CAKE **System**

Computer
Aided
Knowledge
Engineering

based on:
P. Doherty, W. Łukaszewicz, A. Skowron, A. Szałas:
Knowledge Representation Techniques: A Rough Set Approach, Springer-Verlag, 2006.

- **Goal:** reasoning in the framework of distributed and incomplete information.

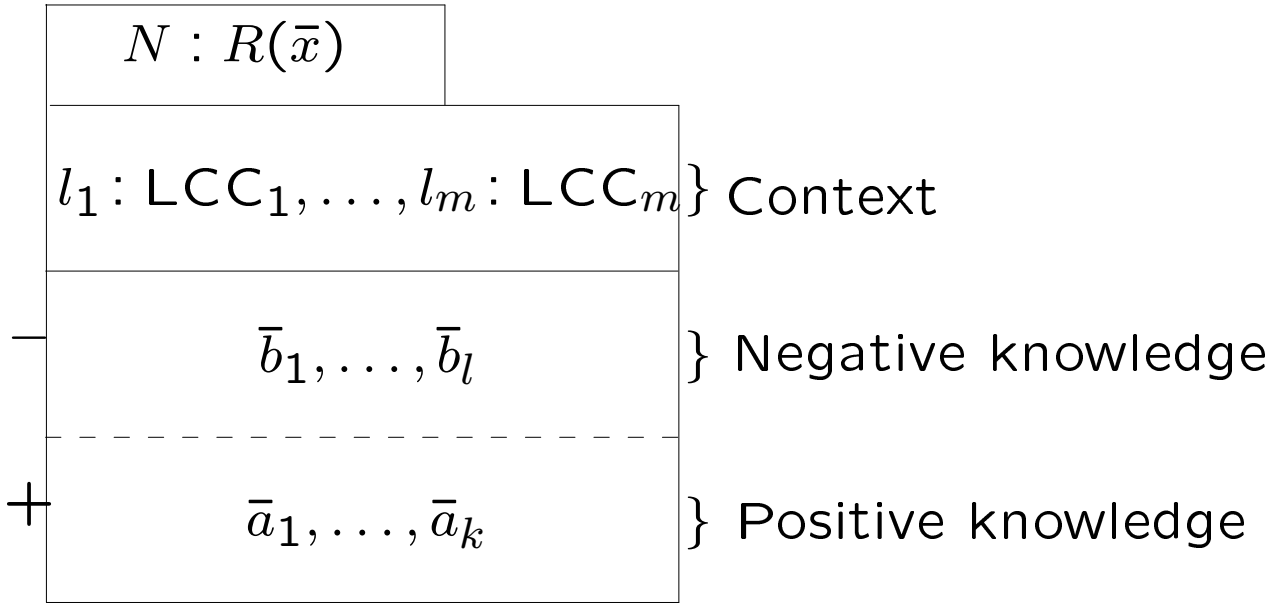
- **Basis:** *a rough relation* paradigm.
  More specifically, a rough relation R is a triple $\langle R^+, R^-, R^\pm \rangle$, where R^+ denotes a set of tuples **known** to satisfy R, R^- denotes a set of tuples **known** not to satisfy R and $R^\pm$ denotes a set of remaining tuples.

- **Visualization:** diagrams representing rough relations.

- **Implementation:** Department of Computer and Information Science of Linköping University.

