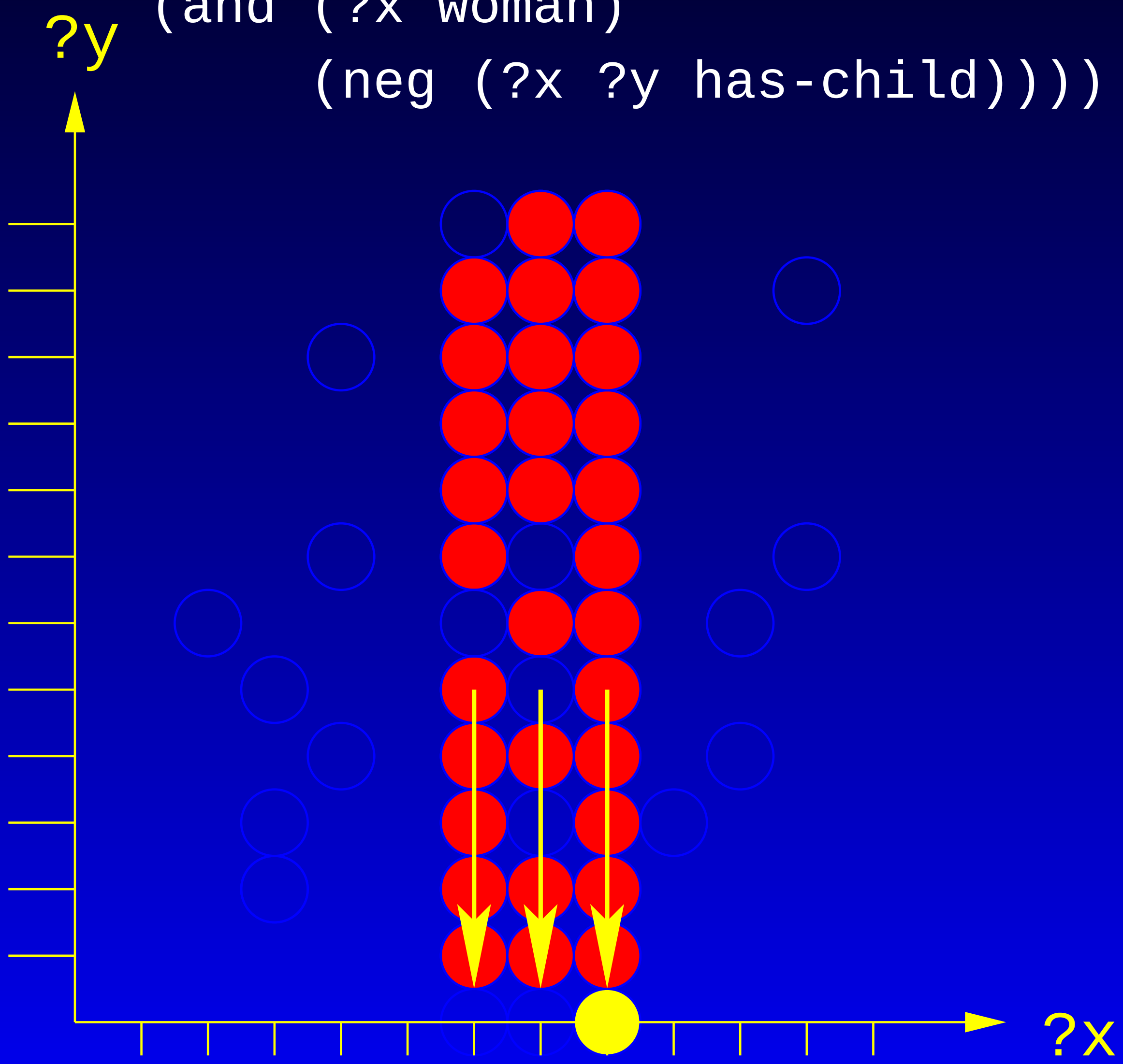
The Projection Operator (2)

```
Q1: (retrieve (?x)
              (and (?x woman)
                   (neg (?x ?y has-child))))
```



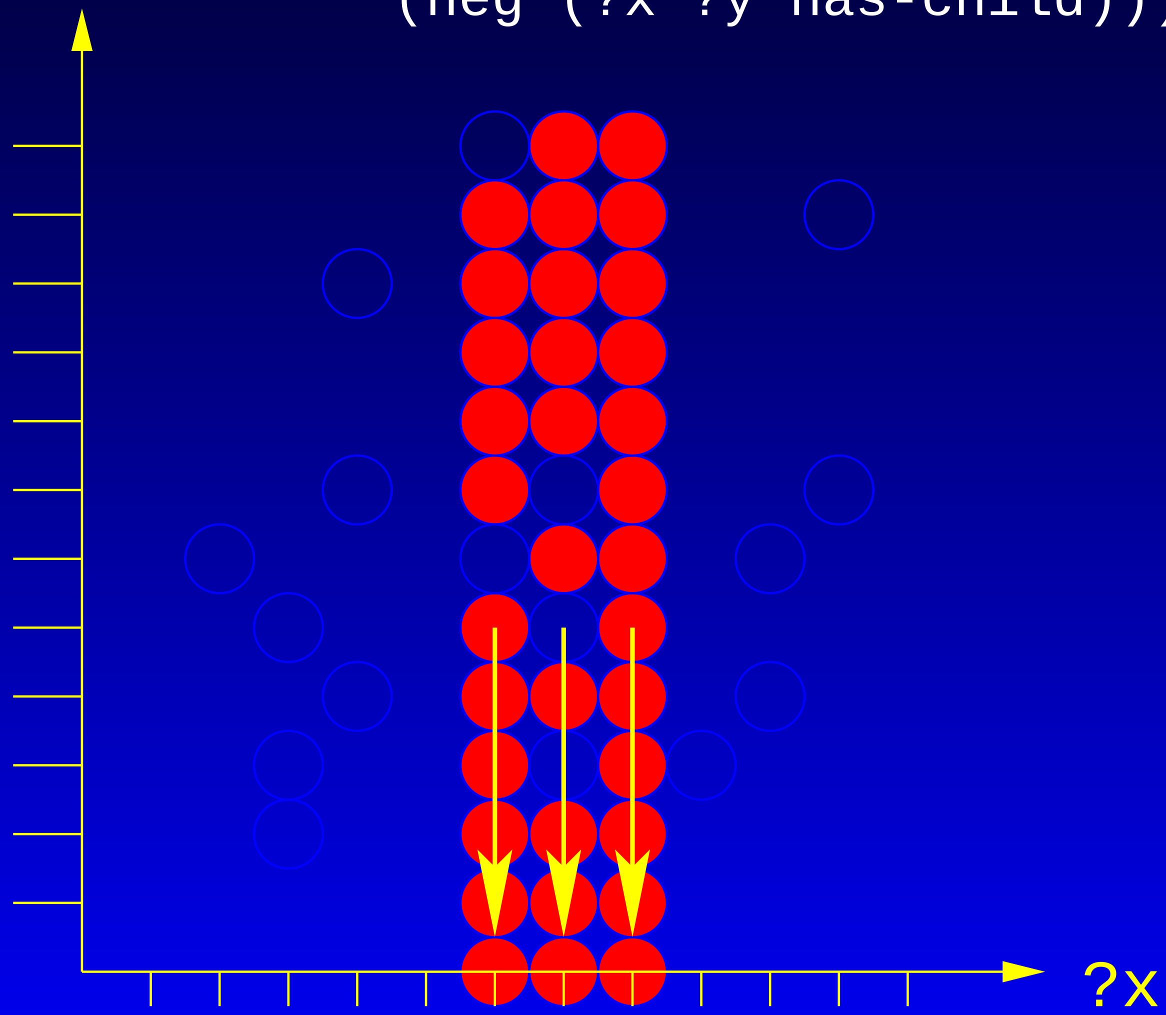
The Projection Operator (2)

```
Q1: (retrieve (?x)
          (and (?x woman)
               (neg (?x ?y has-child))))
```



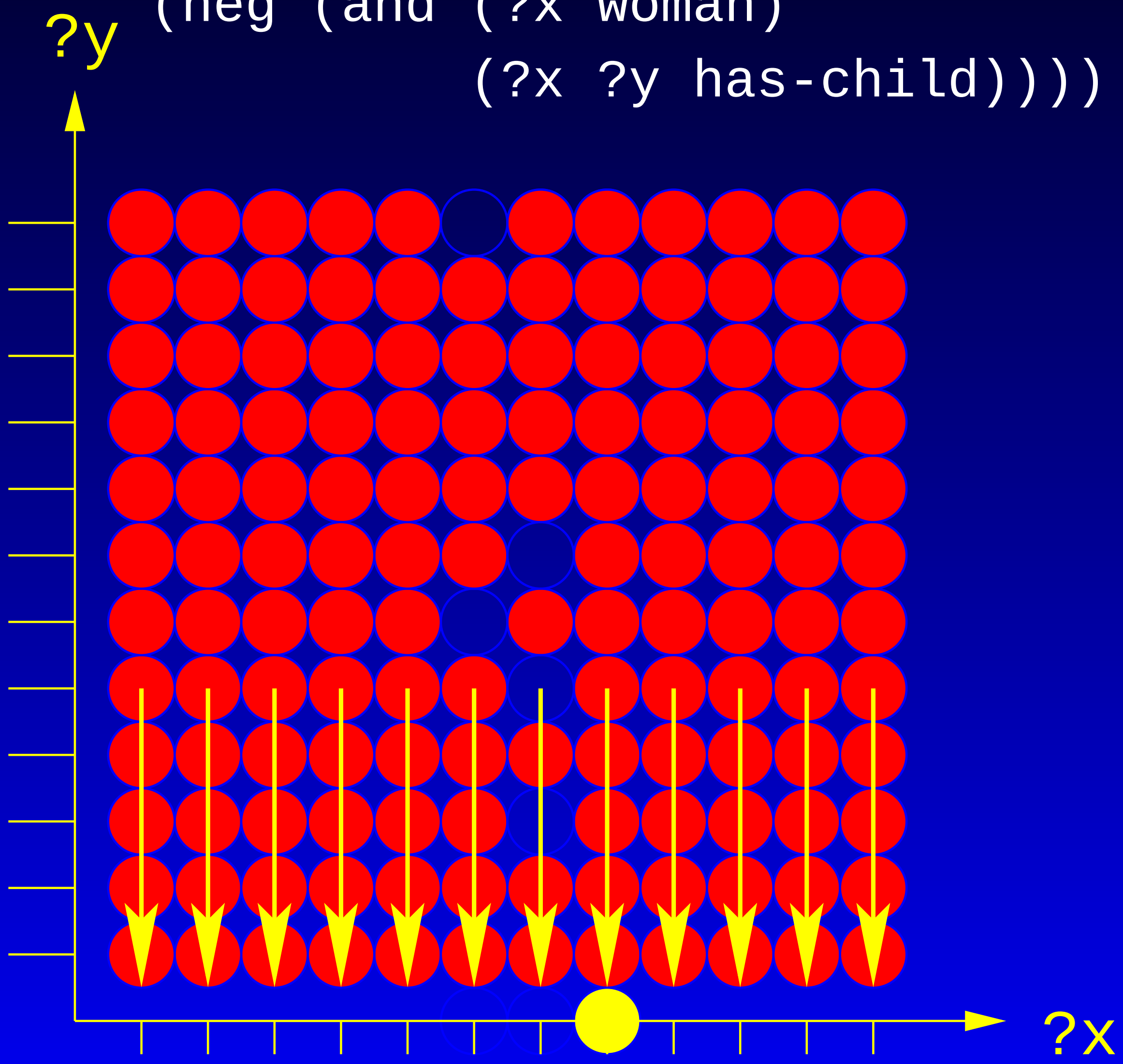
The Projection Operator (2)

