ASP($\mathcal{AC}$): Answer Set Programming with Algebraic Constraints

Thomas Eiter, Rafael Kiesel

Vienna University of Technology

funded by FWF project W1255-N23

22nd of September 2020

FWF
Der Wissenschaftsfonds.

logics LOGICAL METHODS IN COMPUTER SCIENCE

Manifold of ASP Extensions

- Nested Expressions $\alpha \leftarrow \beta$
- Weight Constraints $L \leq \{a_1 = w_1, \dots, \neg a_n = w_n\} \leq U$
- ...with Conditionals $L \leq \{a_1 : c_1 = w_1, \dots, \neg a_n : c_n = w_n\} \leq U$
- Aggregates $T \circ \#F\{X : p(X), q(X, Y)\}$
- Arithmetic Operators $X = Y + Z$
- Choice Constructs $l\{q(X) : p(X)\}u \leftarrow$
- Weak Constraints $:\sim F \;\; [Weight \; @ \; Level]$
- Probabilistic Rules $w : r$
- ...

Manifold of ASP Extensions

- Nested Expressions [Ferraris, 2011]
- Weight Constraints [Niemela *et al.*, 1999]
- ...with Conditionals [Niemela *et al.*, 1999]
- Aggregates [Faber *et al.*, 2011]
- Arithmetic Operators [Lierler, 2014]
- Choice Constructs [Niemela *et al.*, 1999]
- Weak Constraints [Buccafurri *et al.*, 2000]
- Probabilistic Rules [Lee and Yang, 2017]
- ...

Classes of ASP Extensions I

Atom Level
- Nested Expressions
- Weight Constraints
- ...with Conditionals
- Aggregates
- Arithmetic Operators
- Choice Constructs

Model Level
- Weak Constraints
- Probabilistic Rules

Classes of ASP Extensions II

- Model Level: Assign answer sets a weight based on the atoms in it.
  $\hookrightarrow$ Weighted LARS [Eiter and Kiesel, 2020]

Classes of ASP Extensions II

- Model Level: Assign answer sets a weight based on the atoms in it.
  $\hookrightarrow$ Weighted LARS [Eiter and Kiesel, 2020]

- Atom Level: Include atoms in answer sets based on constraints on quantities that depend on the interpretation.
  $\hookrightarrow$ This work

Classes of ASP Extensions II

- Model Level: Assign answer sets a weight based on the atoms in it.
  $\hookrightarrow$ Weighted LARS [Eiter and Kiesel, 2020]

- Atom Level: Include atoms in answer sets based on constraints on quantities that depend on the interpretation.
  $\hookrightarrow$ This work
  $\hookrightarrow$ In ASP the quantities have a non-monotonic dependency!

First-Order Here-and-There Logic

- Signature $\sigma = \langle \mathcal{D}, \mathcal{P}, \mathcal{X}, \mathcal{S}, r \rangle$

First-Order Here-and-There Logic

- Signature $\sigma = \langle \mathcal{D}, \mathcal{P}, \mathcal{X}, \mathcal{S}, r \rangle$
- Syntax $\phi ::= \bot \mid p(\vec{x}) \mid \phi \to \phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x \phi \mid \forall x \phi$

First-Order Here-and-There Logic

- Signature $\sigma = \langle \mathcal{D}, \mathcal{P}, \mathcal{X}, \mathcal{S}, r \rangle$
- Syntax $\phi ::= \bot \mid p(\vec{x}) \mid \phi \rightarrow \phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x \phi \mid \forall x \phi$
- pointed σ-HT-interpretation $\mathcal{I}_w = (\mathcal{I}^H, \mathcal{I}^T, w)$, $\mathcal{I}^H \subseteq \mathcal{I}^T$
- reflexive order $\geq$ on $\{H, T\}$, with $T \geq H$