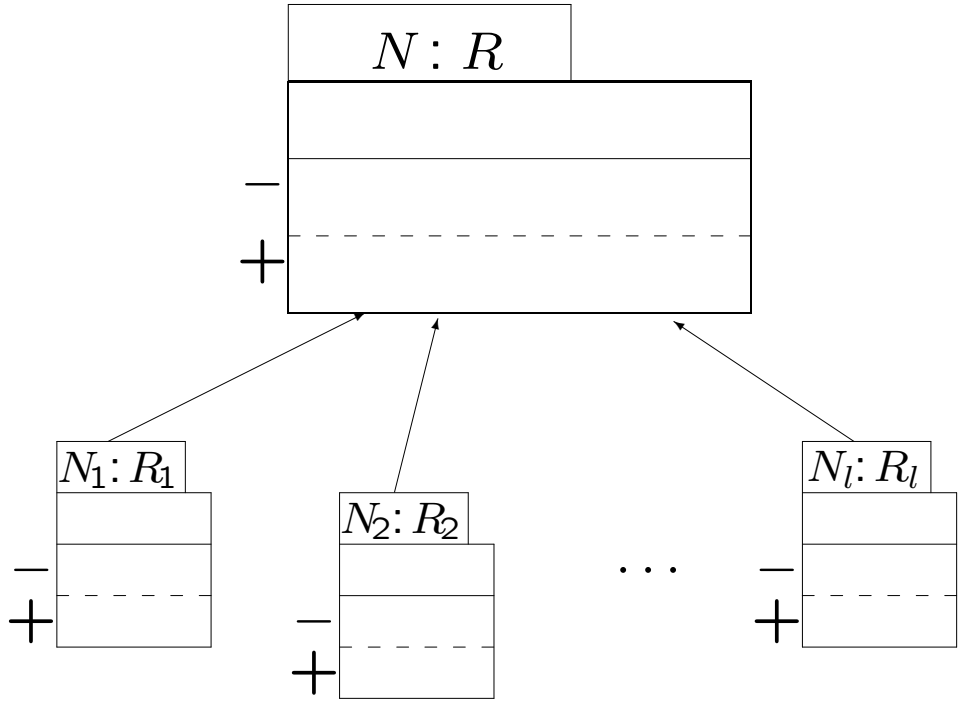
The basic notion of CAKE is that of an *information granule*. Any granule contains information concerning a single rough relation.

Diagram representing a CAKE granule.



If a relation is given by many sources, it should be distributed among many granules. To point out the source of information, any relation is prefixed by a name of a granule.

Diagram representing input relations for a CAKE granule.



Formally, an information granule can be viewed as a set of facts and rules allowing to compute the relation the granule is responsible for.

Example 1

We are given two databases, DB_1 and DB_2, containing information about a relation $C(x, y)$ meaning that a place x is directly connected with a place y.

