Complete Knowledge Assumpt

- Sometimes you want to assume that a dat complete. Any fact not listed is false.
- Example: Assuming a database of *enrol* relations is complete, you can define *emp*
- The definite clause RRS is monotonic: a doesn't invalidate a previous conclusion.
- With the complete knowledge assumptio nonmonotonic: a conclusion can be inva adding more clauses.

CKA: propositional ca

- Suppose the rules for atom a are

$$a \leftarrow b_1.$$
$$\vdots$$
$$a \leftarrow b_n.$$

or equivalently: $a \leftarrow b_1 \vee \ldots \vee b_n$

- Under the CKA, if a is true, one of the b_i

$$a \rightarrow b_1 \vee \ldots \vee b_n.$$

- Under the CKA, the clauses for a mean
Clark's completion:

$$a \leftrightarrow b_1 \vee \ldots \vee b_n$$

CKA: Ground Databas

Example: Consider the relation defined by:

$$student(mary).$$

$$student(john).$$

$$student(ying).$$

The CKA specifies these three are the only stu

$$student(X) \leftrightarrow X = mary \lor X = john \lor$$

To conclude $\neg student(alan)$, you have to be a

$$alan \neq mary \land alan \neq john \land alan \neq y$$

This needs the unique names assumption.

Clark Normal Form

The **Clark normal form** of the clause:

$$p(t_1, \ldots, t_k) \leftarrow B$$

is the clause

$$p(V_1, \ldots, V_k) \leftarrow$$
$$\exists W_1 \ldots \exists W_m \, V_1 = t_1 \wedge \ldots \wedge V_k =$$

where $V_1, \ldots, V_k$ are k different variables that
in the original clause.

$W_1, \ldots, W_m$ are the original variables in the cl

Clark normal form: exan

➤ The Clark normal form of:

$$room(C, room208) \leftarrow$$

$$cs\_course(C) \wedge enrollment(C,$$

is

$$room(X, Y) \leftarrow \exists C \exists E \; X = C \wedge Y =$$

$$cs\_course(C) \wedge enrollment(C,$$

Clark's Completion of a Pre

Put all of the clauses for p into Clark normal f
same set of introduced variables:

$$p(V_1, \ldots, V_k) \leftarrow B_1$$

$$\vdots$$

$$p(V_1, \ldots, V_k) \leftarrow B_n$$

This is the same as: $p(V_1, \ldots, V_k) \leftarrow B_1 \vee$.

Clark's completion of p is the equivalence

$$p(V_1, \ldots, V_k) \leftrightarrow B_1 \vee \ldots \vee B_n,$$

That is, $p(V_1, \ldots, V_k)$ is true if and only if one

Clark's Completion Exan

Given the *mem* function:

$$mem(X, [X|T]).$$

$$mem(X, [H|T]) \leftarrow mem(X, T).$$

the completion is

$$mem(X, Y) \iff (\exists T\ Y = [X|T]) \vee$$

$$(\exists H \exists T\ Y = [H|T] \wedge mem(X, T))$$

Clark's Completion of a

- **Clark's completion** of a knowledge base
completion of every predicate symbol, alo
axioms for equality and inequality.

- If you have a predicate p defined by no cl
knowledge base, the completion is $p \leftrightarrow f$
$\neg p$.

- You can interpret negations in the bodies
means that p is false under the Complete
Assumption. This is called **negation as fa**

Using negation as failu

Previously we couldn't define $empty\_course(C$
database of $enrolled(S, C)$.

This can be defined using negation as failure:

$$
\begin{aligned}
& empty\_course(C) \leftarrow \\
& \qquad course(C) \land \\
& \qquad \sim has\_Enrollment(C). \\
& has\_Enrollment(C) \leftarrow \\
& \qquad enrolled(S, C).
\end{aligned}
$$

Bottom-up NAF proof proc

$C := \{\}$;

repeat

either select "$h \leftarrow b_1 \wedge \ldots \wedge b_m$"$\in KB$ su

$b_i \in C$ for all i, and $h \notin C$;

$C := C \cup \{h\}$

or select h such that

for every rule "$h \leftarrow b_1 \wedge \ldots \wedge$

either for some b_i, $\sim b_i \in C$

or some $b_i = \sim g$ and $g \in C$

$C := C \cup \{\sim h\}$

until no more selections are possible

Negation as failure exam

$p \leftarrow q \wedge \sim r.$

$p \leftarrow s.$

$q \leftarrow \sim s.$

$r \leftarrow \sim t.$

$t.$

$s \leftarrow w.$

Top-Down NAF Proced

- If the proof for a fails, you can conclude
- Failure and success can be defined recurs
  - Suppose you have rules for atom a:

$$a \leftarrow b_1$$
$$\vdots$$
$$a \leftarrow b_n$$

    If each body b_i fails, a fails.

    If one body, b_i succeeds, a succeeds.

  - A body fails if one of the conjuncts in

    A body succeeds if all of the conjunct

  - If a fails, $\sim a$ succeeds. If a succeeds

Free Variables in Negation as

Example:

$$p(X) \leftarrow \sim q(X) \wedge r(X).$$

$$q(a).$$

$$q(b).$$

$$r(a).$$

$$r(c).$$

There is only one answer to the query $?p(X)$,

For calls to negation as failure with free variab

to delay negation as failure goals that contai

until the variables become bound.

Floundering Goals

If the variables never become bound, a negate

flounders.

In this case you can't conclude anything abou

Example: Consider the clauses:

$$p(X) \leftarrow {\sim}q(X)$$

$$q(X) \leftarrow {\sim}r(X)$$

$$r(a)$$

and the query

$$?p(X).$$