Confluence Properties on Open Terms in the First-Order Theory of Rewriting*

Franziska Rapp and Aart Middeldorp

Department of Computer Science, University of Innsbruck, Austria
{franziska.rapp|aart.middeldorp}@uibk.ac.at

Abstract

FORT is a decision and synthesis tool for the first-order theory of rewriting for finite left-linear right-ground rewrite systems. We report on an extension that distinguishes between ground and open terms for properties related to confluence.

1 Introduction

In a recent paper [5] we introduced FORT, a decision and synthesis tool for the first-order theory of rewriting induced by finite left-linear right-ground rewrite systems. In this theory one can express well-known properties like termination (SN), normalization (WN), and confluence (CR), but also properties like strong confluence ($\mathsf{SCR}\colon \forall\, s\, \forall\, t\, \forall\, u\, (s \to t \wedge s \to u \implies \exists\, v\, (t \to^{=} v \wedge u \to^{*} v)))$ and the normal form property ($\mathsf{NFP}\colon \forall\, s\, \forall\, t\, \forall\, u\, (s \to t \wedge s \to^{!} u \implies t \to^{!} u))$. The decision procedure is based on tree automata techniques (Dauchet and Tison [3]). Tree automata operate on *ground* terms. Consequently, variables in formulas range over ground terms and hence the properties that FORT is able to decide are restricted to ground terms. Whereas for termination and normalization this makes no difference, for other properties it does, even for *left-linear right-ground rewrite systems* as will be shown below. This raises the question how one can use FORT to decide properties on open terms. We show that for properties related to confluence it suffices to add one or two fresh constants. We furthermore provide sufficient conditions which obviate the need for additional constants. The proofs of these results are presented in the next section. The results are incorporated in version 0.2 of FORT, which we briefly describe in Section 3. We also provide a few rewrite systems that were synthesized by FORT. Section 4 contains a comparison with AGCP (Aoto and Toyama [1]), a new tool for checking ground-confluence of many-sorted rewrite systems.

We assume familiarity with first-order term rewriting [2]. In this paper we consider the following properties, besides SCR and NFP:

$$\begin{aligned}
\mathsf{CR}\colon&\ \forall\, s\, \forall\, t\, \forall\, u\, (s \to^{*} t \wedge s \to u \implies t \downarrow u)\\
\mathsf{WCR}\colon&\ \forall\, s\, \forall\, t\, \forall\, u\, (s \to t \wedge s \to u \implies t \downarrow u)\\
\mathsf{UN}\colon&\ \forall\, s\, \forall\, t\, \forall\, u\, (s \to^{!} t \wedge s \to^{!} u \implies t = u)\\
\mathsf{UNC}\colon&\ \qquad \forall\, t\, \forall\, u\, (t \leftrightarrow^{*} u \wedge \mathsf{NF}(t) \wedge \mathsf{NF}(u) \implies t = u)
\end{aligned}$$

Let $\mathfrak{P} = \{\mathsf{CR}, \mathsf{SCR}, \mathsf{WCR}, \mathsf{NFP}, \mathsf{UNC}, \mathsf{UN}\}$. In FORT 0.2 these properties are considered over all terms. Let $\mathfrak{R}$ consist of all $(\mathcal{F}, \mathcal{R})$ where $\mathcal{R}$ is a finite left-linear right-ground TRSs over the finite signature $\mathcal{F}$ which contains at least one constant.

*This work is supported by FWF (Austrian Science Fund) project P27528.

2 Ground versus Non-Ground Properties

The properties supported in FORT 0.1 are restricted to *ground* terms. So CR in FORT 0.1 stands for ground-confluence, which is different from confluence, even for left-linear right-ground TRSs. The TRS

$$\mathsf{a} \to \mathsf{b} \qquad \mathsf{f}(x, \mathsf{a}) \to \mathsf{b} \qquad \mathsf{f}(\mathsf{b}, \mathsf{b}) \to \mathsf{b}$$

is ground-confluent since all ground terms rewrite to b, but not confluent: $\mathsf{b} \leftarrow \mathsf{f}(x, \mathsf{a}) \to \mathsf{f}(x, \mathsf{b})$ with normal forms b and $\mathsf{f}(x, \mathsf{b})$. The same example shows that for no property $P \in \mathfrak{P}$, $\mathsf{G}P$ implies P, where $\mathsf{G}P$ denotes the property P restricted to ground terms. So how can we check a property $P \in \mathfrak{P}$ using tree automata techniques? The following result provides the answer.