First-Order Here-and-There Logic

- Signature $\sigma = \langle \mathcal{D}, \mathcal{P}, \mathcal{X}, \mathcal{S}, r \rangle$
- Syntax $\phi ::= \bot \mid p(\vec{x}) \mid \phi \rightarrow \phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x \phi \mid \forall x \phi$
- pointed σ-HT-interpretation $\mathcal{I}_w = (\mathcal{I}^H, \mathcal{I}^T, w)$, $\mathcal{I}^H \subseteq \mathcal{I}^T$
- reflexive order $\geq$ on $\{H, T\}$, with $T \geq H$
- Semantics

$$\mathcal{I}_w \models_\sigma \alpha \wedge \beta \iff \mathcal{I}_w \models_\sigma \alpha \text{ and } \mathcal{I}_w \models_\sigma \beta$$

$$\mathcal{I}_w \models_\sigma \phi \rightarrow \psi \iff \mathcal{I}_{w'} \not\models_\sigma \phi \text{ or } \mathcal{I}_{w'} \models_\sigma \psi \text{ for all } w' \geq w$$

$$\mathcal{I}_w \models_\sigma \exists x \phi(x) \iff \mathcal{I}_w \models_\sigma \phi(\xi), \text{ for some } \xi \in r(s(x))$$

First-Order Here-and-There Logic

- Signature $\sigma = \langle \mathcal{D}, \mathcal{P}, \mathcal{X}, \mathcal{S}, r \rangle$
- Syntax $\phi ::= \bot \mid p(\vec{x}) \mid \phi \rightarrow \phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x \phi \mid \forall x \phi$
- pointed σ-HT-interpretation $\mathcal{I}_w = (\mathcal{I}^H, \mathcal{I}^T, w)$, $\mathcal{I}^H \subseteq \mathcal{I}^T$
- reflexive order $\geq$ on $\{H, T\}$, with $T \geq H$
- Semantics

$$\mathcal{I}_w \models_\sigma \alpha \wedge \beta \iff \mathcal{I}_w \models_\sigma \alpha \text{ and } \mathcal{I}_w \models_\sigma \beta$$

$$\mathcal{I}_w \models_\sigma \phi \rightarrow \psi \iff \mathcal{I}_{w'} \not\models_\sigma \phi \text{ or } \mathcal{I}_{w'} \models_\sigma \psi \text{ for all } w' \geq w$$

$$\mathcal{I}_w \models_\sigma \exists x \phi(x) \iff \mathcal{I}_w \models_\sigma \phi(\xi), \text{ for some } \xi \in r(s(x))$$

- $\mathcal{I}$ is an equilibrium model ϕ if $(\mathcal{I}, \mathcal{I}, H) \models \phi$ and $\nexists \mathcal{I}' \subsetneq \mathcal{I} : (\mathcal{I}', \mathcal{I}, H) \models \phi$

Semirings

A semiring is an algebraic structure $(R, \oplus, \otimes, e_\oplus, e_\otimes)$, s.t.

- $(R, \oplus, e_\oplus)$ is a commutative monoid with neutral element $e_\oplus$
- $(R, \otimes, e_\otimes)$ is a monoid with neutral element $e_\otimes$
- multiplication ($\otimes$) distributes over addition ($\oplus$)
- multiplication by $e_\oplus$ annihilates R

Additionally $\odot \in \{\oplus, \otimes\}$ is invertible if $\forall r : \exists r^i : r \odot r^i = e_\odot$.

Semirings were successfully used to parameterize calculation in [Bistarelli *et al.*, 1997], [Green *et al.*, 2007] and other works.

Semiring Examples

Prominent examples are

- $\mathbb{Q} = (\mathbb{Q}, +, \cdot, 0, 1)$ rational numbers
- $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \mathsf{max}, +, -\infty, 0)$ max-plus
- $\mathcal{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \mathsf{min}, +, \infty, 0)$ min-plus
- $\mathbb{B} = (\{\bot, \top\}, \vee, \wedge, \bot, \top)$ boolean

Semiring Examples

Prominent examples are

- $\mathbb{Q} = (\mathbb{Q}, +, \cdot, 0, 1)$ rational numbers
  $\hookrightarrow$ arithmetics
- $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$ max-plus
- $\mathcal{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$ min-plus
- $\mathbb{B} = (\{\bot, \top\}, \vee, \wedge, \bot, \top)$ boolean