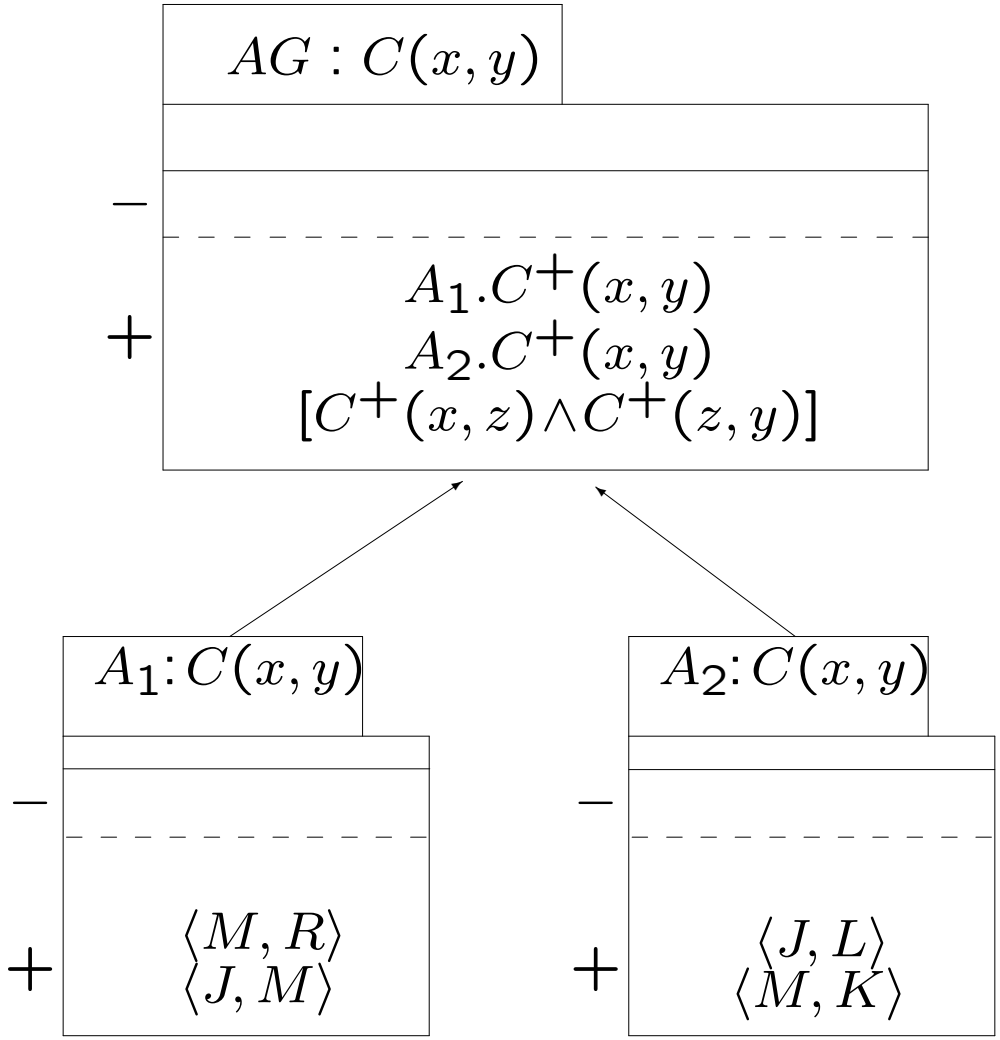
DB_1 contains $C(j, m)$, $C(m, r)$ and DB_2 contains $C(m, k)$, $C(j, l)$.

Let D_1 and D_2 be names of information granules representing these databases.
D_1 consists of $D_1.C^+(j, m)$, $D_1.C^+(m, r)$,
D_2 consists of $D_2.C^+(m, k)$, $D_2.C^+(j, l)$.*

*Only positive and negative parts of relation are explicitly represented in facts and rules.

Now we want to combine information from D_1 and D_2 and to represent indirect connections too.
We introduce a new granule, D, provided with the following rules.

Diagram of Example 1.



and the set of rules defined by this diagram.

$$D_1.C^+(x,y) \wedge \neg D_2.C^-(x,y) \Rightarrow D.C^+(x,y) \quad (1)$$
$$D_1.C^-(x,y) \wedge \neg D_2.C^+(x,y) \Rightarrow D.C^-(x,y) \quad (2)$$
$$D_2.C^+(x,y) \wedge \neg D_1.C^-(x,y) \Rightarrow D.C^+(x,y) \quad (3)$$
$$D_2.C^-(x,y) \wedge \neg D_1.C^+(x,y) \Rightarrow D.C^-(x,y) \quad (4)$$
$$D.C^+(x,z) \wedge D.C^+(z,y) \Rightarrow D.C^+(x,y). \quad (5)$$

The rules (1)-(4) import consistent information from D_1 and D_2.

$\neg D_i.C^+(x,y)$ is equivalent to $D_i.C^-(x,y) \vee D_i.C^{\pm}(x,y)$
$\neg D_i.C^-(x,y)$ is equivalent to $D_i.C^+(x,y) \vee D_i.C^{\pm}(x,y)$.

Using the facts stored by D_1 and D_2, together with the rules (1)-(5), we can infer $D.C^+(j,k)$.

We say that the granule D answers "yes" to the query whether there is a connection between j and k.

Syntax of CAKE

Alphabet of classical first-order logic containing an enumerable set of individual variables and finite sets of individual constants and relation symbols*.

An atomic formula, *C-atom* for short, – expression of the form $A.R^{+}(\overline{u})$ or $A.R^{-}(\overline{u})$, where
A – a name of an information granule,
R – a relation name,
$\overline{u}$ – a tuple of individual variables and/or constant symbols.

