On Circumscription

Walter Forkel
walter.forkel@tu-dresden.de

August 21, 2021

Contents

Abstract

In this document we want to compare the classic version of circumscription [1] against the version proposed in [2]. We choose the murderer world as an example that illustrates the different behaviours.

Definition of predicate circumscription 2.0. Formalized with the *PIE* [1] system.

1 Parallel Predicate Circumscription

$xcirc(M, V, F)$

Defined as

$$F \wedge \neg\exists Args_1\,(F_2 \wedge Min),$$

where

$F_1 := F[M \mapsto M_1]$,
$F_2 := F_1[V \mapsto V_1]$,
sc_to_scsa(M1, F2, Ma),
mac_transfer_clauses_strict_subset(Ma, M, Min),
flatten([M1, V1], Args1).

Circumscription, as defined in [1], takes the original formula and adds additionally the Circumscription Axiom. This axiom says that a model is only a model for the whole expression, iff it satisfies the original formula and additionally there exists no other model that also satisfies the formula, but has a smaller extension of the minimized predicates. The axiom forbids all non-minimal models to be models of the whole expression. The second order quantification is what introduces the non-montonic behavior of circumscription.

2 Simulations Of Interpretations

For the definition of the alternative formulation of circumscription we need to formalize how an interpretation for a formula can be embedded in a parent interpretation and formula. We first introduce the relativization of a formula to then define the simulation of an interpretation.

$relat(P, F)$

Defined as

$$F_1,$$

where

[1] http://cs.christophwernhard.com/pie/

$$F_1 := F^P.$$

Relativization of a formula to a specific (domain) predicate $\Phi/1$ is achieved by relativizing all quantifiers in the formula. Each universial quantifier needs to hold only for the elements of Φ and similarly for each existential quantifier a valid assignment needs to be available using only elements of Φ.

$sim(P, F_0)$

Defined as

$$FRes,$$

where

$F_0 = F,$
mkAtom(P, x, T),
F2 = ex(x, T),
$F_3 := F^P,$
fun_closures(P, F, F4),
[... *unformattable Prolog code*].

An interpretation can be simulated or encoded inside another interpretation. This is accomplished by using a domain predicate P. It needs to be non-empty and the domain needs to be closed under function application.

3 Generalized Parallel Predicate Circumscription

$xcirc_2(M, V, F)$

Defined as

$$\begin{array}{ll} genSpaceAxiom(\Phi) & \wedge \\ sim(\Phi, F) & \wedge \\ genCircAxiom(\Phi, \psi, M, V, F). & \end{array}$$

$genCircAxiom(DomPred_1, DomPred_2, M, V, F)$

Defined as

$$\neg\exists Args\,(CircAxiomT \wedge sim(DomPred_2, F_2) \wedge Min),$$

where

$F_1 := F[M \mapsto M_1]$,
$F_2 := F_1[V \mapsto V_1]$,
sc_to_scsa(M1, F2, Ma),
circ_axiom(F, DomPred1, DomPred2, M, V, CircAxiom),
list2tuple(CircAxiom, CircAxiomT),
$Min := (Ma \subset M)^{DomPred_2, DomPred_1}$,
flatten([DomPred2, M1, V1], Args).

---

genSpaceAxiom(*DomP*)

Defined as

$$F,$$

where

$$\text{spaceAxiom}(\text{nat}, \text{DomP}, \text{F}).$$

The space axioms force the domain to contain infinite elements that are not elements of the given predicate DomP. This is achieved by encoding the natural numbers greater-than relation on the elements of the space predicate. The purpose of this is to ensure that there is always enough space to compare the first interpretation against any other different sized interpretation. Without the space axiom the murderer world example does not work.

In principle one can also let the function symbols vary. Since the PIE system has no build in support for function symbol (un-)skolemization, this cannot be implemented in PIE, yet.

3.1 Tests Of Helper Functions

In this part we want to do some sanity checks on simulations and relativizations. The results match exactly their expected definitions, see [2].

3.1.1 Tests Of Simulation

Test with function symbols and quantor:
Input: $sim(\mathsf{dom}, \forall x\,(\mathsf{m}(\mathsf{f}(x)) \wedge \mathsf{g}(x) \rightarrow \mathsf{ab}(x)))$.
Result of elimination:

$$\forall x\,(\mathsf{dom}(x) \wedge \mathsf{g}(x) \wedge \mathsf{m}(\mathsf{f}(x)) \rightarrow \mathsf{ab}(x)) \wedge$$
$$\forall x\,(\mathsf{dom}(x) \rightarrow \mathsf{dom}(\mathsf{f}(x))).$$

Test with constant function symbols and existential quantor.