Semiring Examples

Prominent examples are

- $\mathbb{Q} = (\mathbb{Q}, +, \cdot, 0, 1)$ rational numbers
- $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$ max-plus
  $\hookrightarrow$ optimisation
- $\mathcal{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$ min-plus
- $\mathbb{B} = (\{\bot, \top\}, \vee, \wedge, \bot, \top)$ boolean

Semiring Examples

Prominent examples are

- $\mathbb{Q} = (\mathbb{Q}, +, \cdot, 0, 1)$ rational numbers
- $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \mathsf{max}, +, -\infty, 0)$ max-plus
- $\mathcal{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \mathsf{min}, +, \infty, 0)$ min-plus
  $\hookrightarrow$ optimisation
- $\mathbb{B} = (\{\bot, \top\}, \vee, \wedge, \bot, \top)$ boolean

Semiring Examples

Prominent examples are

- $\mathbb{Q} = (\mathbb{Q}, +, \cdot, 0, 1)$ rational numbers
- $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \mathsf{max}, +, -\infty, 0)$ max-plus
- $\mathcal{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \mathsf{min}, +, \infty, 0)$ min-plus
- $\mathbb{B} = (\{\bot, \top\}, \vee, \wedge, \bot, \top)$ boolean
  $\hookrightarrow$ boolean constraints

Weighted First-Order Here-and-There Logic I

- Coming from the unweighted version

$$\phi ::= \bot \mid p(\vec{x}) \mid \phi \rightarrow \phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x \phi \mid \forall x \phi$$

- Idea: "Disjunction is addition and conjunction is multiplication"

Weighted First-Order Here-and-There Logic I

- Coming from the unweighted version

$$\phi ::= \bot \mid p(\vec{x}) \mid \phi \rightarrow \phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x \phi \mid \forall x \phi$$

- Idea: "Disjunction is addition and conjunction is multiplication"
- Syntax over a signature σ and semiring $\mathcal{R}$

$$\alpha ::= k \mid x \mid \phi \mid \alpha \rightarrow \alpha \mid \alpha + \alpha \mid \alpha * \alpha \mid -\alpha \mid \alpha^{-1} \mid \Sigma x \alpha \mid \Pi x \alpha,$$

Weighted First-Order Here-and-There Logic II

Semantics w.r.t. a pointed σ-HT-interpretation $\mathcal{I}_w$

$$\llbracket k \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = k, \text{ for } k \in R$$

$$\llbracket \phi \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = \begin{cases} e_{\otimes}, & \text{if } \mathcal{I}_w \models_{\sigma} \phi, \\ e_{\oplus}, & \text{otherwise.} \end{cases}, \text{ for } \sigma\text{-formulas } \phi$$

$$\llbracket -\alpha \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = -\llbracket \alpha \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w)$$

$$\llbracket \alpha^{-\mathbf{1}} \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = \llbracket \alpha \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w)^{-1}$$

$$\llbracket \alpha + \beta \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = \llbracket \alpha \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) \oplus \llbracket \beta \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w)$$

$$\llbracket \alpha * \beta \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = \llbracket \alpha \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) \otimes \llbracket \beta \rrbracket^{\sigma}_{\mathcal{R}}(\mathcal{I}_w)$$

Weighted First-Order Here-and-There Logic II

Semantics w.r.t. a pointed σ-HT-interpretation $\mathcal{I}_w$

$$[\![\alpha \to \beta]\!]^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = \begin{cases} e_{\otimes}, & \text{if } [\![\alpha]\!]^{\sigma}_{\mathcal{R}}(\mathcal{I}_{w'}) = e_{\oplus} \text{ or } [\![\beta]\!]^{\sigma}_{\mathcal{R}}(\mathcal{I}_{w'}) \neq e_{\oplus} \\ & \text{for all } w' \geq w, \\ e_{\oplus}, & \textit{otherwise}. \end{cases}$$