Q1: (retrieve $\boxed{\texttt{(?x)}}$
(and (?x woman)
(neg (?x ?y has-child)))



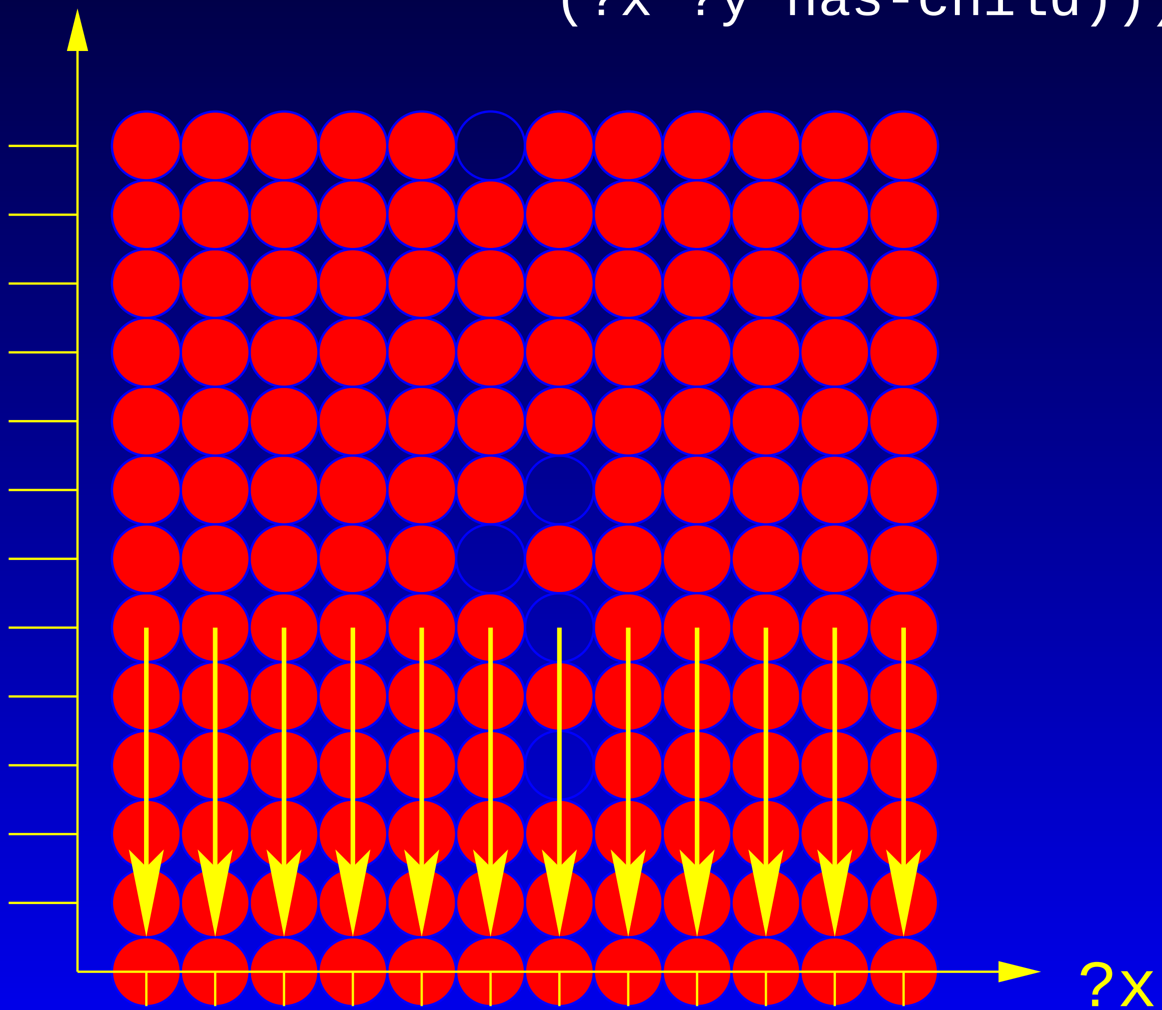
The Projection Operator (2)

```
Q2: (retrieve (?x)
              (neg (and (?x woman)
                        (?x ?y has-child))))
```



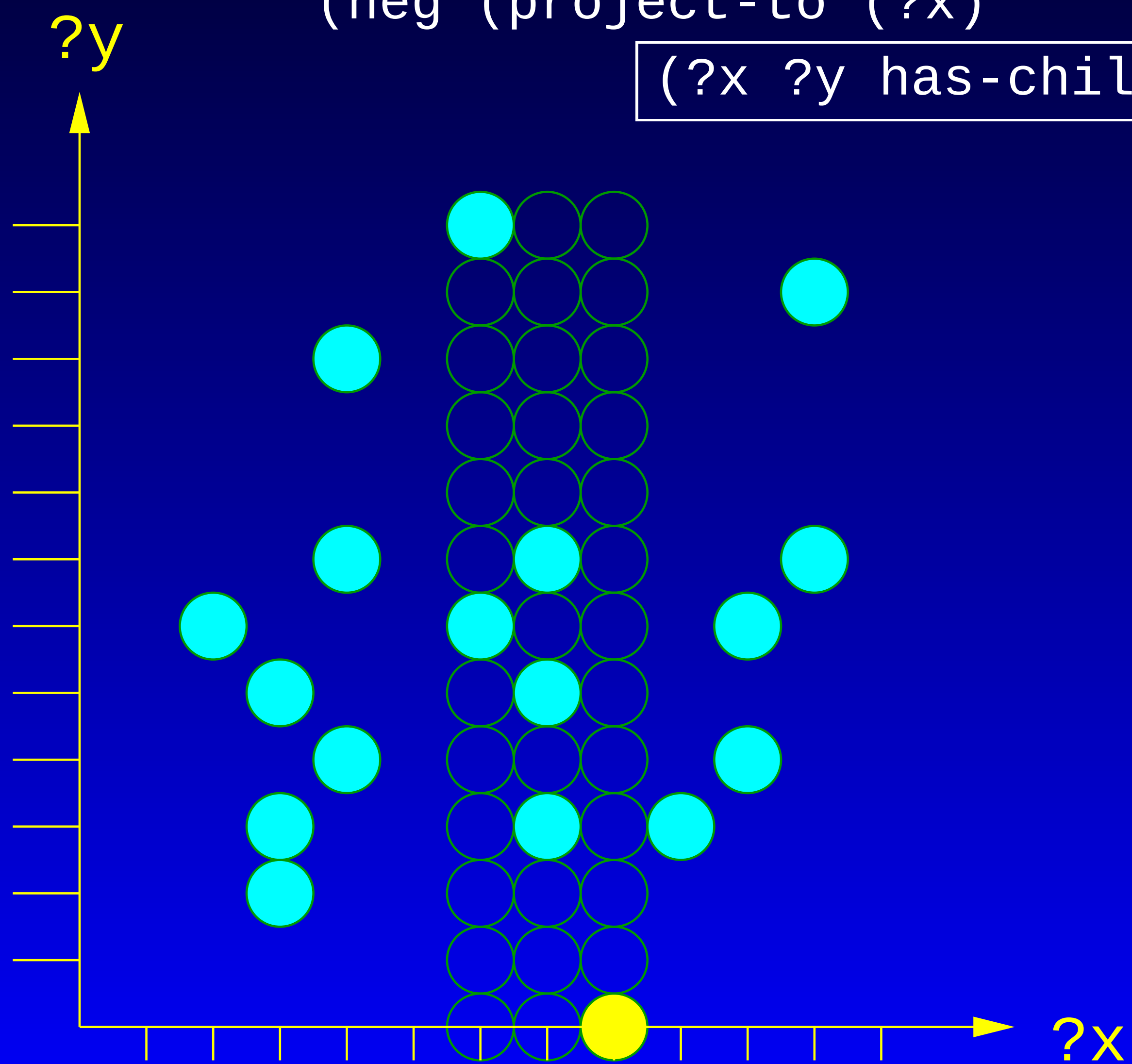
The Projection Operator (2)

- Q2: (retrieve $\boxed{\texttt{(?x)}}$
  (neg (and (?x woman)
  (?x ?y has-child)))



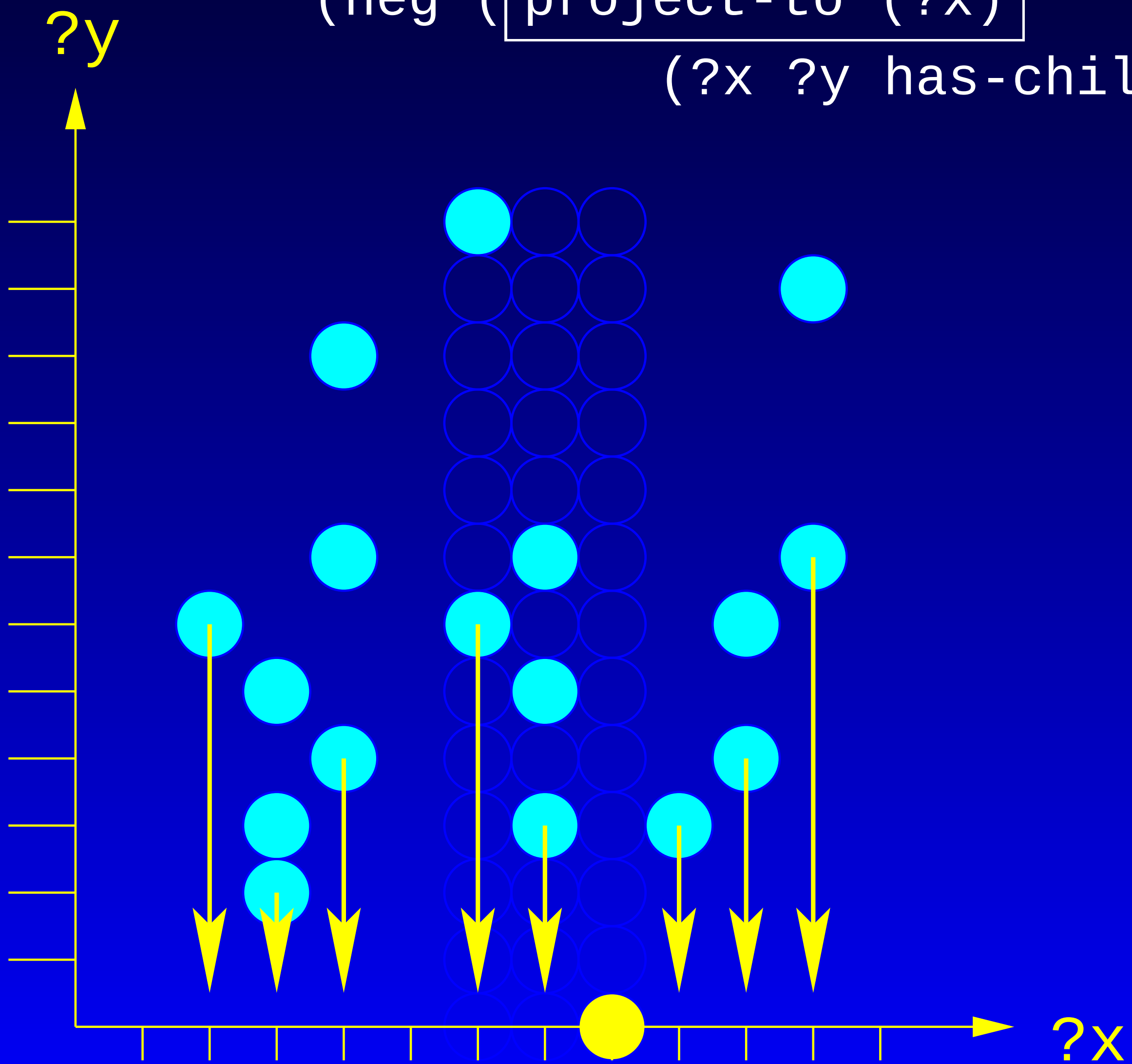
The Projection Operator (2)

```
Q3: (retrieve (?x)
              (and (?x woman)
                   (neg (project-to (?x)
                                    (?x ?y has-child)))))
```