Input: $sim(\mathsf{dom}, \exists x\,(\mathsf{m(f)} \wedge \mathsf{g(s}(x,x)) \rightarrow \mathsf{ab}(x)))$.
Result of elimination:

$$\begin{array}{ll}
\mathsf{dom(f)} & \wedge \\
\forall x\,(\mathsf{dom}(x) \rightarrow \forall y\,(\mathsf{dom}(y) \rightarrow \mathsf{dom(s}(x,y)))) & \wedge \\
\exists x\,(\mathsf{dom}(x) \wedge (\mathsf{g(s}(x,x)) \wedge \mathsf{m(f)} \rightarrow \mathsf{ab}(x))). &
\end{array}$$

3.2 Classic Circumscription Examples

In this section we want to go through some of the examples presented [3]. The examples basically test if primitive formulas yield the intuitively expected results when applying circumscription.

The expanded circumscription formula looks like the following:
Input: $xcirc_2([\mathsf{p}], [], \mathsf{pa})$.

$$\begin{array}{lll}
\mathsf{nat(null)} & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{nat}(y) \wedge \mathsf{gt}(x,y))) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \neg\mathsf{gt}(x, \mathsf{null})) & & \wedge \\
\forall xyz\,(\mathsf{nat}(x) \wedge \mathsf{nat}(y) \wedge \mathsf{nat}(z) \wedge \mathsf{gt}(x,y) \wedge \mathsf{gt}(y,z) \rightarrow \mathsf{gt}(x,z)) & & \wedge \\
\forall xy\,(\mathsf{nat}(x) \wedge \mathsf{nat}(y) \wedge \mathsf{gt}(x,y) \rightarrow \neg\mathsf{gt}(y,x)) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \neg\Phi(x)) & & \wedge \\
\mathsf{pa} & & \wedge \\
\Phi(\mathsf{a}) & & \wedge \\
\neg\exists qr\,(\top & \wedge & \\
\quad r\mathsf{a} & \wedge & \\
\quad q\mathsf{a} & \wedge & \\
\quad \forall x\,(qx \rightarrow (rx \rightarrow \mathsf{p}x)) & \wedge & \\
\quad \neg\forall x\,(\Phi(x) \rightarrow (\mathsf{p}x \rightarrow rx))). & &
\end{array}$$

If we circumscribe a predicate p in the formula $p(a)$ we expect that a is the only member of p. Note that we have to test this relativized to the domain predicate Φ.
Input: $xcirc_2([\mathsf{p}], [], \mathsf{p(a)})$.
Result of elimination:

$$\begin{array}{lll}
\Phi(\mathsf{a}) & & \wedge \\
\mathsf{nat(null)} & & \wedge \\
\mathsf{p(a)} & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\mathsf{gt}(x, \mathsf{null}) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow & \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\Phi(x) \wedge \mathsf{p}(x) \rightarrow x = \mathsf{a}). & &
\end{array}$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, \forall x\,(\mathsf{p}x \rightarrow x = \mathsf{a}))$.

If we circumscribe a predicate p in the formula $\neg p(a)$ we expect that no element is member of p:
Input: $xcirc_2([\mathsf{p}], [], \neg\mathsf{p}(\mathsf{a}))$.
Result of elimination:

$$
\begin{array}{lll}
\forall x\,\exists y\,((\mathsf{nat}(x) \rightarrow \mathsf{nat}(y)) \wedge (\mathsf{nat}(x) \rightarrow \mathsf{gt}(x,y))) & & \wedge \\
\Phi(\mathsf{a}) & & \wedge \\
\mathsf{nat}(\mathsf{null}) & & \wedge \\
\neg\mathsf{p}(\mathsf{a}) & & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{p}(x)) & & \wedge \\
\forall x\,\neg(\mathsf{nat}(x) \wedge \mathsf{gt}(x,\mathsf{null})) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{nat}(y) \wedge \mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x))) & & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow & \\
\quad \forall y\,(\mathsf{nat}(y) & \rightarrow & \\
\quad\quad \forall z\,\neg(\mathsf{nat}(z) \wedge \mathsf{gt}(x,z) \wedge \mathsf{gt}(z,y)) \vee \mathsf{gt}(x,y))). & &
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, \forall x\,\neg\mathsf{p}x)$.

If we circumscribe a predicate p in the formula $p(a) \wedge p(b)$ we expect that a and b are the only members of p. Note that we do not know wheather a and b refer to different objects.
Input: $xcirc_2([\mathsf{p}], [], (\mathsf{p}(\mathsf{a}) \wedge \mathsf{p}(\mathsf{b})))$.
Result of elimination:

$$
\begin{array}{lll}
\Phi(\mathsf{a}) & & \wedge \\
\Phi(\mathsf{b}) & & \wedge \\
\mathsf{nat}(\mathsf{null}) & & \wedge \\
\mathsf{p}(\mathsf{a}) & & \wedge \\
\mathsf{p}(\mathsf{b}) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow & \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\Phi(x) \wedge \mathsf{p}(x) \rightarrow x = \mathsf{a} \vee x = \mathsf{b}). & &
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, (\mathsf{pa} \wedge \mathsf{pb} \wedge \forall x\,(\mathsf{p}x \rightarrow x = \mathsf{a} \vee x = \mathsf{b})))$.

