Friedrich-Alexander-Universität
Erlangen-Nürnberg

Master Thesis Presentation

Formalization and Automation of Company Register Processes via Logic

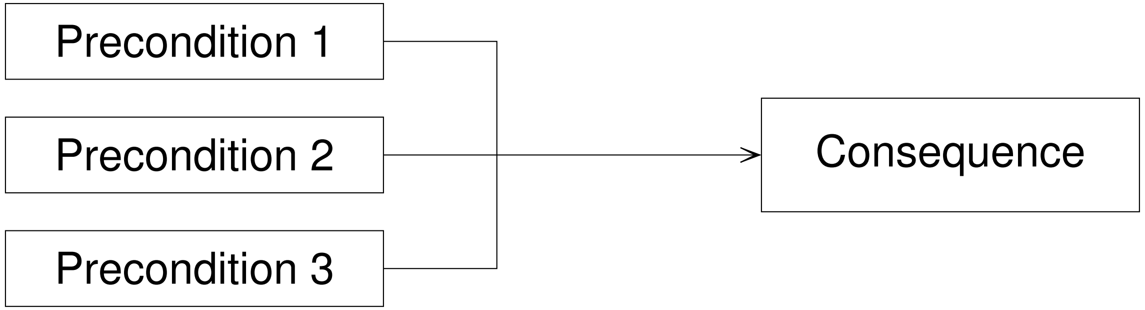
Martin Helmke

20.07.2023

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

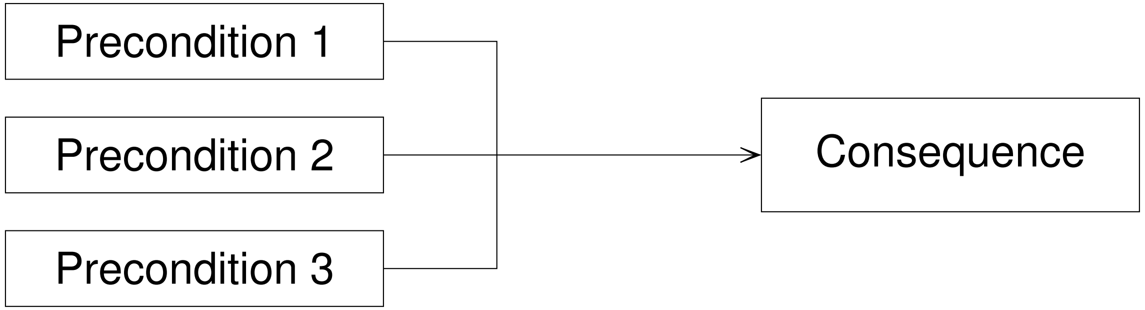
1. **Introduction**
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

AI & Law



$$P_1 \land P_2 \land P_3 \Rightarrow C$$

AI & Law



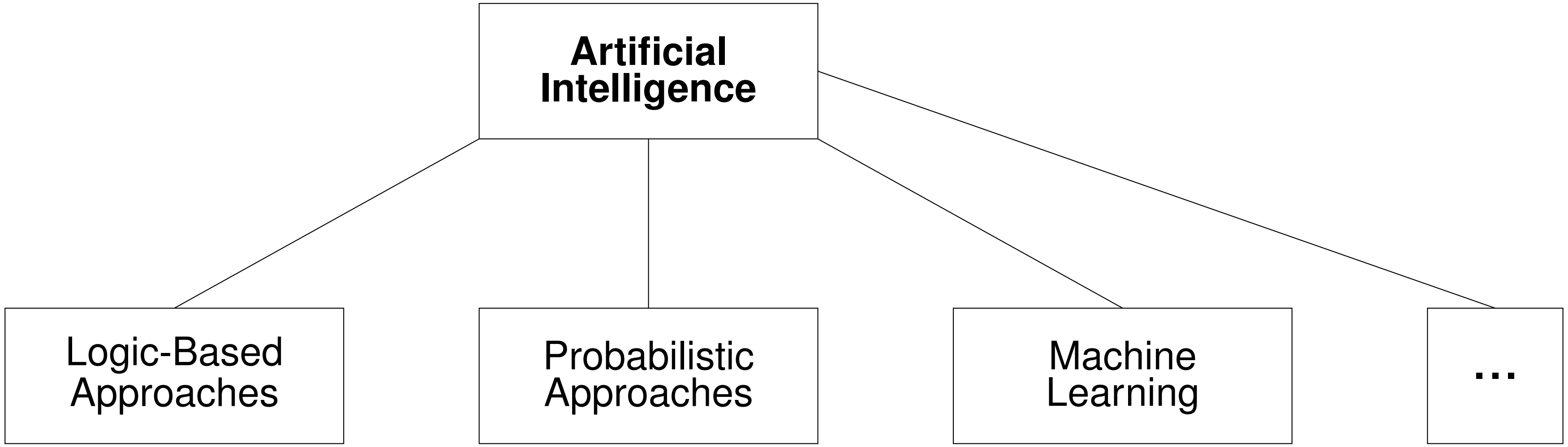
Precondition 1

Precondition 2

Precondition 3

Consequence

$$P_1 \wedge P_2 \wedge P_3 \Rightarrow C$$



Artificial Intelligence

Logic-Based Approaches

Probabilistic Approaches

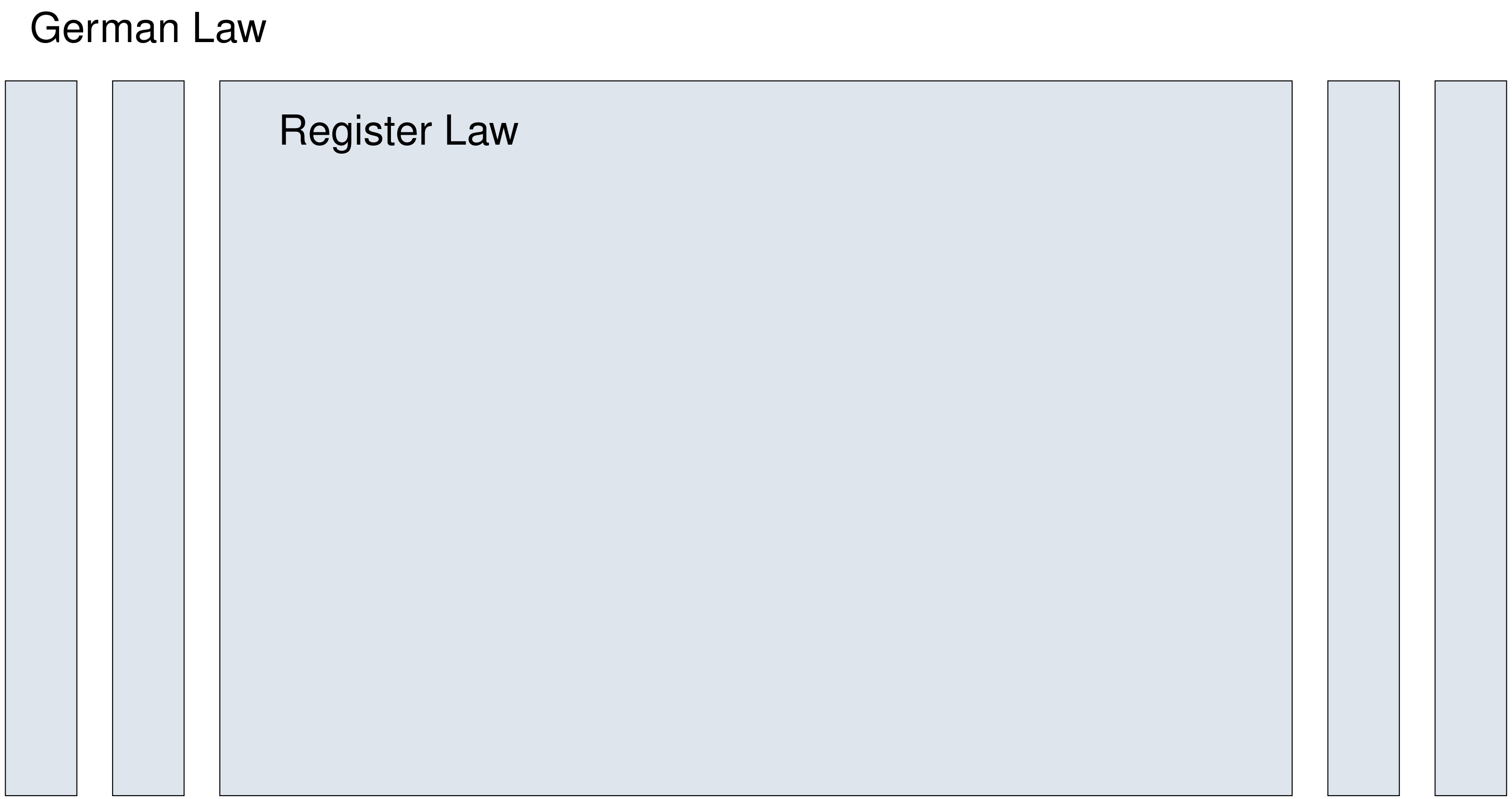
Machine Learning

...