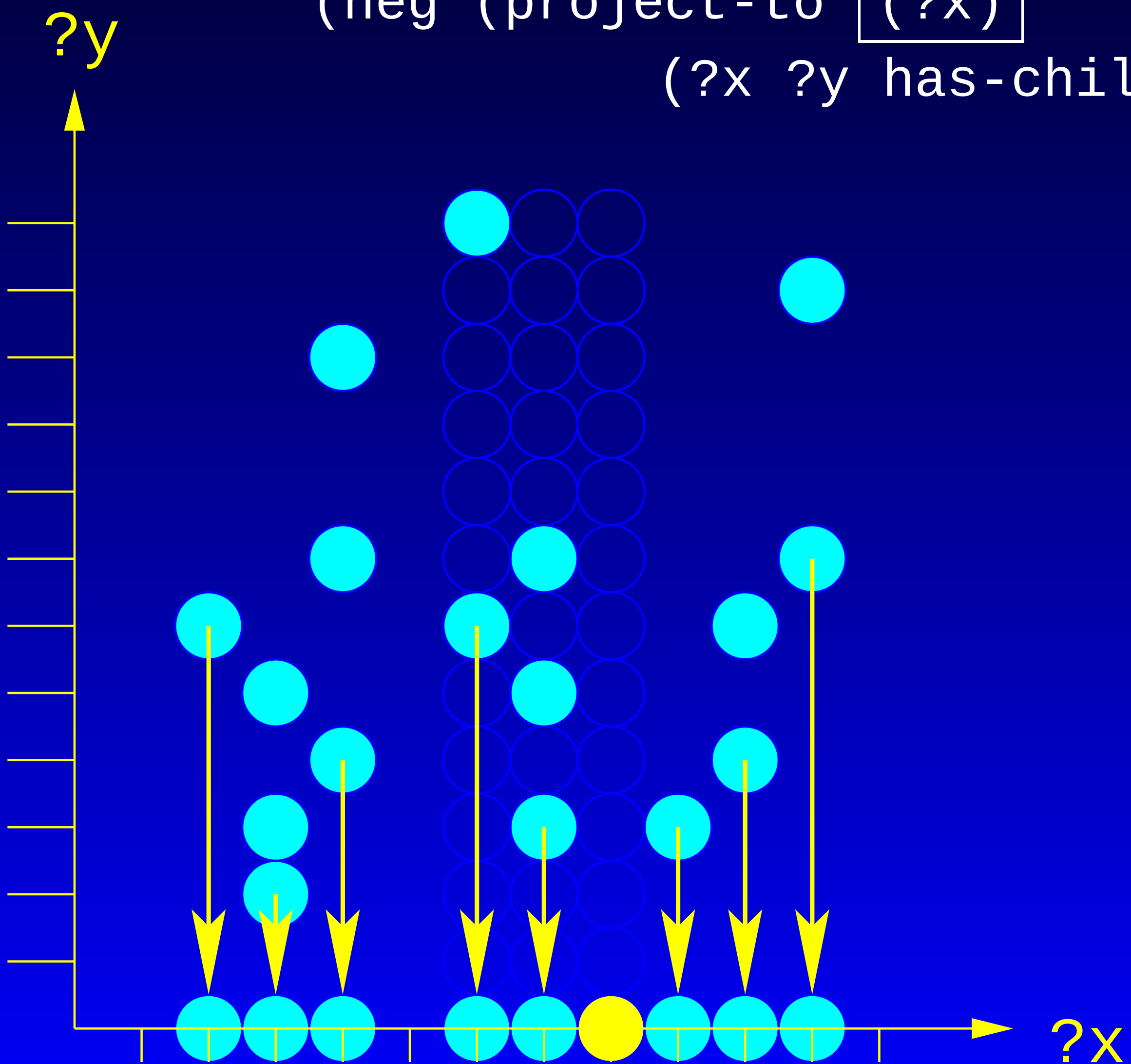
The Projection Operator (2)

```
Q3: (retrieve (?x)
              (and (?x woman)
                   (neg (project-to (?x)
                          (?x ?y has-child)))))
```



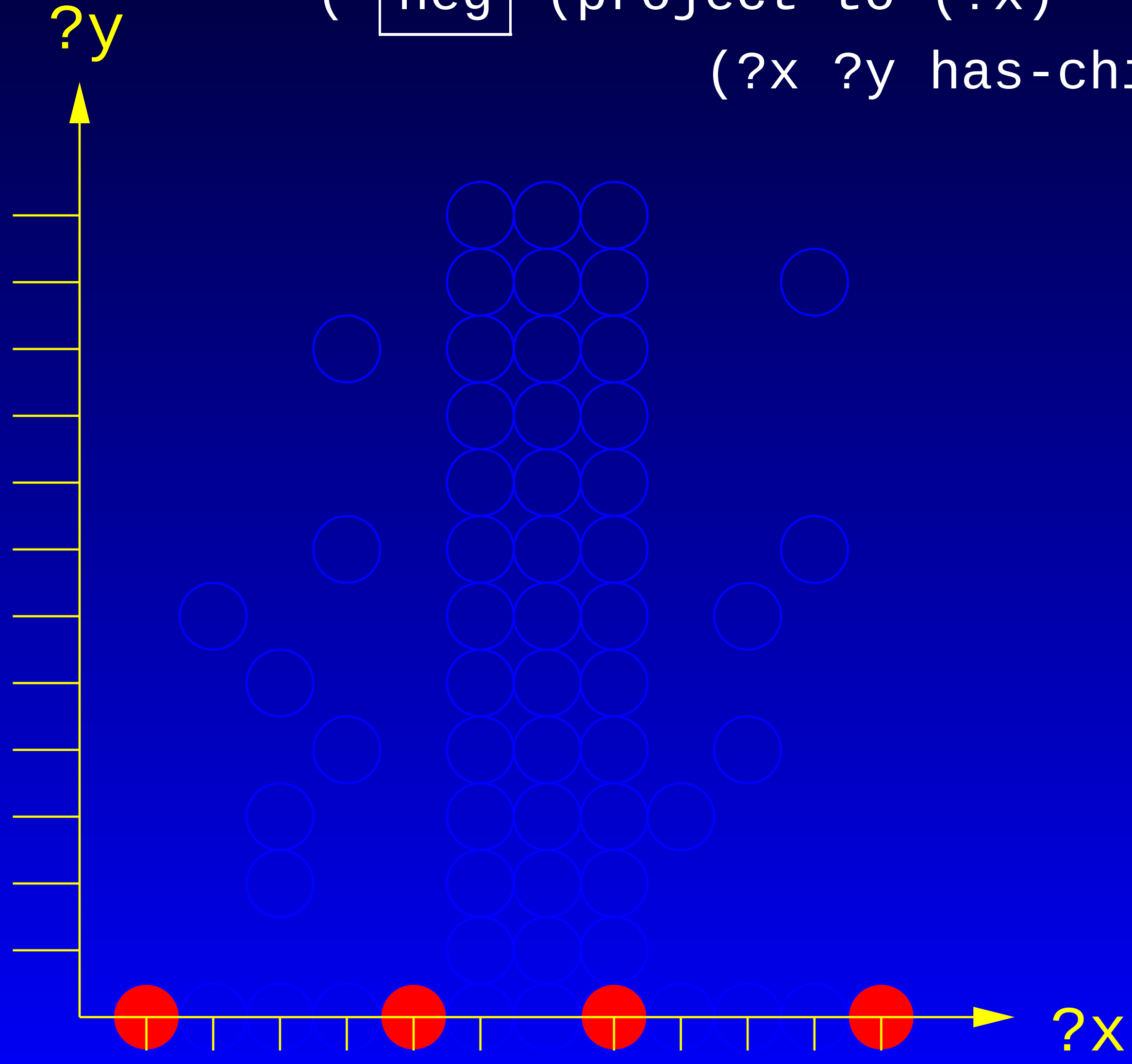
The Projection Operator (2)

```
Q3: (retrieve (?x)
              (and (?x woman)
                   (neg (project-to (?x)
                             (?x ?y has-child)))))
```



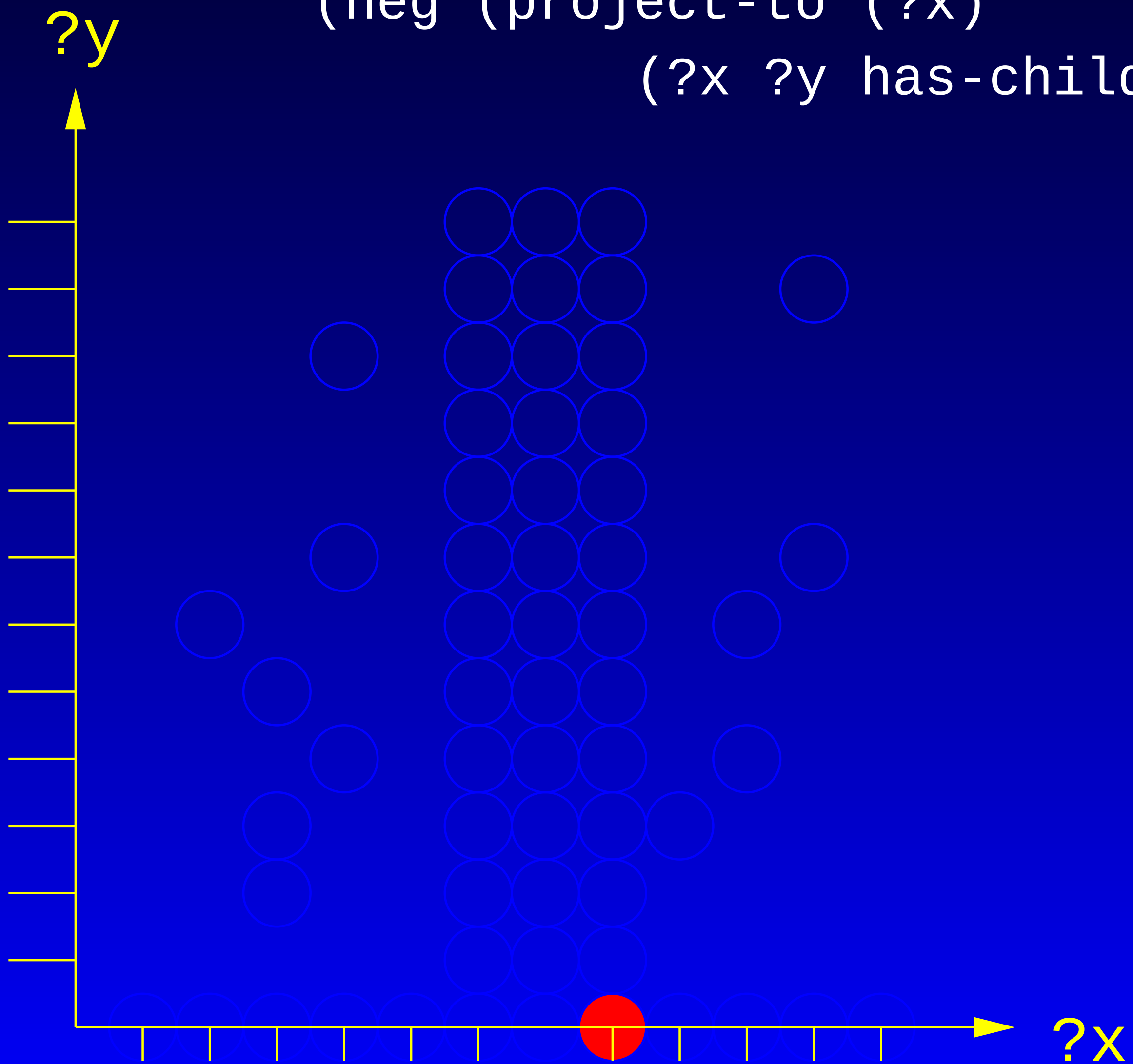
The Projection Operator (2)

```
Q3: (retrieve (?x)
              (and (?x woman)
                   ( neg (project-to (?x)
                           (?x ?y has-child)))))
```