In the case of disjunction we expect that either a or b is are the only members of p. Note that both can be members, if the refer to the same object.
Input: $xcirc_2([\mathsf{p}], [], (\mathsf{p(a)} \vee \mathsf{p(b)}))$.
Result of elimination:

$$
\begin{array}{ll}
\Phi(\mathsf{a}) & \wedge \\
\Phi(\mathsf{b}) & \wedge \\
\mathsf{nat}(\mathsf{null}) & \wedge \\
(\mathsf{p(a)} \vee \mathsf{p(b)}) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(\Phi(x) \wedge \mathsf{p}(x) & \rightarrow \\
\quad (\mathsf{p(a)} \rightarrow x = \mathsf{a} \vee (\neg\mathsf{p(b)} \wedge \mathsf{a} = \mathsf{b})) \wedge (\mathsf{p(b)} \rightarrow x = \mathsf{b} \vee (\neg\mathsf{p(a)} \wedge \mathsf{a} = \mathsf{b}))). &
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, (\forall x\,(\mathsf{p}x \leftrightarrow x = \mathsf{a}) \vee \forall x\,(\mathsf{p}x \leftrightarrow x = \mathsf{b})))$.

3.3 The Murderer World

See [2] for the background story and motivation of the murderer world. The underlying problem is that the domain size influences circumscription: There exists the possibility that only one element exists in the domain. In this case an abnormality can not be avoided. Since entailment is based on all models, we can not conclude that there are no abnormalities, even though it seems reasonable.

---

murderer_world

Defined as

$$
\begin{array}{ll}
\forall y\,(\mathsf{murderer}(y) \wedge \mathsf{good}(y) \rightarrow \mathsf{abnormal}(y)) & \wedge \\
\exists x\,\mathsf{murderer}(x) & \wedge \\
\mathsf{good}(\mathsf{sam}). &
\end{array}
$$

Input: $xcirc_2([\mathsf{abnormal}], [\mathsf{good}, \mathsf{murderer}], murderer\_world)$.

Result of elimination:

$$
\begin{array}{l}
\Phi(\mathsf{sam}) \quad \wedge \\
\mathsf{good}(\mathsf{sam}) \quad \wedge \\
\mathsf{nat}(\mathsf{null}) \quad \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) \quad \wedge \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) \quad \wedge \\
\forall x\,(\mathsf{nat}(x) \quad \rightarrow \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) \quad \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) \quad \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) \quad \wedge \\
\forall x\,(\Phi(x) \wedge \mathsf{good}(x) \wedge \mathsf{murderer}(x) \rightarrow \mathsf{abnormal}(x)) \quad \wedge \\
\forall x,y\,(\Phi(y) \wedge \mathsf{abnormal}(y) \quad \rightarrow \\
\quad (x = \mathsf{sam} \wedge (\mathsf{abnormal}(x) \wedge \mathsf{abnormal}(\mathsf{sam}) \rightarrow x = y)) \quad \vee \\
\quad (x = \mathsf{sam} \wedge (\mathsf{abnormal}(x) \wedge \mathsf{abnormal}(\mathsf{sam}) \rightarrow y = \mathsf{sam}))) \quad \wedge \\
\exists x\,(\Phi(x) \wedge \mathsf{murderer}(x)).
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, (\forall x\,\neg\mathsf{abnormal}(x) \wedge \neg\mathsf{murderer}(\mathsf{sam})))$.

When not using the SpaceAxiom (hence McCarthy's formulation of circumscription), we can not draw the same conclusion anymore:
Input: $xcirc([\mathsf{abnormal}], [\mathsf{good}, \mathsf{murderer}], murderer\_world)$.
Result of elimination:

$$
\begin{array}{l}
\mathsf{good}(\mathsf{sam}) \quad \wedge \\
\forall x\,(\mathsf{good}(x) \wedge \mathsf{murderer}(x) \rightarrow \mathsf{abnormal}(x)) \quad \wedge \\
\forall x\,((\neg\mathsf{abnormal}(\mathsf{sam}) \wedge x = \mathsf{sam}) \quad \vee \\
\quad \forall y\,(\mathsf{abnormal}(y) \rightarrow x = y \wedge y = \mathsf{sam})) \quad \wedge \\
\exists x\,\mathsf{murderer}(x).
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow \forall x\,\neg\mathsf{abnormal}(x) \wedge \neg\mathsf{murderer}(\mathsf{sam})$.

