Consistency in Justification Theory

NMR 2018

Simon Marynissen, Niko Passchyn, Bart Bogaerts and Marc Denecker

October 29, 2018
0 Outline

1 Overview of justification theory

2 Justification theory: an intuition

3 Consistency

4 Conclusion
1 Outline

1 Overview of justification theory

2 Justification theory: an intuition

3 Consistency

4 Conclusion
1 Aims of justification theory

- Unifying framework for describing various semantics of various logics
  - Logic programs
  - Abstract argumentation
  - Inductive definitions
  - Nested definitions
1 Aims of justification theory

- Unifying framework for describing various semantics of various logics
  - Logic programs
  - Abstract argumentation
  - Inductive definitions
  - Nested definitions

- By capturing the underlying types of constructions = justifications
1 Aims of justification theory

- Unifying framework for describing various semantics of various logics
  - Logic programs
  - Abstract argumentation
  - Inductive definitions
  - Nested definitions

- By capturing the underlying types of constructions = justifications

- Give rise to new semantics
1 Aims of justification theory

- Unifying framework for describing various semantics of various logics
  - Logic programs
  - Abstract argumentation
  - Inductive definitions
  - Nested definitions

- By capturing the underlying types of constructions = justifications

- Give rise to new semantics

- Provides ways for seamless integration of various expressive language constructs
  - Aggregates into logic programs
1 Aims of justification theory: computational aspects

Justifications as datastructures in solvers

- Compute unfounded sets in ASP solvers (De Cat, Gebser)
- Check for relevance in complete search algorithms (Jansen)
- Lazy grounding (De Cat, Bogaerts)
2 Outline

1 Overview of justification theory

2 Justification theory: an intuition

3 Consistency

4 Conclusion

5 Consistency in Justification Theory
2 What is justification theory (intuitively)

- Theory (or program) in a logic induces a semantic structure: justification frame: a space of facts, a set of elementary construction steps
What is justification theory (intuitively)

- Theory (or program) in a logic induces a semantic structure: *justification frame*: a space of facts, a set of elementary construction steps

- *justification* := graph of facts constructed by elementary construction steps
  - that shows a type of construction of its facts
  - that embodies a potential reason why its facts are true
2 What is justification theory (intuitively)

- Theory (or program) in a logic induces a semantic structure: *justification frame*: a space of facts, a set of elementary construction steps

- *justification* := graph of facts constructed by elementary construction steps
  - that shows a type of construction of its facts
  - that embodies a potential reason why its facts are true

- A justification is a *good* construction if all its branches are *good*
  - Branch evaluation $B(x_0 \rightarrow x_1 \rightarrow \ldots)$ is a fact, true or false

- Different notions of branch evaluations
  - different sorts of construction
  - different types of semantics
2 Justification status of a fact in an interpretation $\mathcal{I}$

- Interpretations, possibly 4-valued $t$ (true), $f$ (false), $u$ (unknown) and $i$ (inconsistent)
  - Truth order $f \leq_t u \leq_t t$, $f \leq_t i \leq_t t$

- The support value of $J$ for a fact $x$ in $\mathcal{I}$ is the value of the worst branch $x \rightarrow x_1 \rightarrow \ldots$ in $J$ under the branch evaluation $B$.

**Definition**

The *supported value of a fact $x$ in $\mathcal{I}$* is the support value of the “best” justification for $x$.

**Notation:** $SV(\mathcal{I}, x)$
2 A justification frame: example

Example

In a graph \((V, E)\) with nodes \(V\) and edges \(E\), define the nodes reachable from node \(a \in V\):
2 A justification frame: example

Example

In a graph \((V, E)\) with nodes \(V\) and edges \(E\), define the nodes reachable from node \(a \in V\):

- Facts of the justification frame:
  - Facts \(\text{Edge}(v, w)\) and \(\sim\text{Edge}(v, w)\) for \(v, w \in V\)
  - Facts \(\text{Reach}(v)\) and \(\sim\text{Reach}(v)\) for \(v \in V\)
2 A justification frame: example

Example

In a graph \((V, E)\) with nodes \(V\) and edges \(E\), define the nodes reachable from node \(a \in V\):

▶ Facts of the justification frame:

- Facts \(\text{Edge}(v, w)\) and \(\neg\text{Edge}(v, w)\) for \(v, w \in V\)
- Facts \(\text{Reach}(v)\) and \(\neg\text{Reach}(v)\) for \(v \in V\)

▶ Rules of the justification frame:

- \(\text{Reach}(a) \leftarrow t\)
- \(\text{Reach}(v) \leftarrow \text{Reach}(x), \text{Edge}(x, v)\) for \(v, x \in V\)
- \(\neg\text{Reach}(v) \leftarrow \{\neg\text{Reach}(x) \text{ or } \neg\text{Edge}(x, v) \mid x \in V\}\) for all \(v \in V\)
2 A justification frame: example