Register Law

German Law

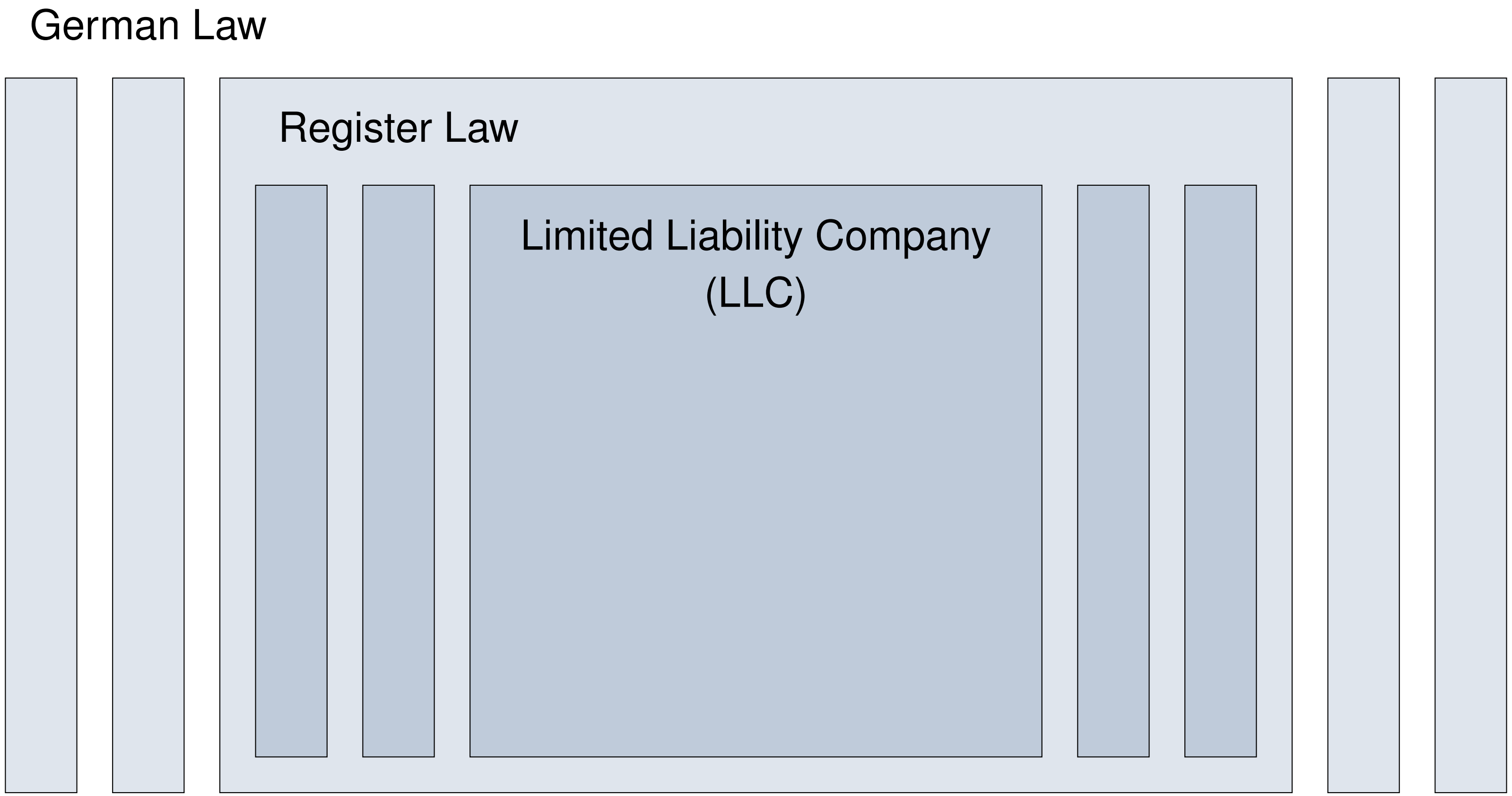
Register Law



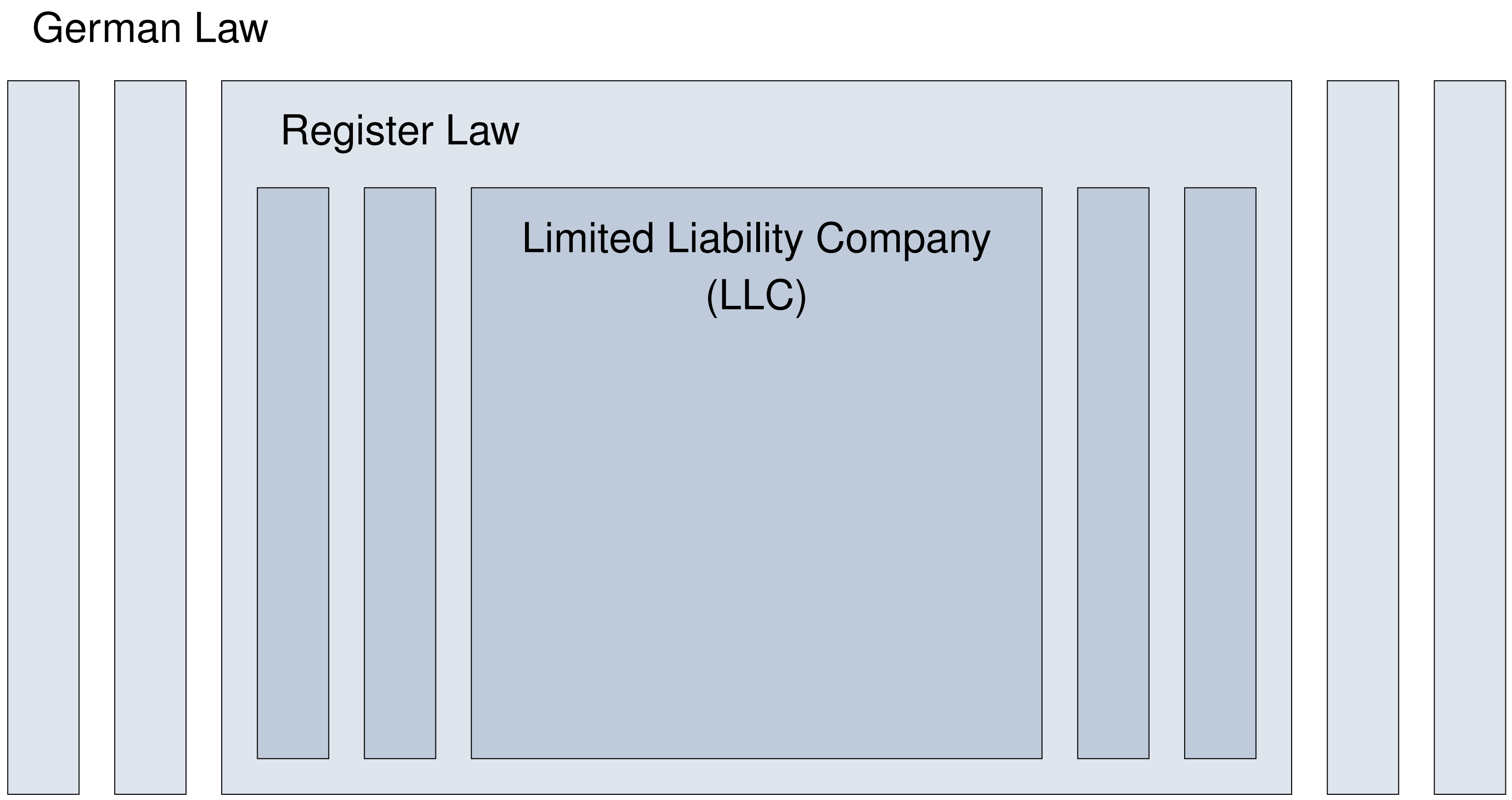
German Law

Register Law

Register Law



Register Law

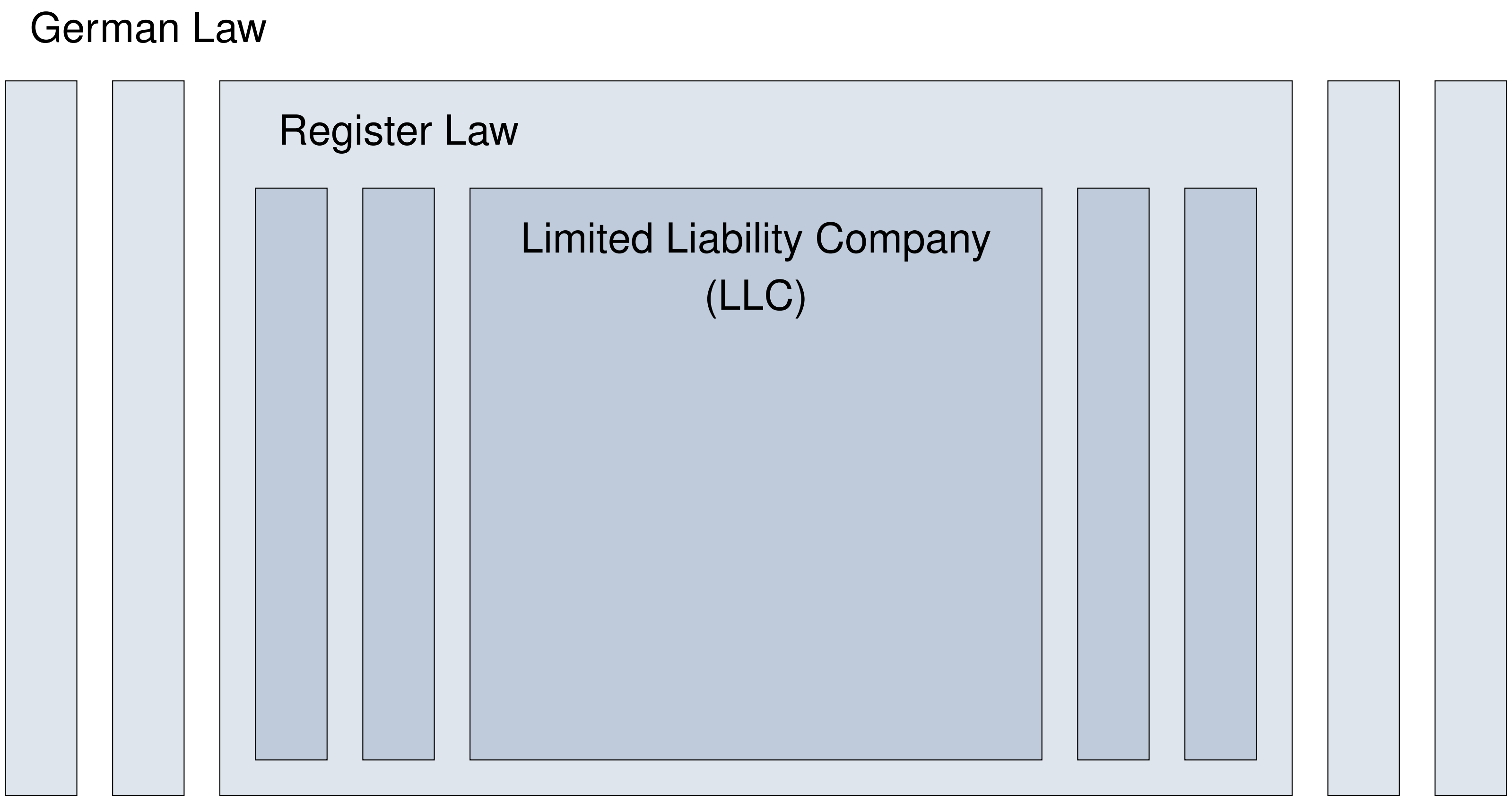


German Law

Register Law

Limited Liability Company (LLC)

Register Law



- 47 LLC processes
- Director appointment
- Requirements check

Research Questions

Sufficiency of Logic

- Is logic a sufficiently powerful means to formalize and automate company register processes?

Necessity of Logic

- Is logic a necessary means to formalize and automate company register processes?

Most Suitable Logic

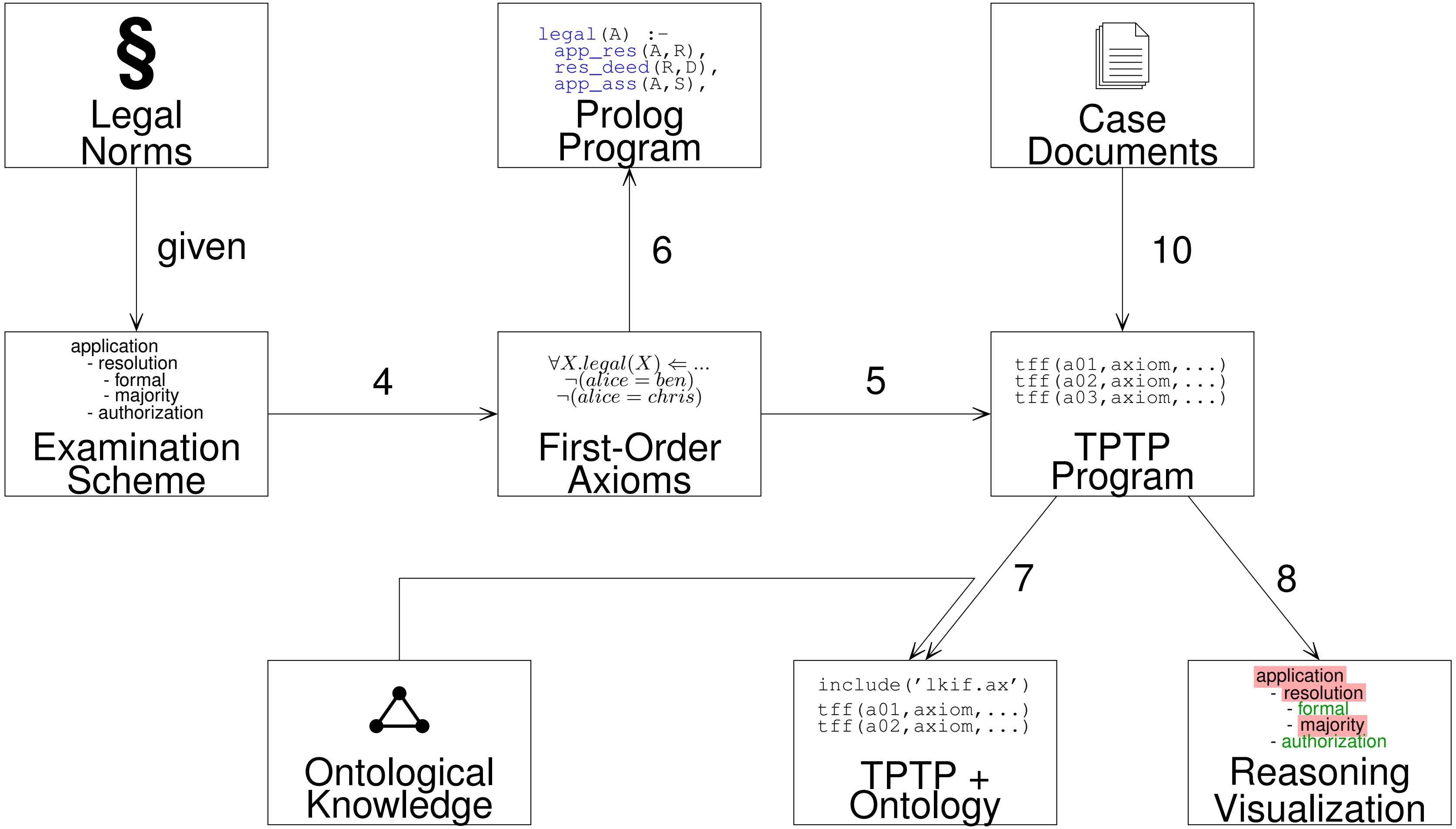
- Which kind of logic is most suitable for formalizing and automating company register processes?