A *literal* is a C-atom or an expression of the form $\neg\alpha$, where α is a C-atom.

*Note that CAKE does not permit function symbols.

A CAKE *rule* – any expression of the form

$$\alpha_1 \wedge \ldots \wedge \alpha_n \Rightarrow \alpha \quad (n \geq 0) \tag{6}$$

where $\alpha_1, \ldots, \alpha_n$ are literals and α is C-atom.
α – the *head* of the rule,
$\alpha_1 \wedge \ldots \wedge \alpha_n$ – its *body*.

A rule with the empty body is called a *fact*. In the sequel, we shall write α instead of $\Rightarrow \alpha$.

Any finite set of rules and facts is called a CAKE *program*.

Definition 1

Let $(x_1, \ldots, x_n)$ be a tuple of all free variables occurring in a rule (6). *A ground instance* of (6) is any rule obtainable from (6) by uniformly replacing each occurrence of x_i by c_i, where c_i is an individual constant, for $1 \leq i \leq n$. A set of all ground instances of all rules from a program Π is called the *ground instance* of Π. ▪

Voting

Problem: different granules may provide contradictory information about a particular relation. A granule which combines information from contradictory sources (as D in Example 1) has to solve the inconsistency problem. In Example 1 the problem is solved by the rules (1)-(4).

In general case, i.e. if there are many different granules responsible for a relation, an additional granule, called voting granule, is introduced and equipped with rules solving inconsistencies.

The *standard voting mechanism* is based on the following idea.

- if at least one granule responsible for relation R answers *True* to the query and none of the granules answers *False*, the final answer to the query is *True*

- if at least one granule answers *False* to the query and none answers *True*, the final answer to the query is *False*.

- otherwise, the answer to the query is *Unknown*.

Formalization

M – the set of all granules delivering the relation R. V – the CAKE voting granule attached to relation R

$$\bigvee_{N \in M} N.R^{+}(\overline{x}) \wedge \bigwedge_{N \in M} \neg N.R^{-}(\overline{x}) \Rightarrow V.R^{+}(\overline{x})$$
$$\bigvee_{N \in M} N.R^{-}(\overline{x}) \wedge \bigwedge_{N \in M} \neg N.R^{+}(\overline{x}) \Rightarrow V.R^{-}(\overline{x})$$

The standard voting policy may be changed, for instance, by providing priorities among granules.

Semantics of CAKE
Stratified semantics (fixpoint semantics) of CAKE programs.

Definition 2

A CAKE program Π is said to be *semi-positive* if, whenever a literal l occurs negatively in the body of a rule of Π, then the relation symbol of l occurs in facts only.▪

Π – a semi-positive CAKE program, K – a set of ground C-atoms.

Definition 3
A ground C-atom α is *an immediate consequence* of Π and K if either $\alpha \in K$ or $\alpha_1 \wedge \ldots \wedge \alpha_n \Rightarrow \alpha$ is a ground instance of a rule from Π such that

1. if α_i is a positive literal, then $\alpha_i \in K$ and
2. if α_i is negative literal of the form $\neg\beta_i$, then $\beta_i \notin K$.

operator $\mathcal{C}_\Pi$: $\mathcal{C}_\Pi(K)$ – the set of all immediate consequences of Π and any set of ground C-atoms K.

Let Π be a semi-positive CAKE program. The set of consequences of Π is the smallest fixpoint, written lfp, of the operator $\mathcal{C}_\Pi$. For a semi-positive CAKE program Π, lfp($\mathcal{C}_\Pi$) always exists and is given by the following construction:

X:=K, where K is the set of all facts from Π;

while X$\neq \mathcal{C}_\Pi$(X) do X:=$\mathcal{C}_\Pi$(X).

The above semantics can be generalized to stratified CAKE programs.