$$[\![\Sigma x \alpha(x)]\!]^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = \bigoplus_{\xi \in \mathrm{supp}_{\oplus}(\alpha(x), \mathcal{I}_w)} [\![\alpha(\xi)]\!]^{\sigma}_{\mathcal{R}}(\mathcal{I}_w)^{1}$$

$$[\![\Pi x \alpha(x)]\!]^{\sigma}_{\mathcal{R}}(\mathcal{I}_w) = \bigotimes_{\xi \in \mathrm{supp}_{\otimes}(\alpha(x), \mathcal{I}_w)} [\![\alpha(\xi)]\!]^{\sigma}_{\mathcal{R}}(\mathcal{I}_w)^{1}$$

[1] When $\mathrm{supp}_{\odot}(\alpha(x), \mathcal{I}_w)$ is finite.

Example

- Maximization over a unary predicate p:

$$\mathrm{MAX}\{X : p(X)\} = [\![\Sigma X\ p(X) * X]\!]_{\mathcal{R}_{\max}}$$

where $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$.

Example

- Maximization over a unary predicate p:

$$\mathrm{MAX}\{X : p(X)\} = [\![\Sigma X \; p(X) * X]\!]_{\mathcal{R}_{\max}}$$

where $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$.

- The range of the sort of X must be $\mathbb{R} \cup \{-\infty\}$!

Example

- Maximization over a unary predicate p:

$$\mathrm{MAX}\{X : p(X)\} = [\![\Sigma X\ p(X) * X]\!]_{\mathcal{R}_{\max}}$$

where $\mathcal{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$.

- The range of the sort of X must be $\mathbb{R} \cup \{-\infty\}$!
- We get

$$\begin{aligned} & [\![\Sigma X\ p(X) * X]\!]_{\mathcal{R}_{\max}}(\mathcal{I}_w) \\ = & \max_{\sigma \in \mathsf{supp}_{-\infty}(p(X)*X, \mathcal{I}_w)} [\![p(\sigma)]\!]_{\mathcal{R}_{\max}}(\mathcal{I}_w) + \sigma \\ = & \max_{\sigma \in \mathbb{R},\ \text{s.t. } p(\sigma) \in \mathcal{I}^w} \sigma \end{aligned}$$

Algebraic Constraints

- Algebraic Constraints, $k \sim_{\mathcal{R}} \alpha$ or $x \sim_{\mathcal{R}} \alpha$ where
  - $\mathcal{R}$ a semiring
  - α a weighted formula
  - $k \in R$ and x is a variable
  - $\sim \in \{>, \geq, =, \leq, <, \not>, \not\geq, \neq, \not\leq, \not<\}$

Algebraic Constraints

- Algebraic Constraints, $k \sim_{\mathcal{R}} \alpha$ or $x \sim_{\mathcal{R}} \alpha$ where
  - $\mathcal{R}$ a semiring
  - α a weighted formula
  - $k \in R$ and x is a variable
  - $\sim \in \{>, \geq, =, \leq, <, \not>, \not\geq, \neq, \not\leq, \not<\}$
- Satisfaction of $k \sim_{\mathcal{R}} \alpha$ w.r.t. $\mathcal{I}_w$:

$$\mathcal{I}_w \models k \sim_{\mathcal{R}} \alpha \iff k \sim [\![\alpha]\!]_{\mathcal{R}}(\mathcal{I}_{w'}) \text{ for } w' \geq w$$

Algebraic Constraints

- Algebraic Constraints, $k \sim_{\mathcal{R}} \alpha$ or $x \sim_{\mathcal{R}} \alpha$ where
  - $\mathcal{R}$ a semiring
  - α a weighted formula
  - $k \in R$ and x is a variable
  - $\sim \in \{>, \geq, =, \leq, <, \not>, \not\geq, \neq, \not\leq, \not<\}$
- Satisfaction of $k \sim_{\mathcal{R}} \alpha$ w.r.t. $\mathcal{I}_w$:

$$\mathcal{I}_w \models k \sim_{\mathcal{R}} \alpha \iff k \sim [\![\alpha]\!]_{\mathcal{R}}(\mathcal{I}_{w'}) \text{ for } w' \geq w$$