Contribution

- **Formalization** in many-sorted first-order logic with conventions
- **Practical Implementation** in TPTP and Prolog
- **Reasoning Traceability** by subdividing the preconditions and checking which are valid or not
- **Ontology Incorporation** of LKIF Core and LegalRuleML
- **General Suitability of Logic** can be concluded
- **Limited Suitability of First-Order Logic** due to montonicity and semi-decidability

Thesis Structure

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in FOL
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Introduction
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion



§

Legal Norms

given

application
- resolution
- formal
- majority
- authorization

Examination Scheme

4

$\forall X.legal(X) \Leftarrow \ldots$
$\neg(alice = ben)$
$\neg(alice = chris)$

First-Order Axioms

6

```
legal(A) :-
  app_res(A,R),
  res_deed(R,D),
  app_ass(A,S),
```

Prolog Program

5

Case Documents

10

```
tff(a01,axiom,...)
tff(a02,axiom,...)
tff(a03,axiom,...)
```

TPTP Program

7

8

Ontological Knowledge

```
include('lkif.ax')
tff(a01,axiom,...)
tff(a02,axiom,...)
```

TPTP + Ontology

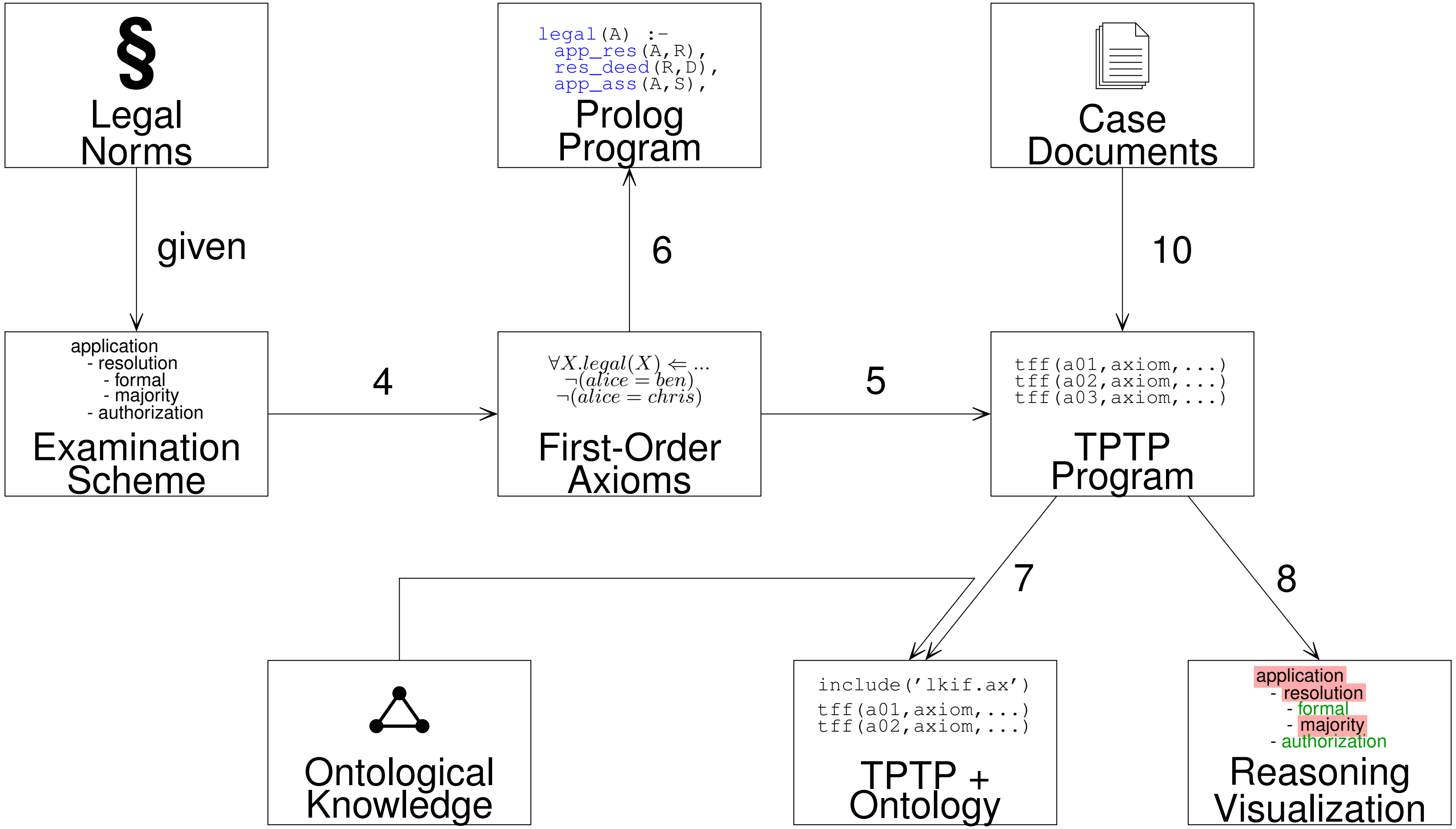
application
- resolution
- formal
- majority
- authorization

Reasoning Visualization

Thesis Structure

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in FOL
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Introduction
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

→ Evaluation sections



§ Legal Norms

given

```
legal(A) :-
  app_res(A,R),
  res_deed(R,D),
  app_ass(A,S),
```

Prolog Program

6

Case Documents

10

application
- resolution
  - formal
  - majority
- authorization

Examination Scheme

4

$\forall X. legal(X) \Leftarrow ...$
$\neg(alice = ben)$
$\neg(alice = chris)$

First-Order Axioms

5

```
tff(a01,axiom,...)
tff(a02,axiom,...)
tff(a03,axiom,...)
```

TPTP Program

7

8

Ontological Knowledge

```
include('lkif.ax')
tff(a01,axiom,...)
tff(a02,axiom,...)
```

TPTP + Ontology

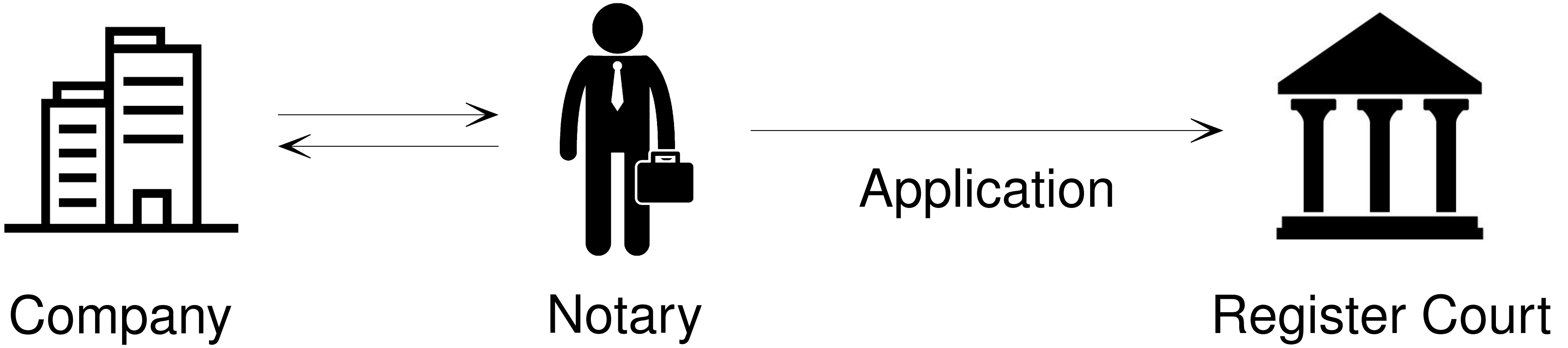
application
- resolution
  - formal
  - majority
- authorization

Reasoning Visualization

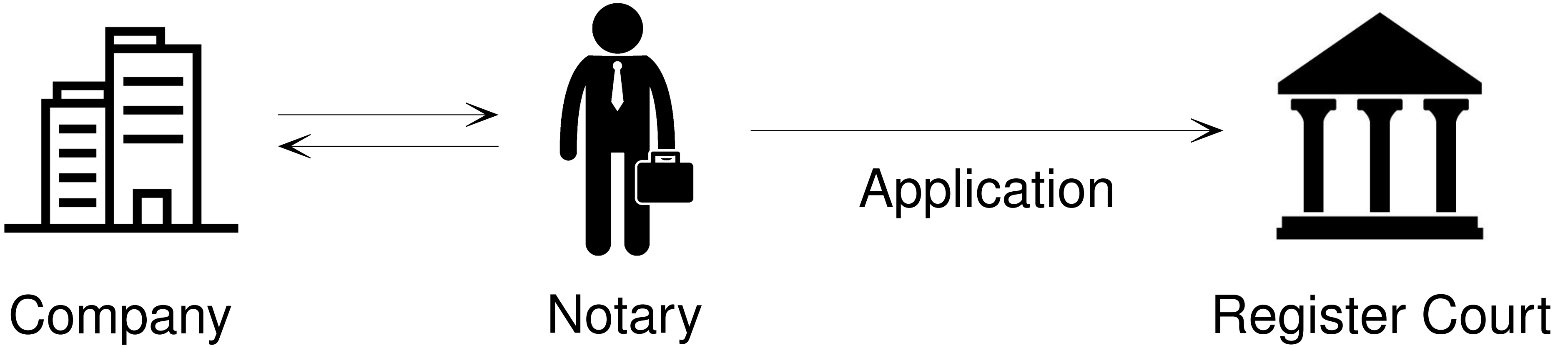
Friedrich-Alexander-Universität
Erlangen-Nürnberg

1. Introduction
2. **Use Case: Director Application**
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

Register Applications

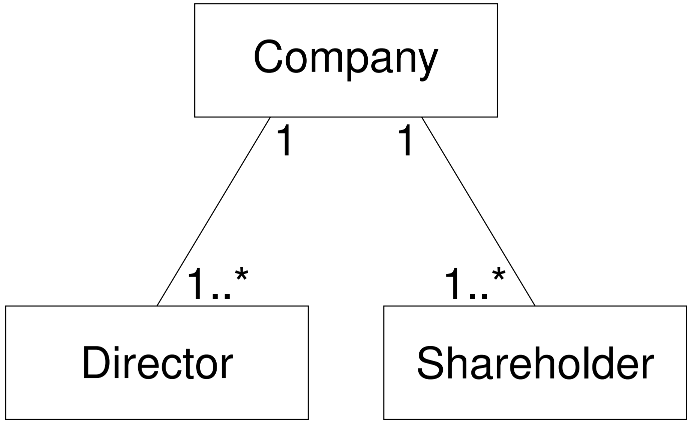


Register Applications

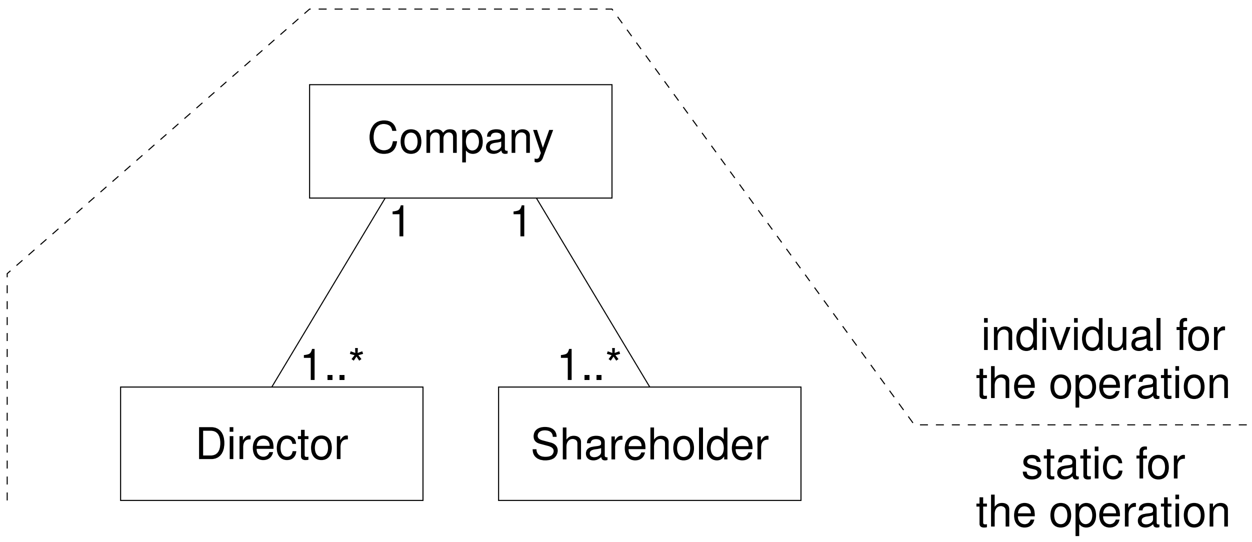


- Company formation
- Amendment to articles of association
- Retirement of a shareholder
- **Appointment of a new director**
- Resignation of a director
- Relocation of the business address
- Increase in share capital
- Company liquidation
- ...