3.4 Fixing The Domain Again

So far we introduced a space axiom, which forces the domain to be countably infinite. This allows the interpretations of the circumscribed formula to be minimized even between varying domains. Sometimes this may not be desired and comparison should just be possible between interpretations with the same domain. The generalized version of circumscription can be used to simulate a fixed domain in the following way:

$xcirc2FixedDomain(M, V, F, P)$

Defined as

$$xcirc_2(M, V, (\forall x\, Px \land F)),$$

where

$$\mathrm{mkAtom(P, x, Px)}.$$

$xcirc2FixedDomain(M, V, F)$

Defined as

$$xcirc2FixedDomain(M, V, F, P),$$

where

$$\mathrm{logform\_gen\_predicate(P)}.$$

Again if we fix the domain, we can not draw the same conclusion anymore.
Input: $xcirc2FixedDomain([\mathsf{abnormal}], [\mathsf{good}, \mathsf{murderer}], murderer\_world)$.
Result of elimination:

$$
\begin{array}{l}
\Phi(\mathsf{sam}) \land \\
\mathsf{good}(\mathsf{sam}) \land \\
\mathsf{nat}(\mathsf{null}) \land \\
(\$\mathsf{p}_1(\mathsf{sam}) \rightarrow \\
\quad \forall x\, (\$\mathsf{p}_1(x) \land \Phi(x) \rightarrow \\
\qquad (x = \mathsf{sam} \land \neg(\mathsf{abnormal}(x) \land \mathsf{abnormal}(\mathsf{sam}))) \lor \\
\qquad \forall y\, (\Phi(y) \land \mathsf{abnormal}(y) \rightarrow \\
\qquad\quad (x = y \land y = \mathsf{sam} \land \$\mathsf{p}_1(y)) \lor \\
\qquad\quad (x = \mathsf{sam} \land (\mathsf{abnormal}(x) \land \mathsf{abnormal}(\mathsf{sam}) \rightarrow x = y)) \lor \\
\qquad\quad (x = \mathsf{sam} \land (\mathsf{abnormal}(x) \land \mathsf{abnormal}(\mathsf{sam}) \rightarrow y = \mathsf{sam}))))) \land \\
\forall x\, (\mathsf{nat}(x) \rightarrow \exists y\, (\mathsf{gt}(x, y) \land \mathsf{nat}(y))) \land \\
\forall x\, \neg(\mathsf{gt}(x, \mathsf{null}) \land \mathsf{nat}(x)) \land \\
\forall x\, (\mathsf{nat}(x) \rightarrow \\
\quad \forall y\, (\mathsf{gt}(x, y) \land \mathsf{nat}(y) \rightarrow \forall z\, (\mathsf{gt}(y, z) \land \mathsf{nat}(z) \rightarrow \mathsf{gt}(x, z)))) \land \\
\forall x\, (\mathsf{nat}(x) \rightarrow \forall y\, \neg(\mathsf{gt}(x, y) \land \mathsf{gt}(y, x) \land \mathsf{nat}(y))) \land \\
\forall x\, \neg(\Phi(x) \land \mathsf{nat}(x)) \land \\
\forall x\, (\Phi(x) \rightarrow \$\mathsf{p}_1(x)) \land \\
\forall x\, (\Phi(x) \land \mathsf{good}(x) \land \mathsf{murderer}(x) \rightarrow \mathsf{abnormal}(x)) \land \\
\exists x\, (\Phi(x) \land \mathsf{murderer}(x)).
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, (\forall x\, \neg\mathsf{abnormal}(x) \wedge \neg\mathsf{murderer}(\mathsf{sam})))$.

If we add the artifially fixed domain to the varying predicates, it again becomes effectless and we again can conclude that Sam is not a murderer.
Input: $xcirc2FixedDomain([\mathsf{abnormal}], [\mathsf{good}, \mathsf{murderer}, \mathsf{fixed\_dom}], murderer\_world, \mathsf{fixed\_d}$
Result of elimination:

$$
\begin{array}{ll}
\Phi(\mathsf{sam}) & \wedge \\
\mathsf{good}(\mathsf{sam}) & \wedge \\
\mathsf{nat}(\mathsf{null}) & \wedge \\
\forall x\, (\mathsf{nat}(x) \rightarrow \exists y\, (\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\, \neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\, (\mathsf{nat}(x) & \rightarrow \\
\quad \forall y\, (\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\, (\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) & \wedge \\
\forall x\, (\mathsf{nat}(x) \rightarrow \forall y\, \neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\, \neg(\Phi(x) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\, (\Phi(x) \rightarrow \mathsf{fixed\_dom}(x)) & \wedge \\
\forall x\, (\Phi(x) \wedge \mathsf{good}(x) \wedge \mathsf{murderer}(x) \rightarrow \mathsf{abnormal}(x)) & \wedge \\
\forall x, y\, (\Phi(y) \wedge \mathsf{abnormal}(y) & \rightarrow \\
\quad (x = \mathsf{sam} \wedge (\mathsf{abnormal}(x) \wedge \mathsf{abnormal}(\mathsf{sam}) \rightarrow x = y)) & \vee \\
\quad (x = \mathsf{sam} \wedge (\mathsf{abnormal}(x) \wedge \mathsf{abnormal}(\mathsf{sam}) \rightarrow y = \mathsf{sam}))) & \wedge \\
\exists x\, (\Phi(x) \wedge \mathsf{murderer}(x)). &
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, (\forall x\, \neg\mathsf{abnormal}(x) \wedge \neg\mathsf{murderer}(\mathsf{sam})))$.

Note that in a similar way it is possible to exploit fixed predicates to refix function symbols.

3.5 Simulating Domain Circumscription

Is it possible to emulate domain circumscription? We already have domain predicates, why not simply minimize them? For this we need to omit the function closure for fixed functions.

$xdomcirc_2(V, F)$

Defined as

$$sim(\mathsf{Dom_1}, F) \wedge \neg\exists Args\, (A \wedge \neg\forall x\, (\mathsf{Dom_1}(x) \rightarrow \mathsf{Dom_2}(x))),$$

where

$$genCircAxiom(\mathsf{Dom_1}, \mathsf{Dom_2}, [], V, F) = \neg\exists Args\, A.$$

Using this we now force the domain to have just one element:

$$\begin{array}{ll} \mathsf{pa} & \wedge \\ \mathsf{Dom_1(a)} & \wedge \\ \neg\exists qr\,(\top \wedge r\mathsf{a} \wedge q\mathsf{a} \wedge \top \wedge \neg\forall x\,(\mathsf{Dom_1}(x) \rightarrow qx)). & \end{array}$$

Input: $xdomcirc_2([\mathsf{p}], \mathsf{p(a)})$.
Result of elimination:

$$\mathsf{Dom_1(a)} \wedge \mathsf{p(a)} \wedge \forall x\,(\mathsf{Dom_1}(x) \rightarrow x = \mathsf{a}).$$

Failed to validate this formula: $last\_result \rightarrow relat(\mathsf{Dom_1}, \forall x\, x = \mathsf{a})$.

In the case of the murderer world this means that Sam is the only existing entity.
Input: $xdomcirc_2([\mathsf{abnormal}, \mathsf{good}, \mathsf{murderer}], murderer\_world)$.
Result of elimination:

$$\begin{array}{ll} \mathsf{Dom_1(sam)} & \wedge \\ \mathsf{good(sam)} & \wedge \\ \forall x\,(\mathsf{Dom_1}(x) \wedge \mathsf{good}(x) \wedge \mathsf{murderer}(x) \rightarrow \mathsf{abnormal}(x)) & \wedge \\ \forall x, y\,(\mathsf{Dom_1}(y) \rightarrow x = y \vee y = \mathsf{sam}) & \wedge \\ \exists x\,(\mathsf{Dom_1}(x) \wedge \mathsf{murderer}(x)). & \end{array}$$

Failed to validate this formula: $last\_result \rightarrow relat(\mathsf{Dom_1}, \forall x\, x = \mathsf{sam})$.

3.6 Prioritized Circumscription With Domain Circumscription

Domain circumscription can be activated by adding the domain predicate Φ to the minimized predicates and if needed adding the required tuples to the given partial order.

In the following we extend the definition of xcirc2 by allowing a partial order as first argument.

$xcirc_2(Ord, M, V, F)$

Defined as

$$\begin{array}{ll} genSpaceAxiom(\Phi) & \wedge \\ sim(\Phi, F) & \wedge \\ genCircAxiom(\Phi, \psi, Ord, M, V, F). & \end{array}$$

genCircAxiom($DomPred_1$, $DomPred_2$, Ord, M, V, F)

Defined as

$$\neg\exists Args\,(CircAxiomT \wedge sim(DomPred_2, F_2) \wedge Min_1 \wedge Min_2),$$

where

delete(M, DomPred1, M0),
$F_1 := F[M_0 \mapsto M_1]$,
$F_2 := F_1[V \mapsto V_1]$,
sc_to_scsa(M1, F2, Ma),
[... *unformattable Prolog code*],
circ_axiom(F, DomPred1, DomPred2, M0, V, CircAxiom),
list2tuple(CircAxiom, CircAxiomT),
prio_strictness(DomPred1, DomPred2, M, Ma1, Min1),
prio_subset_rel(DomPred1, DomPred2, Ord, M, Ma1, Min2),
flatten([DomPred2, M1, V1], Args).