Lemma 1. *If* $(\mathcal{F}, \mathcal{R}) \in \mathfrak{R}$ *then*

1. $(\mathcal{F}, \mathcal{R}) \vDash P \iff (\mathcal{F} \cup \{c\}, \mathcal{R}) \vDash \mathsf{G}P$ *for all* $P \in \mathfrak{P} \setminus \{\mathsf{UNC}\}$
2. $(\mathcal{F}, \mathcal{R}) \vDash \mathsf{UNC} \iff (\mathcal{F} \cup \{c, c'\}, \mathcal{R}) \vDash \mathsf{GUNC}$

with fresh constants c *and* c'.

Proof. For the only-if directions we observe that all properties $P \in \mathfrak{P}$ are preserved under signature extension [4]. Moreover, $(\mathcal{G}, \mathcal{R}) \vDash P$ implies $(\mathcal{G}, \mathcal{R}) \vDash \mathsf{G}P$ for all TRSs $(\mathcal{G}, \mathcal{R})$ and properties $P \in \mathfrak{P}$. For the if-direction, we first consider $P \in \mathfrak{P} \setminus \{\mathsf{UNC}\}$. Suppose $(\mathcal{F} \cup \{c\}, \mathcal{R}) \vDash \mathsf{G}P$ and let σ be the substitution that maps all variables to the constant c. Because $\mathcal{R}$ is left-linear and c does not appear in the rules of $\mathcal{R}$, the following property holds for all terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$:

(a) if $t\sigma \to_{\mathcal{R}} u$ then $t \to_{\mathcal{R}} u'$ with $u'\sigma = u$.

Moreover,

(b) if $t \to_{\mathcal{R}} u$ and $p \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(u)$ then $u(p) = t(p)$.

This property relies on the right-groundness of $\mathcal{R}$, which entails that the redex contracted in $t \to_{\mathcal{R}} u$ cannot be above position p. The above properties allow us to prove $(\mathcal{F}, \mathcal{R}) \vDash P$ for $P \in \{\mathsf{CR}, \mathsf{SCR}, \mathsf{WCR}\}$. Here we consider $P = \mathsf{SCR}$ and let $s \to_{\mathcal{R}} t$ and $s \to_{\mathcal{R}} u$. Closure under substitutions yields $s\sigma \to_{\mathcal{R}} t\sigma$ and $s\sigma \to_{\mathcal{R}} u\sigma$. Because $(\mathcal{F} \cup \{c\}, \mathcal{R})$ satisfies GSCR, we obtain a ground term $v \in \mathcal{T}(\mathcal{F} \cup \{c\})$ such that $s\sigma \to_{\mathcal{R}}^{=} v$ and $t\sigma \to_{\mathcal{R}}^{*} v$. Property (a) yields terms $v_1, v_2 \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that $t \to_{\mathcal{R}}^{=} v_1$ and $u \to_{\mathcal{R}}^{*} v_2$ with $v_1\sigma = v = v_2\sigma$. If $v_1 \neq v_2$ then there must be a position $p \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(v_1) \cap \mathcal{P}\mathsf{os}_{\mathcal{V}}(v_2)$ such that $v_1(p) \neq v_2(p)$. Repeated application of (b) yields $v_1(p) = t(p) = s(p)$ and $v_2(p) = u(p) = s(p)$, which is impossible. Hence $v_1 = v_2$ and thus $(\mathcal{F}, \mathcal{R}) \vDash \mathsf{SCR}$. The proofs for $P = \mathsf{CR}$ and $P = \mathsf{WCR}$ are very similar. For $P \in \{\mathsf{UN}, \mathsf{NFP}\}$ we need the following additional observation:

(c) if t is a normal form then $t\sigma$ is a normal form.