The Projection Operator (2)

```
Q3: (retrieve (?x)
              (and (?x woman)
                   (neg (project-to (?x)
                             (?x ?y has-child))))) ✓
```



. . . Some Syntactic Sugar

- Due to the new projection operator, some "special syntax" from the older nRQL (DL '04) can now be expressed
- `(?x (has-known-successor has-child))`
  = `(project-to (?x) (?x ?y has-child))`
- `(?x NIL has-child)` (borrowed from LOOM)
  = `(neg`
  `(?x (has-known-successor has-child)))`
  = `(neg`
  `(project-to (?x) (?x ?y has-child)))`
- now expressible in terms of `project-to`

Querying OWL KBs

- OWL datatype properties:

```
<owl:Class rdf:ID="Person">
  <rdfs:label>person</rdfs:label>
</owl:Class>

<owl:DatatypeProperty rdf:ID="age">
  <rdfs:domain rdf:resource="#Person" />
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#integer" />
</owl:DatatypeProperty>

<Person rdf:about="http://www.test.com/michael">
   <age>34</age>
</Person>
```

nRQL & Datatype Properties

- Idea: handle OWL DTP like concrete domain attributes

```
? (retrieve
    (?x
      (datatype-fillers
         (|http://www.test.com/test.owl#age| ?x)))
    (?x (some |http://www.test.com/test.owl#age|
              (and (min 30) (max 35)))))

> (((?X |http://www.test.com/michael|)
    ((:TOLD-VALUE
     (|http://www.test.com/test.owl#age| ?X)) (34))))
```

- Extended RacerPro concept syntax (expressions like `(and (min 30) (max 35))` only recognized by nRQL)

nRQL & Annotation Properties

```
<owl:AnnotationProperty rdf:ID="my-comment">
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#person"/>
</owl:AnnotationProperty>

<person rdf:ID="i">
  <my-comment rdf:datatype=
       "http://www.w3.org/2001/XMLSchema#string"
  >My comment</my-comment>
</person>
```

- A special head projection operator `annotations (told-value)` is provided by nRQL
- Similar to querying for datatype properties

Expressivity Problems

- Access to "data values" in OWL docs (fillers of datatype/annotation properties) is restricted
- from the DL perspective, only the (extended) RacerPro concept expression language can be used
- How to retrieve all individuals which have (CD attribute or DTP) fillers containing substring x?
- Solution: maintain a data substrate in parallel to an ABox
- the data substrate is used to automatically "mirror" the ABox
- offer query access to this substrate by means of a hybrid query language - nRQL

Hybrid Queries - Data Substrate

```
(retrieve (?x ?*name ?*age)
   (and (?x (and |http://...#person|
                 (an |http://...#age|)))
        (?*x ?*name |http://...#name|)
        (?*name ( (:predicate (search "wessel"))
                  ((:predicate (search "michael"))
                   (:predicate (search "achim")))))
        (?*x ?*age |http://...#age|)
        (?*age ((:predicate (< 40))))))
```

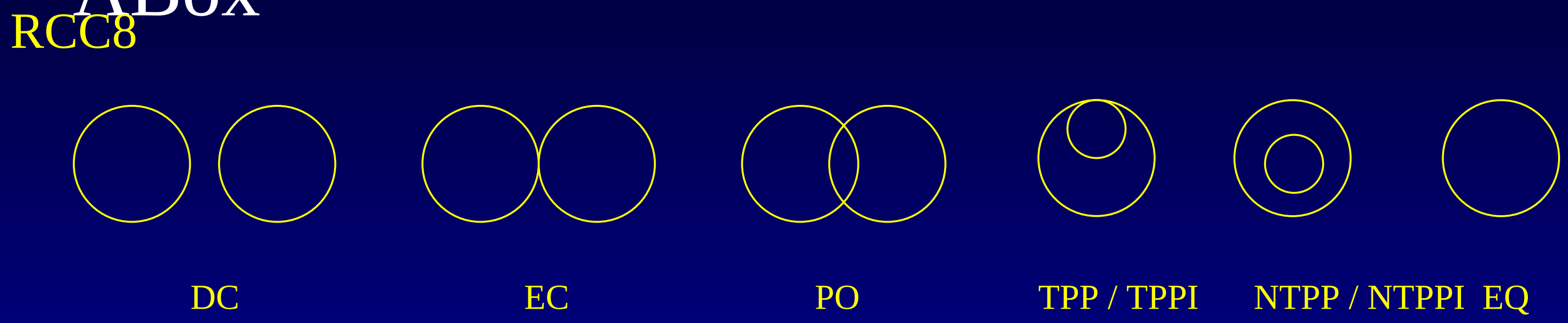
- New sort of variables: `*?x (*$?x)`, ranging over data nodes
- Data nodes can also be data values in OWL documents
- Data nodes/edges have descriptive labels: kind, role, property, . . .
- Notion of entailment for labels of nodes/edges
- Data query atoms are in pos. CNF & contain literals and predicates.

The RCC Substrate

- Associates an RCC8 (RCC5) network with an ABox

The RCC Substrate

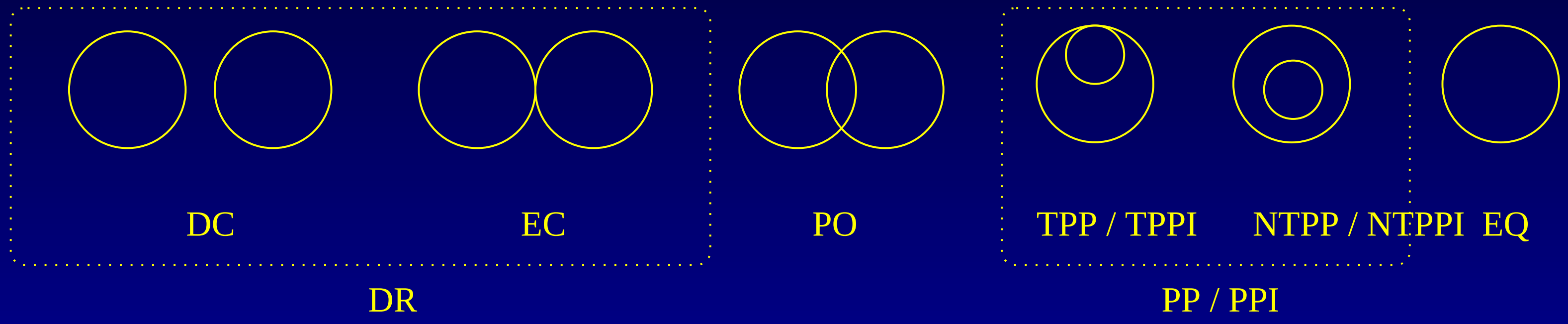
- Associates an RCC8 (RCC5) network with an ABox



The RCC Substrate

- Associates an RCC8 (RCC5) network with an ABox

RCC8



RCC5

The RCC Substrate

- Associates an RCC8 (RCC5) network with an ABox
- (Some) ABox individuals have corresponding nodes in the network, representing “spatial characteristics”

The RCC Substrate

- Associates an RCC8 (RCC5) network with an ABox
- (Some) ABox individuals have corresponding nodes in the network, representing “spatial characteristics”
- An RCC8 relation is a disjunction of RCC8 base relations, e.g. $\{DC, EC\} \rightarrow$ “coarser knowledge” concerning spatial relationship

The RCC Substrate

- Associates an RCC8 (RCC5) network with an ABox
- (Some) ABox individuals have corresponding nodes in the network, representing “spatial characteristics”
- An RCC8 relation is a disjunction of RCC8 base relations, e.g. $\{DC, EC\} \rightarrow$ “coarser knowledge” concerning spatial relationship
- RCC8: <u>8 base relations,</u> $2^8 = 256$ relations

The RCC Substrate

- Associates an RCC8 (RCC5) network with an ABox
- (Some) ABox individuals have corresponding nodes in the network, representing "spatial characteristics"
- An RCC8 relation is a disjunction of RCC8 base relations, e.g. $\{DC, EC\} \rightarrow$ "coarser knowledge" concerning spatial relationship
- RCC8: 8 base relations, $2^8 = 256$ relations
- JEPD property for base relations = jointly exhaustive, pairwise disjoint

The RCC Substrate

- Associates an RCC8 (RCC5) network with an ABox
- (Some) ABox individuals have corresponding nodes in the network, representing “spatial characteristics”
- An RCC8 relation is a disjunction of RCC8 base relations, e.g. $\{DC, EC\} \rightarrow$ “coarser knowledge” concerning spatial relationship
- RCC8: 8 base relations, $2^8 = 256$ relations
- JEPD property for base relations = jointly exhaustive, pairwise disjoint
- No knowledge about relationship: use disjunction of all 8 base relations

The RCC Substrate (2)

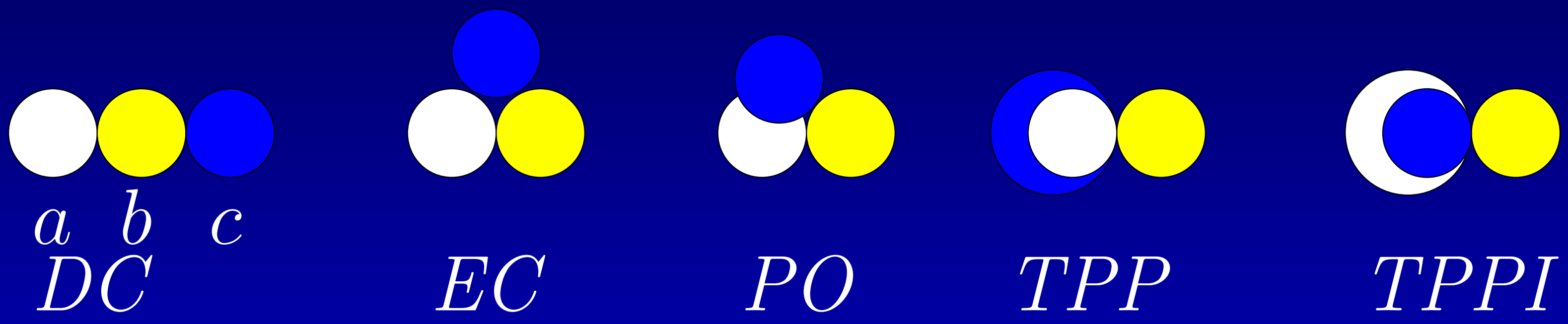
- Charateristics of RCC relations captured by "composition table"