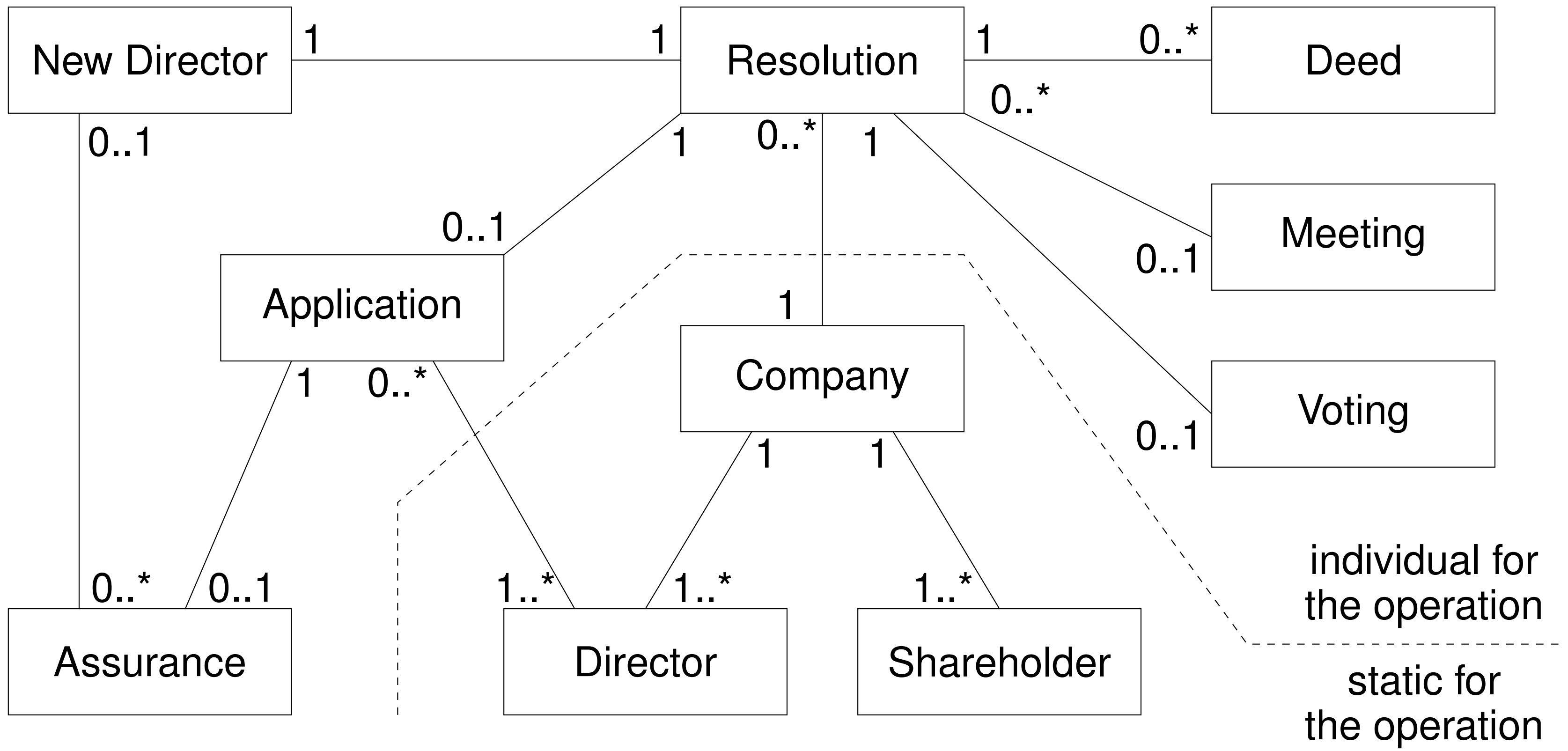
Director Appointment: Overview



Director Appointment: Overview



Director Appointment: Overview



Director Appointment: Examination Scheme

The application is legal.

1. The resolution is made legally.
   - 1.1. The resolution is formally legal.
     - case 1.1.1. The resolution is made in a meeting.
     - case 1.1.2. There is no meeting.
   - 1.2. The resolution is made with a majority vote.
     - case 1.2.1. No special AoA majority requirement.
     - case 1.2.2. Special AoA majority requirement.
2. The deed of the director's appointment is attached.
   - case 2.1. The deed is original.
   - case 2.2. The deed is a certified copy.
3. The new director has signed an assertion for being qualified.
4. The applying directors are authorized to represent the company.
   - case 4.1. One applicant has sole representation power.
   - case 4.2. One applicant has modified representation power and another director is present.
   - case 4.3. All directors perform the application.

Director Appointment: Examination Scheme

The application is legal.

1. The resolution is made legally.

1.1. The resolution is formally legal.

case 1.1.1. The resolution is made in a meeting.

case 1.1.2. There is no meeting.

1.2. The resolution is made with a majority vote.

case 1.2.1. No special AoA majority requirement.

case 1.2.2. Special AoA majority requirement.

2. The deed of the director's appointment is attached.

case 2.1. The deed is original.

case 2.2. The deed is a certified copy.

3. The new director has signed an assertion for being qualified.

4. The applying directors are authorized to represent the company.

case 4.1. One applicant has sole representation power.

case 4.2. One applicant has modified representation power and another director is present.

case 4.3. All directors perform the application.

- Contains entities from diagram
- Tree-shaped structure
- Scheme root:
  Is application legal?

Director Appointment: Examination Scheme

The application is legal.

1. The resolution is made legally.

1.1. The resolution is formally legal.

case 1.1.1. The resolution is made in a meeting.
case 1.1.2. There is no meeting.

1.2. The resolution is made with a majority vote.

case 1.2.1. No special AoA majority requirement.
case 1.2.2. Special AoA majority requirement.

2. The deed of the director's appointment is attached.

case 2.1. The deed is original.
case 2.2. The deed is a certified copy.

3. The new director has signed an assertion for being qualified.

4. The applying directors are authorized to represent the company.

case 4.1. One applicant has sole representation power.
case 4.2. One applicant has modified representation power and another director is present.
case 4.3. All directors perform the application.

- ▶ Contains entities from diagram
- ▶ Tree-shaped structure
- ▶ Scheme root: Is application legal?
- ▶ Necessary conditions
- ▶ Sufficient conditions
- ▶ Case distinctions

Director Appointment: Examination Scheme

The application is legal.

1. The resolution is made legally.

1.1. The resolution is formally legal.

case 1.1.1. The resolution is made in a meeting.

case 1.1.2. There is no meeting.

1.2. The resolution is made with a majority vote.

case 1.2.1. No special AoA majority requirement.

case 1.2.2. Special AoA majority requirement.

2. The deed of the director's appointment is attached.

case 2.1. The deed is original.

case 2.2. The deed is a certified copy.

3. The new director has signed an assertion for being qualified.

4. The applying directors are authorized to represent the company.

case 4.1. One applicant has sole representation power.

case 4.2. One applicant has modified representation power and another director is present.

case 4.3. All directors perform the application.

- Contains entities from diagram
- Tree-shaped structure
- Scheme root:
  Is application legal?
- Necessary conditions
- Sufficient conditions
- Case distinctions
- Implicit assumptions

Evaluation

- Examination Scheme as Basis
- Moderate Complexity of the Director Appointment
- Conjunction- & Disjunction-Based Scheme
- Implicit Assumptions
- Operation-Related & Operation-Independent Details

1. Introduction
2. Use Case: Director Application
3. **Technical Preliminaries**
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

Logic Infrastructures

TPTP

- Library of logic problems for theorem provers
- Support for several logics
- Language for phrasing formulas
- Several theorem provers
- First-order standard semantics

Prolog

- Logic programming
- Restriction to Horn clauses with universally quantified variables: $\forall X.\ P_1 \land P_2 \land ... \Rightarrow C$
- Language for phrasing facts and rules
- Interpreter SWI-Prolog
- Database semantics

TPTP

Language Syntax

```
tff(a01, axiom, app_res(app1, res1)).
tff(a02, axiom, is_res_legal(res1)).
tff(a03, axiom,
  ! [A: app] : (
      is_app_legal(A)
    <=
    ? [R: res] : (
        app_res(A, R)
        is_res_legal(R)
    )
  )
).
tff(theorem_in_question, conjecture,
  is_app_legal(app1)
).
```

Theorem Provers

- cvc5
- iProver
- Leo-III
- Princess
- Vampire

First-Order Dialects

- We use TFF (Typed First-Order Form)
- instead of FOF (First-Order Form)

Prolog

Language Syntax

```
app_res(app1, res1).
is_res_legal(res1).
is_app_legal(A) :-
  app_res(A, R),
  is_res_legal(R).

?- is_app_legal(app1).
```

Interpreter

Evaluation

- **Infrastructures for Implementation**
- **Limited Expressiveness of Prolog** (Horn clauses)

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. **Formalization in First-Order Logic**
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

Juridical Syllogism

Base of a judge’s decision making:

- **Major Premise** according to a legal norm
- **Minor Premise** according to the facts under consideration
- **Conclusion** resulting from the subsumption

Juridical Syllogism

Base of a judge’s decision making:

- **Major Premise** according to a legal norm
- **Minor Premise** according to the facts under consideration
- **Conclusion** resulting from the subsumption

Example:

- **Major Premise**: Whoever physically assaults another person, incurs a penalty.
- **Minor Premise**: A punches B in the face.
- **Conclusion**: A incurs a penalty.

General Set-Up

- Many-sorted first-order logic

  $$\forall_s X.\mathbf{A} \equiv \forall X.(\mathcal{Q}_s(X) \Rightarrow \mathbf{A})$$

  $$\exists_s X.\mathbf{A} \equiv \exists X.(\mathcal{Q}_s(X) \wedge \mathbf{A})$$

  where $\mathcal{Q}_s(\cdot)$ indicates whether the argument is of sort s

- Law-related vs. case-related axioms

- Predicates with at most two arguments
  - **Case facts**: binary or monadic
  - **Validity rules**: monadic
  - **Relations**: binary

- Root rule predicate (root in scheme): $is\_application\_legal \rightarrow$ Validity rule

- Root individual $app1$

Validity Rule Axioms (1)

- Example: Validity rule for $is\_application\_legal$
- Validity rule specifies the validity property of an individual

Validity Rule Axioms (1)

$\forall_{app} A.\ ($

$)$

- Example: Validity rule for $is\_application\_legal$
- Validity rule specifies the validity property of an individual
- Sorted universal quantification

Validity Rule Axioms (1)

$$
\begin{aligned}
&\forall_{app} A.\ (\\
&\quad (\\
&\quad) \Rightarrow is\_application\_legal(A)\\
&)
\end{aligned}
$$

- Example: Validity rule for $is\_application\_legal$
- Validity rule specifies the validity property of an individual
- Sorted universal quantification
- Validity is implied by the preconditions

Validity Rule Axioms (1)

$$
\begin{array}{l}
\forall_{app} A.\ (\\
\quad (\\
\qquad \exists_{res} R.\ \exists_{deed} D.\ \exists_{ass} S.\ (\\
\qquad\qquad is\_resolution\_legal(R) \\
\qquad\quad \wedge\ is\_deed\_legal(D) \\
\qquad\quad \wedge\ is\_assurance\_legal(S) \\
\qquad\quad \wedge\ are\_applicants\_authorized(A) \\
\qquad) \\
\quad) \Rightarrow is\_application\_legal(A) \\
)
\end{array}
$$

- Example: Validity rule for $is\_application\_legal$
- Validity rule specifies the validity property of an individual
- Sorted universal quantification
- Validity is implied by the preconditions
- Conjunction or disjunction of preconditions
- Optional: Existential quantification

Validity Rule Axioms (1)

$$
\begin{array}{l}
\forall_{app} A.\ (\\
\quad (\\
\quad\quad \exists_{res} R.\ \exists_{deed} D.\ \exists_{ass} S.\ (\\
\quad\quad\quad\quad is\_resolution\_legal(R) \\
\quad\quad\quad \wedge\ is\_deed\_legal(D) \\
\quad\quad\quad \wedge\ is\_assurance\_legal(S) \\
\quad\quad\quad \wedge\ are\_applicants\_authorized(A) \\
\quad\quad\quad \wedge\ application\_resolution(A, R) \\
\quad\quad\quad \wedge\ resolution\_deed(R, D) \\
\quad\quad\quad \wedge\ application\_assurance(A, S) \\
\quad\quad) \\
\quad) \Rightarrow is\_application\_legal(A) \\
)
\end{array}
$$

- Example: Validity rule for $is\_application\_legal$
- Validity rule specifies the validity property of an individual
- Sorted universal quantification
- Validity is implied by the preconditions
- Conjunction or disjunction of preconditions
- Optional: Existential quantification
- Navigation via relations

Validity Rule Axioms (1)

$$
\begin{aligned}
&\forall_{app} A.\ (\\
&\quad (\\
&\qquad \exists_{res} R.\ \exists_{deed} D.\ \exists_{ass} S.\ (\\
&\qquad\qquad is\_resolution\_legal(R)\\
&\qquad \ \wedge\ is\_deed\_legal(D)\\
&\qquad \ \wedge\ is\_assurance\_legal(S)\\
&\qquad \ \wedge\ are\_applicants\_authorized(A)\\
&\qquad \ \wedge\ application\_resolution(A, R)\\
&\qquad \ \wedge\ resolution\_deed(R, D)\\
&\qquad \ \wedge\ application\_assurance(A, S)\\
&\qquad)\\
&\quad) \Rightarrow is\_application\_legal(A)\\
&)
\end{aligned}
$$

- Example: Validity rule for $is\_application\_legal$
- Validity rule specifies the validity property of an individual
- Sorted universal quantification
- Validity is implied by the preconditions
- Conjunction or disjunction of preconditions
- Optional: Existential quantification
- Navigation via relations

$\rightarrow$ similar to Prolog Horn clause

Validity Rule Axioms (2)

Necessary conditions (with relations)

$$
\begin{aligned}
&\forall_{app} A.\ (\\
&\quad \exists_{res} R.\ \exists_{deed} D.\ \exists_{ass} S.\ (\\
&\qquad\quad is\_resolution\_legal(R)\\
&\qquad \wedge\ is\_deed\_legal(D)\\
&\qquad \wedge\ is\_assurance\_legal(S)\\
&\qquad \wedge\ are\_applicants\_authorized(A)\\
&\qquad \wedge\ application\_resolution(A, R)\\
&\qquad \wedge\ resolution\_deed(R, D)\\
&\qquad \wedge\ application\_assurance(A, S)\\
&\quad) \Rightarrow is\_application\_legal(A)\\
&)
\end{aligned}
$$

Necessary conditions (without relations)

$$
\begin{aligned}
&\forall_{res} R.\ (\\
&\quad (\quad is\_resolution\_formally\_legal(R)\\
&\qquad \wedge\ has\_resolution\_majority(R)\\
&\quad) \Rightarrow is\_resolution\_legal(R)\\
&)
\end{aligned}
$$

Sufficient conditions & case distinctions

$$
\begin{aligned}
&\forall_{deed} D.\ (\\
&\quad (\quad deed\_format(D, original)\\
&\qquad \vee\ deed\_format(D, certifiedcopy)\\
&\quad) \Rightarrow is\_deed\_legal(D)\\
&)
\end{aligned}
$$

World-Closing: Predicate Completion (1)

Rule axiom

$$
\begin{array}{l}
\forall_{com} C.\ (\\
\quad \forall_{sh} S.\ (\\
\qquad shareholder\_company(S, C)\\
\qquad \Rightarrow\\
\qquad shareholder\_consents(S)\\
\quad) \Rightarrow all\_shareholders\_consent(C)\\
)
\end{array}
$$

Case facts

$shareholder\_company(alice, com1).$

$shareholder\_company(ben, com1).$

$shareholder\_company(chris, com1).$

$shareholder\_consents(alice).$

$shareholder\_consents(ben).$

$shareholder\_consents(chris).$

World-Closing: Predicate Completion (1)

Rule axiom

$$
\begin{aligned}
&\forall_{com} C.\ (\\
&\quad \forall_{sh} S.\ (\\
&\qquad shareholder\_company(S, C)\\
&\qquad \Rightarrow\\
&\qquad shareholder\_consents(S)\\
&\quad) \Rightarrow all\_shareholders\_consent(C)\\
&)
\end{aligned}
$$

Case facts

$shareholder\_company(alice, com1).$
$shareholder\_company(ben, com1).$
$shareholder\_company(chris, com1).$
$shareholder\_company(dana, com1).$

$shareholder\_consents(alice).$
$shareholder\_consents(ben).$
$shareholder\_consents(chris).$

- $all\_shareholders\_consent(com1)$ is not entailed by the rule axiom and the case facts!
- Conclusiveness of knowledge must also be stated.