Consider $P = \mathsf{UN}$ and let $s \to_{\mathcal{R}}^{!} t$ and $s \to_{\mathcal{R}}^{!} u$ with $s \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. We obtain $s\sigma \to_{\mathcal{R}}^{!} t\sigma$ and $s\sigma \to_{\mathcal{R}}^{!} u\sigma$ from (c), and thus $t\sigma = u\sigma$ because $(\mathcal{F} \cup \{c\}, \mathcal{R})$ satisfies GUN. We need to show $t = u$. If this does not hold then there must be a position $p \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(t) \cap \mathcal{P}\mathsf{os}_{\mathcal{V}}(u)$ such that $t(p) \neq u(p)$. This contradicts $t(p) = s(p)$ and $u(p) = s(p)$, which we obtain from (b). Next consider $P = \mathsf{NFP}$. So let $s \to_{\mathcal{R}} t$ and $s \to_{\mathcal{R}}^{!} u$. We obtain $s\sigma \to_{\mathcal{R}} t\sigma$ and $s\sigma \to_{\mathcal{R}}^{!} u\sigma$ as before. Hence $t\sigma \to^{*} u\sigma$ because GNFP holds. From property (a) we obtain a term u' such that $t \to_{\mathcal{R}}^{*} u'$ and $u'\sigma = u\sigma$. Let p be any position in $\mathcal{P}\mathsf{os}_{\mathcal{V}}(u') \cap \mathcal{P}\mathsf{os}_{\mathcal{V}}(u)$. Repeated application of property (b) yields $u'(p) = t(p) = s(p) = u(p)$. Hence $u' = u$ and thus $t \to_{\mathcal{R}}^{*} u$ as desired.

Finally we consider $P = \mathsf{UNC}$. So suppose $(\mathcal{F} \cup \{c, c'\}, \mathcal{R}) \vDash \mathsf{GUNC}$ and let $t \leftrightarrow^*_{\mathcal{R}} u$ with normal forms $t, u \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. If t and u are ground then $t = u$ by GUNC. If one of the two terms is ground, say t, and $t \neq u$ then $t \neq u\sigma$ and $t \leftrightarrow^*_{\mathcal{R}} u\sigma$ for the same substitution σ as before, contradicting GUNC. If both t and u are non-ground and $t \neq u$ then, because $t\sigma = u\sigma$ by GUNC and (c), there has to be a position $p \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(t) \cap \mathcal{P}\mathsf{os}_{\mathcal{V}}(u)$ such that $t(p) \neq u(p)$. In this case a contradiction is obtained by considering the substitution σ' that maps $t(p)$ to c and all other variables to c'. □

The following example shows that adding a single fresh constant is insufficient for UNC.

Example 1. *The left-linear right-ground TRS $\mathcal{R}$ consisting of the rules*

$$\mathsf{a} \to \mathsf{b} \qquad \mathsf{f}(x, \mathsf{a}) \to \mathsf{f}(\mathsf{b}, \mathsf{b}) \qquad \mathsf{f}(\mathsf{b}, x) \to \mathsf{f}(\mathsf{b}, \mathsf{b}) \qquad \mathsf{f}(\mathsf{f}(x, y), z) \to \mathsf{f}(\mathsf{b}, \mathsf{b})$$

does not satisfy UNC *since* $\mathsf{f}(x, \mathsf{b}) \leftarrow \mathsf{f}(x, \mathsf{a}) \to \mathsf{f}(\mathsf{b}, \mathsf{b}) \leftarrow \mathsf{f}(y, \mathsf{a}) \to \mathsf{f}(y, \mathsf{b})$ *is a conversion between distinct normal forms. Adding a single fresh constant* c *is not enough to violate* GUNC *as the last two rewrite rules ensure that* $\mathsf{f}(\mathsf{c}, \mathsf{b})$ *is the only ground instance of* $\mathsf{f}(x, \mathsf{b})$ *that is a normal form. Adding another fresh constant* c'*,* GUNC *is lost:* $\mathsf{f}(\mathsf{c}, \mathsf{b}) \leftarrow \mathsf{f}(\mathsf{c}, \mathsf{a}) \to \mathsf{f}(\mathsf{b}, \mathsf{b}) \leftarrow \mathsf{f}(\mathsf{c}', \mathsf{a}) \to \mathsf{f}(\mathsf{c}', \mathsf{b})$*.*