The RCC Substrate (2)

- Charateristics of RCC relations captured by "composition table"
- Inference patterns of the form $R(x, y) \circ S(y, z) \rightarrow T_1(x, z) \vee \cdots \vee T_n(x, z)$

The RCC Substrate (2)

- Charateristics of RCC relations captured by "composition table"
- Inference patterns of the form $R(x, y) \circ S(y, z) \rightarrow T_1(x, z) \vee \cdots \vee T_n(x, z)$



The RCC Substrate (2)

RCC5 Composition Table

∘	DR(a,b)	PO(a,b)	EQ(a,b)	PPI(a,b)	PP(a,b)
DR(b,c)	*	DR PO PPI	DR	DR PO PPI	DR
PO(b,c)	DR PO PP	*	PO	PO PPI	DR PO PP
EQ(b,c)	DR	PO	EQ	PPI	PP
PP(b,c)	DR PO PP	PO PP	PP	PO EQ PP PPI	PP
PPI(b,c)	DR	DR PO PPI	PPI	PPI	*

The RCC Substrate (2)

- Charateristics of RCC relations captured by "composition table"
- Inference patterns of the form $R(x,y) \circ S(y,z) \rightarrow T_1(x,z) \vee \cdots \vee T_n(x,z)$
- Notions of consistency:
  - Constraint satisfaction: find an instantiation of the network with base relations such that the composition table axioms are satisfied (requires search, exponential)

The RCC Substrate (2)

- Charateristics of RCC relations captured by "composition table"
- Inference patterns of the form
  $R(x, y) \circ S(y, z) \rightarrow T_1(x, z) \vee \cdots \vee T_n(x, z)$
- Notions of consistency:
  - Polynominal methods (incomplete): Constraint propagation, "path or 3-consistency algorithm": for all x, y, z, compute
    $T(x, z) =_{def} T(x, z) \cap R(x, y) \circ S(y, z)$
    until fixpoint is reached

The RCC Substrate (2)

- Charateristics of RCC relations captured by "composition table"
- Inference patterns of the form $R(x,y) \circ S(y,z) \rightarrow T_1(x,z) \vee \cdots \vee T_n(x,z)$
- Notions of consistency:
  - Spatial realizabilty: find a spatial model (not considered here)

The RCC Substrate (3)

- nRQL based on notion of logical consequence: a binding to a variable is only established if this binding holds in <u>all</u> models (“certain answer”)

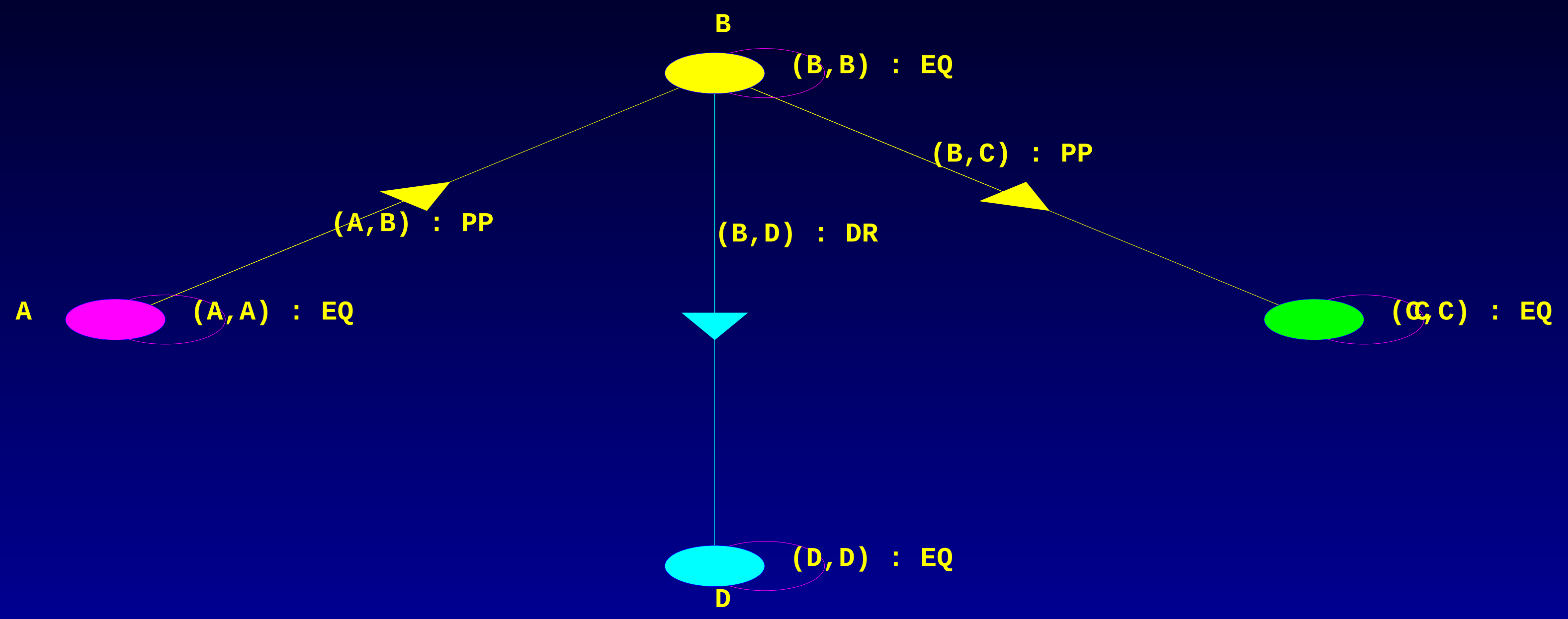
The RCC Substrate (3)

- nRQL based on notion of logical consequence: a binding to a variable is only established if this binding holds in <u>all</u> models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?

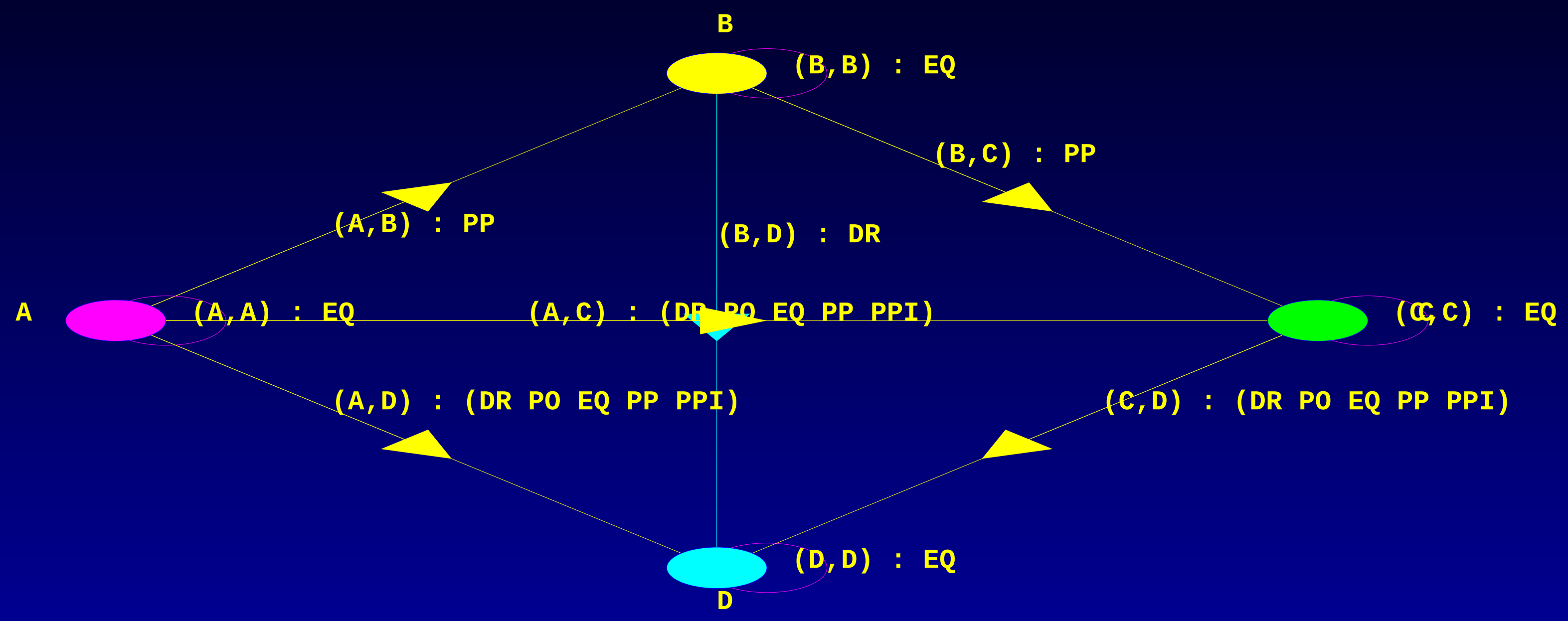
The RCC Substrate (3)

- nRQL based on notion of logical consequence: a binding to a variable is only established if this binding holds in <u>all</u> models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations

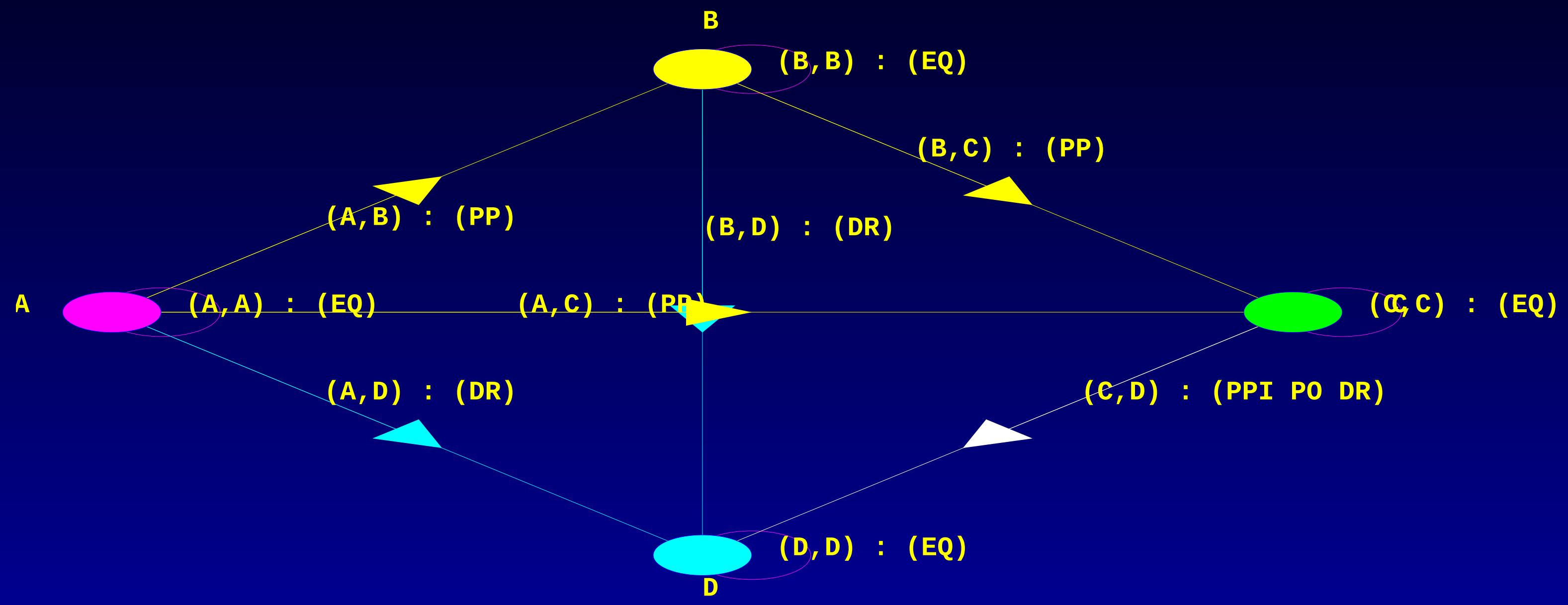
The RCC Substrate (3)