World-Closing: Predicate Completion (2)

Facts without predicate completion

$$\begin{aligned}
&shareholder\_company(alice, com1).\\
&shareholder\_company(ben, com1).\\
&shareholder\_company(chris, com1).
\end{aligned}$$

Facts with predicate completion

$$\begin{array}{l}
\forall_{sh} S.\ \forall_{com} C.\ (\\
\quad shareholder\_company(S, C)\\
\quad \iff\\
\quad (\quad (S = alice \quad \wedge\ C = com1)\\
\quad\ \ \vee\ \ (S = ben \quad \wedge\ C = com1)\\
\quad\ \ \vee\ \ (S = chris \quad \wedge\ C = com1)\\
\quad)\\
)
\end{array}$$

Further Conventions

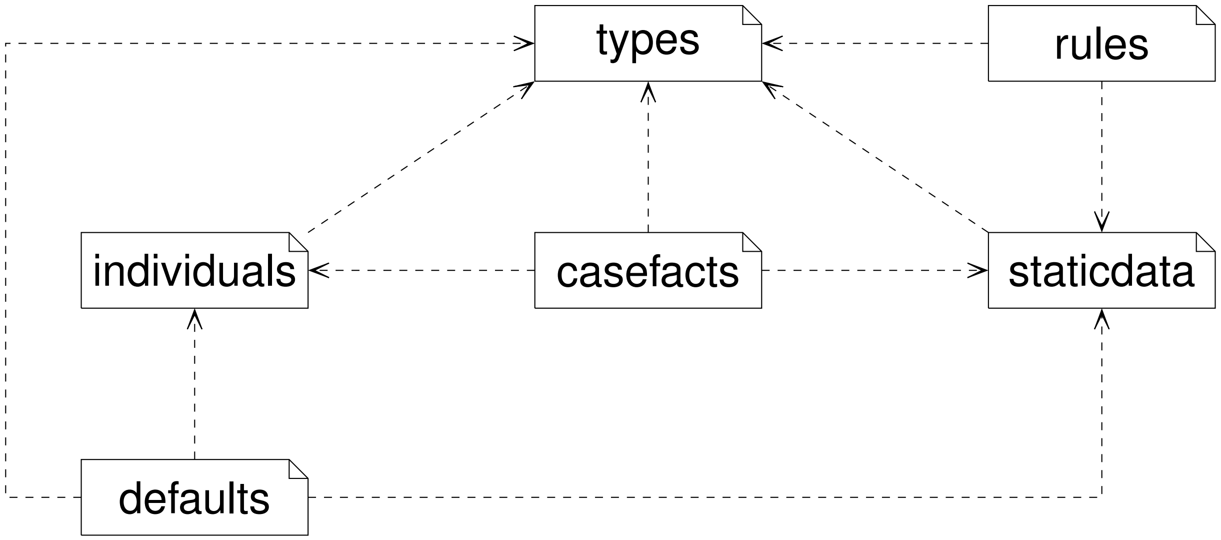
- No Individual for Non-Existent Entity
- Binary Representation Pattern
- Conclusive Sort Domains
- Inequality Declarations
- Default Values
- Assumed Subconditions

Evaluation

- Successful Realization
- Many-Sorted Conciseness
- General Rules vs. Case-Related Facts
- Conventions for Rule Definition
- Deformation due to World-Closing
- Weightiness due to World-Closing
- High Manual Effort

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. **Implementation in TPTP**
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

Program Assembling & Prover Invocation



```
~$ vampire -t 5 -p off program.p
% Running in auto input_syntax mode. Trying
  TPTP
% Refutation found. Thanks to Tanya!
% SZS status Theorem for program
% ------------------------------
```

program.p

```
include('types.ax').
include('staticdata.ax').
include('rules.ax').
include('individuals.ax').
include('casefacts.ax').
include('defaults.ax').

tff(theorem_in_question,
  conjecture,
  is_application_legal(app1)
).
```

Prover Performance

	cvc5	iProver	Leo-III	Princess	Vampire
Trivial Program (valid)	✓	✓	✓	✓	✓
Trivial Program (falsifiable)	✓	✓	✓	✗	✓
File Inclusion	✓	✗	✓	✓	✓
Arithmetic (valid)	✓	✓	✗	✓	✓
Arithmetic (falsifiable)	✓	✗	✗	✗	✗
Predicate Completion	✗	✓	✓	✓	✓
Runtime	✓	✓	✗	✗	✓

Prover Performance

	cvc5	iProver	Leo-III	Princess	Vampire
Trivial Program (valid)	✓	✓	✓	✓	✓
Trivial Program (falsifiable)	✓	✓	✓	✗	✓
File Inclusion	✓	✗	✓	✓	✓
Arithmetic (valid)	✓	✓	✗	✓	✓
Arithmetic (falsifiable)	✓	✗	✗	✗	✗
Predicate Completion	✗	✓	✓	✓	✓
Runtime	✓	✓	✗	✗	✓

Evaluation

- **Qualified Implementation Success**

- **Rejected Prover Tools**

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. **Implementation in Prolog**
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

Program Assembling & Interpreter Invocation

program.pl

```
:-[rules].
:-[individuals].
:-[casefacts].
:-[defaults].
```

```
~$ swipl -s program.pl -g "is_application_legal(app1)" -g halt
ERROR: -g is_application_legal(app1): false
```

Differences to TPTP

No built-in sorts

- No sort/type system
- Emulation via explicit predicate

No world-closing

- Closed-world assumption
- Unique name assumption

List operations

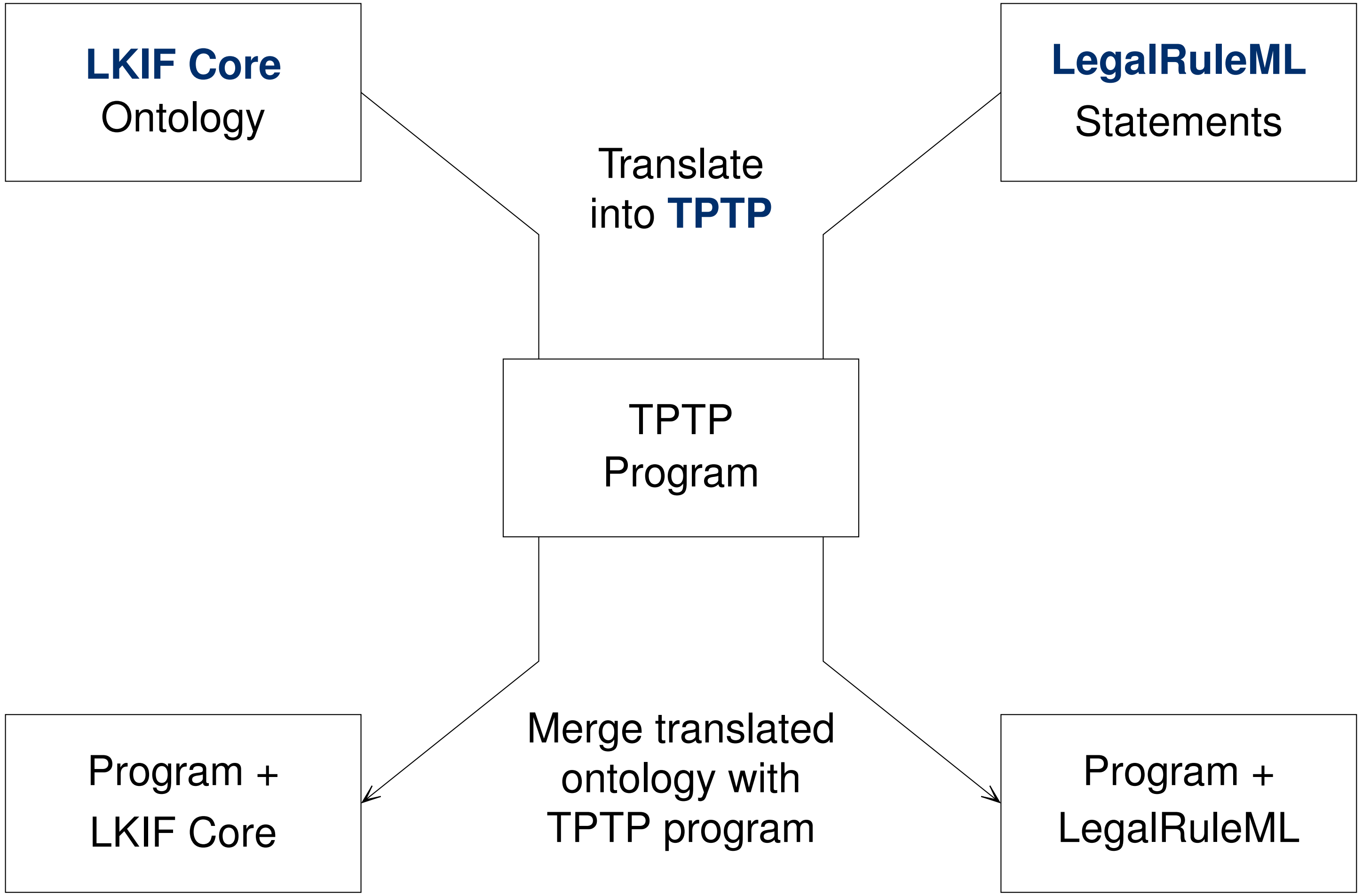
- Restriction to universally quantified Horn clauses
- Many rules (e. g. $all\_shareholders\_consent$) can be implemented using list operations

Evaluation

- **Feasibility of Implementation**
- **Indirect Axiom Translation**
- **No World-Closing**
- **Lack of Sorts**
- **Excellent Tool Support**

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

General Approach



LKIF Core Ontology

LegalRuleML Statements

Translate into **TPTP**

TPTP Program

Merge translated ontology with TPTP program

Program + LKIF Core

Program + LegalRuleML

LKIF Core

- Core Ontology
- Phrased in Web Ontology Language (OWL) which is XML-based
- 200 abstract concepts in 13 modules
- Translation to TPTP (FOF) via *FOWL* (by Flügel et al.)
- Practical issues of merging TFF (implementation) and FOF (ontology)

LegalRuleML

- Markup language for modelling legal knowledge and rules
- Semantic web standards (IRI)
- XML structure: metadata, statements, context
- Support for deontic operators and defeasibility
- Limited translation TPTP by Steen & Fuenmayor; not practically performed

Evaluation

- **Knowledge Reuse**
- **Minor Practical Issues**
- **Participation in the Semantic Web**

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. Conclusion

Motivation & General Approach

How to make a conclusion comprehensible?