Definition 4

By a *stratification* of a CAKE program Π we mean a sequence of CAKE programs $\Pi^1, \ldots, \Pi^n$, with Π^1 possibly empty, such that

- $\Pi^1 \cup \ldots \cup \Pi^n = \Pi$ and for all $1 \leq i \neq j \leq n$, $\Pi^i \cap \Pi^j = \emptyset$
- for any relation X of Π and $1 \leq i \leq n$:
  - if X occurs positively in a rule's body in Π^i, then all the rules with X in their heads are in $\Pi^1 \cup \ldots \cup \Pi^i$
  - if X occurs in a rule's body in Π^i under negation, then all the rules with X in their heads are in $\Pi^1 \cup \ldots \cup \Pi^{i-1}$.

▪

Given a stratification $\Pi^1, \ldots, \Pi^n$ of Π, each Π^i is called a *stratum* of the stratification. A program is called *stratified* if it has a stratification.

Let Π be a stratified CAKE program and let $\Pi^1, \ldots, \Pi^n$ be its stratification. We define an operator $\mathcal{CS}_\Pi(X)$ by

$$\begin{aligned}
\mathcal{CS}_\Pi(X) &= \mathcal{CS}^n_\Pi(X), \text{ where} \\
\mathcal{CS}^1_\Pi(X) &= \mathcal{C}_{\Pi^1}(X) \\
\mathcal{CS}^i_\Pi(X) &= \mathcal{C}_{\Pi^i \cup \mathsf{lfp}(\mathcal{CS}^{i-1}_\Pi)}(X).
\end{aligned}$$

The set of consequences of Π is the smallest fixpoint of an operator $\mathcal{CS}_\Pi$. Its existence follows from the fact that, for $1 < i \leq n$, the program $\Pi^i \cup \mathsf{lfp}(\mathcal{CS}^{i-1}_\Pi)$ is semi-positive.

Well-founded Semantics for CAKE **Programs**

Well-founded semantics is based on a notion of three-valued interpretation, in which the truth value of facts can be *True*, *False* or *Unknown*. In the sequel, we shall consider only finite interpretations, i.e. interpretations providing truth values to all ground C-atoms in a given CAKE program Π.

Three-valued interpretations.

Definition 5
Let $ATM(\Pi)$ be a set of all ground C-atoms in a CAKE program Π. A *3-valued interpretation* **S** of (Π) is a mapping from $ATM(\Pi)$ into the set {*True*,*Unknown*,*False*}. ▪

We denote by $\mathbf{S}^0$, $\mathbf{S}^{1/2}$, $\mathbf{S}^1$ the set of C-atoms in $ATM(\Pi)$ whose truth value is *False*, *Unknown* and *True* respectively.

Representation of a 3-valued interpretation – the list of the positive and negative facts only.
For instance, if $\mathbf{S}(P^+(a)) = \mathit{True}$, $\mathbf{S}(P^+(b)) = \mathit{Unknown}$, $\mathbf{S}(P^-(b)) = \mathit{Unknown}$ and $\mathbf{S}(P^-(a)) = \mathit{False}$ then it may be represented as $\mathbf{S}$=$\{P^+(a), \neg P^-(a)\}$.

There is a natural ordering $\prec$ among 3-valued interpretations of a program (Π) defined by

$\mathbf{S} \prec \mathbf{T}$ iff $\mathbf{S}^1 \subseteq \mathbf{T}^1$ and $\mathbf{T}^0 \subseteq \mathbf{S}^0$.

The least element wrt $\prec$, denoted by $\perp$, is the interpretation where all atoms are false.

Given a 3-valued interpretation **S** we can extend it in a standard way to Boolean combinations of atoms:

$$
\begin{aligned}
\mathbf{S}(A \wedge B) &= \min(\mathbf{S}(A), \mathbf{S}(B)) \\
\mathbf{S}(A \vee B) &= \max(\mathbf{S}(A), \mathbf{S}(B)), \\
\mathbf{S}(\neg A) &= \begin{cases} \mathit{True} & \text{if } \mathbf{S}(A) = \mathit{False} \\ \mathit{Unknown} & \text{if } \mathbf{S}(A) = \mathit{Unknown} \\ \mathit{False} & \text{if } \mathbf{S}(A) = \mathit{True} \end{cases}
\end{aligned}
$$

where $\min, \max$ are defined according to the truth ordering $\mathit{False} < \mathit{Unknown} < \mathit{True}$.