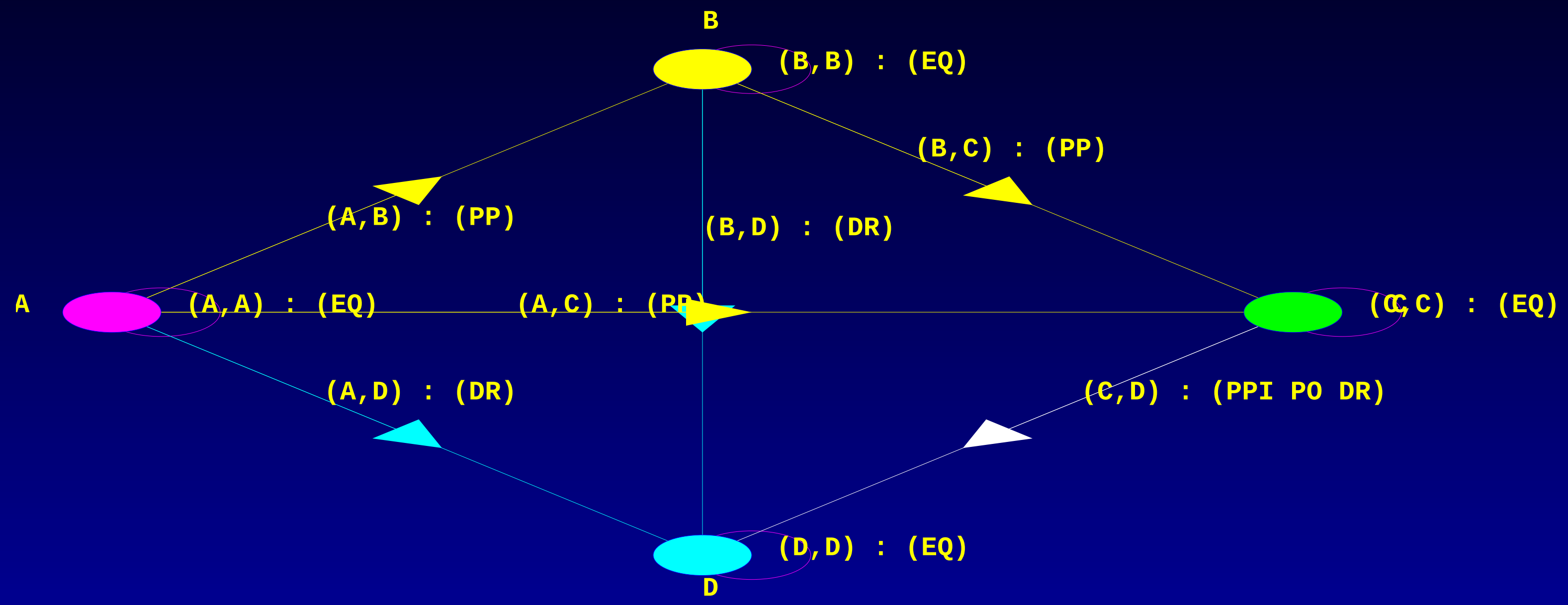
The RCC Substrate (3)



The RCC Substrate (3)



The RCC Substrate (3)



$$\models DR(A, D)$$
$$\models PP(A, C)$$
$$\models \{PPI, PO, DR\}(C, D)$$
$$\models \ldots$$

The RCC Substrate (3)

- nRQL based on notion of logical consequence: a binding to a variable is only established if this binding holds in <u>all</u> models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable

The RCC Substrate (3)

- nRQL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Much better RCC checkers are known

The RCC Substrate (3)

- nRQL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Much better RCC checkers are known
- RacerPro handbook: RCC substrate is an "experimental, non-essential add on"

The RCC Substrate (3)

- nRQL based on notion of logical consequence: a binding to a variable is only established if this binding holds in <u>all</u> models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Much better RCC checkers are known
- RacerPro handbook: RCC substrate is an "experimental, non-essential add on"

The RCC Substrate (4)

The RCC Substrate (4)

```
(in-rcc-box geo-example :rcc5)
(implies city bavarian-city)
(implies software-department
computer-science-department)
(implies computer-science-department
university-department)
(implies computer-science-department it-specialists)
(implies it-company it-specialists)
(implies it-company company)
```

The RCC Substrate (4)

```
(rcc-node germany country country)
(rcc-node hamburg city city)
(rcc-node munich bavarian-city bavarian-city)
(rcc-node harburg district district)
(rcc-node sts software-department
software-department)
(rcc-node docomo-euro-labs company it-company)
```

The RCC Substrate (4)

```
(rcc-related germany hamburg :ppi)
(rcc-related germany munich :ppi)
(rcc-related hamburg harburg :ppi)
(rcc-related hamburg munich :dr)
(rcc-related munich docomo-euro-labs :ppi)
(rcc-related hamburg sts :ppi)
```

The RCC Substrate (4)

```
> (retrieve (?x ?y) (and (?x country) (?*x ?*y :ppi)
                         (?y it-specialists)))
(((?X GERMANY) (?Y DOCOMO-EURO-LABS))
 ((?X GERMANY) (?Y STS)))
```

The RCC Substrate (4)

```
> (retrieve (?x ?y) (and (?x country) (?*x ?*y :ppi)
                         (?y it-specialists)))
(((?X GERMANY) (?Y DOCOMO-EURO-LABS))
 ((?X GERMANY) (?Y STS)))
```

```
> (retrieve (?x ?y) (and (?*x ?*y :dr) (?x it-specialists)
                         (?y it-specialists)))
(((?X DOCOMO-EURO-LABS) (?Y STS))
 ((?X STS) (?Y DOCOMO-EURO-LABS)))
```

The RCC Substrate (4)

```
> (retrieve (?x ?y) (and (?x country) (?*x ?*y :ppi)
                         (?y it-specialists)))
(((?X GERMANY) (?Y DOCOMO-EURO-LABS))
 ((?X GERMANY) (?Y STS)))
```

```
> (retrieve (?x ?y) (and (?*x ?*y :dr) (?x it-specialists)
                         (?y it-specialists)))
(((?X DOCOMO-EURO-LABS) (?Y STS))
 ((?X STS) (?Y DOCOMO-EURO-LABS)))
```

```
> (retrieve (?x ?y)
            (and (?*x ?*y :dr) (?x it-specialists)
                 (?y it-specialists)
                 (?*y ?*z :pp) (?z bavarian-city)))
(((?X STS) (?Y DOCOMO-EURO-LABS))
((?X DOCOMO-EURO-LABS) (?Y STS)))
```

RCC Substrate & OWL?

- OWL DL = $\mathcal{SHOIN}(Dn)$
- RacerPro supports $\mathcal{SHIN}(Dn)$ (Nominals, $\mathcal{O}$, approximated)
- An OWL KB can be mirrored into the data substrate (additional retrieval predicates possible)
- Idea: declare a set of object properties as spatial relationships, then automatically create an RCC substrate from an OWL KB

$\Rightarrow$ "RCC substrate" addition for OWL

- Already requested by a user of RacerPro, but not implemented yet

Reasoning about Queries

- Query consistency

Reasoning about Queries

- Query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts

Reasoning about Queries

- Query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiabilty seperately

Reasoning about Queries

- Query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiabilty seperately
  - ABox assertions: construct an ABox from the conjuncts, replacing variables with individuals, check for ABox satisfiability

Reasoning about Queries

- Query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiabilty seperately
  - ABox assertions: construct an ABox from the conjuncts, replacing variables with individuals, check for ABox satisfiability
  - RCC conjuncts: construct an RCC substrate and check its consistency

Reasoning about Queries

- Query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiabilty seperately
  - ABox assertions: construct an ABox from the conjuncts, replacing variables with individuals, check for ABox satisfiability
  - RCC conjuncts: construct an RCC substrate and check its consistency
  - Conjecture: somehow “weak” since no interaction, but quite useful in this scenario