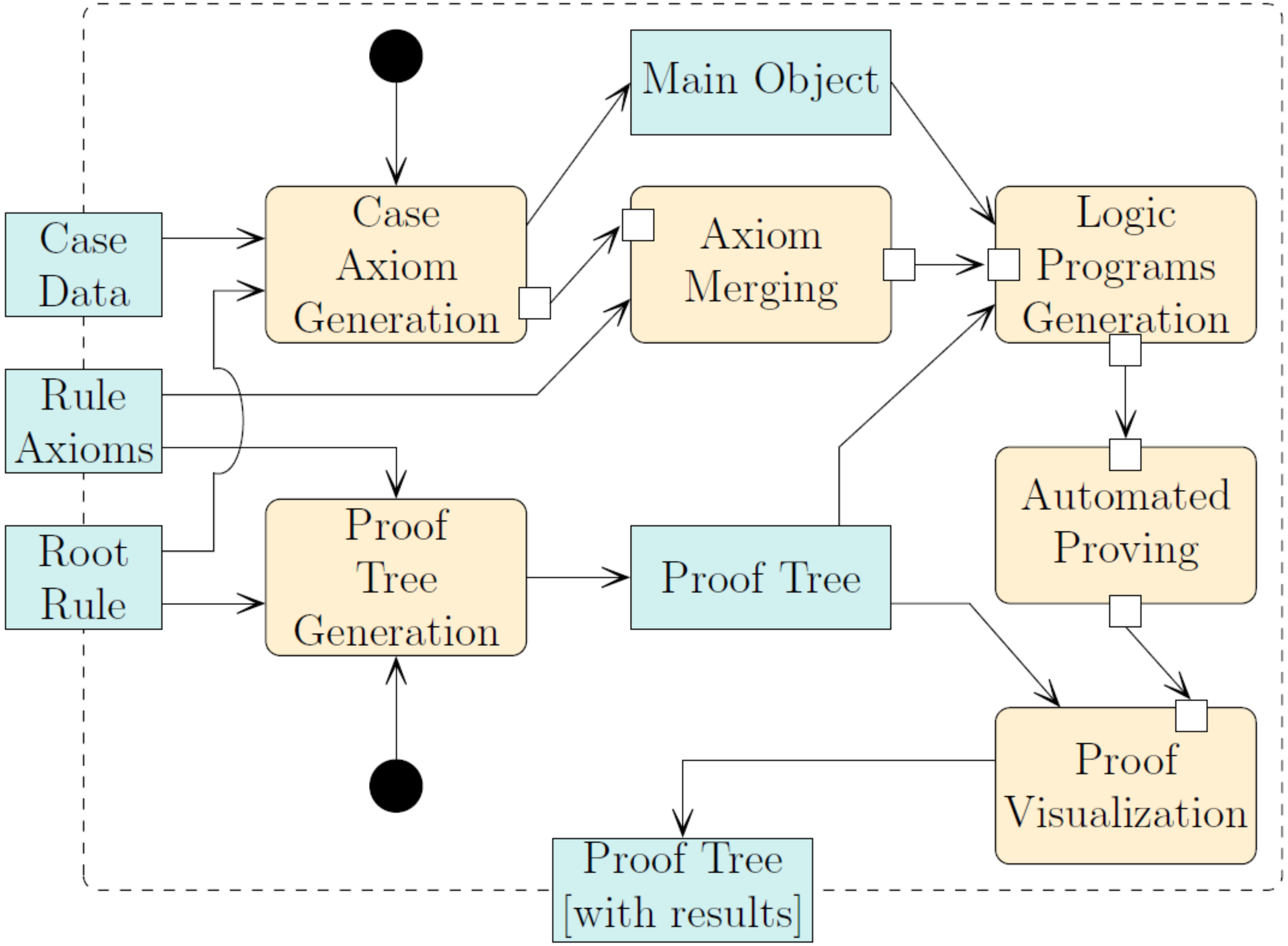
- Symbolic AI allows reconstructing a conclusion
- Prover output is very complex and only available for positive proofs
- Recursively subdivide rule axioms (by exploiting the conventions)
- Separately check the subordinate rules/conditions
- Visualize the checked rules

→ implemented for TPTP

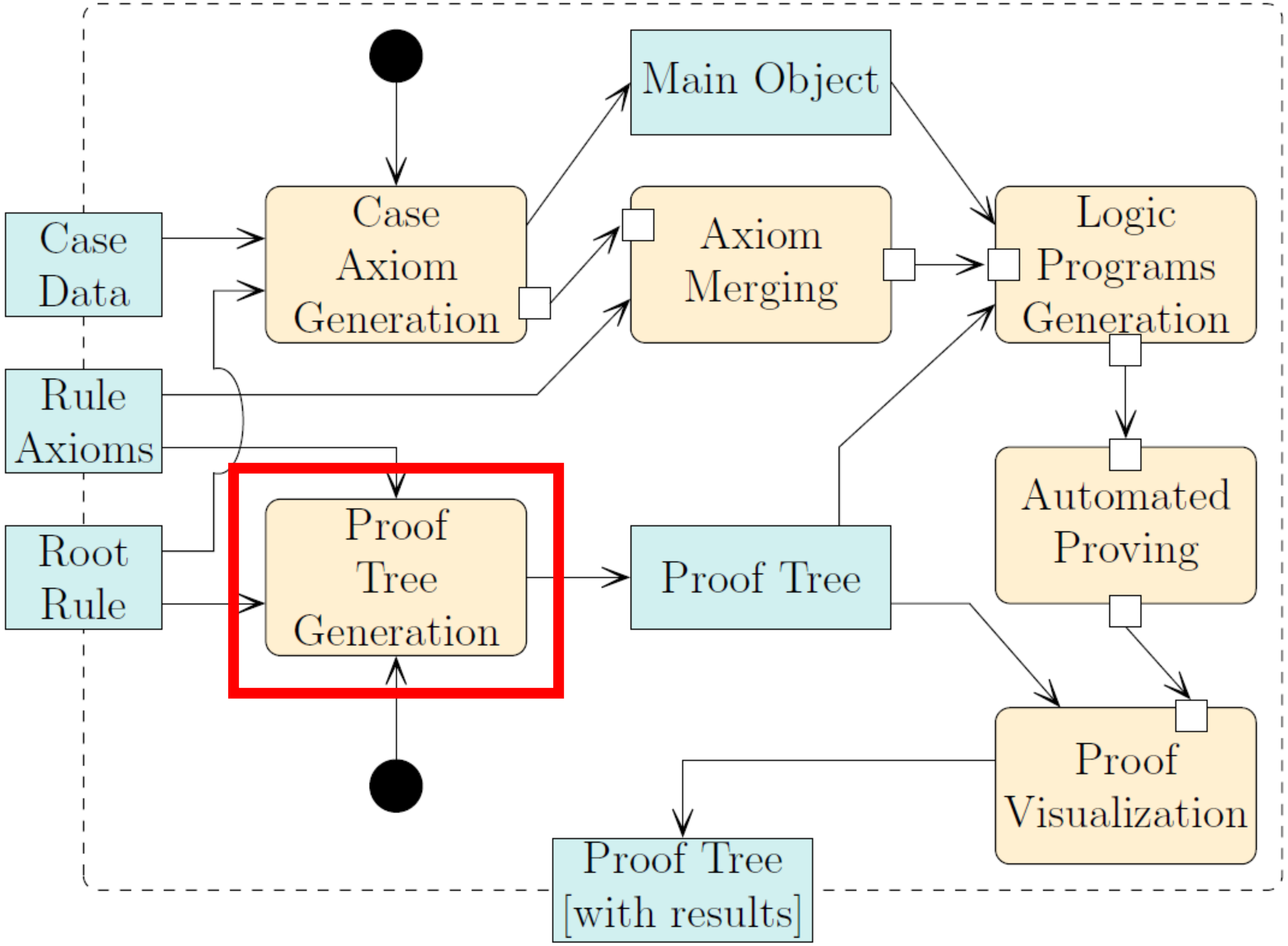
Case Data File

```
{
  "company": {
    "shareholders": [ { "name": "alice", "votes": 50001 },
                      { "name": "ben",   "votes": 39999 },
                      { "name": "chris", "votes": 10000 } ],
    "directors": [  { "name": "jacob", "representationpower": ""          },
                    { "name": "kate",  "representationpower": "modified"  },
                    { "name": "chris", "representationpower": "sole"      } ],
    "majorityrequirement": 0.60
  },
  "resolution": {
    "new_director": { "name": "luca", "representationpower": "joint"     },
    "meeting": { "occurred": true, "format": personal },
    "voting": { "yes_votes": 60001, "no_votes": 39999, "abstentions": 0 }
  },
  "application": {
    "assurance_signed": true,
    "deed_format": "original",
    "applicants": [ "jacob", "kate" ]
  }
}
```

Program Components



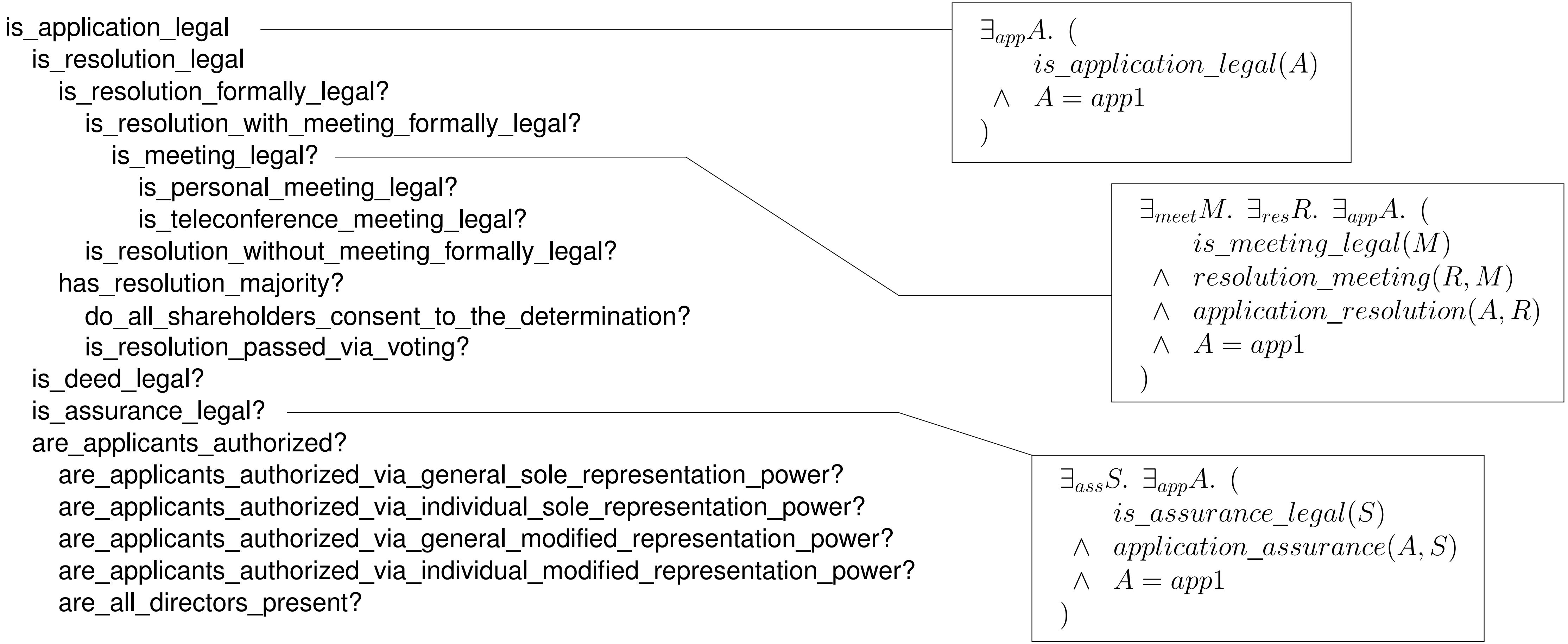
Program Components



Proof Tree Generation

```
is_application_legal
  is_resolution_legal
    is_resolution_formally_legal?
      is_resolution_with_meeting_formally_legal?
        is_meeting_legal?
          is_personal_meeting_legal?
          is_teleconference_meeting_legal?
      is_resolution_without_meeting_formally_legal?
    has_resolution_majority?
      do_all_shareholders_consent_to_the_determination?
      is_resolution_passed_via_voting?
  is_deed_legal?
  is_assurance_legal?
  are_applicants_authorized?
    are_applicants_authorized_via_general_sole_representation_power?
    are_applicants_authorized_via_individual_sole_representation_power?
    are_applicants_authorized_via_general_modified_representation_power?
    are_applicants_authorized_via_individual_modified_representation_power?
    are_all_directors_present?
```