Definition 6
Given a CAKE program Π and an interpretation **S** of Π, the *positivized version* of Π wrt **S** is the program obtained from ground instance of Π by replacing each negative literal $\neg A$ by $\mathbf{S}(\neg A)$ (i.e. *False*, *True* or *Unknown*). Such program will be called a 3-CAKE program.*

Example 2 *Consider a ground* CAKE *program* Π *given by*

$$\begin{aligned}
\neg R^+(a) &\Rightarrow P^+(a) \\
\neg R^+(a) \wedge P^+(a) &\Rightarrow Q^+(a) \\
\neg T^+(a) &\Rightarrow S^+(a) \\
Q^+(a) \wedge \neg S^+(a) &\Rightarrow T^+(a) \\
\neg T^+(a) \wedge P^+(a) \wedge S^+(a) &\Rightarrow U^+(a)
\end{aligned}$$

*Note that 3-CAKE programs do not contain negation symbols.

and the interpretation

$$\perp = \{\neg P^+(a), \neg Q^+(a), \neg R^+(a), \neg S^+(a), \neg T^+(a), \neg U^+(a)\}.$$

The positivized version of Π wrt $\perp$ is the 3-CAKE program, Π_+, given by

$$\begin{aligned}
\mathit{True} &\Rightarrow P^+(a) \\
\mathit{True} \wedge P^+(a) &\Rightarrow Q^+(a) \\
\mathit{True} &\Rightarrow S^+(a) \\
Q^+(a) \wedge \mathit{True} &\Rightarrow T^+(a) \\
\mathit{True} \wedge P^+(a) \wedge S^+(a) &\Rightarrow U^+(a).
\end{aligned}$$

▪

Computing well-founded semantics

Definition 7

Let Π be a 3-CAKE program. The *3-valued immediate consequence operator, 3-*$\mathcal{C}_\Pi$, on an interpretation **S** of Π is a mapping defined as follows, where $A \in ATM(\Pi)$:

$$3-\mathcal{C}_\Pi(A) \stackrel{\text{def}}{=} \begin{cases} \textit{True} & \text{if } A \text{ is a fact or there is a rule of the form} R_1 \wedge \cdots \wedge R_k \Rightarrow A \text{ such that } \mathbf{S}(R_1 \wedge \ldots \wedge R_k) = 1 \\ \textit{False} & \text{if there is no rule with } A \text{ in its head or, for each rule of the form } R_1 \wedge \cdots \wedge R_k \Rightarrow A, \ \mathbf{S}(R_1 \wedge \ldots \wedge R_k) = 0 \\ \textit{Unknown} & \text{otherwise.} \end{cases}$$

▪

The immediate consequence operator 3-$\mathcal{C}_\Pi$ has following property.

Theorem 1
Let Π be a 3-CAKE program. Then the sequence $\{(3\text{-}\mathcal{C}_\Pi)^i(\bot)\}_{i>0}$ is non-decreasing and converges to the least fixpoint of 3-$\mathcal{C}_\Pi$. ▪

Let Π be a CAKE program and let **S** be a 3-valued interpretation. We write 3-$\mathcal{CS}_\Pi(\mathbf{S})$ to denote the least fixpoint of 3-$\mathcal{C}_{\Pi_+}$, where Π_+ is the positivized version of the ground instance of the program Π wrt **S**.

Example 2 (continued)

We compute 3-$\mathcal{CS}_{\Pi}(\bot)$.