Input: $xcirc_2([], [\mathsf{p}], [], \mathsf{p}(\mathsf{a}))$.
Result of elimination:

$$\begin{array}{ll}
\Phi(\mathsf{a}) & \wedge \\
\mathsf{nat}(\mathsf{null}) & \wedge \\
\mathsf{p}(\mathsf{a}) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x, y) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\mathsf{gt}(x, \mathsf{null}) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow \\
\quad \forall y\,(\mathsf{gt}(x, y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y, z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x, z)))) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x, y) \wedge \mathsf{gt}(y, x) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(\Phi(x) \wedge \mathsf{p}(x) \rightarrow x = \mathsf{a}). &
\end{array}$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, \forall x\,(\mathsf{p}x \rightarrow x = \mathsf{a}))$.

Input: $xcirc_2([(\mathsf{p}, \mathsf{q})], [\mathsf{p}, \mathsf{q}], [], (\mathsf{p}(\mathsf{a}) \wedge \mathsf{q}(\mathsf{a})))$.

Result of elimination:

$$
\begin{array}{ll}
\Phi(\mathsf{a}) & \wedge \\
\mathsf{nat}(\mathsf{null}) & \wedge \\
\mathsf{p}(\mathsf{a}) & \wedge \\
\mathsf{q}(\mathsf{a}) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(x = \mathsf{a} \vee (\neg(\Phi(x) \wedge \mathsf{p}(x)) \wedge \neg(\Phi(x) \wedge \mathsf{q}(x)))). &
\end{array}
$$

3.6.1 Prioritized Circumscription With Disjunction

Input: $xcirc_2([],[\mathsf{p}],[],(\mathsf{p}(\mathsf{a}) \vee \mathsf{p}(\mathsf{b})))$.
Result of elimination:

$$
\begin{array}{lll}
\Phi(\mathsf{a}) & & \wedge \\
\Phi(\mathsf{b}) & & \wedge \\
\mathsf{nat}(\mathsf{null}) & & \wedge \\
(\mathsf{p}(\mathsf{a}) \vee \mathsf{p}(\mathsf{b})) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow & \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\Phi(x) \wedge \mathsf{p}(x) & \rightarrow & \\
\quad (\mathsf{p}(\mathsf{a}) \rightarrow x = \mathsf{a} \vee (\neg\mathsf{p}(\mathsf{b}) \wedge \mathsf{a} = \mathsf{b})) \wedge (\mathsf{p}(\mathsf{b}) \rightarrow x = \mathsf{b} \vee (\neg\mathsf{p}(\mathsf{a}) \wedge \mathsf{a} = \mathsf{b}))). & &
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, (\forall x\,(\mathsf{p}x \rightarrow x = \mathsf{a}) \vee \forall x\,(\mathsf{p}x \rightarrow x = \mathsf{b})))$.

Unfortunately, disjunction does not work with priorization so far:
Input: $xcirc_2([(\mathsf{p},\mathsf{q})],[\mathsf{p},\mathsf{q}],[],(\mathsf{p}(\mathsf{a}) \vee \mathsf{q}(\mathsf{a})))$.

Result of elimination:

$$
\begin{array}{ll}
\mathsf{\Phi(a)} & \wedge \\
\mathsf{nat(null)} & \wedge \\
(\mathsf{p(a)} \vee \mathsf{q(a)}) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) & \wedge \\
\forall x\,\neg(\mathsf{\Phi}(x) \wedge \mathsf{nat}(x)) & \wedge \\
\forall x\,(\mathsf{\Phi}(x) \wedge \mathsf{p}(x) & \rightarrow \\
\quad \neg\mathsf{q(a)} \wedge (\mathsf{p(a)} \rightarrow x = \mathsf{a}) \wedge \forall y\,\neg(\mathsf{\Phi}(y) \wedge \mathsf{p}(y))) & \wedge \\
\forall x\,(\mathsf{\Phi}(x) \wedge \mathsf{q}(x) \rightarrow \neg\mathsf{p(a)} \wedge (\mathsf{q(a)} \rightarrow x = \mathsf{a})). &
\end{array}
$$

Failed to validate this formula: $last\_result \rightarrow relat(\mathsf{\Phi}, (\forall x\,\neg\mathsf{p}x \wedge \forall x\,(\mathsf{q}x \rightarrow x = \mathsf{a})))$.