- Allow algebraic constraints in heads and bodies of $\mathcal{AC}$-rules

Example

- Weighted sums with a global or local weight using $\mathcal{AC}$-rules:

$$\mathrm{l\_sum}(Y) \leftarrow Y =_{\mathbb{Q}} \mathrm{ind}(I) * \mathrm{l\_weight}(I, W) * W$$
$$\mathrm{g\_sum}(Y) \leftarrow \mathrm{g\_weight}(W), Y =_{\mathbb{Q}} \mathrm{ind}(I) * W$$

Example

- Weighted sums with a global or local weight using $\mathcal{AC}$-rules:

$$\mathrm{l\_sum}(Y) \leftarrow Y =_{\mathbb{Q}} \mathrm{ind}(I) * \mathrm{l\_weight}(I, W) * W$$
$$\mathrm{g\_sum}(Y) \leftarrow \mathrm{g\_weight}(W), Y =_{\mathbb{Q}} \mathrm{ind}(I) * W$$

Example

- Weighted sums with a global or local weight using $\mathcal{AC}$-rules:

$$\mathrm{l\_sum}(Y) \leftarrow Y =_{\mathbb{Q}} \mathrm{ind}(I) * \mathrm{l\_weight}(I, W) * W$$
$$\mathrm{g\_sum}(Y) \leftarrow \mathrm{g\_weight}(W), Y =_{\mathbb{Q}} \mathrm{ind}(I) * W$$

- local variables are Σ-quantified
  global variable are $\forall$-quantified

Example

- Weighted sums with a global or local weight using $\mathcal{AC}$-rules:

$$\text{l\_sum}(Y) \leftarrow Y =_{\mathbb{Q}} \text{ind}(I) * \text{l\_weight}(I, W) * W$$
$$\text{g\_sum}(Y) \leftarrow \text{g\_weight}(W), Y =_{\mathbb{Q}} \text{ind}(I) * W$$

- local variables are Σ-quantified
  global variable are $\forall$-quantified
- Semantics given by the FO-HT-sentences:

$$\forall Y \; (Y =_{\mathbb{Q}} \Sigma I \Sigma W \; \text{ind}(I) * \text{l\_weight}(I, W) * W) \rightarrow \text{l\_sum}(Y)$$
$$\forall Y \forall W \; \text{g\_weight}(W) \wedge (Y =_{\mathbb{Q}} \Sigma I \; \text{ind}(I) * W) \rightarrow \text{g\_sum}(Y)$$

Constructs Captured

- Nested Expressions
- Weight Constraints
- ... with Conditionals
- Aggregates
- Arithmetic Operators
- Choice Constructs

Constructs Captured

- Nested Expressions
- Weight Constraints
- ...with Conditionals
- Aggregates
- Arithmetic Operators
- Choice Constructs

} can all be expressed!

Constructs Captured

- Nested Expressions
- Weight Constraints
- . . . with Conditionals
- Aggregates
- Arithmetic Operators
- Choice Constructs

} can all be expressed!

- Minimized Guesses as a new construct

Minimized Guesses vs. Choice Constructs

- Choice Construct:

$$5\{\mathrm{accept}(X) : \mathrm{possible}(X)\} \leftarrow$$

$\hookrightarrow$ Any interpretation that accepts five or more elements can be stable

Minimized Guesses vs. Choice Constructs

- Choice Construct:

$$5\{\mathrm{accept}(X) : \mathrm{possible}(X)\} \leftarrow$$

$\hookrightarrow$ Any interpretation that accepts five or more elements can be stable

- Minimized Guess:

$$5 \leq_{\mathbb{N}} \neg\neg\mathrm{possible}(X) * (\mathrm{possible}(X) \rightarrow \mathrm{accept}(X)) \leftarrow$$

$\hookrightarrow$ Only interpretations that accepts exactly five elements can be stable

Language Aspects

- Domain independence is undecidable...