3-$\mathcal{C}_{\Pi_+}(\bot) = \{P^+(a), S^+(a), \neg Q^+(a), \neg T^+(a), \neg U^+(a), \neg R^+(a)\}$

(3-$\mathcal{C}_{\Pi_+})^2(\bot) = \{P^+(a), S^+(a), Q^+(a), \neg T^+(a), U^+(a), \neg R^+(a)\}$

(3-$\mathcal{C}_{\Pi_+})^3(\bot) = \{P^+(a), S^+(a), Q^+(a), T^+(a), U^+(a), \neg R^+(a)\}$

(3-$\mathcal{C}_{\Pi_+})^4(\bot) =$ (3-$\mathcal{C}_{\Pi_+})^3(\bot)$.

Thus, 3-$\mathcal{CS}_{\Pi}(\bot) = \{R^+(a), S^+(a), Q^+(a), T^+(a), U^+(a), \neg R^+(a)\}$.▪

Definition of well-founded semantics of a CAKE program.

Let Π be a ground CAKE program. We define the following sequence of interpretations:

$$\begin{aligned} \mathbf{S}_0 &= \bot \\ \mathbf{S}_{i+1} &= \text{3-}\mathcal{CS}_\Pi(\mathbf{S}_i). \end{aligned}$$

It can be shown that, for all $i > 0$,

$$\mathbf{S}_0 \prec \mathbf{S}_2 \ldots \prec \mathbf{S}_{2i} \prec \mathbf{S}_{2i+2} \prec \ldots \prec \mathbf{S}_{2i+1} \prec \mathbf{S}_{2i-1} \prec \ldots \prec \mathbf{S}_1.$$

The even subsequence is non-decreasing and the odd one is non-increasing. Since there are finitely many three-valued interpretations for a given program, each of those subsequences becomes constant at some point.

Let $\mathbf{S}_*$ denote the limit of the subsequence $\{\mathbf{S}_{2i}\}_{i\geq 0}$ and let $\mathbf{S}^*$ denote the limit of the subsequence $\{\mathbf{S}_{2i+1}\}_{i\geq 0}$. A three-valued interpretation of P, denoted by $\mathbf{S}^*_*(A)$, is defined by

$$\mathbf{S}^*_*(A) = \begin{cases} \textit{True} & \text{if } \mathbf{S}_*(A) = \mathbf{S}^*(A) = 1 \\ \textit{False} & \text{if } \mathbf{S}_*(A) = \mathbf{S}^*(A) = 0 \\ \textit{Unknown} & \text{otherwise} \end{cases}$$

The above interpretation is called the *well-founded model* of a program Π. The well-founded semantics assigns to a given CAKE program Π the well-founded model of Π.

Example 2 (continued)

We now compute the well-founded model for the program Π.

$$
\begin{aligned}
\mathbf{S}_1 &= 3\text{-}\mathcal{CS}_\Pi(\bot) = \{P^+(a), S^+(a), Q^+(a), T^+(a), U^+(a), \\
&\quad \neg R^+(a)\} \\
\mathbf{S}_2 &= 3\text{-}\mathcal{CS}_\Pi(\mathbf{S}_1) = \{P^+(a), Q^+(a), \neg R^+(a), \neg S^+(a), \\
&\quad \neg T^+(a), \neg U^+(a)\} \\
\mathbf{S}_3 &= 3\text{-}\mathcal{CS}_\Pi(\mathbf{S}_2) = \mathbf{S}_1 \\
\mathbf{S}_4 &= 3\text{-}\mathcal{CS}_\Pi(\mathbf{S}_3) = \mathbf{S}_2.
\end{aligned}
$$

$\mathbf{S}_* = \mathbf{S}_4 = \{P^+(a), Q^+(a), \neg R^+(a), \neg S^+(a), \neg T^+(a), \neg U^+(a)\}$

and

$\mathbf{S}^* = \mathbf{S}_3 = \{P^+(a), S^+(a), Q^+(a), T^+(a), U^+(a), \neg R^+(a)\}$.

Thus, $\mathbf{S}^*_* = \{P^+(a), Q^+(a), \neg R^+(a)\}$. ∎