3.6.2 Prioritized Circumscription With Conjunction In The Murderer World

This time we want to minimize abnormal/1 again, but additionally we want to have as few as possible murderers and good people.
Input: $xcirc_2([(\mathsf{abnormal}, \mathsf{good}), (\mathsf{abnormal}, \mathsf{murderer})], [\mathsf{abnormal}, \mathsf{good}, \mathsf{murderer}], [], murderer$

Result of elimination:

$$
\begin{array}{l}
\Phi(\mathsf{sam}) \\
\mathsf{good}(\mathsf{sam}) \\
\mathsf{nat}(\mathsf{null}) \\
\forall x\,(\mathsf{nat}(x) \to \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) \\
\forall x\,(\mathsf{nat}(x) \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \to \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \to \mathsf{gt}(x,z)))) \\
\forall x\,(\mathsf{nat}(x) \to \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) \\
\forall x\,(\Phi(x) \wedge \mathsf{good}(x) \wedge \mathsf{murderer}(x) \to \mathsf{abnormal}(x)) \\
\forall x\,((\neg\mathsf{abnormal}(x) \wedge x = \mathsf{sam}) \\
\quad (\neg\mathsf{abnormal}(\mathsf{sam}) \wedge x = \mathsf{sam}) \\
\quad (\neg\mathsf{good}(x) \\
\quad\ x = \mathsf{sam} \\
\quad\ (\mathsf{abnormal}(x) \wedge \mathsf{abnormal}(\mathsf{sam}) \\
\quad\ \forall y\,(\Phi(y) \wedge \mathsf{abnormal}(y) \to x = y \vee y = \mathsf{sam}))) \\
\quad ((\mathsf{murderer}(x) \to \neg\mathsf{murderer}(\mathsf{sam}) \wedge x = \mathsf{sam}) \\
\quad\ ((\neg\mathsf{abnormal}(x) \wedge x = \mathsf{sam}) \\
\quad\ (\neg\mathsf{abnormal}(\mathsf{sam}) \wedge x = \mathsf{sam}) \\
\quad\ \forall y\,(\Phi(y) \wedge \mathsf{abnormal}(y) \to (x = y \wedge x = \mathsf{sam}) \vee (x = \mathsf{sam} \wedge y = \mathsf{sam})))) \\
\quad ((\mathsf{murderer}(x) \\
\quad\ (\neg\mathsf{murderer}(\mathsf{sam}) \wedge x = \mathsf{sam}) \vee \\
\quad\ \forall y\,(\Phi(y) \wedge \mathsf{murderer}(y) \to x = y)) \\
\quad\ ((\neg\mathsf{abnormal}(x) \wedge x = \mathsf{sam}) \\
\quad\ (\neg\mathsf{abnormal}(\mathsf{sam}) \wedge x = \mathsf{sam}) \\
\quad\ \forall y\,(\Phi(y) \wedge \mathsf{abnormal}(y) \to (x = y \wedge x = \mathsf{sam}) \vee (x = \mathsf{sam} \wedge y = \mathsf{sam}))) \\
\quad\ ((\neg\mathsf{good}(x) \wedge x = \mathsf{sam}) \vee \forall y\,(\Phi(y) \wedge \mathsf{good}(y) \to y = \mathsf{sam})))) \\
\exists x\,(\Phi(x) \wedge \mathsf{murderer}(x)).
\end{array}
$$

Still there should be no abnormal person, additionally there should be the minimum number of good and murderers in the solution.

Failed to validate this formula: $last\_result \to relat(\Phi, (\forall x\,\neg\mathsf{abnormal}(x) \wedge \neg\mathsf{murderer}(\mathsf{sam})))$.

Failed to validate this formula: $last\_result \to relat(\Phi, \forall x\,(\mathsf{good}(x) \to x = \mathsf{sam}))$.

Failed to validate this formula: $last\_result \to relat(\Phi, \exists x\,\forall y\,(\mathsf{murderer}(y) \to x = y))$.

On the other hand the domain can still have arbitrary size, causing the assertion below to fail:

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, \exists xy\,(x \neq y \wedge \forall z\,(z = x \vee z = y)))$.

3.7 Combining Domain and Predicate Circumscription

We can combine domain and predicate circumscription by adding the domain predicate Φ to the minimized predicates and minimizing it with least priority:
Input: $xcirc_2([(\mathsf{p}, \Phi)], [\mathsf{p}, \Phi], [], \mathsf{p}(\mathsf{a}))$.
Result of elimination:

$$
\begin{array}{lll}
\Phi(\mathsf{a}) & & \wedge \\
\mathsf{nat}(\mathsf{null}) & & \wedge \\
\mathsf{p}(\mathsf{a}) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x, y) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\mathsf{gt}(x, \mathsf{null}) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\mathsf{nat}(x) & \rightarrow & \\
\quad \forall y\,(\mathsf{gt}(x, y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y, z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x, z)))) & & \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x, y) \wedge \mathsf{gt}(y, x) \wedge \mathsf{nat}(y))) & & \wedge \\
\forall x\,\neg(\Phi(x) \wedge \mathsf{nat}(x)) & & \wedge \\
\forall x\,(\Phi(x) \rightarrow x = \mathsf{a}) & & \wedge \\
\forall x\,(\Phi(x) \wedge \mathsf{p}(x) \rightarrow x = \mathsf{a}). & &
\end{array}
$$