For termination (SN) and normalizion (WN) there is no need to add fresh constants, since these properties hold if and only if they hold for all ground terms. For other properties that can be expressed in the first-order theory of rewriting, one or two fresh constants may be insufficient. Consider e.g. the formula φ:

$$\neg\, \exists\, s\, \exists\, t\, \exists\, u\, \forall\, v\, (v \leftrightarrow^* s \,\vee\, v \leftrightarrow^* t \,\vee\, v \leftrightarrow^* u)$$

which is satisfied on open terms (with respect to any $(\mathcal{F}, \mathcal{R}) \in \mathfrak{R}$). For the TRS consisting of the rule $\mathsf{f}(x) \to \mathsf{a}$ and two additional constants c and c', φ does not hold for ground terms because every ground term is convertible to a, c or c'. It is tempting to believe that adding a fresh unary symbol g in addition to a fresh constant c, in order to create infinitely many ground normal forms which can replace variables that appear in open terms, is sufficient for any property P. The formula $\forall\, s\, \forall\, t\, (s \to t \implies s \xrightarrow{\epsilon} t)$ and the TRS consisting of the rule $\mathsf{a} \to \mathsf{b}$ show that this is incorrect.

It is interesting to note that the two properties in the preceding paragraph are not *component-closed* [6], unlike the properties in $\mathfrak{P}$. This observation can be used to generalize Lemma 1 to confluence-related properties outside $\mathfrak{P}$. The following result shows that for the properties in $\mathfrak{P}$ it is not always necessary to add fresh constants. Here a *monadic* signature consists of constants and unary function symbols.

Lemma 2. *Let $(\mathcal{F}, \mathcal{R}) \in \mathfrak{R}$ such that $\mathcal{R}$ is ground or $\mathcal{F}$ is monadic. For all $P \in \mathfrak{P}$*

$$(\mathcal{F}, \mathcal{R}) \vDash P \quad \iff \quad (\mathcal{F}, \mathcal{R}) \vDash \mathsf{G}P$$

Proof. First assume that $\mathcal{R}$ is ground. In this case only ground subterms can be rewritten. Given a term $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, we write $t = C[\![t_1, \ldots, t_n]\!]$ if $t = C[t_1, \ldots, t_n]$ and $t_1, \ldots, t_n$ are the maximal ground subterms of t. So all variables appearing in t occur in C. The following property is obvious:

1. if $t = C[\![t_1, \ldots, t_n]\!] \to^*_{\mathcal{R}} u$ then $u = C[\![u_1, \ldots, u_n]\!]$ and $t_i \to^*_{\mathcal{R}} u_i$ for all $1 \leqslant i \leqslant n$.

Suppose $(\mathcal{F}, \mathcal{R}) \vDash \mathsf{GCR}$ and consider $s \to^*_{\mathcal{R}} t$ and $s \to^*_{\mathcal{R}} u$ with $s \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. Writing $s = C[\![s_1, \ldots, s_n]\!]$, we obtain $t = C[\![t_1, \ldots, t_n]\!]$ and $u = C[\![u_1, \ldots, u_n]\!]$ with $s_i \to^*_{\mathcal{R}} t_i$ and $s_i \to^*_{\mathcal{R}} u_i$ for all $1 \leqslant i \leqslant n$. GCR yields $t_i \downarrow u_i$ for all $1 \leqslant i \leqslant n$. Hence $t \downarrow u$ as desired. The proofs for the other properties in $\mathfrak{P}$ are equally easy. For UNC note that $\leftrightarrow^*_{\mathcal{R}}$ coincides with $\rightarrow^*_{\mathcal{R}\,\cup\,\mathcal{R}^{-1}}$ for the *ground* TRS $\mathcal{R} \cup \mathcal{R}^{-1}$, where $\mathcal{R}^{-1}$ is obtained from $\mathcal{R}$ by reversing the rewrite rules.