Example

In a graph \((V, E)\) with nodes \(V\) and edges \(E\), define the nodes reachable from node \(a \in V\):

▶ Facts of the justification frame:
  • Facts \(\text{Edge}(v, w)\) and \(\neg\text{Edge}(v, w)\) for \(v, w \in V\)
  • Facts \(\text{Reach}(v)\) and \(\neg\text{Reach}(v)\) for \(v \in V\)

▶ Rules of the justification frame:
  • \(\text{Reach}(a) \leftarrow t\)
  • \(\text{Reach}(v) \leftarrow \text{Reach}(x), \text{Edge}(x, v)\) for \(v, x \in V\)
  • \(\neg\text{Reach}(v) \leftarrow \{\neg\text{Reach}(x) \text{ or } \neg\text{Edge}(x, v) | x \in V\}\) for all \(v \in V\)

▶ Elements \(\text{Edge}(v, w)\) correspond to parameters
2 A justification frame: example

Example

In a graph \((V, E)\) with nodes \(V\) and edges \(E\), define the nodes reachable from node \(a \in V\):

▶ Facts of the justification frame:
- Facts \(\text{Edge}(v, w)\) and \(\sim\text{Edge}(v, w)\) for \(v, w \in V\)
- Facts \(\text{Reach}(v)\) and \(\sim\text{Reach}(v)\) for \(v \in V\)

▶ Rules of the justification frame:
- \(\text{Reach}(a) \leftarrow t\)
- \(\text{Reach}(v) \leftarrow \text{Reach}(x), \text{Edge}(x, v)\) for \(v, x \in V\)
- \(\sim\text{Reach}(v) \leftarrow \{\sim\text{Reach}(x) \text{ or } \sim\text{Edge}(x, v) \mid x \in V\}\) for all \(v \in V\)

▶ Elements \(\text{Edge}(v, w)\) correspond to parameters

▶ A specific graph corresponds to an interpretation \(\mathcal{I}\)
Let the graph \((V, E)\) be:

![Graph Diagram](image)

- \(\text{Reach}(a) \leftarrow t\)
- \(\text{Reach}(v) \leftarrow \text{Reach}(x), \text{Edge}(x, v)\) for \(v, x \in V\)
- \(\neg\text{Reach}(v) \leftarrow \{\neg\text{Reach}(x) \text{ or } \neg\text{Edge}(x, v) \mid x \in V\}\) for all \(v \in V\)

Part of a justification:
Under the suitable branch evaluation $B$:

- finite branches evaluate to their leaf
- infinite branches of positive facts: evaluate to $f$
  \[ \text{Reach}(d) \rightarrow \text{Reach}(d) \rightarrow \cdots \] is mapped to $f$
- infinite branches of negative facts evaluate to $t$
  \[ \sim\text{Reach}(d) \rightarrow \sim\text{Reach}(d) \rightarrow \cdots \] is mapped to $t$

$SV(\mathcal{I}, \text{Reach}(c)) = t$, $SV(\mathcal{I}, \sim\text{Reach}(d)) = t$
2 Another example

Let \((A, R)\) be an abstract argumentation frame.
2 Another example

- Let \((A, R)\) be an abstract argumentation frame
- \(a\) attacks \(b\) \((aRb)\) can be modelled with the rule
  
  \[ \sim b \leftarrow a \]
2 Another example

- Let \((A, R)\) be an abstract argumentation frame

- \(a\) attacks \(b\) \((aRb)\) can be modelled with the rule

\[
\sim b \leftarrow a
\]

- Denecker et al. (2015) give characterisations of admissible, stable, preferred, complete and grounded sets in terms fixed points of an operator associated with the supported value
2 Justification theory for logic programs

- Various semantics
  - Clarks completion
  - Kripke-Kleene
  - Stable (answer set)
  - Well-founded
2 Justification theory for logic programs

- Various semantics
  - Clarks completion
  - Kripke-Kleene
  - Stable (answer set)
  - Well-founded

- All four can be captured in justification theory with various branch evaluations
3 Outline

1 Overview of justification theory

2 Justification theory: an intuition

3 Consistency

4 Conclusion
3 Conflicting justification status

- Both $x$ and $\sim x$ have a supported value in $\mathcal{I}$
3 Conflicting justification status

- Both $x$ and $\sim x$ have a supported value in $\mathcal{I}$

- These values should not conflict, otherwise the semantics is defect
3 Conflicting justification status