Proof Tree Generation



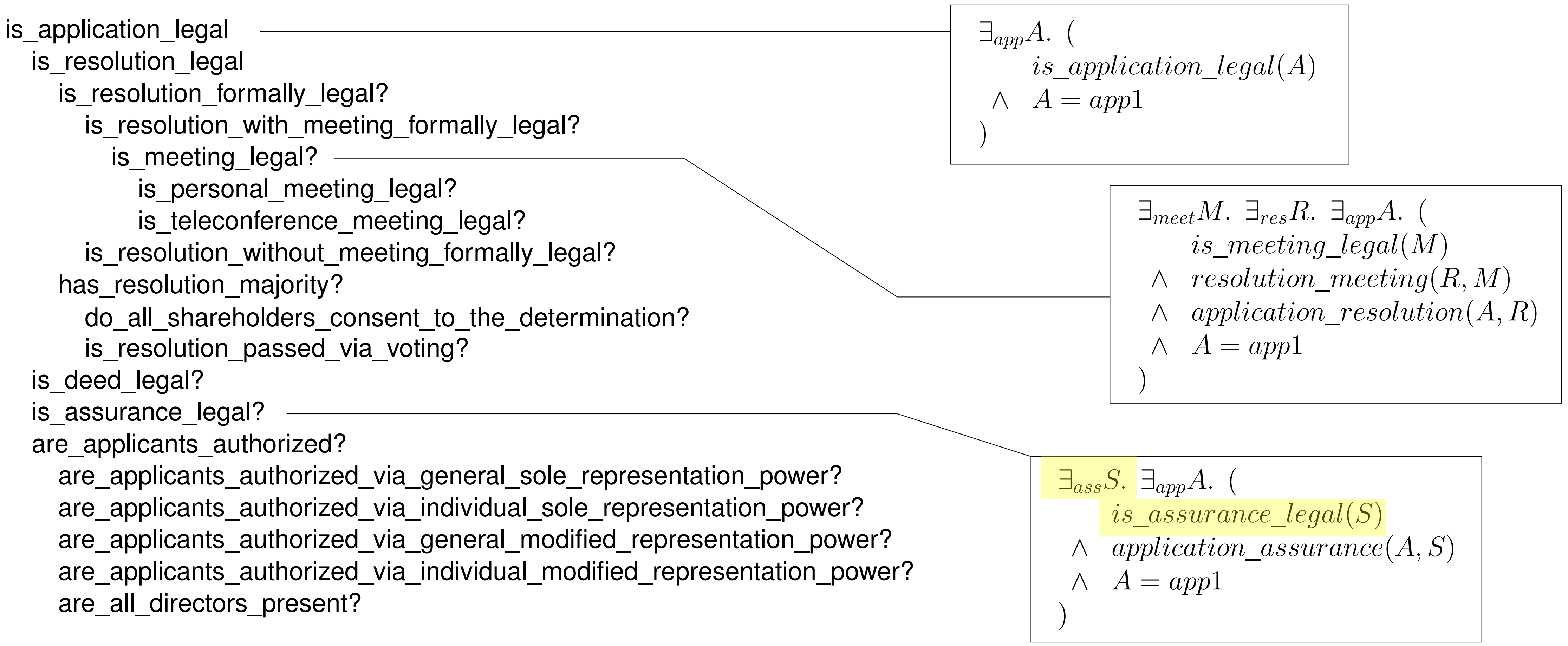
- is_application_legal
  - is_resolution_legal
    - is_resolution_formally_legal?
      - is_resolution_with_meeting_formally_legal?
        - is_meeting_legal?
          - is_personal_meeting_legal?
          - is_teleconference_meeting_legal?
      - is_resolution_without_meeting_formally_legal?
    - has_resolution_majority?
      - do_all_shareholders_consent_to_the_determination?
      - is_resolution_passed_via_voting?
  - is_deed_legal?
  - is_assurance_legal?
  - are_applicants_authorized?
    - are_applicants_authorized_via_general_sole_representation_power?
    - are_applicants_authorized_via_individual_sole_representation_power?
    - are_applicants_authorized_via_general_modified_representation_power?
    - are_applicants_authorized_via_individual_modified_representation_power?
    - are_all_directors_present?

$$\begin{aligned}&\exists_{app} A.\ (\\ &\qquad is\_application\_legal(A)\\ &\ \wedge\quad A = app1\\ &)\end{aligned}$$

$$\begin{aligned}&\exists_{meet} M.\ \exists_{res} R.\ \exists_{app} A.\ (\\ &\qquad is\_meeting\_legal(M)\\ &\ \wedge\quad resolution\_meeting(R, M)\\ &\ \wedge\quad application\_resolution(A, R)\\ &\ \wedge\quad A = app1\\ &)\end{aligned}$$

$$\begin{aligned}&\exists_{ass} S.\ \exists_{app} A.\ (\\ &\qquad is\_assurance\_legal(S)\\ &\ \wedge\quad application\_assurance(A, S)\\ &\ \wedge\quad A = app1\\ &)\end{aligned}$$

Proof Tree Generation



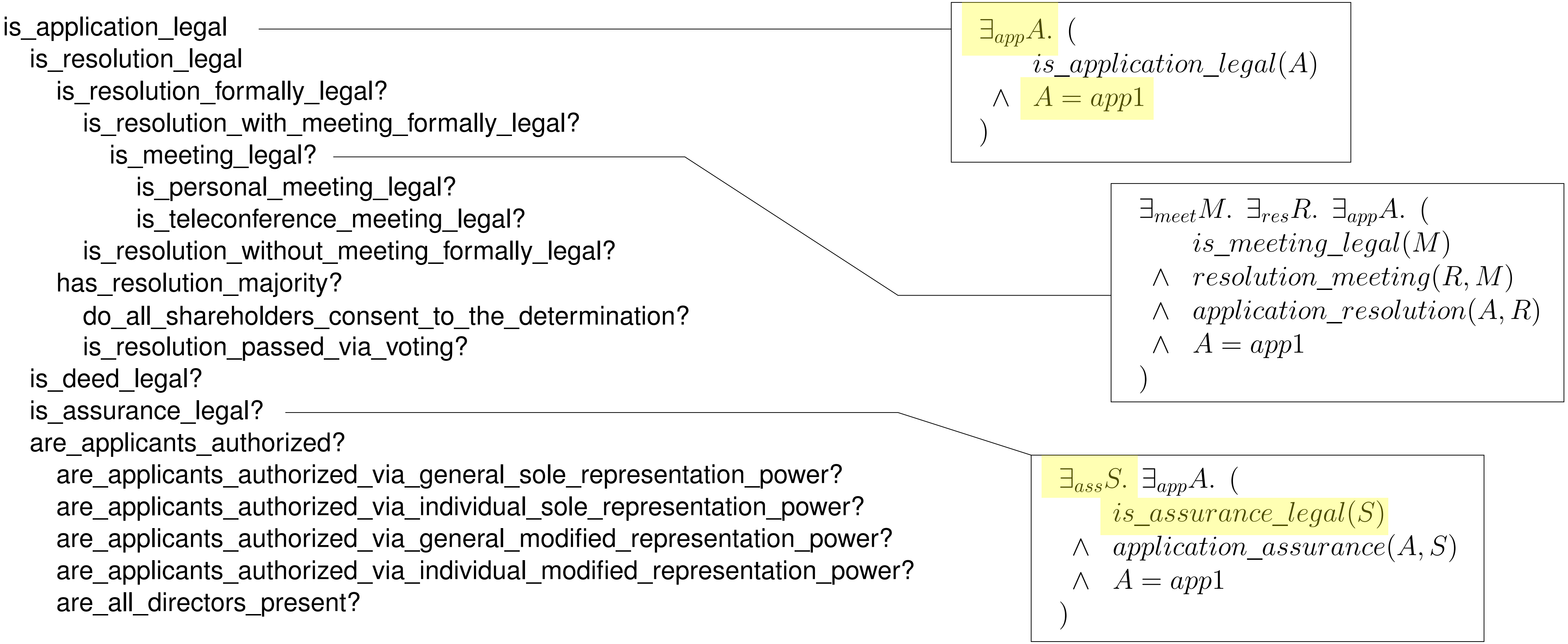
- is_application_legal
  - is_resolution_legal
    - is_resolution_formally_legal?
      - is_resolution_with_meeting_formally_legal?
        - is_meeting_legal?
          - is_personal_meeting_legal?
          - is_teleconference_meeting_legal?
      - is_resolution_without_meeting_formally_legal?
    - has_resolution_majority?
      - do_all_shareholders_consent_to_the_determination?
      - is_resolution_passed_via_voting?
  - is_deed_legal?
  - is_assurance_legal?
  - are_applicants_authorized?
    - are_applicants_authorized_via_general_sole_representation_power?
    - are_applicants_authorized_via_individual_sole_representation_power?
    - are_applicants_authorized_via_general_modified_representation_power?
    - are_applicants_authorized_via_individual_modified_representation_power?
    - are_all_directors_present?

$$\begin{aligned}&\exists_{app} A.\ (\\ &\quad is\_application\_legal(A)\\ &\land\quad A = app1\\ &)\end{aligned}$$

$$\begin{aligned}&\exists_{meet} M.\ \exists_{res} R.\ \exists_{app} A.\ (\\ &\quad is\_meeting\_legal(M)\\ &\land\quad resolution\_meeting(R, M)\\ &\land\quad application\_resolution(A, R)\\ &\land\quad A = app1\\ &)\end{aligned}$$

$$\begin{aligned}&\exists_{ass} S.\ \exists_{app} A.\ (\\ &\quad is\_assurance\_legal(S)\\ &\land\quad application\_assurance(A, S)\\ &\land\quad A = app1\\ &)\end{aligned}$$

Proof Tree Generation



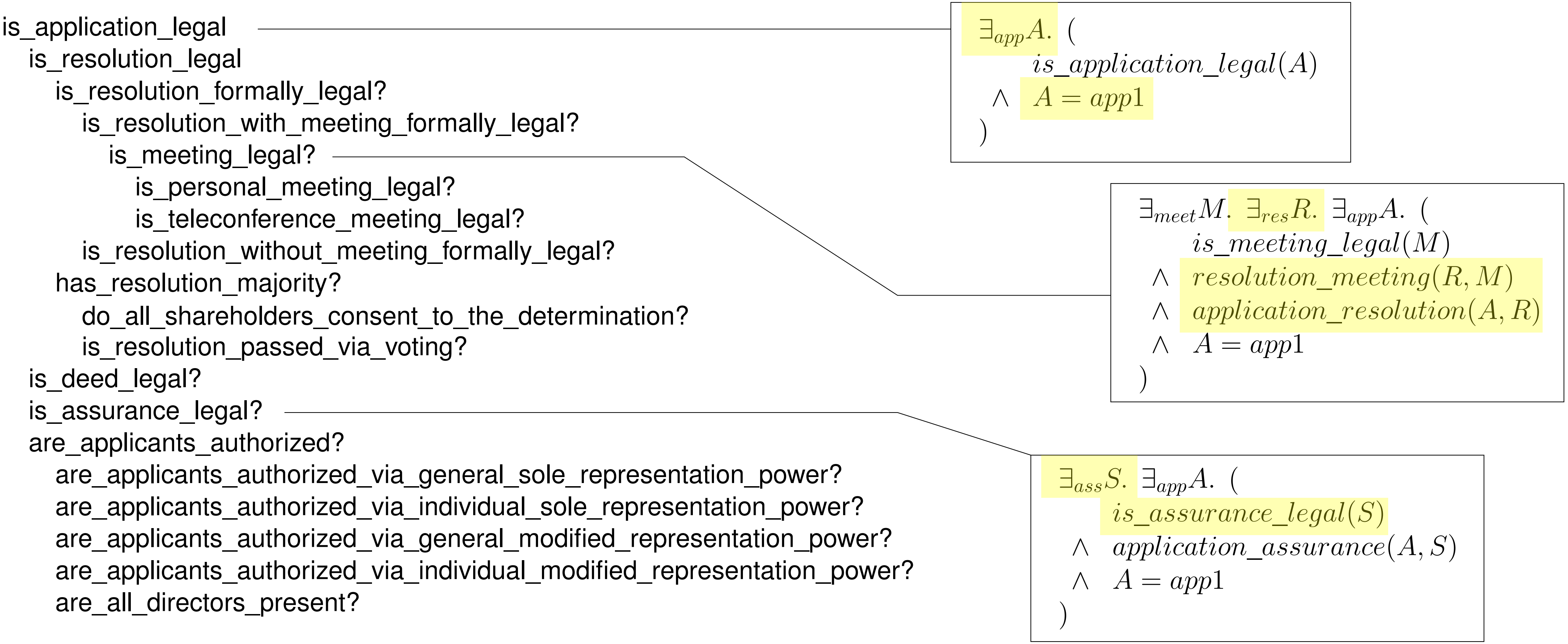
- is_application_legal
  - is_resolution_legal
    - is_resolution_formally_legal?
      - is_resolution_with_meeting_formally_legal?
        - is_meeting_legal?
          - is_personal_meeting_legal?
          - is_teleconference_meeting_legal?
      - is_resolution_without_meeting_formally_legal?
    - has_resolution_majority?
      - do_all_shareholders_consent_to_the_determination?
      - is_resolution_passed_via_voting?
  - is_deed_legal?
  - is_assurance_legal?
  - are_applicants_authorized?
    - are_applicants_authorized_via_general_sole_representation_power?
    - are_applicants_authorized_via_individual_sole_representation_power?
    - are_applicants_authorized_via_general_modified_representation_power?
    - are_applicants_authorized_via_individual_modified_representation_power?
    - are_all_directors_present?

$$\begin{aligned}&\exists_{app} A.\ (\\ &\qquad is\_application\_legal(A)\\ &\ \wedge\ \ A = app1\\ &)\end{aligned}$$

$$\begin{aligned}&\exists_{meet} M.\ \exists_{res} R.\ \exists_{app} A.\ (\\ &\qquad is\_meeting\_legal(M)\\ &\ \wedge\ \ resolution\_meeting(R, M)\\ &\ \wedge\ \ application\_resolution(A, R)\\ &\ \wedge\ \ A = app1\\ &)\end{aligned}$$

$$\begin{aligned}&\exists_{ass} S.\ \exists_{app} A.\ (\\ &\qquad is\_assurance\_legal(S)\\ &\ \wedge\ \ application\_assurance(A, S)\\ &\ \wedge\ \ A = app1\\ &)\end{aligned}$$