Reasoning about Queries

- Query consistency

  $\Rightarrow$ Reduction to appropriate ABox / RCC consistency checks

Reasoning about Queries

- Query consistency

  $\Rightarrow$ Reduction to appropriate ABox / RCC consistency checks

- Hybrid query containment

Reasoning about Queries

- Query consistency

  $\Rightarrow$ Reduction to appropriate ABox / RCC consistency checks

- Hybrid query containment

$$query(?germany, ?city, ?sea) \leftarrow$$

$$germany(?germany^*), federal\_division(?division^*),$$

$$german\_city(?city^*), (baltic\_see \sqcup north\_sea)(?sea^*),$$

$$\boxed{\text{PPI(?germany,?division), PPI(?division,?city),}}$$

$$\boxed{\text{DR(?division,?sea)}}$$

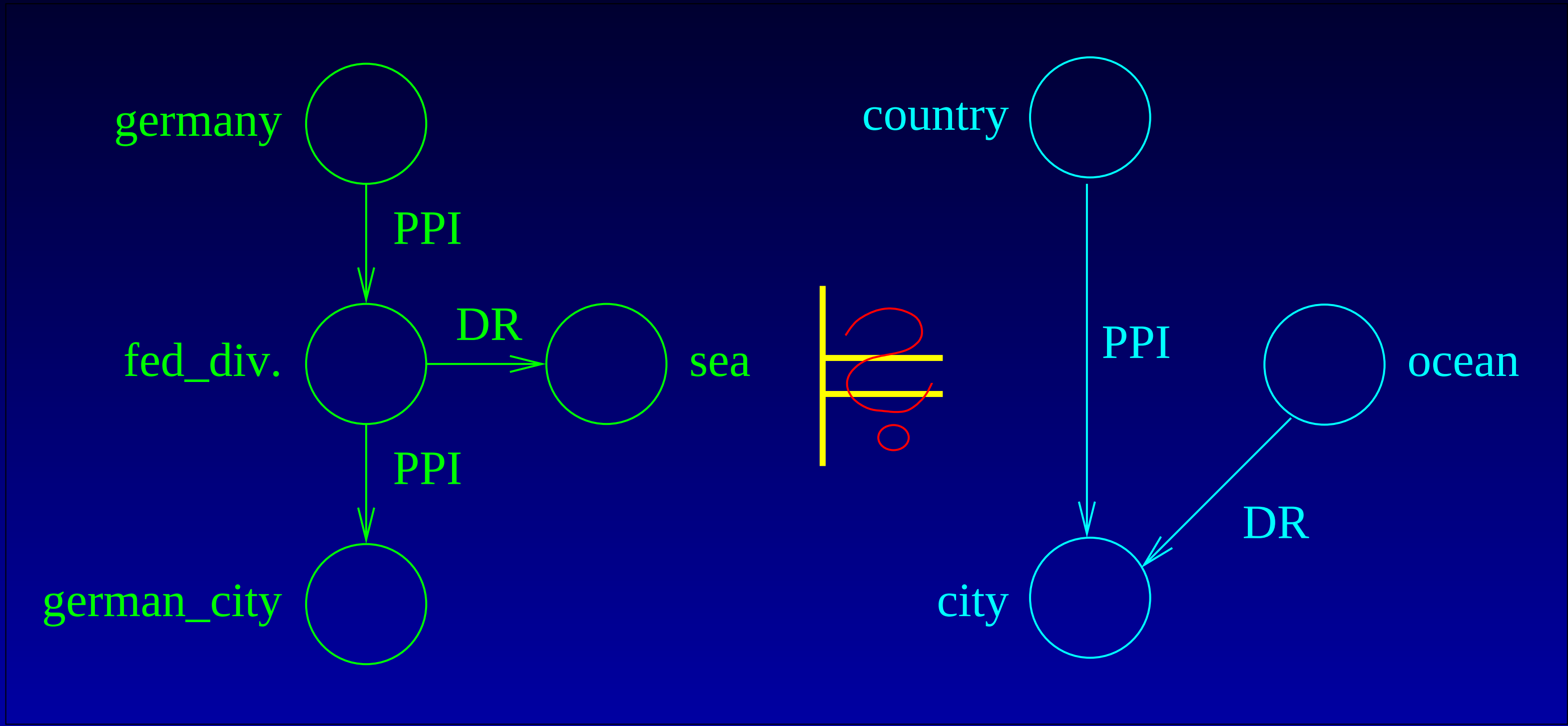
$$\models$$

$$query(?country, ?city, ?ocean) \leftarrow$$

$$country(?country^*), city(?city^*), ocean(?ocean^*),$$

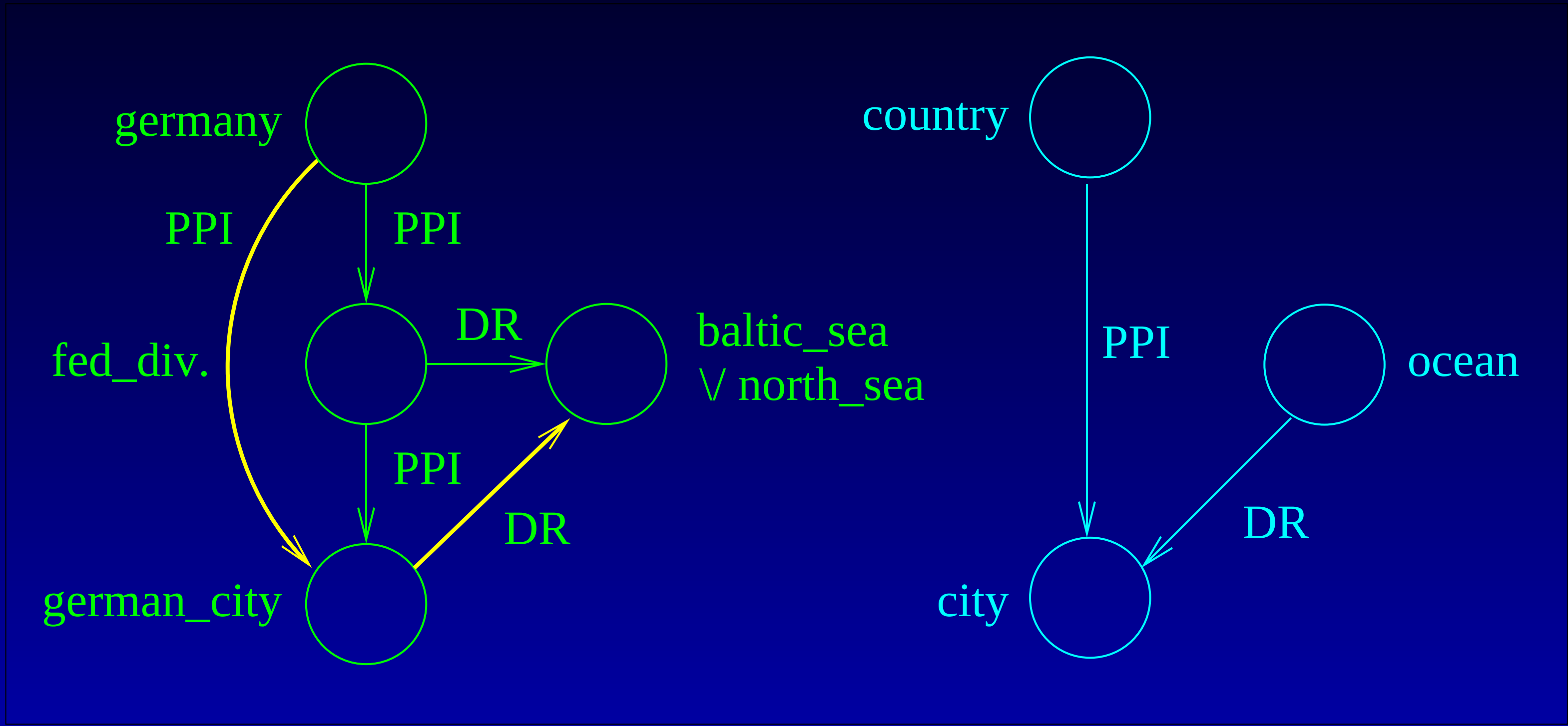
$$\boxed{\text{DR(?ocean,?city), PPI(?country,?city)}}$$

Reasoning about Queries



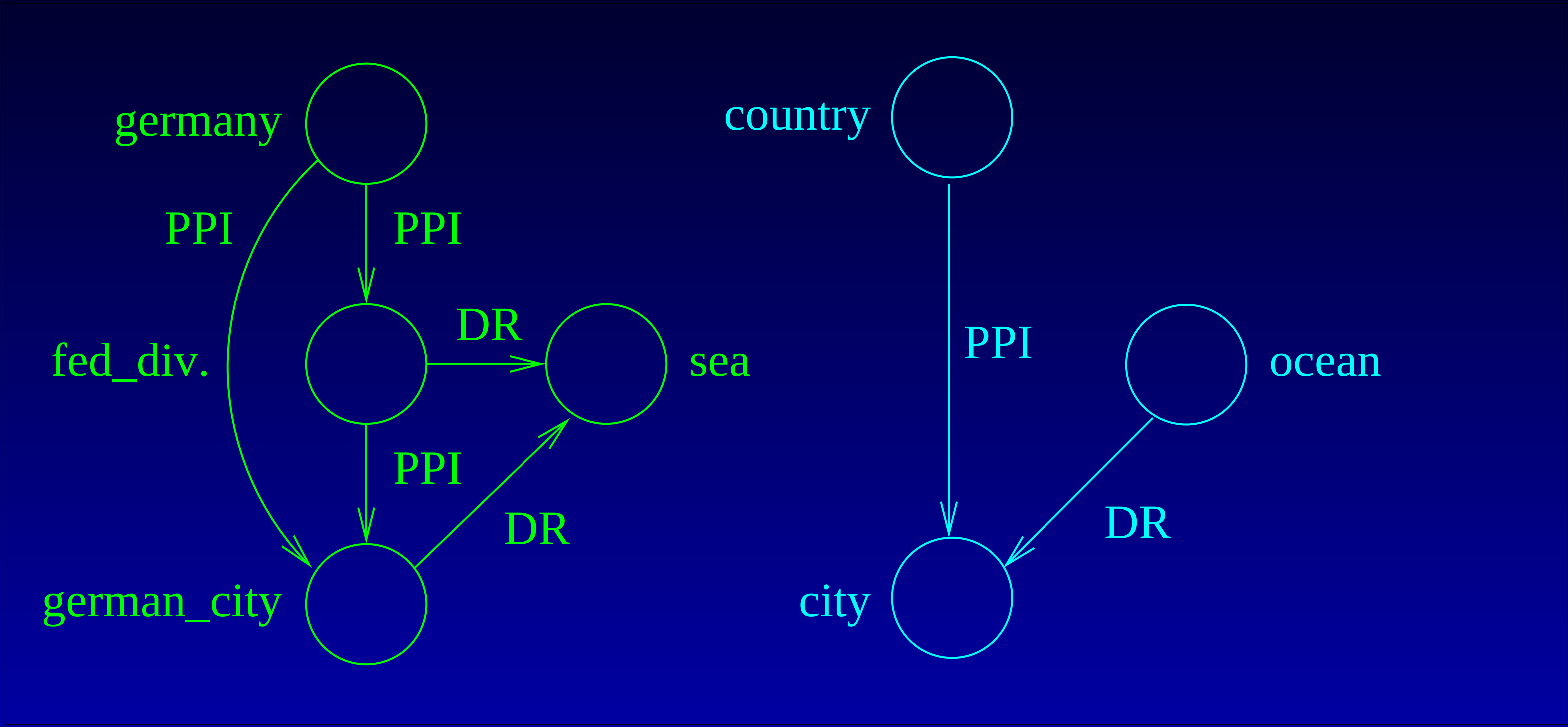
Two queries - does Green entail Blue?