The expected solution is that only constant a is contained in p and the domain has size one:

Failed to validate this formula: $last\_result \rightarrow relat(\Phi, \forall x\,(x = \mathsf{a} \wedge \mathsf{p}x))$.

In the case of the murderer world we could state that we want to minimze abnormal and the domain. Since they are in conflict, the domain should still be forced to be of at least size two.
Input: $xcirc_2([(\mathsf{abnormal}, \Phi)], [\mathsf{abnormal}, \Phi], [\mathsf{good}, \mathsf{murderer}], murderer\_world)$.

Result of elimination:

$$
\begin{array}{l}
\mathsf{\Phi(sam)} \wedge \\
\mathsf{good(sam)} \wedge \\
\mathsf{nat(null)} \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \exists y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y))) \wedge \\
\forall x\,\neg(\mathsf{gt}(x,\mathsf{null}) \wedge \mathsf{nat}(x)) \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \\
\quad \forall y\,(\mathsf{gt}(x,y) \wedge \mathsf{nat}(y) \rightarrow \forall z\,(\mathsf{gt}(y,z) \wedge \mathsf{nat}(z) \rightarrow \mathsf{gt}(x,z)))) \wedge \\
\forall x\,(\mathsf{nat}(x) \rightarrow \forall y\,\neg(\mathsf{gt}(x,y) \wedge \mathsf{gt}(y,x) \wedge \mathsf{nat}(y))) \wedge \\
\forall x\,\neg(\mathsf{\Phi}(x) \wedge \mathsf{nat}(x)) \wedge \\
\forall x\,(\mathsf{\Phi}(x) \wedge \mathsf{good}(x) \wedge \mathsf{murderer}(x) \rightarrow \mathsf{abnormal}(x)) \wedge \\
\forall x\,((\neg\mathsf{\Phi}(x) \\
\quad ((\neg\mathsf{abnormal}(x) \wedge x = \mathsf{sam}) \\
\quad\ (\neg\mathsf{abnormal(sam)} \wedge x = \mathsf{sam}) \\
\quad\ \forall y\,(\mathsf{\Phi}(y) \wedge \mathsf{abnormal}(y) \rightarrow (x = y \wedge x = \mathsf{sam}) \vee (x = \mathsf{sam} \wedge y = \mathsf{sam})))) \\
\quad (\neg\mathsf{abnormal}(x) \wedge x = \mathsf{sam}) \\
\quad (\neg\mathsf{abnormal(sam)} \wedge x = \mathsf{sam}) \\
\quad ((\mathsf{\Phi}(x) \rightarrow \forall y\,(\mathsf{\Phi}(y) \rightarrow x = y \vee y = \mathsf{sam})) \\
\quad\ ((\neg\mathsf{abnormal}(x) \wedge x = \mathsf{sam}) \\
\quad\ (\neg\mathsf{abnormal(sam)} \wedge x = \mathsf{sam}) \\
\quad\ \forall y\,(\mathsf{\Phi}(y) \wedge \mathsf{abnormal}(y) \rightarrow (x = y \wedge x = \mathsf{sam}) \vee (x = \mathsf{sam} \wedge y = \mathsf{sam}))))) \wedge \\
\exists x\,(\mathsf{\Phi}(x) \wedge \mathsf{murderer}(x)).
\end{array}
$$

Since we are putting the highest priority on minimizing abnormal, there should still be no abnormalities. When minimizing abnormal and the domain, the expected solution is, that still there is no abnormality and therefore there has to be at least one other individual than Sam. On the other hand, the domain should be as small as possible. This implies that there exist exactly two people: Sam and the murderer:

3.8 Auxiliary Macros

---

last_result

Defined as

$$X,$$

where

last_ppl_result(X).

last_form_result

Defined as

$$X,$$

where

last_ppl_form_result(X).

References

[1] John McCarthy. Circumscription—a form of non-monotonic reasoning. In *Readings in Artificial Intelligence*, pages 466–472. Elsevier, 1981.

[2] Walter Forkel. Circumscription revisited. diploma thesis, TU Dresden, 2017.

[3] Vladimir Lifschitz. Computing circumscription. In *IJCAI*, volume 85, pages 121–127, 1985.

Index