Proof Tree Generation



- is_application_legal
  - is_resolution_legal
    - is_resolution_formally_legal?
      - is_resolution_with_meeting_formally_legal?
        - is_meeting_legal?
          - is_personal_meeting_legal?
          - is_teleconference_meeting_legal?
      - is_resolution_without_meeting_formally_legal?
    - has_resolution_majority?
      - do_all_shareholders_consent_to_the_determination?
      - is_resolution_passed_via_voting?
  - is_deed_legal?
  - is_assurance_legal?
  - are_applicants_authorized?
    - are_applicants_authorized_via_general_sole_representation_power?
    - are_applicants_authorized_via_individual_sole_representation_power?
    - are_applicants_authorized_via_general_modified_representation_power?
    - are_applicants_authorized_via_individual_modified_representation_power?
    - are_all_directors_present?

$$\exists_{app} A.\ (\ \mathit{is\_application\_legal}(A) \land A = \mathit{app}1\ )$$

$$\exists_{meet} M.\ \exists_{res} R.\ \exists_{app} A.\ (\ \mathit{is\_meeting\_legal}(M) \land \mathit{resolution\_meeting}(R, M) \land \mathit{application\_resolution}(A, R) \land A = \mathit{app}1\ )$$

$$\exists_{ass} S.\ \exists_{app} A.\ (\ \mathit{is\_assurance\_legal}(S) \land \mathit{application\_assurance}(A, S) \land A = \mathit{app}1\ )$$

Visualization

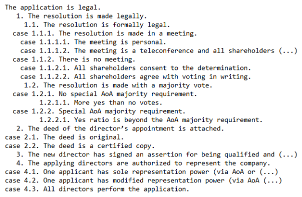
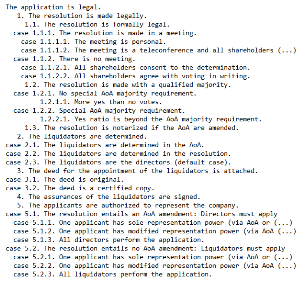
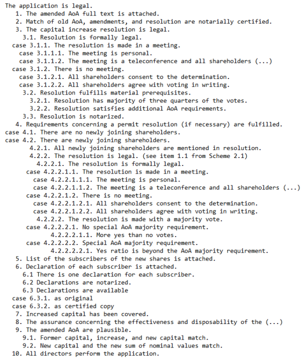
- ✗ is_application_legal?
  - ✗ is_resolution_legal?
    - ✓ is_resolution_formally_legal?
      - ✓ is_resolution_with_meeting_formally_legal?
        - ✓ is_meeting_legal?
          - ✗ is_personal_meeting_legal?
          - ✓ is_teleconference_meeting_legal?
      - ✗ is_resolution_without_meeting_formally_legal?
    - ✗ has_resolution_majority?
      - ✗ do_all_shareholders_consent_to_the_determination?
      - ✗ is_resolution_passed_via_voting?
  - ✓ is_deed_legal?
  - ✓ is_assurance_legal?
  - ✓ are_applicants_authorized?
    - ✗ are_applicants_authorized_via_general_sole_representation_power?
    - ✗ are_applicants_authorized_via_individual_sole_representation_power?
    - ✗ are_applicants_authorized_via_general_modified_representation_power?
    - ✓ are_applicants_authorized_via_individual_modified_representation_power?
    - ✗ are_all_directors_present?

Evaluation

- **Traceability of Reasoning**
- **Redundant Proofs**
- **Reusability for Other Processes**

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. **Adaption to Other Processes**
10. Case Data Retrieval
11. Conclusion

Process Comparison

	Director Appointment	Company Dissolution	Capital Increase
Scheme appearance	 The application is legal. 1. The resolution is made legally. 1.1. The resolution is formally legal. case 1.1.1. The resolution is made in a meeting. case 1.1.1.1. The meeting is personal. case 1.1.1.2. The meeting is a teleconference and all shareholders (...) case 1.1.2. There is no meeting. case 1.1.2.1. All shareholders consent to the determination. case 1.1.2.2. All shareholders agree with voting in writing. 1.2. The resolution is made with a majority vote. case 1.2.1. No special AoA majority requirement. 1.2.1.1. More yes than no votes. case 1.2.2. Special AoA majority requirement. 1.2.2.1. Yes ratio is beyond the AoA majority requirement. 2. The deed of the director's appointment is attached. case 2.1. The deed is original. case 2.2. The deed is a certified copy. 3. The new director has signed an assertion for being qualified and (...) 4. The applying directors are authorized to represent the company. case 4.1. One applicant has sole representation power (via AoA or (...) case 4.2. One applicant has modified representation power (via AoA (...) case 4.3. All directors perform the application.	 The application is legal. 1. The resolution is made legally. 1.1. The resolution is formally legal. case 1.1.1. The resolution is made in a meeting. case 1.1.1.1. The meeting is personal. case 1.1.1.2. The meeting is a teleconference and all shareholders (...) case 1.1.2. There is no meeting. case 1.1.2.1. All shareholders consent to the determination. case 1.1.2.2. All shareholders agree with voting in writing. 1.2. The resolution is made with a qualified majority. case 1.2.1. No special AoA majority requirement. 1.2.1.1. More yes than no votes. case 1.2.2. Special AoA majority requirement. 1.2.2.1. Yes ratio is beyond the AoA majority requirement. 1.3. The resolution is notarized if the AoA are amended. 2. The liquidators are determined. case 2.1. The liquidators are determined in the AoA. case 2.2. The liquidators are determined in the resolution. case 2.3. The liquidators are the directors (default case). 3. The deed for the appointment of the liquidators is attached. case 3.1. The deed is original. case 3.2. The deed is a certified copy. 4. The assurances of the liquidators are signed. 5. The applicants are authorized to represent the company. case 5.1. The resolution entails an AoA amendment: Directors must apply case 5.1.1. One applicant has sole representation power (via AoA or (...) case 5.1.2. One applicant has modified representation power (via AoA (...) case 5.1.3. All directors perform the application. case 5.2. The resolution entails no AoA amendment: Liquidators must apply case 5.2.1. One applicant has sole representation power (via AoA or (...) case 5.2.2. One applicant has modified representation power (via AoA (...) case 5.2.3. All liquidators perform the application.	 The application is legal. 1. The amended AoA full text is attached. 2. Match of old AoA, amendments, and resolution are notarially certified. 3. The capital increase resolution is legal. 3.1. Resolution is formally legal. case 3.1.1. The resolution is made in a meeting. case 3.1.1.1. The meeting is personal. case 3.1.1.2. The meeting is a teleconference and all shareholders (...) case 3.1.2. There is no meeting. case 3.1.2.1. All shareholders consent to the determination. case 3.1.2.2. All shareholders agree with voting in writing. 3.2. Resolution fulfills material prerequisites. 3.2.1. Resolution has majority of three quarters of the votes. 3.2.2. Resolution satisfies additional AoA requirements. 3.3. Resolution is notarized. 4. Requirements concerning a permit resolution (if necessary) are fulfilled. case 4.1. There are no newly joining shareholders. case 4.2. There are newly joining shareholders. 4.2.1. All newly joining shareholders are mentioned in resolution. 4.2.2. The resolution is legal. (see item 1.1 from Scheme 2.1) 4.2.2.1. The resolution is formally legal. case 4.2.2.1.1. The resolution is made in a meeting. case 4.2.2.1.1.1. The meeting is personal. case 4.2.2.1.1.2. The meeting is a teleconference and all shareholders (...) case 4.2.2.1.2. There is no meeting. case 4.2.2.1.2.1. All shareholders consent to the determination. case 4.2.2.1.2.2. All shareholders agree with voting in writing. 4.2.2.2. The resolution is made with a majority vote. case 4.2.2.2.1. No special AoA majority requirement. 4.2.2.2.1.1. More yes than no votes. case 4.2.2.2.2. Special AoA majority requirement. 4.2.2.2.2.1. Yes ratio is beyond the AoA majority requirement. 5. List of the subscribers of the new shares is attached. 6. Declaration of each subscriber is attached. 6.1 There is one declaration for each subscriber. 6.2 Declarations are notarized. 6.3 Declarations are available case 6.3.1. as original case 6.3.2. as certified copy 7. Increased capital has been covered. 8. The assurance concerning the effectiveness and disposability of the (...) 9. The amended AoA are plausible. 9.1. Former capital, increase, and new capital match. 9.2. New capital and the new sum of nominal values match. 10. All directors perform the application.
Rule axioms	24	28	38
Characteristics		▶ Liquidators which can be directors or explicitly stated ▶ Directors or liquidators can be responsible applicants	▶ Complex dependencies between individuals ▶ Many "all X must Y" rules ▶ Plausibility checks that sum up and compare values
Limitations		iProver & Vampire cannot handle non-valid conjectures	First-order logic cannot sum over a predicate's extension

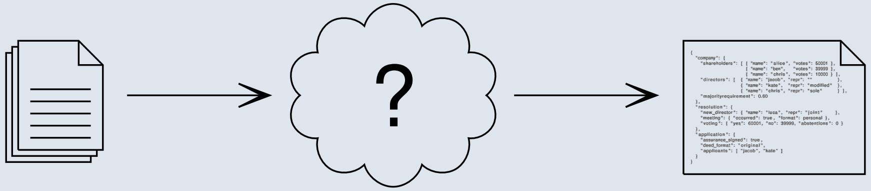
Evaluation

- Adaptability of Developments
- Generality of Conclusions
- Restricted Expressiveness of First-Order Logic
- Semi-Decidability
- Necessity of Ontologies
- Great Support by Prolog

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. **Case Data Retrieval**
11. Conclusion

Approaches to Obtain Structured Case Data

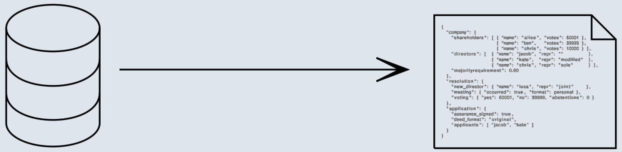
Text-Based Data Extraction



```
{
  "company": {
    "shareholders": [ { "name": "alice", "votes": 50001 },
                      { "name": "ben",   "votes": 39999 },
                      { "name": "chris", "votes": 10000 } ],
    "directors": [  { "name": "jacob", "repr": ""          },
                    { "name": "kate",  "repr": "modified"  },
                    { "name": "chris", "repr": "sole"      } ],
    "majorityrequirement": 0.60
  },
  "resolution": {
    "new_director": { "name": "luca", "repr": "joint"     },
    "meeting": { "occurred": true , "format": personal },
    "voting": { "yes": 60001, "no": 39999, "abstentions": 0 }
  },
  "application": {
    "assurance_signed": true ,
    "deed_format": "original",
    "applicants": [ "jacob", "kate" ]
  }
}
```

- ▶ Regular Expressions
- ▶ Probabilistic Methods (HMM, CRF)
- ▶ Grammar-Based Methods

Structured Data Resources



```
{
  "company": {
    "shareholders": [ { "name": "alice", "votes": 50001 },
                      { "name": "ben",   "votes": 39999 },
                      { "name": "chris", "votes": 10000 } ],
    "directors": [  { "name": "jacob", "repr": ""          },
                    { "name": "kate",  "repr": "modified"  },
                    { "name": "chris", "repr": "sole"      } ],
    "majorityrequirement": 0.60
  },
  "resolution": {
    "new_director": { "name": "luca", "repr": "joint"     },
    "meeting": { "occurred": true , "format": personal },
    "voting": { "yes": 60001, "no": 39999, "abstentions": 0 }
  },
  "application": {
    "assurance_signed": true ,
    "deed_format": "original",
    "applicants": [ "jacob", "kate" ]
  }
}
```

- ▶ Notary’s Internal Database
- ▶ Commercial Register

1. Introduction
2. Use Case: Director Application
3. Technical Preliminaries
4. Formalization in First-Order Logic
5. Implementation in TPTP
6. Implementation in Prolog
7. Ontology Incorporation
8. Reasoning Visualization
9. Adaption to Other Processes
10. Case Data Retrieval
11. **Conclusion**

Sufficiency of Logic

Pros

- Successful Formalization
- Successful Implementation

Cons

- Automated Reasoning Issues
- Expressiveness-Decidability-Tradeoff

Sufficiency of Logic

Pros

- Successful Formalization
- Successful Implementation

Cons

- Automated Reasoning Issues
- Expressiveness-Decidability-Tradeoff

Conclusion

- Works out on the whole
- Great majority of cases can be formalized

Necessity of Logic

Pros

- Ontology Incorporation
- Reasoning Traceability

Cons

- Formalization Effort

Necessity of Logic

Pros

- Ontology Incorporation
- Reasoning Traceability

Cons

- Formalization Effort

Conclusion

- No conclusive answer
- Benefits of logic as a benchmark for further research

Suitability of First-Order Logic

Pros

- Successful Formalization
- Successful Implementation
- Ontology Incorporation
- Reasoning Traceability

Cons

- Monotonicity
- Semi-Decidability
- Expressiveness

Suitability of First-Order Logic

Pros

- Successful Formalization
- Successful Implementation
- Ontology Incorporation
- Reasoning Traceability

Cons

- Monotonicity
- Semi-Decidability
- Expressiveness

Conclusion

- First-order logic is not the best option
- Non-monotonic logic might be better

Future Work

- Non-Monotonic Logics
- Exploring Alternatives to Logic
- Extensions for Reasoning Visualization
- Case Data Retrieval
- Automated Formalization
- Examination of TPTP Prover Limitations
- Ontology Incorporation for Assumptions