▶ Both $x$ and $\sim x$ have a supported value in $\mathcal{I}$

▶ These values should not conflict, otherwise the semantics is defect
  • We want that $SV(\mathcal{I}, \sim x) = \sim SV(\mathcal{I}, x)$
  • $(\neg f = t, \neg t = f, \sim u = u$ and $\sim i = i)$
3 Conflicting justification status

- Both $x$ and $\sim x$ have a supported value in $\mathcal{I}$

- These values should not conflict, otherwise the semantics is defect
  
  - We want that $SV(\mathcal{I}, \sim x) = \sim SV(\mathcal{I}, x)$
  - $(\sim f = t, \sim t = f, \sim u = u$ and $\sim i = i)$
  - If $SV(\mathcal{I}, \sim x) = \sim SV(\mathcal{I}, x)$ for all $x$, then $SV(\mathcal{I}, \cdot)$ is also an interpretation
3 Resolving conflicting status

Theorem

*If the rules for \( x \) and \( \sim x \) are “complementary”, then*

\[
SV(\mathcal{I}, x) \leq_t \sim SV(\mathcal{I}, \sim x)
\]
3 Resolving conflicting status

Theorem

If the rules for $x$ and $\sim x$ are “complementary”, then

$$SV(I, x) \leq_t \sim SV(I, \sim x)$$

Example of complementary

Abstract Argumentation Frame: $A = \{a, b, c\}$ with
$R = \{(a, b), (c, b), (c, a), (a, c)\}$
3 Resolving conflicting status

Theorem

*If the rules for $x$ and $\sim x$ are “complementary”, then*

$$SV(I, x) \leq t \sim SV(I, \sim x)$$

Example of complementary

Abstract Argumentation Frame: $A = \{a, b, c\}$ with $R = \{(a, b), (c, b), (c, a), (a, c)\}$

- Rules $\{\sim b \leftarrow a, \sim b \leftarrow c, \sim a \leftarrow c, \sim c \leftarrow a\}$
3 Resolving conflicting status

Theorem

*If the rules for* $x$ *and* $\sim x$ *are “complementary”, then*

$$SV(\mathcal{I}, x) \leq_t \sim SV(\mathcal{I}, \sim x)$$

**Example of complementary**

Abstract Argumentation Frame: $A = \{a, b, c\}$ with

$R = \{(a, b), (c, b), (c, a), (a, c)\}$

- Rules $\{\sim b \leftarrow a, \sim b \leftarrow c, \sim a \leftarrow c, \sim c \leftarrow a\}$

- But also rules $\{b \leftarrow \sim a, \sim c \quad a \leftarrow \sim c \quad c \leftarrow \sim a\}$
3 Resolving conflicting status

▶ What about other direction?
3 Resolving conflicting status

▶ What about other direction?

▶ Does not always hold
3 Resolving conflicting status

- What about other direction?
- Does not always hold

Theorem

For the branch evaluations capturing Clarks completion, Kripke-Kleene, stable and well-founded semantics and “complementary” rules we have

$$SV(\mathcal{I}, x) = \neg SV(\mathcal{I}, \neg x)$$

- Proof uses heavy machinery and clever “pasting” of justifications
4 Outline

1 Overview of justification theory
2 Justification theory: an intuition
3 Consistency
4 Conclusion
4 Future work

- We proved this consistency result only for particular branch evaluations

- Can we find a general property of branch evaluations so that \[ \text{SV}(I, x) = \sim \text{SV}(I, \sim x) \]?

- What other formalisms can be expressed in justification theory?
4 Future work

- We proved this consistency result only for particular branch evaluations.

- Can we find a general property of branch evaluations so that $SV(\mathcal{I}, x) = \sim SV(\mathcal{I}, \sim x)$?
4 Future work

- We proved this consistency result only for particular branch evaluations.

- Can we find a general property of branch evaluations so that $SV(I, x) = \sim SV(I, \sim x)$?

- What other formalisms can be expressed in justification theory?
4 Conclusions

▶ Justification as a description of a construction

▶ A flexible theory
4 Conclusions

- Justification as a description of a construction

- A flexible theory
  - Captures various semantics of formalisms
    - Logic programming semantics
    - Abstract argumentation framework
    - ...
4 Conclusions

- Justification as a description of a construction

- A flexible theory
  - Captures various semantics of formalisms
    - Logic programming semantics
    - Abstract argumentation framework
    - ...
  - Used in computational tools
4 Conclusions

- Justification as a description of a construction
- A flexible theory
  - Captures various semantics of formalisms
    - Logic programming semantics
    - Abstract argumentation framework
    - ...
  - Used in computational tools
- Consistency result for particular semantics
  - Clarks completion
  - Kripke-Kleene
  - Stable
  - Well-founded