Language Aspects

- Domain independence is undecidable...
- ... but expressive *safe* fragment is domain independent

Language Aspects

- Domain independence is undecidable...
- ... but expressive *safe* fragment is domain independent

Theorem (Strong Equivalence)

For any Π_1, Π_2 programs, $\Pi_1 \equiv_s \Pi_2$ iff Π_1 has the same HT-models, i.e. satisfying pointed HT-interpretations, as Π_2.

Complexity

Theorem (Ground Complexity)

For variable-free programs over efficiently encoded semirings

- *MC and (propositional) SE are co-NP-complete.*
- *SAT is Σ_2^p-complete.*

Theorem (Non-ground Complexity)

For safe programs over efficiently encoded semirings

- *MC is in EXPTIME, both co-NP^{PP}-hard and NP^{PP}-hard and*
- *SAT and SE are undecidable.*
- *Over $\mathbb{N}$, MC is co-$NP^{NP^{PP}}$-complete*

Conclusion & Outlook

- Subsume many previous extensions and add new constructs

Conclusion & Outlook

- Subsume many previous extensions and add new constructs
  $\hookrightarrow$ constructs in a uniform language

Conclusion & Outlook

- Subsume many previous extensions and add new constructs
  $\hookrightarrow$ constructs in a uniform language
  $\hookrightarrow$ others [Cabalar *et al.*, 2020], [Son *et al.*, 2007] have abstract semantics but leave syntax open

Conclusion & Outlook

- Subsume many previous extensions and add new constructs
  $\hookrightarrow$ constructs in a uniform language
  $\hookrightarrow$ others [Cabalar *et al.*, 2020], [Son *et al.*, 2007] have abstract semantics but leave syntax open
- No increase in ground complexity

Conclusion & Outlook

- Subsume many previous extensions and add new constructs
  $\hookrightarrow$ constructs in a uniform language
  $\hookrightarrow$ others [Cabalar *et al.*, 2020], [Son *et al.*, 2007] have abstract semantics but leave syntax open
- No increase in ground complexity
- Work towards an implementation
  - finitely ground fragment
  - restrictions on weighted formulas

Stefano Bistarelli, Ugo Montanari, and Francesca Rossi.
Semiring-based constraint logic programming.
In *IJCAI (1)*, pages 352–357, 1997.

Francesco Buccafurri, Nicola Leone, and Pasquale Rullo.
Enhancing disjunctive datalog by constraints.
IEEE Transactions on Knowledge and Data Engineering, 12(5):845–860, 2000.

Pedro Cabalar, Jorge Fandinno, Torsten Schaub, and Philipp Wanko.
A uniform treatment of aggregates and constraints in hybrid ASP.
arXiv preprint arXiv:2003.04176, 2020.

Thomas Eiter and Rafael Kiesel.
Weighted lars for quantitative stream reasoning.
In *Proc. ECAI'20*, 2020.

Wolfgang Faber, Gerald Pfeifer, and Nicola Leone.
Semantics and complexity of recursive aggregates in answer set programming.
Artificial Intelligence, 175(1):278–298, 2011.

Paolo Ferraris.
Logic programs with propositional connectives and aggregates.
ACM Transactions on Computational Logic (TOCL), 12(4):25, 2011.

Todd J Green, Grigoris Karvounarakis, and Val Tannen.
Provenance semirings.
In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40. ACM, 2007.

Joohyung Lee and Zhun Yang.
Lpmln, weak constraints, and p-log.

In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Yuliya Lierler.
Relating constraint answer set programming languages and algorithms.
Artificial Intelligence, 207:1–22, 2014.

Ilkka Niemela, Patrik Simons, and Timo Soininen.
Stable model semantics of weight constraint rules.
In *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 317–331. Springer, 1999.

Tran Cao Son, Enrico Pontelli, and Phan Huy Tu.
Answer sets for logic programs with arbitrary abstract constraint atoms.
Journal of Artificial Intelligence Research, 29:353–389, 2007.