Next suppose that $\mathcal{F}$ is monadic. Let $(\mathcal{F}, \mathcal{R}) \vDash \mathsf{G}P$ and let σ be the substitution that maps all variables to some arbitrary but fixed ground term. In this case the following property holds:

2. if $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and $t \to u$ then $u \in \mathcal{T}(\mathcal{F})$ and $t\sigma \to u$.

We consider $P = \mathsf{NFP}$ and $P = \mathsf{UNC}$ and leave the proof for the other properties to the reader. Let $s \to_{\mathcal{R}} t$ and $s \to^!_{\mathcal{R}} u$. We obtain $s\sigma \to_{\mathcal{R}} t$ and $s\sigma \to^!_{\mathcal{R}} u$ from property 2. (Note that $s \neq u$.) Hence $t \to^*_{\mathcal{R}} u$ follows from GNFP. Let $t \leftrightarrow^*_{\mathcal{R}} u$ with normal forms t and u. If t and u are ground terms then we obtain $t = u$ from GUNC (after applying the substitution σ to all intermediate terms in the conversion between t and u). Otherwise, the conversion between t and u must be empty due to property 2 and the fact that t and u are normal forms. Hence also in this case $t = u$. ☐

FORT indeed benefits from this optimization. Checking for GCR of the TRS

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(x))) \to \mathsf{a} \quad \mathsf{f}(\mathsf{f}(\mathsf{a})) \to \mathsf{a} \quad \mathsf{f}(\mathsf{a}) \to \mathsf{a} \quad \mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{g}(x)))) \to \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{f}(\mathsf{a})) \to \mathsf{a} \quad \mathsf{g}(\mathsf{a}) \to \mathsf{a}$$

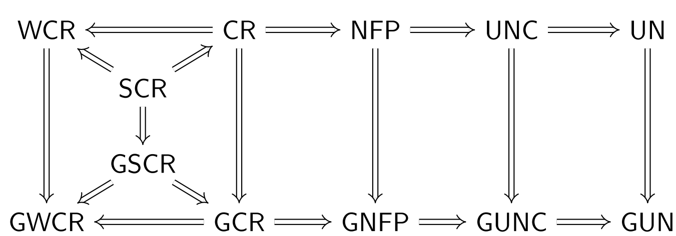
whose signature is monadic takes 1.73 seconds if a fresh constant is added, compared to 0.85 seconds if Lemma 2 is used.

3 Synthesis Experiments with FORT 0.2

The results of the previous section are incorporated in version 0.2 of FORT. Compared to version 0.1, the properties in $\mathfrak{P}$ now refer to open terms and we reserve $\mathsf{G}P$ with $P \in \mathfrak{P}$ for properties on ground terms. The property SCR, which is new in version 0.2, can also be used for parallel rewriting (SCR($\nrightarrow$)) and the same holds for the diamond property ($\diamond(\nrightarrow)$), which is another addition in FORT 0.2. Further additions can be found in the online documentation of FORT. Precompiled binaries to run FORT 0.2 from the command line are available from

http://cl-informatik.uibk.ac.at/software/FORT

We report on some synthesis experiments that we performed in FORT 0.2, based on the following diagram which summarizes the relationships between properties P and $\mathsf{G}P$ for $P \in \mathfrak{P}$:



The following TRSs were produced by FORT 0.2 on the given formulas when restricting the signature (using the option `-f "f:2 a:0 b:0"`) to a binary function symbol f and two constants a and b:

GWCR & ~WCR & ~GCR	$\mathsf{a} \to \mathsf{b}$	$\mathsf{f}(x, \mathsf{a}) \to \mathsf{a}$	$\mathsf{a} \to \mathsf{f}(\mathsf{a}, \mathsf{a})$
GCR & ~CR & ~GSCR	$\mathsf{a} \to \mathsf{b}$	$\mathsf{f}(x, \mathsf{a}) \to \mathsf{b}$	$\mathsf{b} \to \mathsf{f}(\mathsf{a}, \mathsf{a})$
GNFP & ~NFP & ~GCR	$\mathsf{a} \to \mathsf{b}$	$\mathsf{f}(x, \mathsf{a}) \to \mathsf{f}(\mathsf{a}, \mathsf{a})$	$\mathsf{f}(\mathsf{b}, \mathsf{b}) \to \mathsf{f}(\mathsf{a}, \mathsf{a})$
GUNC & ~UNC & ~GNFP	$\mathsf{a} \to \mathsf{a}$	$\mathsf{f}(x, \mathsf{a}) \to \mathsf{a}$	$\mathsf{f}(\mathsf{b}, x) \to \mathsf{b}$

tool	yes (∅ time)	no (∅ time)	maybe (∅ time)	timeout	total time
AGCP	8 (0.02 s)	–	56 (0.19 s)	1	71 s
FORT	42 (0.37 s)	14 (3.31 s)	–	9	602 s

Table 1: Comparison of AGCP and FORT 0.2 on 65 left-linear right-ground TRSs.

The reader is encouraged to verify that the synthesized TRSs indeed satisfy the indicated properties. We do not know whether there exist TRSs over the restricted signature that satisfy `GUN & ~UN & ~GUNC`. Human expertise was used to produce a witness over a larger signature, which was subsequently simplified using the decision mode of FORT 0.2:

$$\begin{array}{lllll} \mathsf{b} \to \mathsf{a} & \mathsf{c} \to \mathsf{c} & \mathsf{d} \to \mathsf{c} & \mathsf{f}(x, \mathsf{a}) \to \mathsf{A} & \mathsf{f}(x, \mathsf{A}) \to \mathsf{A} \\ \mathsf{b} \to \mathsf{c} & & \mathsf{d} \to \mathsf{e} & \mathsf{f}(x, \mathsf{e}) \to \mathsf{A} & \mathsf{f}(\mathsf{c}, x) \to \mathsf{A} \end{array}$$

4 Comparison

The tool AGCP[1] uses rewriting induction to automatically prove ground-confluence of many-sorted TRSs (Aoto & Toyama [1]). In Table 1 we compare FORT 0.2 and AGCP on the 65 left-linear right-ground TRSs from the combined confluence[2] and termination[3] problem databases. These TRSs were presented to AGCP as many-sorted TRSs having exactly one sort. It is interesting to note that there is no difference between confluence and ground-confluence on this database. We used a 60 seconds time limit. Unlike FORT, AGCP is not restricted to left-linear right-ground TRSs. Moreover, AGCP is much faster than FORT. In the near future, we plan to extend FORT to many-sorted TRSs in order to allow a fairer comparison to AGCP.

Acknowledgments Discussions with Bertram Felgenhauer and Vincent van Oostrom helped to improve the paper. The same holds for the remarks by the anonymous reviewers.

References

[1] T. Aoto and Y. Toyama. Ground confluence prover based on rewriting induction. In *Proc. 1st International Conference on Formal Structures for Computation and Deduction*, volume 52 of *Leibniz International Proceedings in Informatics*, pages 33:1–33:12, 2016. doi: 10.4230/LIPIcs.FSCD.2016.33.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[3] M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *Proc. 5th IEEE Symposium on Logic in Computer Science*, pages 242–248, 1990. doi: 10.1109/LICS.1990.113750.

[4] A. Middeldorp. *Modular Properties of Term Rewriting Systems*. PhD thesis, Vrije Universiteit, Amsterdam, 1990.

[5] F. Rapp and A. Middeldorp. Automating the first-order theory of left-linear right-ground term rewrite systems. In *Proc. 1st International Conference on Formal Structures for Computation and Deduction*, volume 52 of *Leibniz International Proceedings in Informatics*, pages 36:1–36:12, 2016. doi: 10.4230/LIPIcs.FSCD.2016.36.

[6] H. Zantema. Termination of term rewriting: Interpretation and type elimination. *JSC*, 17(1):23–50, 1994. doi: 10.1006/jsco.1994.1003.

[1] http://www.nue.ie.niigata-u.ac.jp/tools/agcp/
[2] http://cops.uibk.ac.at/
[3] http://termination-portal.org/wiki/TPDB