

B3-1 配信型情報源統合環境における統合要求記述の正当性の検証

質問・応答

質問[1]

情報が来ていないときは代数式の評価結果が空になってしまい、充足可能性を満たさないと判定されてしまうのではないか？

応答[1]

充足可能性の検証は、代数式が常に空になってしまわないかを事前に調べるだけであって、配信処理を行なっている最中に調べるわけではない。

質問[2]

タイムスタンプ関数は、あの5つだけで十分なのか？

応答[2]

それについては検討する余地があると思うが、とりあえず一般的な要求を記述する上では十分だと考えている。

質問[3]

「Aチームが勝ったら配信して欲しい」というような要求は書けるのか？

応答[3]

試合の結果を配信する情報源があるとすれば、選択条件に「Aチームが勝ったら」と書くことで実現可能である。

質問[4]

「Aチームが勝ったら配信して欲しい」とすべきところを「負けたら」と間違えてしまったような場合はわかるのか？

応答[4]

当然、そのようなミスを検出することも必要である。統合要求記述は基本的にリレーショナル代数式であるので、従来のチェック手法を用いることができると考えている。記述ミスを減らすという意味では、GUIを用いて統合要求記述を生成することも有効であると思われる。

質問[5]

いつ情報が到着するか分からない情報源もあるのではないか？

応答[5]

もちろんそのような情報源は存在する。情報源の性質記述は、到着時刻がある程度分かっている情報源が対象としたものであるが、到着時刻が分からない情報源の性質記述は条件PROPにtrueを与えることで対処する。

質問[6]

到着した情報をいったん蓄積するメリットは何か？

応答[6]

配信型情報源から情報が到着するタイミングは情報源ごとにバラバラである。これらを統合しようと思ったら、先に来たほうを蓄積し、情報が揃うまで待たなくてはならない。