Reasoning about Queries

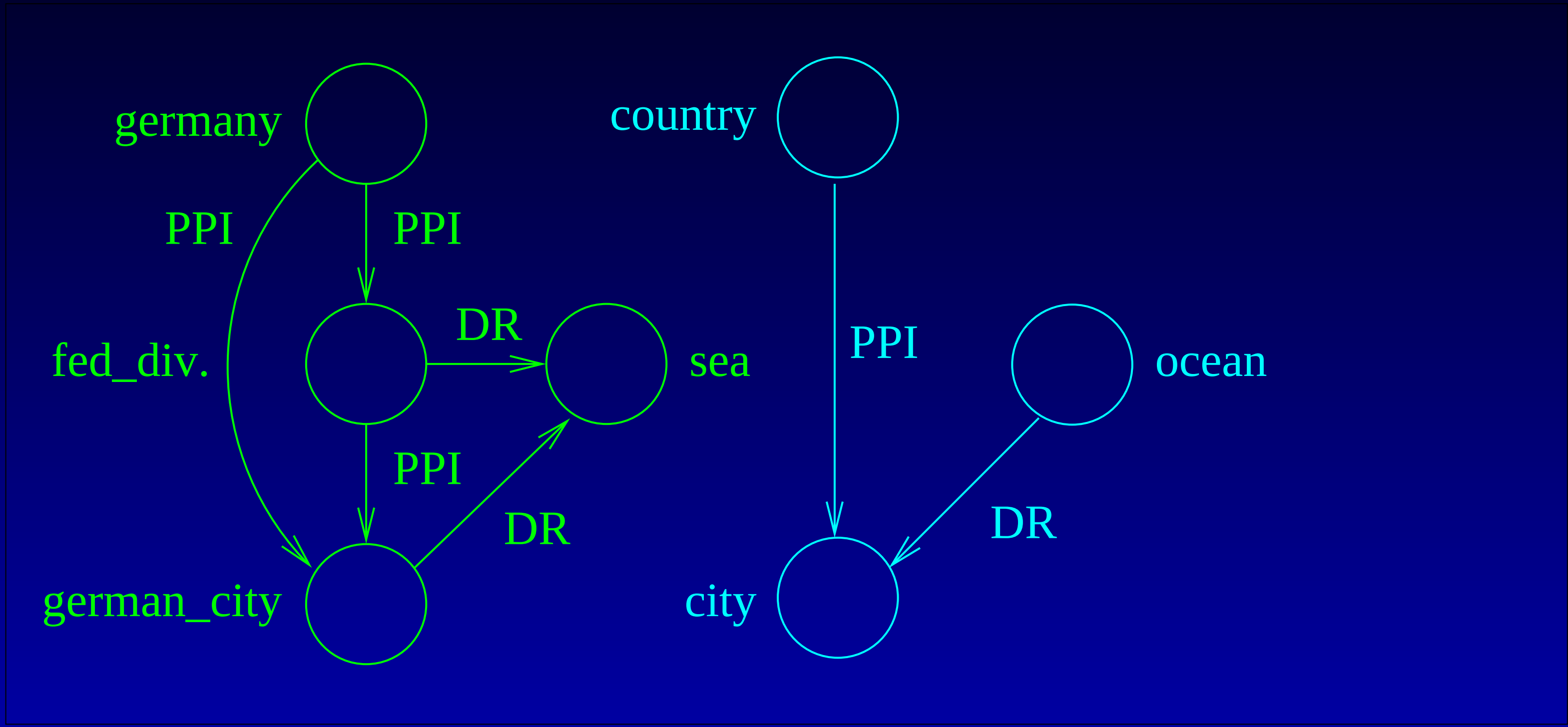


Adding entailed constraints for Green

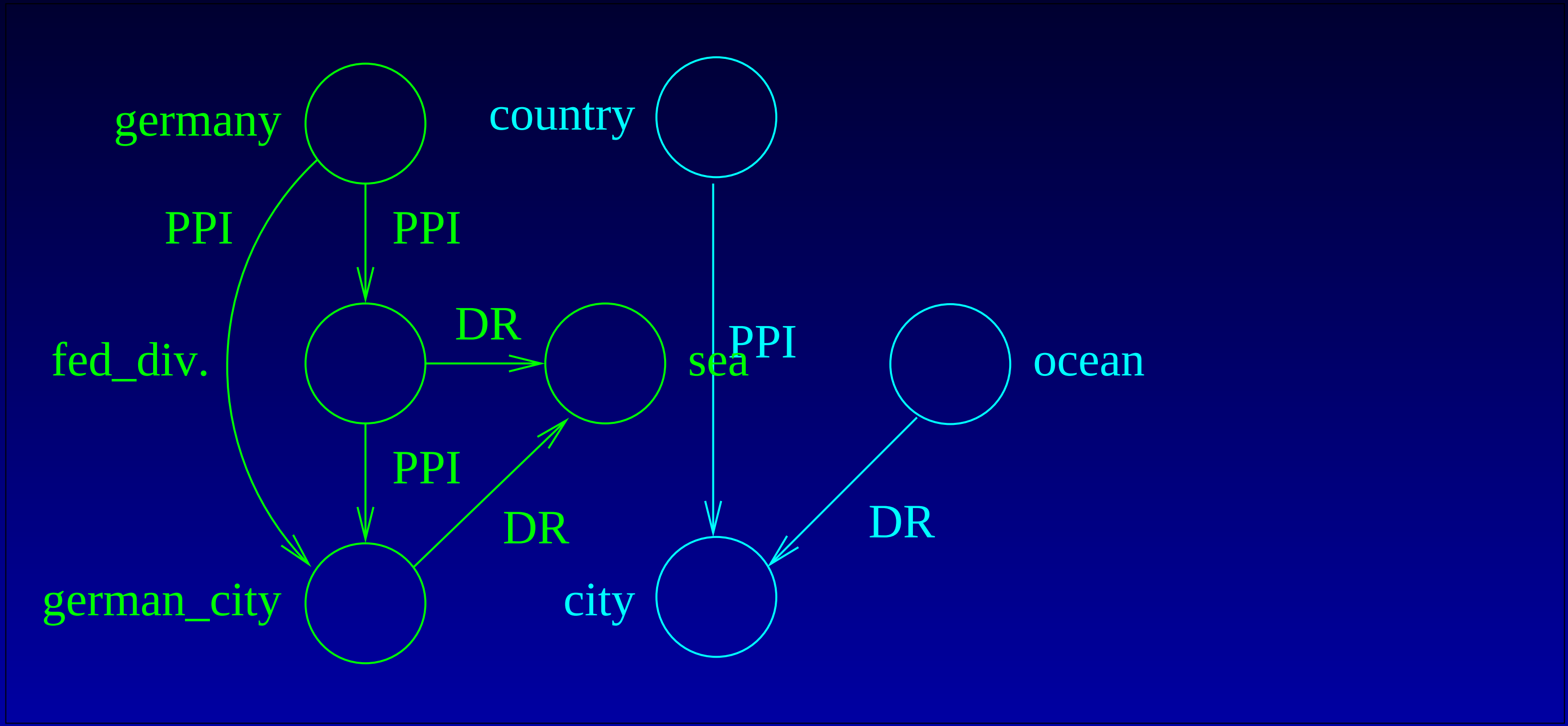
Reasoning about Queries



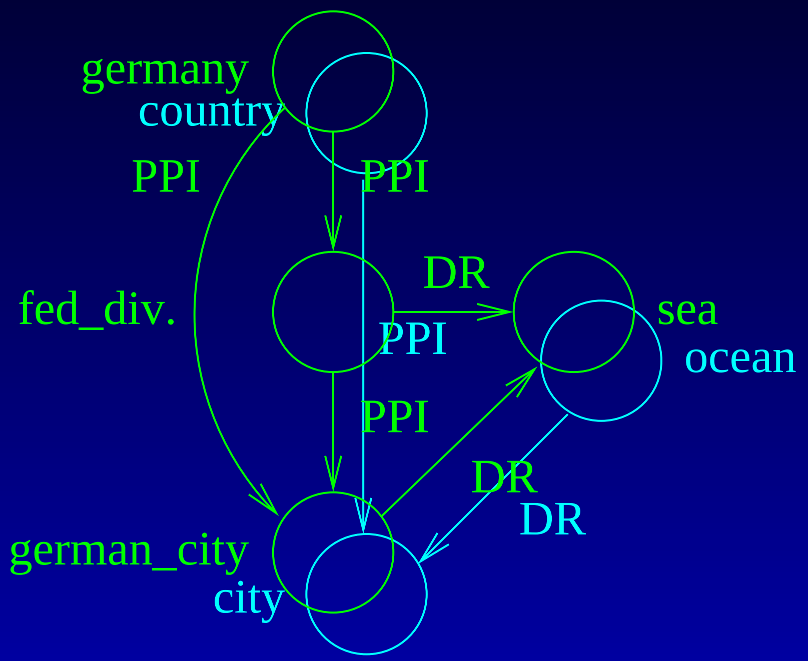
Reasoning about Queries



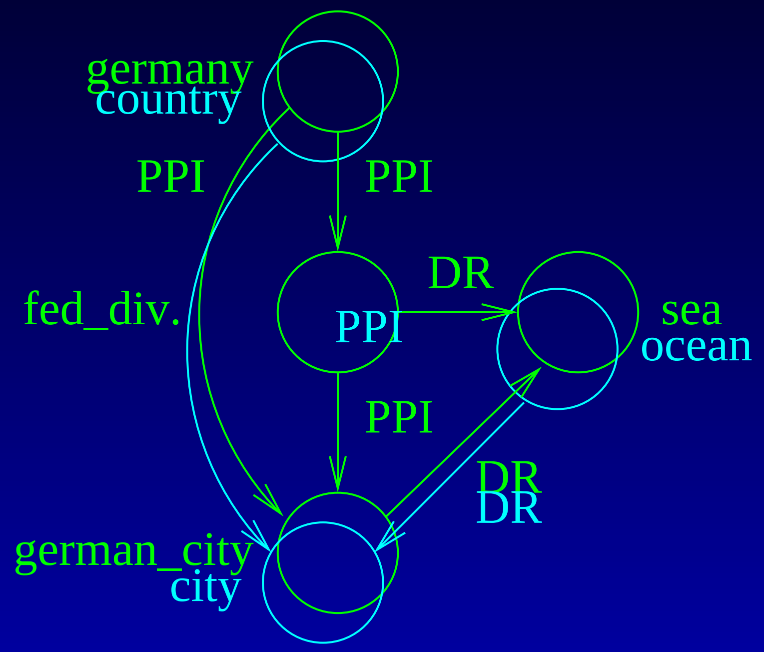
Reasoning about Queries



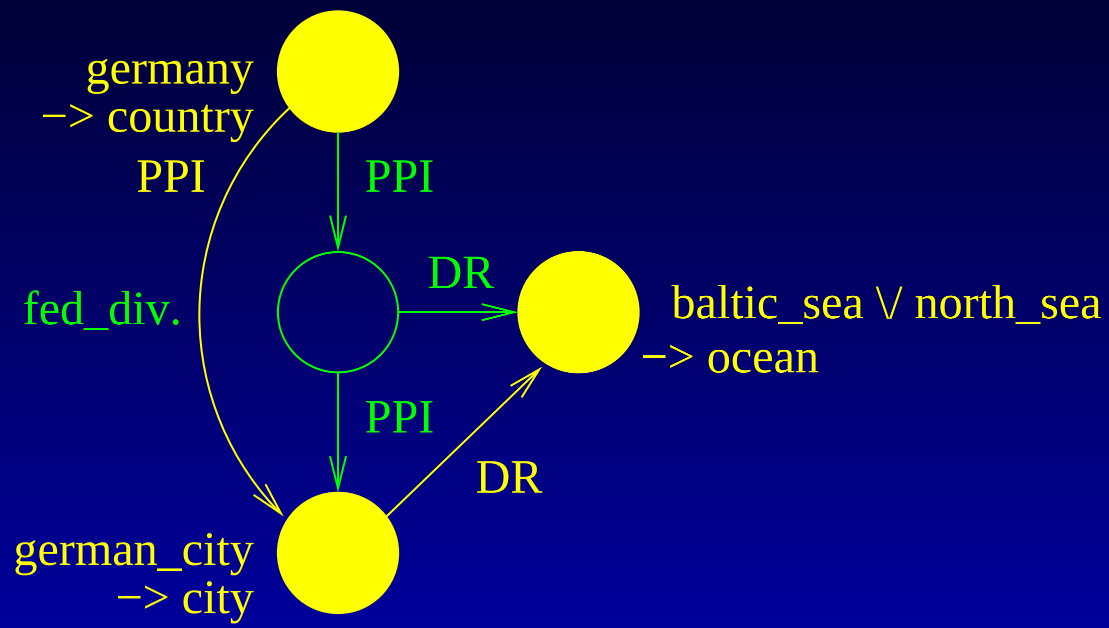
Reasoning about Queries



Reasoning about Queries



Reasoning about Queries



Match - Green is more specific than Blue

Reasoning about Queries

- Query consistency

  $\Rightarrow$ Reduction to appropriate ABox / RCC consistency checks

- Hybrid query containment

  $\Rightarrow$ By reduction to query consistency

Reasoning about Queries

- Query consistency

  $\Rightarrow$ Reduction to appropriate ABox / RCC consistency checks

- Hybrid query containment

  $\Rightarrow$ By reduction to query consistency

Conclusion

- Handling of space by logical means is difficult
- Standard-DLs or languages such as OWL are not tailored for this purpose
- Good for thematic aspects
- Tailored spatial languages should be decidable (conflict: expressiveness vs. decidability)
- The OWL "standard ontologies" for GIS (found on the web) can be understood as data schema specifications for GIS data, reasoning concerning the spatial aspects is not addressed
- RacerPro supports "spatial reasoning" only in the query answering engine

Thanks

for your

attention!